Practical 1

Aim: The "Caesar Box," or "Caesar Cipher," is one of the earliest known ciphers. Developed around 100 BC, it was used by Julius Caesar to send secret messages to his generals in the field. In the event that one of his messages got intercepted, his opponent could not read them. This obviously gave him a great strategic advantage. Caesar shifted each letter of his message few letters to the right to produce what could be called the ciphertext. The ciphertext is what the enemy would see instead of the true message. So, for example, if Caesar’s messages were written in the English alphabet, and shift by 3 then each letter “A” in the message would become a "D," the "B’s" would become "E’s," and the "X's" become "A’s." This type of cipher is appropriately called a “shift cipher.” Implement the cipher in any programming language of your choice. Perform encryption, decryption. Discuss and try some possible attacks on traditional Caesar cipher.

Source Code :

#Encryption

```
def encrypt(text,key):
    result = ""
    text = "".join(text.split())
    for i in range(len(text)):
        char = text[i]
        if (char.isupper()):
            result += chr((ord(char) + key + 65) % 26 + 65)
        else:
            result += chr((ord(char) + key + 97) % 26 + 97)
    return "Your Encrypted Code is ", result


text = input("Please enter your message: ")
key = int(input("Enter your secret key:"))


print("Plain Text: ", text)
print("Shifted Pattern: ", key)
print("Cipher Text: ",encrypt(text,key))
print("Dhyey Joshi ~ 19DCS043")
```

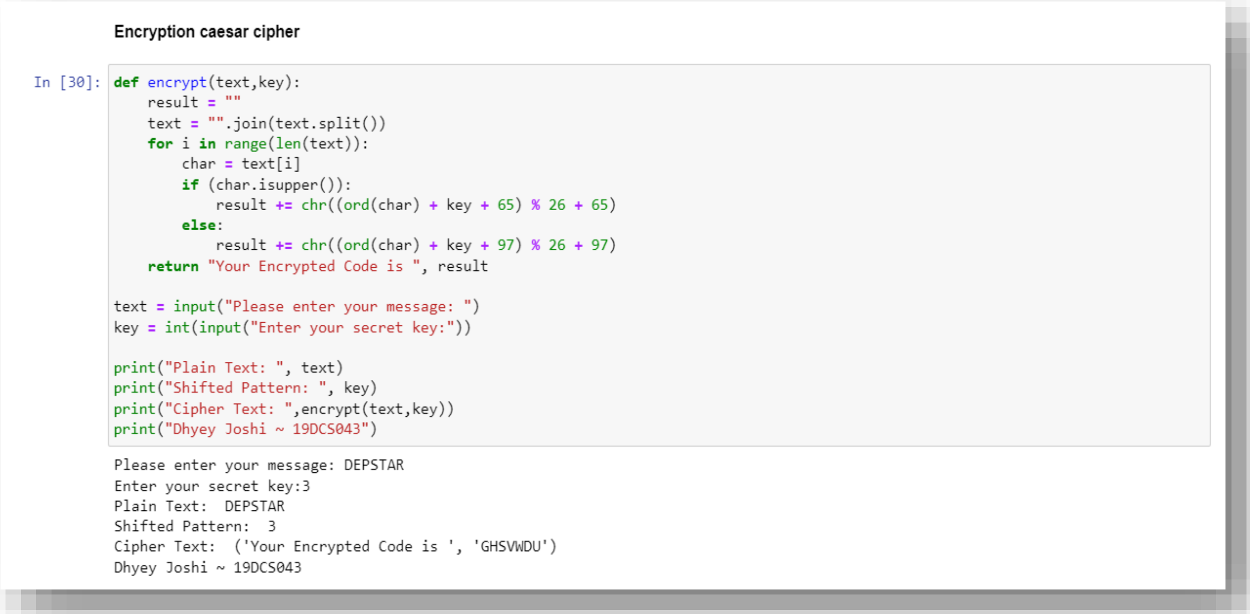
#Decryption

```
text = input("Please enter your cipher message: ")
key = int(input("Enter your secret key:"))

def decrypt(cipher,k):
    result = ""
    cipher = "".join(cipher.split())
    for i in range(len(cipher)):
        char = cipher[i]
        if (char.isupper()):
            result += chr((ord(char) - k + 65) % 26 + 65)
        else:
            result += chr((ord(char) - k + 97) % 26 + 97)
    return "Your Encrypted Code is ", result

print("Cipher Text: ", cipher)
print("Shifted Pattern: ", k)
print("Plain Text: ",decrypt(cipher,k))
print("Dhyey Joshi ~ 19DCS043")
```

#The Brute Force Attack on Caesar Cipher

```
#The Brute-Force Attack
#Decryption
message = "GHSVWDU"
LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
for key in range(len(LETTERS)):
  translated = ''
  for symbol in message:
    if symbol in LETTERS:
      num = LETTERS.find(symbol)
      num = num - key
      if num < 0:
         num = num + len(LETTERS)
      translated = translated + LETTERS[num]
    else:
        translated = translated + symbol
  print('Key #%s: %s' % (key, translated))
print("Dhyey Joshi ~ 19DCS043")
```

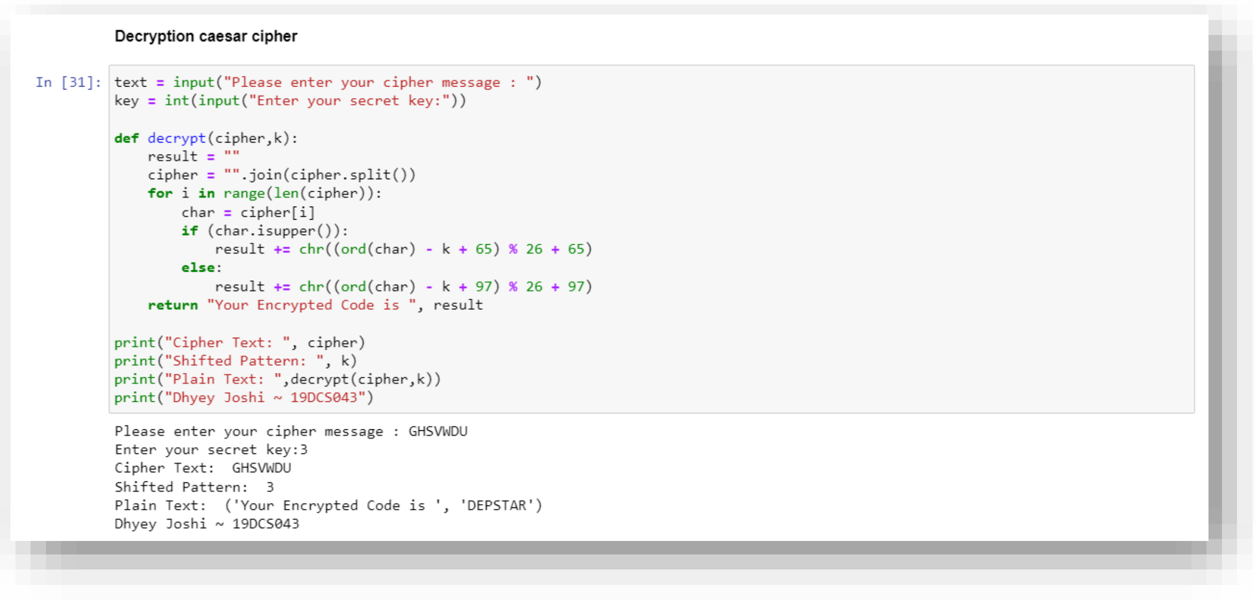
Output Screenshot:



Encryption caesar cipher

```
In [30]: def encrypt(text,key):
             result = ""
             text = "".join(text.split())
             for i in range(len(text)):
                 char = text[i]
                 if (char.isupper()):
                     result += chr((ord(char) + key + 65) % 26 + 65)
                 else:
                     result += chr((ord(char) + key + 97) % 26 + 97)
             return "Your Encrypted Code is ", result

         text = input("Please enter your message: ")
         key = int(input("Enter your secret key:"))

         print("Plain Text: ", text)
         print("Shifted Pattern: ", key)
         print("Cipher Text: ",encrypt(text,key))
         print("Dhyey Joshi ~ 19DCS043")
```

```
Please enter your message: DEPSTAR
Enter your secret key:3
Plain Text:  DEPSTAR
Shifted Pattern:  3
Cipher Text:  ('Your Encrypted Code is ', 'GHSVWDU')
Dhyey Joshi ~ 19DCS043
```

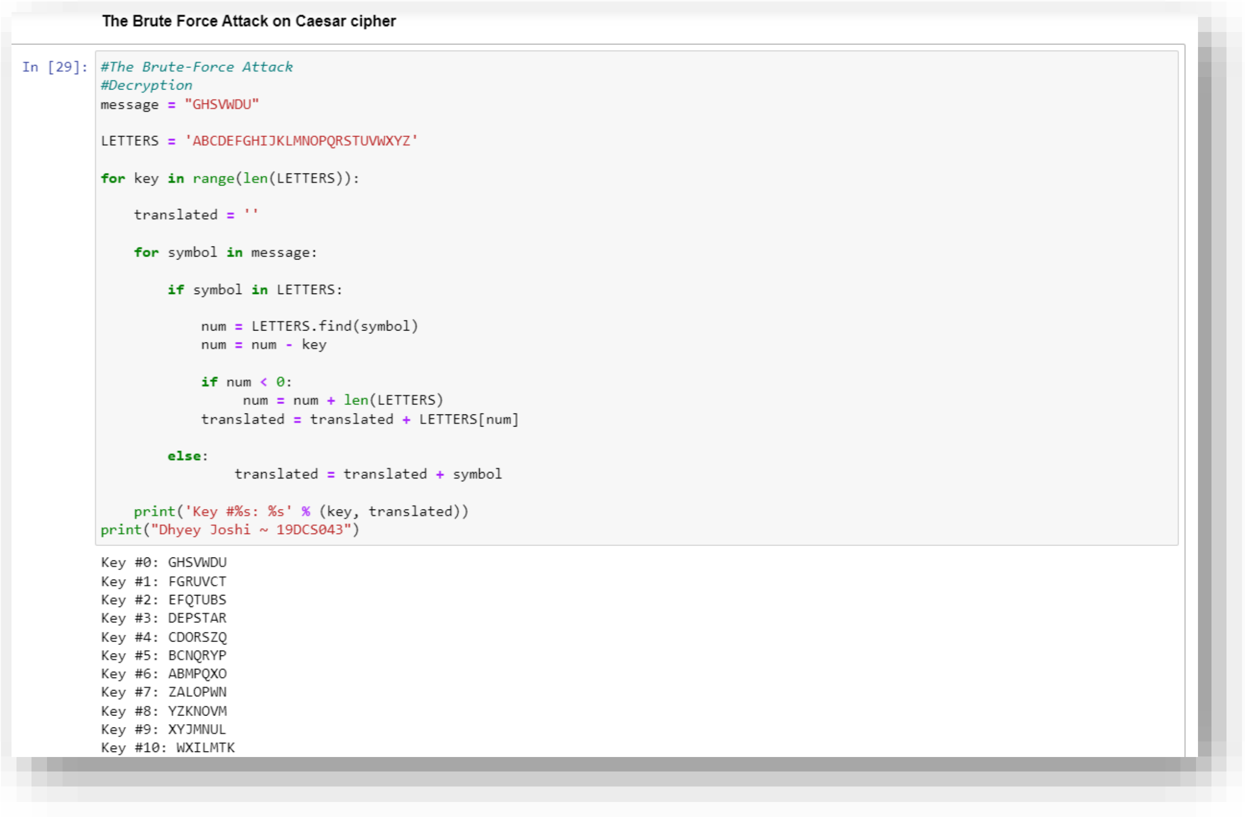


Decryption caesar cipher

In [31]:

```
text = input("Please enter your cipher message : ")
key = int(input("Enter your secret key:"))

def decrypt(cipher,k):
    result = ""
    cipher = "".join(cipher.split())
    for i in range(len(cipher)):
        char = cipher[i]
        if (char.isupper()):
            result += chr((ord(char) - k + 65) % 26 + 65)
        else:
            result += chr((ord(char) - k + 97) % 26 + 97)
    return "Your Encrypted Code is ", result

print("Cipher Text: ", cipher)
print("Shifted Pattern: ", k)
print("Plain Text: ",decrypt(cipher,k))
print("Dhyey Joshi ~ 19DCS043")
```

```
Please enter your cipher message : GHSVWDU
Enter your secret key:3
Cipher Text:  GHSVWDU
Shifted Pattern:  3
Plain Text:  ('Your Encrypted Code is ', 'DEPSTAR')
Dhyey Joshi ~ 19DCS043
```

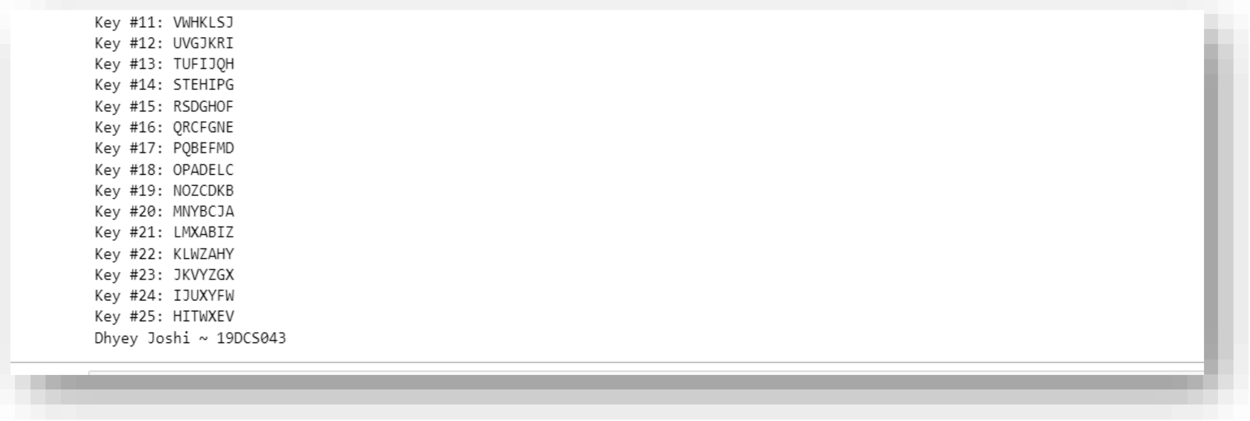


The Brute Force Attack on Caesar cipher

In [29]:

```
#The Brute-Force Attack
#Decryption
message = "GHSVWDU"

LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

for key in range(len(LETTERS)):

    translated = ''

    for symbol in message:

        if symbol in LETTERS:

            num = LETTERS.find(symbol)
            num = num - key

            if num < 0:
                 num = num + len(LETTERS)
            translated = translated + LETTERS[num]

        else:
                translated = translated + symbol

    print('Key #%s: %s' % (key, translated))
print("Dhyey Joshi ~ 19DCS043")
```

```
Key #0: GHSVWDU
Key #1: FGRUVCT
Key #2: EFQTUBS
Key #3: DEPSTAR
Key #4: CDORSZQ
Key #5: BCNQRYP
Key #6: ABMPQXO
Key #7: ZALOPWN
Key #8: YZKNOVM
Key #9: XYJMNUL
Key #10: WXILMTK
```



```
Key #11: VWHKLSJ
Key #12: UVGJKRI
Key #13: TUFIJQH
Key #14: STEHIPG
Key #15: RSDGHOF
Key #16: QRCFGNE
Key #17: PQBEFMD
Key #18: OPADELC
Key #19: NOZCDKB
Key #20: MNYBCJA
Key #21: LMXABIZ
Key #22: KLWZAHY
Key #23: JKVYZGX
Key #24: IJUXYFW
Key #25: HITWXEV
Dhyey Joshi ~ 19DCS043
```

CONCLUSION:

From this practical, we learnt the concept of encryption & decryption of Caesar Cipher and also learnt how to implement it as shown in the above codes.

Practical 2

Aim: The Playfair cipher was predominantly used by British forces during the Second Boer War (1899-1902) and World War I (1914- 1918). Soldier from field wants to send message to base. Implement the cipher to encrypt and decrypt message.

Encrypt message: Hiroshima

Use key: pearlharbour

THEORY:

- The Playfair cipher or Playfair square or Wheatstone–Playfair cipher is a manual symmetric encryption technique and was the first literal digram substitution cipher.
- The technique encrypts pairs of letters (bigrams or digrams), instead of single letters as in the simple substitution cipher. The Playfair is thus significantly harder to break since the frequency analysis used for simple substitution ciphers does not work with it. The frequency analysis of bigrams is possible, but considerably more difficult.

The Playfair Cipher Encryption Algorithm:

The Algorithm consists of 2 steps:

1) Generate the key Square(5×5):
The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I. The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.
2) Algorithm to encrypt the plain text: The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.

The Playfair Cipher Decryption Algorithm:

The Algorithm consists of 2 steps:

1. Generate the key Square(5×5) at the receiver's end:
The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet

(usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I. The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.

2. Algorithm to decrypt the ciphertext: The ciphertext is split into pairs of two letters (digraphs).

Terminology:

- Plaintext: It is the original message that is to be encrypted. It is also known as a message.
- Ciphertext: It is an encrypted message.
- Cipher: It is an algorithm for transforming plaintext to ciphertext.
- Key: It is the key to encrypt or decrypt the plaintext. It is known only to the sender and receiver. It is filled character by character in the matrix that is called key-table or key-matrix.
- Encipher: The process of converting plaintext into ciphertext is called encipher. Decipher: The process of removing ciphertext from plaintext is called decipher. Cryptanalysis: It is the study of the methods and principles of deciphering ciphertext without knowing the key.

Source Code :

#Encryption

```
key=input("Enter key")
key=key.replace(" ", "")
key=key.upper()
def matrix(x,y,initial):
    return [[initial for i in range(x)] for j in range(y)]

result=list()
for c in key: #storing key
    if c not in result:
        if c=='J':
            result.append('I')
        else:
            result.append(c)
```

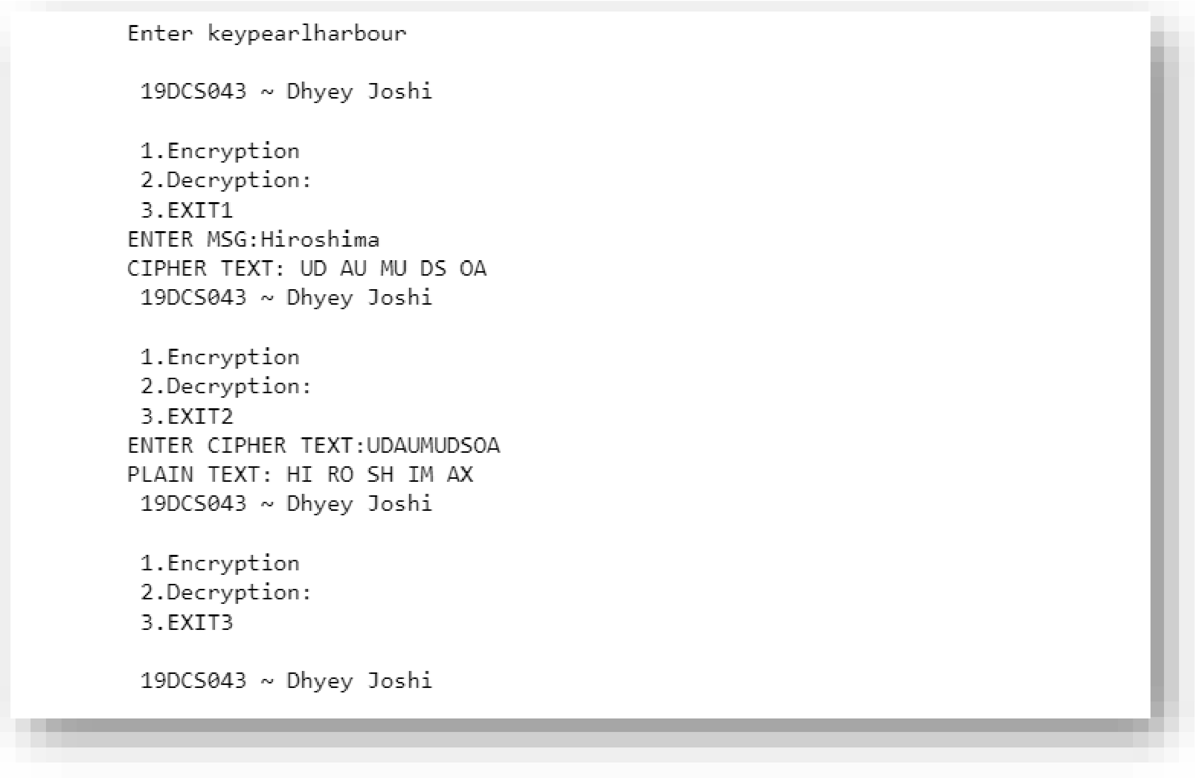
```
flag=0
for i in range(65,91): #storing other character
    if chr(i) not in result:
        if i==73 and chr(74) not in result:
            result.append("I")
            flag=1
        elif flag==0 and i==73 or i==74:
            pass
        else:
            result.append(chr(i))
k=0
my_matrix=matrix(5,5,0) #initialize matrix
for i in range(0,5): #making matrix
    for j in range(0,5):
        my_matrix[i][j]=result[k]
        k+=1

def locindex(c): #get location of each character
    loc=list()
    if c=='J':
        c='I'
    for i ,j in enumerate(my_matrix):
        for k,l in enumerate(j):
            if c==l:
                loc.append(i)
                loc.append(k)
                return loc
```

```
def encrypt():  #Encryption
    msg=str(input("ENTER MSG:"))
    msg=msg.upper()
    msg=msg.replace(" ", "")
    i=0
    for s in range(0,len(msg)+1,2):
        if s<len(msg)-1:
            if msg[s]==msg[s+1]:
                msg=msg[:s+1]+'X'+msg[s+1:]
    if len(msg)%2!=0:
        msg=msg[:]+'X'
    print("CIPHER TEXT:",end=' ')
    while i<len(msg):
        loc=list()
        loc=locindex(msg[i])
        loc1=list()
        loc1=locindex(msg[i+1])
        if loc[1]==loc1[1]:

print("{}{}".format(my_matrix[(loc[0]+1)%5][loc[1]],my_matrix[(loc1[0]+1)%5][loc1[1]]),end
=' ')
        elif loc[0]==loc1[0]:

print("{}{}".format(my_matrix[loc[0]][(loc[1]+1)%5],my_matrix[loc1[0]][(loc1[1]+1)%5]),end
=' ')
        else:
            print("{}{}".format(my_matrix[loc[0]][loc1[1]],my_matrix[loc1[0]][loc[1]]),end=' ')
        i=i+2
```

#Decryption

```
def decrypt():  #decryption
    msg=str(input("ENTER CIPHER TEXT:"))
    msg=msg.upper()
    msg=msg.replace(" ", "")
    print("PLAIN TEXT:",end=' ')
    i=0
    while i<len(msg):
        loc=list()
        loc=locindex(msg[i])
        loc1=list()
        loc1=locindex(msg[i+1])
        if loc[1]==loc1[1]:
            print("{}{}".format(my_matrix[(loc[0]-1)%5][loc[1]],my_matrix[(loc1[0]-
1)%5][loc1[1]]),end=' ')
        elif loc[0]==loc1[0]:
            print("{}{}".format(my_matrix[loc[0]][(loc[1]-1)%5],my_matrix[loc1[0]][(loc1[1]-
1)%5]),end=' ')
        else:
            print("{}{}".format(my_matrix[loc[0]][loc1[1]],my_matrix[loc1[0]][loc[1]]),end=' ')
        i=i+2

while(1):
    choice=int(input("\n 1.Encryption \n 2.Decryption: \n 3.EXIT"))
    if choice==1:
        encrypt()
    elif choice==2:
        decrypt()
    elif choice==3:
        exit()
    else:
        print("Choose correct choice")
```

Output Screenshot:



```
Enter keypearlharbour

 19DCS043 ~ Dhyey Joshi

 1.Encryption
 2.Decryption:
 3.EXIT1
ENTER MSG:Hiroshima
CIPHER TEXT: UD AU MU DS OA
 19DCS043 ~ Dhyey Joshi

 1.Encryption
 2.Decryption:
 3.EXIT2
ENTER CIPHER TEXT:UDAUMUDSOA
PLAIN TEXT: HI RO SH IM AX
 19DCS043 ~ Dhyey Joshi

 1.Encryption
 2.Decryption:
 3.EXIT3

 19DCS043 ~ Dhyey Joshi
```

CONCLUSION:

From this practical, we learnt the concept of encryption & decryption of Playfair Cipher and also learnt how to implement it as shown in the above codes.

Practical 3

Aim: The Rail Fence Cipher was invented in ancient times. It was used by the Greeks, who created a special tool, called scytale, to make message encryption and decryption easier. The letters are arranged in a way which is similar to the shape of the top edge of the rail fence. If king Leonidas want to send message to Sparta as “300 achieved glory at hot gate, unite for Greece” then what will be ciphertext when it is encrypted using 3 rows. Also implement decryption of message.

THEORY:

- The rail fence cipher is a transposition cipher that jumbles up the order of the letters of a message using a basic algorithm.
- The Rail Fence Cipher Encryption Algorithm:
- In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence. The key corresponds to the number of rails.
- When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus, the alphabets of the message are written in a zig-zag manner.
- After each alphabet has been written, the individual rows are combined to obtain the cipher-text.
- After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

The Rail Fence Cipher Decryption Algorithm:

- Hence, rail matrix can be constructed accordingly. Once we’ve got the matrix we can figure-out the spots where texts should be placed (using the same way of moving diagonally up and down alternatively).
- Then, we fill the cipher-text row wise. After filling it, we traverse the matrix in zig-zag manner to obtain the original text.

The Security of Rail Fence Cipher:

- Ciphertexts produced by transposition ciphers are relatively easy to recognize, because the frequency distribution of all ciphertext alphabet letters is the same as in plain messages written in the same language.
- Due to the small number of possible keys, the Rail Fence Cipher can be broken quite easily by using brute force attacks. The attacker should check all the practicable numbers of rail levels, that might have been using during encryption.

Source Code :

#Rail Fence Cipher --> Encryption

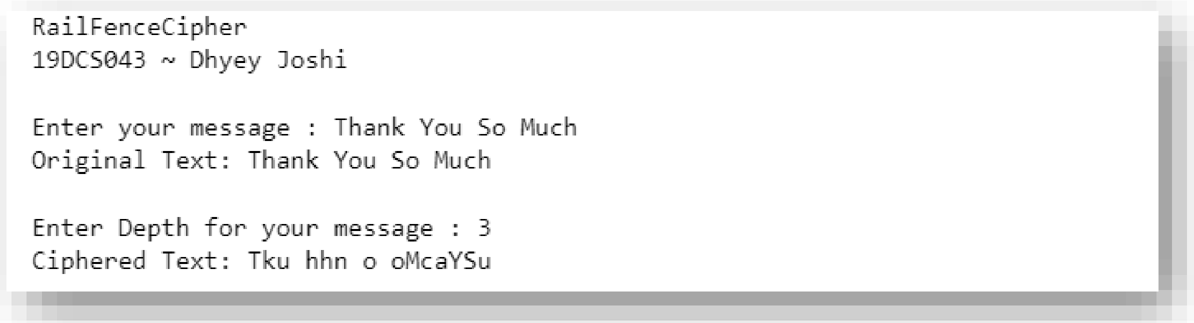
```
def main():
    print( "RailFenceCipher" )
    print( "19DCS043 ~ Dhyey Joshi" )
    print()
    clearText = str(input("Enter your message : "))
    print ("Original Text: " + clearText)
    print()
    key = int(input("Enter Depth for your message : "))
    cipherText = cipher(clearText, key)
    print ("Ciphered Text: {0}".format(cipherText))
    return
def cipher(clearText, key):
    result = ""
    matrix = [["" for x in range(len(clearText))] for y in range(key)]
    increment = 1
    row = 0
    col = 0
    for c in clearText:
        if row + increment < 0 or row + increment >= len(matrix):
            increment = increment * -1
        matrix[ row ][ col ] = c
        row += increment
        col += 1
    for list in matrix:
        result += "".join(list)
    return result
```

```
main()
```

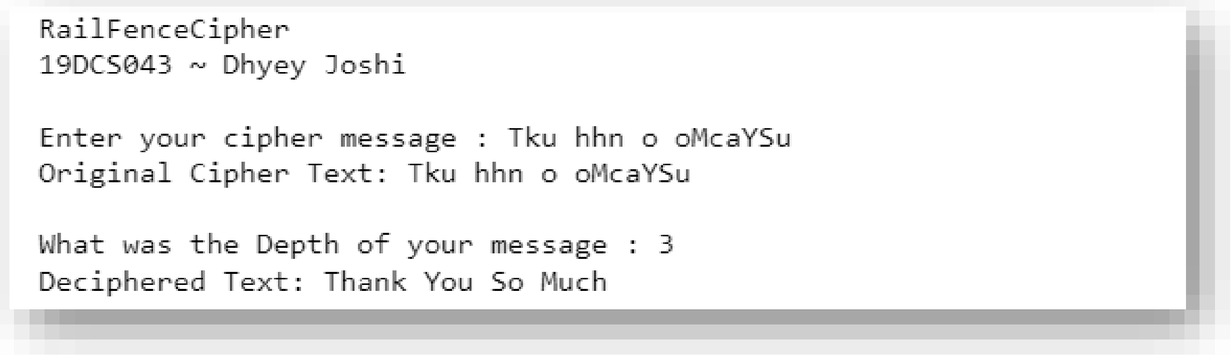
#Rail Fence Cipher --> Decryption

```
def main():
    print( "RailFenceCipher" )
    print( "19DCS043 ~ Dhyey Joshi" )
    print()
    cipherText = str(input("Enter your cipher message : "))
    print ("Original Cipher Text: " + cipherText)
    print()
    key = int(input("What was the Depth of your message : "))
    decipherText = decipher(cipherText, key)
    print ("Deciphered Text: {0}".format(decipherText))
    return
def decipher(cipherText, key):
    result = ""
    matrix = [["" for x in range(len(cipherText))] for y in range(key)]
    idx = 0
    increment = 1
    for selectedRow in range(0, len(matrix)):
        row = 0
        for col in range(0, len(matrix[ row ])):
            if row + increment < 0 or row + increment >= len(matrix):
                increment = increment * -1
            if row == selectedRow:
                matrix[row][col] += cipherText[idx]
                idx += 1
            row += increment
    matrix = transpose( matrix )
    for list in matrix:
        result += "".join(list)
    return result
def transpose( m ):
    result = [ [ 0 for y in range( len(m) ) ] for x in range( len(m[0]) ) ]
    for i in range( len(m) ):
        for j in range( len(m[0]) ):
            result[ j ][ i ] = m[ i ][ j ]
    return result
main()
```

Output Screenshot:



```
RailFenceCipher
19DCS043 ~ Dhyey Joshi

Enter your message : Thank You So Much
Original Text: Thank You So Much

Enter Depth for your message : 3
Ciphered Text: Tku hhn o oMcaYSu
```



```
RailFenceCipher
19DCS043 ~ Dhyey Joshi

Enter your cipher message : Tku hhn o oMcaYSu
Original Cipher Text: Tku hhn o oMcaYSu

What was the Depth of your message : 3
Deciphered Text: Thank You So Much
```

CONCLUSION:

From this practical, we learnt the concept of encryption & decryption of Rail Fence Cipher and also learnt how to implement it as shown in the above codes.

Practical 4

Aim: RSA algorithm is used by Salim to transfer session key to Anarkali. He suspects that Akbar is performing man in middle attack he chose to use 1024 bit prime numbers. Hint: you may choose to use big integer in java.

THEORY:

RSA Encryption Algorithm:

RSA is the most common public-key algorithm, named after its inventors Rivest, Shamir, and Adelman (RSA). Public Key encryption algorithm is also called the Asymmetric algorithm. Asymmetric algorithms are those algorithms in which sender and receiver use different keys for encryption and decryption. Each sender is assigned a pair of keys:

- Public key
- Private key

The Public key is used for encryption, and the Private Key is used for decryption. Decryption cannot be done using a public key. The two keys are linked, but the private key cannot be derived from the public key. The public key is well known, but the private key is secret and it is known only to the user who owns the key. It means that everybody can send a message to the user using user's public key. But only the user can decrypt the message using his private key.

The Public key algorithm operates in the following manner:

- The data to be sent is encrypted by sender A using the public key of the intended receiver
- B decrypts the received ciphertext using its private key, which is known only to B. B replies to A encrypting its message using A's public key.
- A decrypts the received ciphertext using its private key, which is known only to him.

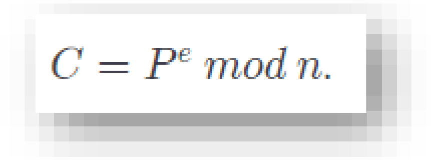
RSA algorithm uses the following procedure to generate public and private keys:

- Select two large prime numbers, p and q.

- Multiply these numbers to find n = p x q, where n is called the modulus for encryption and decryption.
- Choose a number e less than n, such that n is relatively prime to (p - 1) x (q -1). It means that e and (p - 1) x (q - 1) have no common factor except 1. Choose "e" such that 1<e < φ (n), e is prime to φ (n),
- gcd (e,d(n)) =1
- If n = p x q, then the public key is <e, n>. A plaintext message m is encrypted using public key <e, n>. To find ciphertext from the plain text following formula is used to get ciphertext C.
- C = me mod n
- Here, m must be less than n. A larger message (>n) is treated as a concatenation of messages, each of which is encrypted separately.
- To determine the private key, we use the following formula to calculate the d such that:
- De mod {(p - 1) x (q - 1)} = 1
- Or
- De mod φ (n) = 1
- The private key is <d, n>. A ciphertext message c is decrypted using private key <d, n>. To calculate plain text m from the ciphertext c following formula is used to get plain text m.
- m = cd mod n

For Encryption

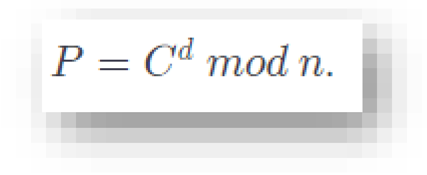
Given a plaintext PP, represented as a number, the ciphertext CC is calculated as:



$$C = P^e \; mod \; n.$$

For Decryption

Using the private key (n,d)(n,d), the plaintext can be found using:

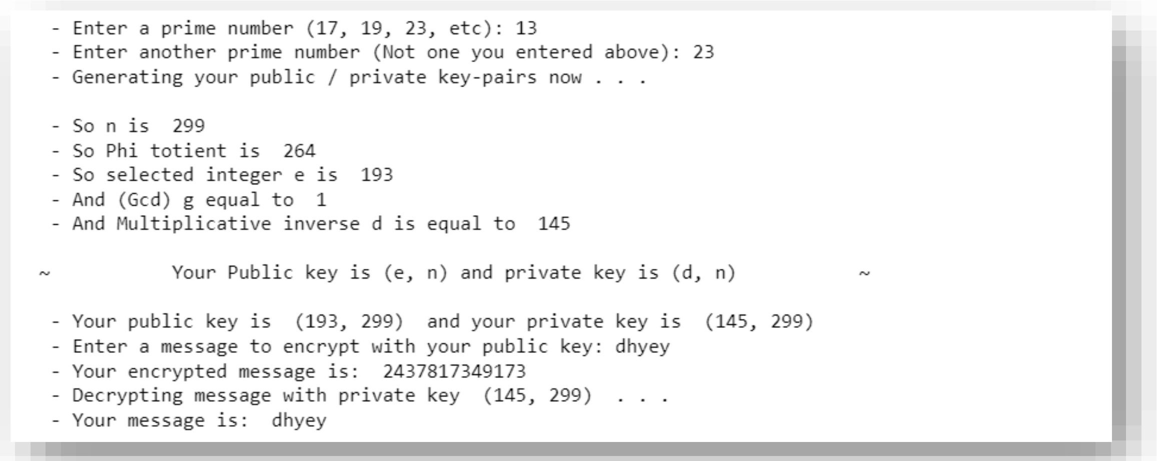


$P = C^d \; mod \, n.$

Source Code :

```
#RSA Algorithm
import random
'''
Euclid's algorithm for determining the greatest common divisor
Use iteration to make it faster for larger integers
'''
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

'''
Euclid's extended algorithm for finding the multiplicative inverse of two numbers
'''
def multiplicative_inverse(e, phi):
    d = 0
    x1 = 0
    x2 = 1
    y1 = 1
    temp_phi = phi

    while e > 0:
        temp1 = temp_phi//e
        temp2 = temp_phi - temp1 * e
        temp_phi = e
        e = temp2

        x = x2 - temp1 * x1
        y = d - temp1 * y1

        x2 = x1
        x1 = x
        d = y1
        y1 = y

    if temp_phi == 1:
        return d + phi
'''
Tests to see if a number is prime.
'''
def is_prime(num):
    if num == 2:
        return True
    if num < 2 or num % 2 == 0:
        return False
    for n in range(3, int(num**0.5)+2, 2):
        if num % n == 0:
            return False
    return True

def generate_key_pair(p, q):
    if not (is_prime(p) and is_prime(q)):
        raise ValueError('Both numbers must be prime.')
    elif p == q:
        raise ValueError('p and q cannot be equal')
    print(" ")
    # step :1  n = pq
    n = p * q
    print(" - So n is ",n)

    # step :2 Phi is the totient of n
    phi = (p-1) * (q-1)
    print(" - So Phi totient is ",phi)

    # step 3: Choose an integer e such that e and phi(n) are coprime
    e = random.randrange(1, phi)

    # Checking Using Euclid's Algorithm to verify that e and phi(n) are coprime
    g = gcd(e, phi)
    while g != 1:
        e = random.randrange(1, phi)
        g = gcd(e, phi)

    print(" - So selected integer e is ",e)
    print(" - And (Gcd) g equal to ",g)



    # step 4: Use Extended Euclid's Algorithm to generate the private key
    d = multiplicative_inverse(e, phi)
    print(" - And Multiplicative inverse d is equal to ",d)

    # Return public and private key_pair
    # Public key is (e, n) and private key is (d, n)
    print(" ")
    print("~          Your Public key is (e, n) and private key is (d, n)          ~")
    print(" ")
    return ((e, n), (d, n))

def encrypt(pk, plaintext):
    # Unpack the key into it's components
    key, n = pk
    # Convert each letter in the plaintext to numbers based on the character using a^b mod m
    cipher = [pow(ord(char), key, n) for char in plaintext]
    # Return the array of bytes
    return cipher

def decrypt(pk, ciphertext):
    # Unpack the key into its components
    key, n = pk
    # Generate the plaintext based on the ciphertext and key using a^b mod m
    aux = [str(pow(char, key, n)) for char in ciphertext]
    # Return the array of bytes as a string
    plain = [chr(int(char2)) for char2 in aux]
    return ''.join(plain)

if __name__ == '__main__':
    p = int(input(" - Enter a prime number (17, 19, 23, etc): "))
    q = int(input(" - Enter another prime number (Not one you entered above): "))

    print(" - Generating your public / private key-pairs now . . .")

    public, private = generate_key_pair(p, q)

    print(" - Your public key is ", public, " and your private key is ", private)

    message = input(" - Enter a message to encrypt with your public key: ")
    encrypted_msg = encrypt(public, message)

    print(" - Your encrypted message is: ", ''.join(map(lambda x: str(x), encrypted_msg)))

    print(" - Decrypting message with private key ", private, " . . .")

    print(" - Your message is: ", decrypt(private, encrypted_msg))
```

Output Screenshot:



```
  - Enter a prime number (17, 19, 23, etc): 13
  - Enter another prime number (Not one you entered above): 23
  - Generating your public / private key-pairs now . . .

  - So n is  299
  - So Phi totient is  264
  - So selected integer e is  193
  - And (Gcd) g equal to  1
  - And Multiplicative inverse d is equal to  145

 ~           Your Public key is (e, n) and private key is (d, n)          ~

  - Your public key is  (193, 299)  and your private key is  (145, 299)
  - Enter a message to encrypt with your public key: dhyey
  - Your encrypted message is:  2437817349173
  - Decrypting message with private key  (145, 299)  . . .
  - Your message is:  dhyey
```

CONCLUSION:

From this practical, we learnt the concept of encryption & decryption using RSA Algorithm and also learnt how to implement it as shown in the above codes.

Practical 5

Aim: The transmission of information needs to be secure over the communication channel and the data has to be confidential. Study and implement the practical approach for Steganography.

THEORY:

Steganography is the practice of hiding a secret message inside of (or even on top of) something that is not secret.

The purpose of steganography is to conceal and deceive.

It is a form of covert communication and can involve the use of any medium to hide messages.

It’s not a form of cryptography, because it doesn’t involve scrambling data or using a key. Instead, it is a form of data hiding and can be executed in clever ways, where cryptography is a science that largely enables privacy, steganography is a practice that enables secrecy – and deceit.

Using DOS commands:

For example, think we have to hide a text file Message.txt with the image file Modi.jpg and combine them in a new file as result.jpg, where result.jpg is our output file which contains the text hidden in the image file.

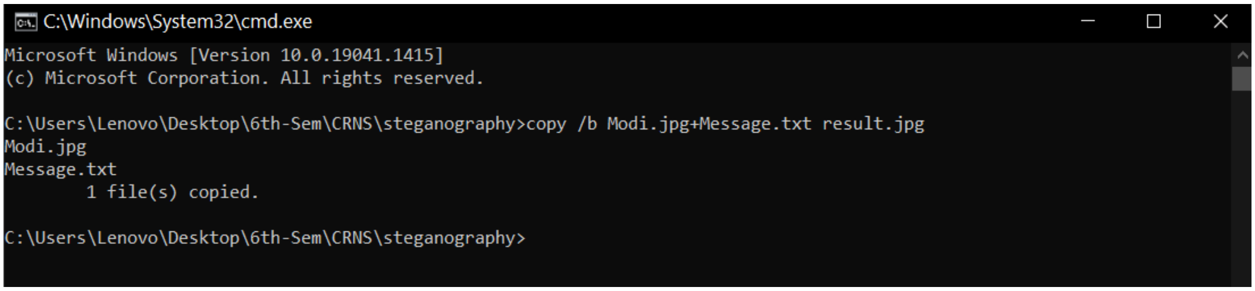
- Open Run command window by pressing win + r .

- Open command prompt by typing cmd and press ok
- Enter the directory where you have your files.

Then type the command:

copy /b Modi.jpg + Message.txt result.jpg

Now our file is successfully hidden in a new file.



C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19041.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo\Desktop\6th-Sem\CRNS\steganography>copy /b Modi.jpg+Message.txt result.jpg
Modi.jpg
Message.txt
        1 file(s) copied.

C:\Users\Lenovo\Desktop\6th-Sem\CRNS\steganography>
```

When you normally open the output file result.jpg you will see the image file but how to view your hidden text file Right click the image file.

Select open with notepad. Successfully opened! In the last of the notepad you'll find the content of the text file.

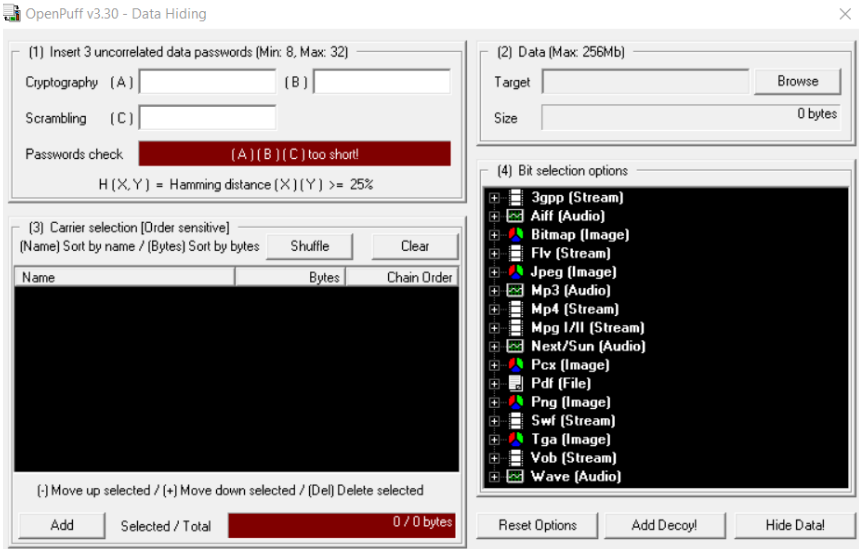
Using OpenPuff Tool:

OpenPuff is a professional steganography tool with unique features, suitable for highly sensitive data covert transmission.

OpenPuff is a prevailing data hiding application made easy, safe and free that allows you to hide data into encrypted files in order to send it to other users. This application aims to protect both secret messages and the persons who are exchanging messages.

Data Hiding:

Step 1: Open the OpenPuff. You see a configuration window when you click on the hide button in the main interface.

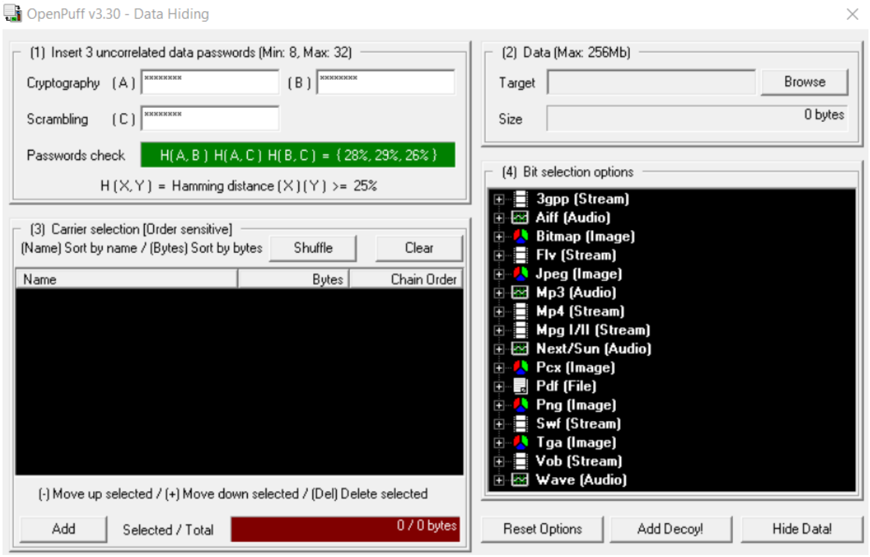


Passwords check: (A)(B)(C) too short!

Selected / Total: 0 / 0 bytes

Size: 0 bytes

Step 2: To specify a password under (1) with a minimum length of eight characters and a maximum length of 32 characters.



Cryptography (A) ******** (B) ********

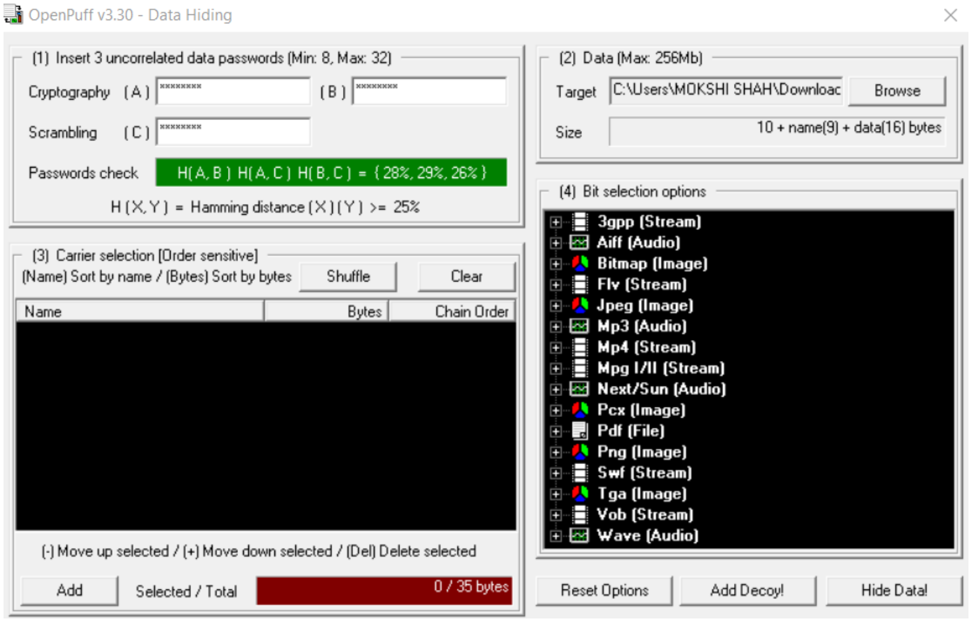
Scrambling (C) ********

Passwords check: H(A, B) H(A, C) H(B, C) = { 28%, 29%, 26% }

Selected / Total: 0 / 0 bytes

Size: 0 bytes

Step 3: A target file is selected under (2). The maximum size may not exceed 256 Megaybtes.


OpenPuff v3.30 - Data Hiding

(1) Insert 3 uncorrelated data passwords (Min: 8, Max: 32)

Cryptography (A) ******** (B) ********

Scrambling (C) ********

Passwords check: H(A, B) H(A, C) H(B, C) = {28%, 29%, 26%}

H(X, Y) = Hamming distance (X)(Y) >= 25%

(2) Data (Max: 256Mb)

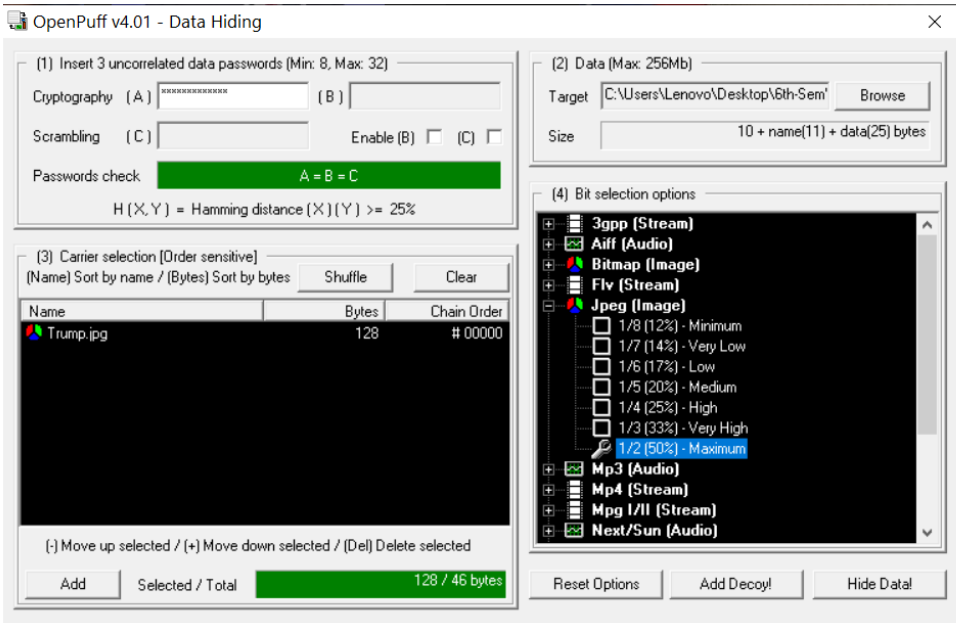
Target: C:\Users\MOKSHI SHAH\Downloac

Size: 10 + name(9) + data(16) bytes

(3) Carrier selection [Order sensitive]

Selected / Total: 0 / 35 bytes

Step 4: Once you have selected the target file you need to select one or multiple carriers under (3). Carriers are the files the data gets added to. The bytes added to each carrier file are displayed immediately after they have been added.


OpenPuff v4.01 - Data Hiding

(1) Insert 3 uncorrelated data passwords (Min: 8, Max: 32)

Cryptography (A) ************* (B)

Scrambling (C) Enable (B) □ (C) □

Passwords check: A = B = C

H(X, Y) = Hamming distance (X)(Y) >= 25%

(2) Data (Max: 256Mb)

Target: C:\Users\Lenovo\Desktop\6th-Sem'

Size: 10 + name(11) + data(25) bytes

(3) Carrier selection [Order sensitive]

Name	Bytes	Chain Order
Trump.jpg	128	# 00000

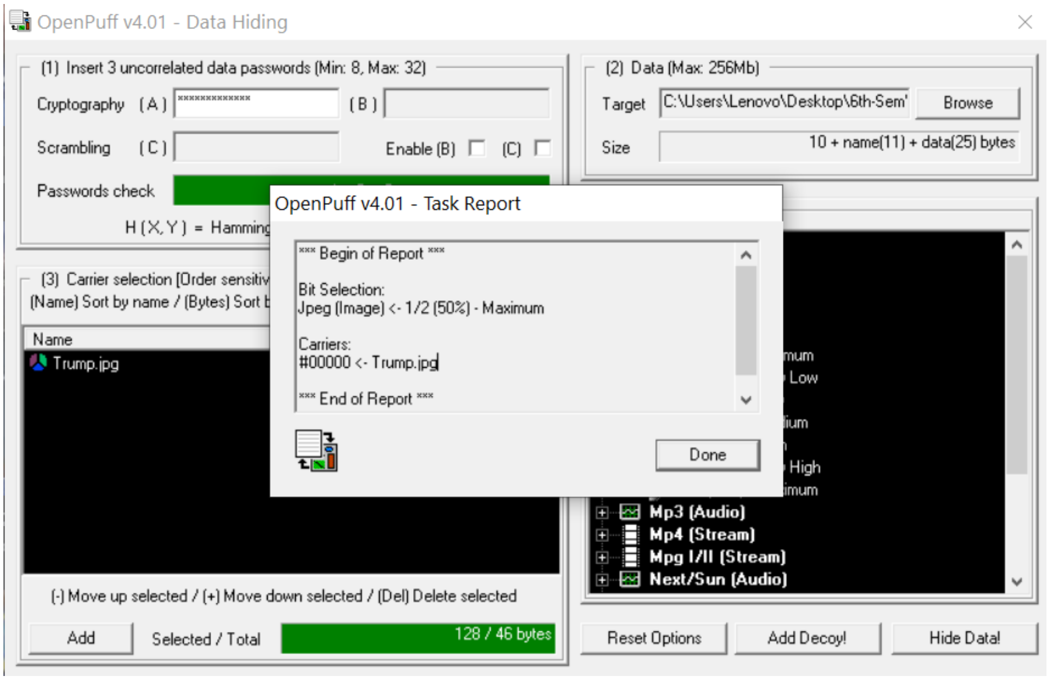
Selected / Total: 128 / 46 bytes

(4) Bit selection options

Jpeg (Image)
- □ 1/8 (12%) - Minimum
- □ 1/7 (14%) - Very Low
- □ 1/6 (17%) - Low
- □ 1/5 (20%) - Medium
- □ 1/4 (25%) - High
- □ 1/3 (33%) - Very High
- 1/2 (50%) - Maximum

You need to make sure that the carrier's available byte size exceeds the size of the selected file that you want to hide. For that, you can make changes to the bit selection screen (4).

Step 5: A click on Hide Data processes the files by adding the data of the selected file that you want to hide to all of them. A save window is displayed automatically to store the processed files in a different folder than the original files.



OpenPuff v4.01 - Data Hiding

(1) Insert 3 uncorrelated data passwords (Min: 8, Max: 32)

Cryptography (A) ************** (B)

Scrambling (C) Enable (B) □ (C) □

Passwords check

H (X, Y) = Hamming

(2) Data (Max: 256Mb)

Target C:\Users\Lenovo\Desktop\6th-Sem'

Size 10 + name(11) + data(25) bytes

(3) Carrier selection [Order sensitiv

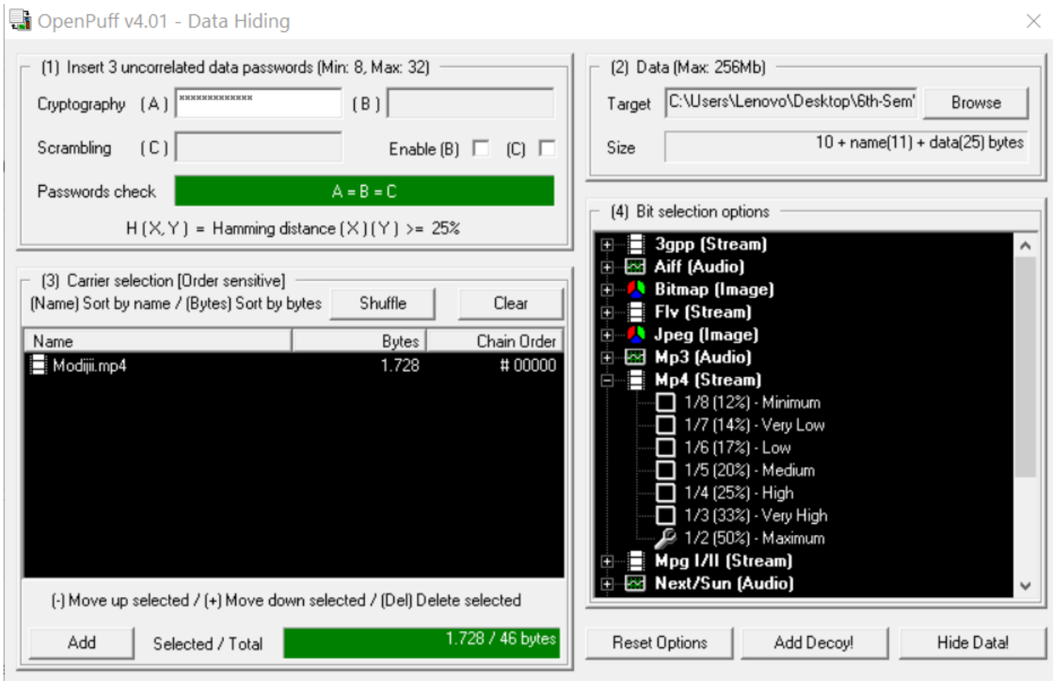
(Name) Sort by name / (Bytes) Sort b

Name
Trump.jpg

Selected / Total 128 / 46 bytes

OpenPuff v4.01 - Task Report

```
*** Begin of Report ***

Bit Selection:
Jpeg (Image) <- 1/2 (50%) - Maximum

Carriers:
#00000 <- Trump.jpg

*** End of Report ***
```



OpenPuff v4.01 - Data Hiding

(1) Insert 3 uncorrelated data passwords (Min: 8, Max: 32)

Cryptography (A) ************** (B)

Scrambling (C) Enable (B) □ (C) □

Passwords check A = B = C

H (X, Y) = Hamming distance (X) (Y) >= 25%

(2) Data (Max: 256Mb)

Target C:\Users\Lenovo\Desktop\6th-Sem'

Size 10 + name(11) + data(25) bytes

(3) Carrier selection [Order sensitive]

(Name) Sort by name / (Bytes) Sort by bytes

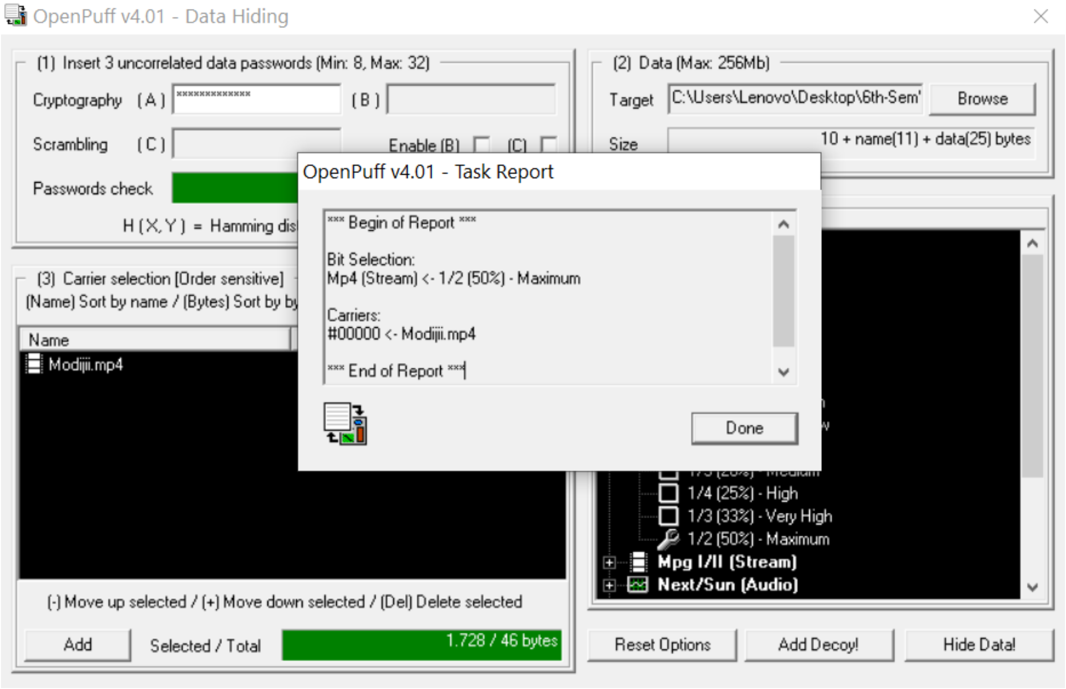
Name	Bytes	Chain Order
Modiji.mp4	1.728	# 00000

Selected / Total 1.728 / 46 bytes

(4) Bit selection options

- 3gpp (Stream)
- Aiff (Audio)
- Bitmap (Image)
- Flv (Stream)
- Jpeg (Image)
- Mp3 (Audio)
- Mp4 (Stream)
  - □ 1/8 (12%) - Minimum
  - □ 1/7 (14%) - Very Low
  - □ 1/6 (17%) - Low
  - □ 1/5 (20%) - Medium
  - □ 1/4 (25%) - High
  - □ 1/3 (33%) - Very High
  - 1/2 (50%) - Maximum
- Mpg I/II (Stream)
- Next/Sun (Audio)

Browse
Shuffle
Clear
(-) Move up selected / (+) Move down selected / (Del) Delete selected
Add
Reset Options
Add Decoy!
Hide Data!



OpenPuff v4.01 - Data Hiding

(1) Insert 3 uncorrelated data passwords (Min: 8, Max: 32)

Cryptography (A) *************

(2) Data (Max: 256Mb)

Target C:\Users\Lenovo\Desktop\6th-Sem'

Size 10 + name(11) + data(25) bytes

(3) Carrier selection [Order sensitive]

(Name) Sort by name / (Bytes) Sort by by

Name: Modiji.mp4

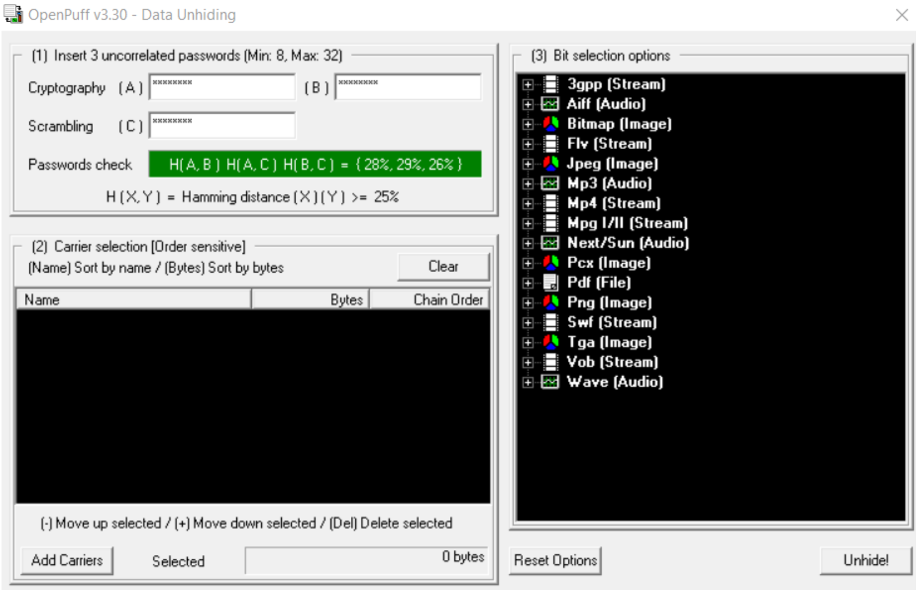
Selected / Total 1.728 / 46 bytes

OpenPuff v4.01 - Task Report

```
*** Begin of Report ***

Bit Selection:
Mp4 (Stream) <- 1/2 (50%) - Maximum

Carriers:
#00000 <- Modiji.mp4

*** End of Report ***
```

Data Unhiding:

Step 1: The unhide process basically reverses the process. You still need to enter the password that you have used to protect the data.


OpenPuff v3.30 - Data Unhiding
(1) Insert 3 uncorrelated passwords (Min: 8, Max: 32)
(2) Carrier selection [Order sensitive]
(Name) Sort by name / (Bytes) Sort by bytes
Clear
Name
Bytes
Chain Order
(-) Move up selected / (+) Move down selected / (Del) Delete selected
Add Carriers
(3) Bit selection options
3gpp (Stream)
Aiff (Audio)
Bitmap (Image)
Flv (Stream)
Jpeg (Image)
Mp3 (Audio)
Mp4 (Stream)
Mpg I/II (Stream)
Next/Sun (Audio)
Pcx (Image)
Pdf (File)
Png (Image)
Swf (Stream)
Tga (Image)
Vob (Stream)
Wave (Audio)
Reset Options
Unhide!

Cryptography (A) ******** (B) ********

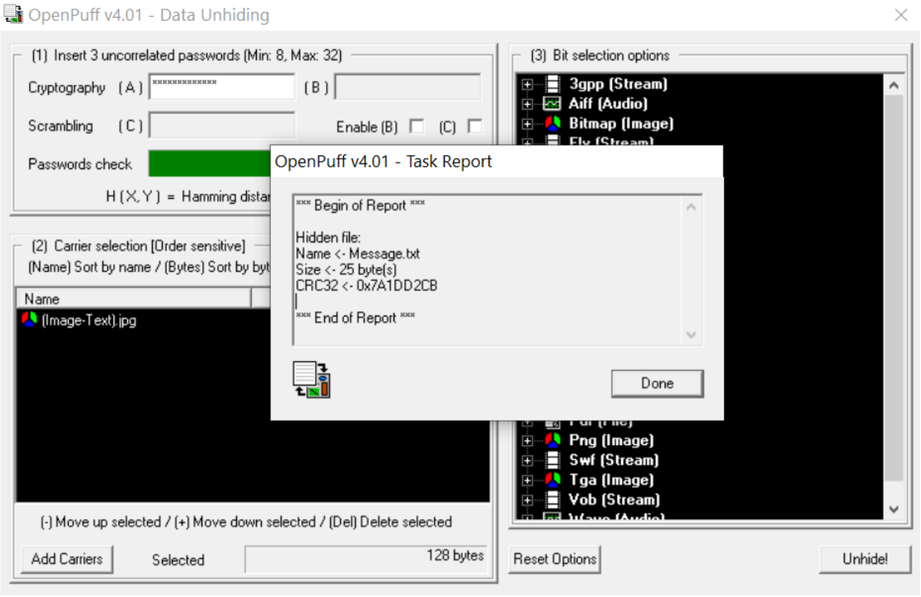
Scrambling (C) ********

Passwords check H(A, B) H(A, C) H(B, C) = (28%, 29%, 26%)

H (X, Y) = Hamming distance (X) (Y) >= 25%

Selected 0 bytes

Step 2: Select all carrier files and the bit selection that you have selected. You are then presented with a save as window to select a folder to save the hidden file to.


OpenPuff v4.01 - Data Unhiding
(1) Insert 3 uncorrelated passwords (Min: 8, Max: 32)
Enable (B) □ (C) □
(2) Carrier selection [Order sensitive]
(Name) Sort by name / (Bytes) Sort by byt
Name
(-) Move up selected / (+) Move down selected / (Del) Delete selected
Add Carriers
(3) Bit selection options
3gpp (Stream)
Aiff (Audio)
Bitmap (Image)
Flv (Stream)
Pdf (File)
Png (Image)
Swf (Stream)
Tga (Image)
Vob (Stream)
Wave (Audio)
Reset Options
Unhide!
OpenPuff v4.01 - Task Report
Done

Cryptography (A) ************* (B)

Scrambling (C)

Passwords check

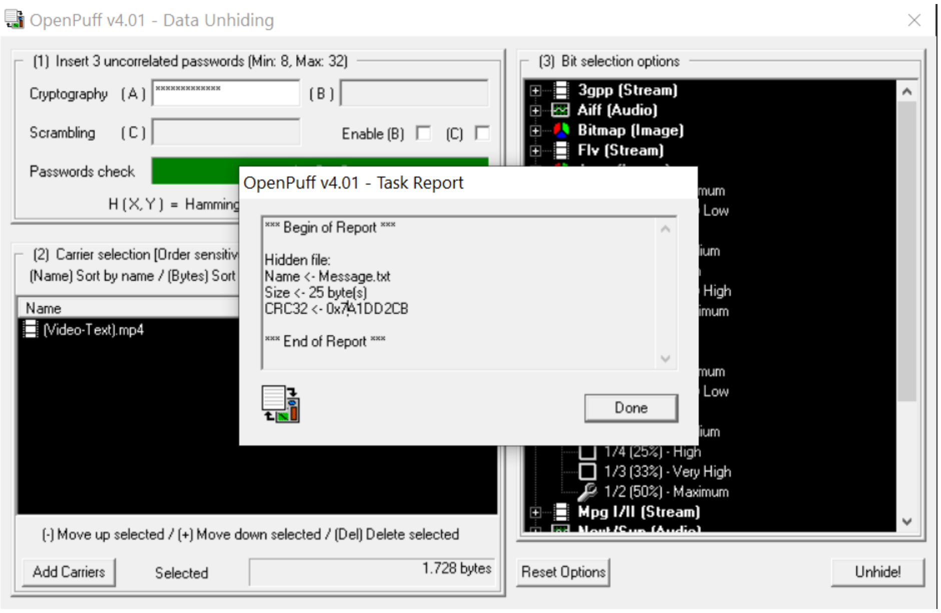
H (X, Y) = Hamming distan

(Image-Text).jpg

Selected 128 bytes

```
*** Begin of Report ***

Hidden file:
Name <- Message.txt
Size <- 25 byte(s)
CRC32 <- 0x7A1DD2CB

*** End of Report ***
```

Step 3: The data is successfully unhidden and the original file will be available in the folder you saved it.


OpenPuff v4.01 - Data Unhiding

(1) Insert 3 uncorrelated passwords (Min: 8, Max: 32)

Cryptography (A) ************* (B)

Scrambling (C) Enable (B) □ (C) □

Passwords check

(2) Carrier selection

Name

(Video-Text).mp4

Add Carriers Selected 1.728 bytes

(3) Bit selection options

OpenPuff v4.01 - Task Report

```
*** Begin of Report ***

Hidden file:
Name <- Message.txt
Size <- 25 byte(s)
CRC32 <- 0x7A1DD2CB

*** End of Report ***
```

Famous 6 Steganography Tools:

Tool	Description
Stegosuite	Hide text inside any image
Stegohide	Hide secret file in image or audio file.
Xiao Steganography	Free software that can be used to hide secret files in BMP images or in WAV files.
SSuite Picsel	Portable application to hide text inside image file
OpenPuff	Tool to conceal files in image, audio & flash files
Camouflage	Tool that lets you hide any type of file inside of file.

CONCLUSION:

From this practical, we learnt the concept of steganography and also learnt how to implement it with the help of command prompt as well as OpenPuff tool.

Aim: Implement GPG for windows.

THEORY:

The Windows installation has become significantly easier in the recent past. There is now a relatively intelligent installer that will do most of the work for you. This procedure is based on the documentation for Enigmail.

Note that if you have a previous installation of GnuPG that was done prior to the new installation, you should follow Enigmail's instructions for removing old GnuPG versions before proceeding.

- Download the Windows installer from the GnuPG Download page. As of this writing 1.4.2 is the latest version.
- Run the installer.
- Add the installation directory to your path so you may just type "gpg" from a command line rather than "C:\Program Files\GnuPG\gpg":
    - Go to Start -> Settings -> Control Panel -> System -> Advanced -> Environment Variables
    - Choose "Path", and select "Edit", and to the very end of the value add ;C:\Program Files\GnuPG (note the preceding semi-colon). Click OK until you're out of the System dialog box.
    - In order for this to take effect you must close any open command windows and start a new one.
- From a command line (Start -> Run -> type "cmd" -> OK), type gpg --version. If gnupg gives you this:

```
gpg (GnuPG) 1.4.1

   Copyright (C) 2005 Free Software Foundation, Inc.

   This program comes with ABSOLUTELY NO WARRANTY.

   This is free software, and you are welcome to redistribute it

   under certain conditions. See the file COPYING for details.

Home: C:/Documents and Settings/bporter/application Data/GnuPG

     Supported algorithms:

     Pubkey: RSA, RSA-E, RSA-S, ELG-E, DSA
```

```
Cipher: 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH

Hash: MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512

Compression: Uncompressed, ZIP, ZLIB, BZIP2
```

- Then you're all done. However, if it says:

```
'gpg' is not recognized as an internal or external command
```

- Then something went wrong. If that's the case, you can try typing in:

```
"C:\Program Files\GnuPG\gpg" --version
```

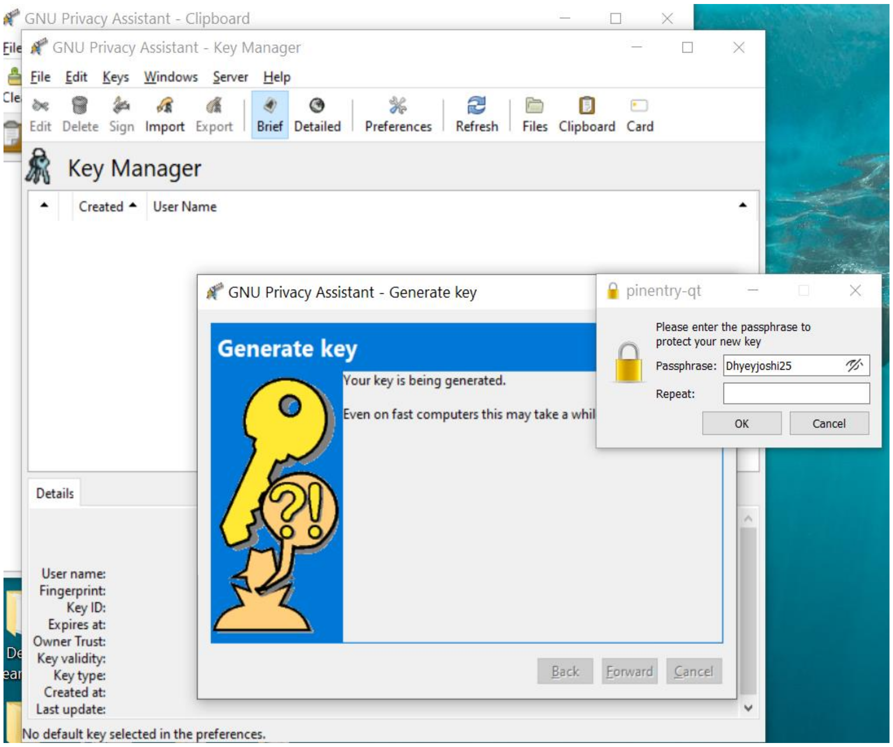
```
(note the quotes). If that works, you only messed up
changing your path, if it doesn't work, try re-installing.
If your path isn't working but you're sure you did it
right, try logging out and logging back in.
```

GPG

- GNU Privacy Guard is a free-software replacement for Symantec's PGP cryptographic software suite. The software is compliant with RFC 4880, the IETF standards-track specification of OpenPGP. Modern versions of PGP are interoperable with GnuPG and other OpenPGP-compliant systems.
- Gpg4win enables users to securely transport emails and files with the help of encryption and digital signatures. Encryption protects the contents against an unwanted party reading it.
- Digital signatures make sure that it was not modified and comes from a specific sender.
- Gpg4win supports both relevant cryptography standards, OpenPGP and S/MIME (X.509), and is the official GnuPG distribution for Windows. It is maintained by the developers of GnuPG. Gpg4win and the software included with Gpg4win are Free software.

PROCESS:

Step 1: After installing, enter email and create public and private key for sender.


GNU Privacy Assistant - Clipboard
GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Edit Delete Sign Import Export Brief Detailed Preferences Refresh Files Clipboard Card
Key Manager
Created User Name
Details
GNU Privacy Assistant - Generate key
Generate key
Back Forward Cancel
pinentry-qt
OK Cancel

Your key is being generated.

Even on fast computers this may take a whil

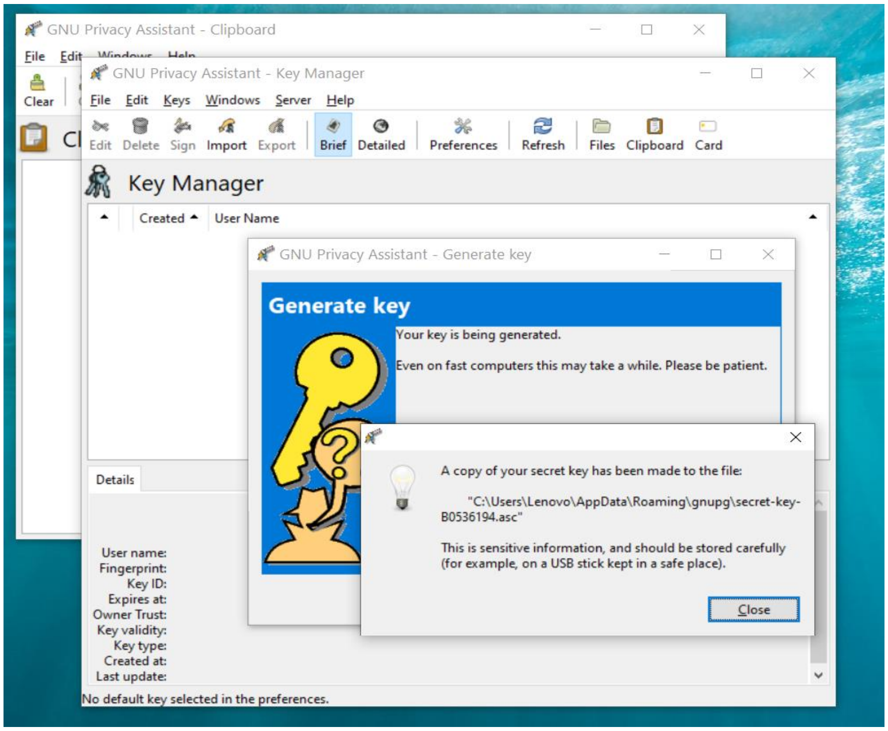
Please enter the passphrase to protect your new key

Passphrase: Dhyeyjoshi25

Repeat:

User name:
Fingerprint:
Key ID:
Expires at:
Owner Trust:
Key validity:
Key type:
Created at:
Last update:

No default key selected in the preferences.


GNU Privacy Assistant - Clipboard
File Edit Windows Help
Clear
GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Edit Delete Sign Import Export Brief Detailed Preferences Refresh Files Clipboard Card
Key Manager
Created User Name
Details
GNU Privacy Assistant - Generate key
Generate key
Close

Your key is being generated.

Even on fast computers this may take a while. Please be patient.

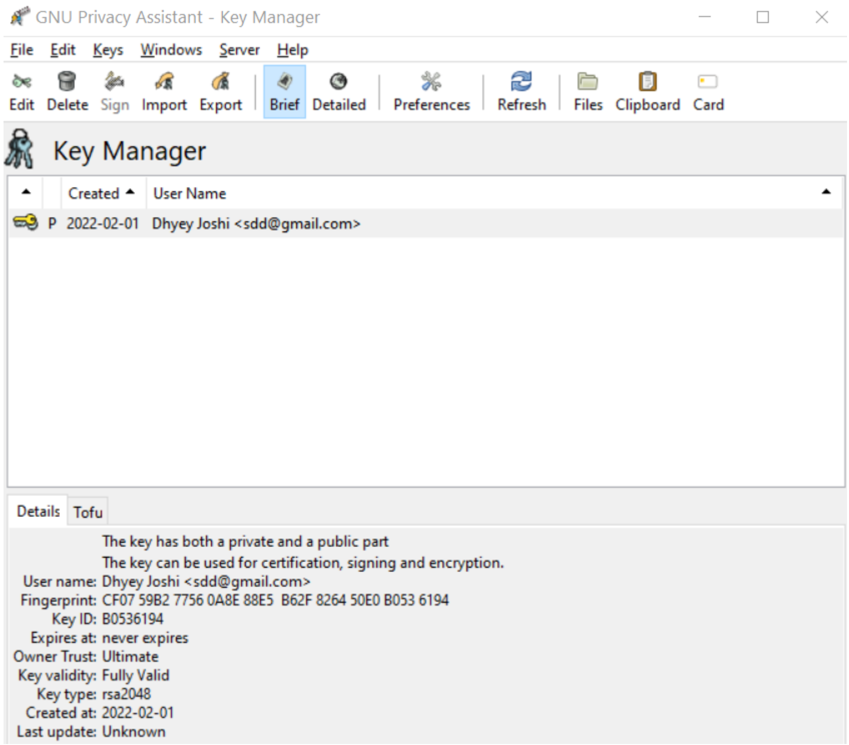
A copy of your secret key has been made to the file:

"C:\Users\Lenovo\AppData\Roaming\gnupg\secret-key-B0536194.asc"

This is sensitive information, and should be stored carefully (for example, on a USB stick kept in a safe place).

User name:
Fingerprint:
Key ID:
Expires at:
Owner Trust:
Key validity:
Key type:
Created at:
Last update:

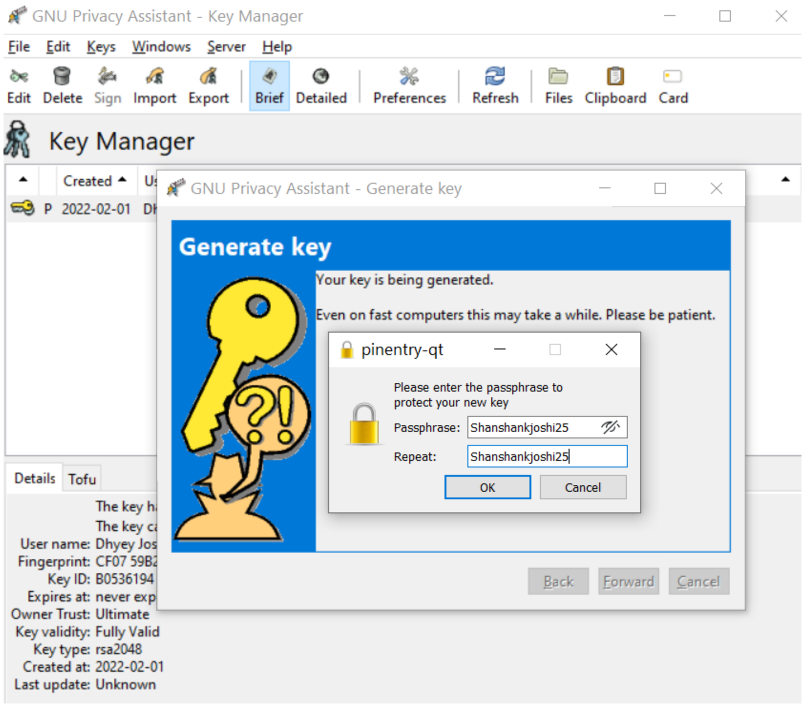
No default key selected in the preferences.


GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Edit Delete Sign Import Export Brief Detailed Preferences Refresh Files Clipboard Card
Key Manager
Created User Name
Details Tofu

P 2022-02-01 Dhyey Joshi <sdd@gmail.com>

The key has both a private and a public part
The key can be used for certification, signing and encryption.
User name: Dhyey Joshi <sdd@gmail.com>
Fingerprint: CF07 59B2 7756 0A8E 88E5 B62F 8264 50E0 B053 6194
Key ID: B0536194
Expires at: never expires
Owner Trust: Ultimate
Key validity: Fully Valid
Key type: rsa2048
Created at: 2022-02-01
Last update: Unknown

Step 2: Create Second user, enter email and create public and private key for receiver.


GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Edit Delete Sign Import Export Brief Detailed Preferences Refresh Files Clipboard Card
Key Manager
Created
Details Tofu
GNU Privacy Assistant - Generate key
Generate key
pinentry-qt
OK
Cancel
Back
Forward
Cancel

P 2022-02-01 Dh

Your key is being generated.

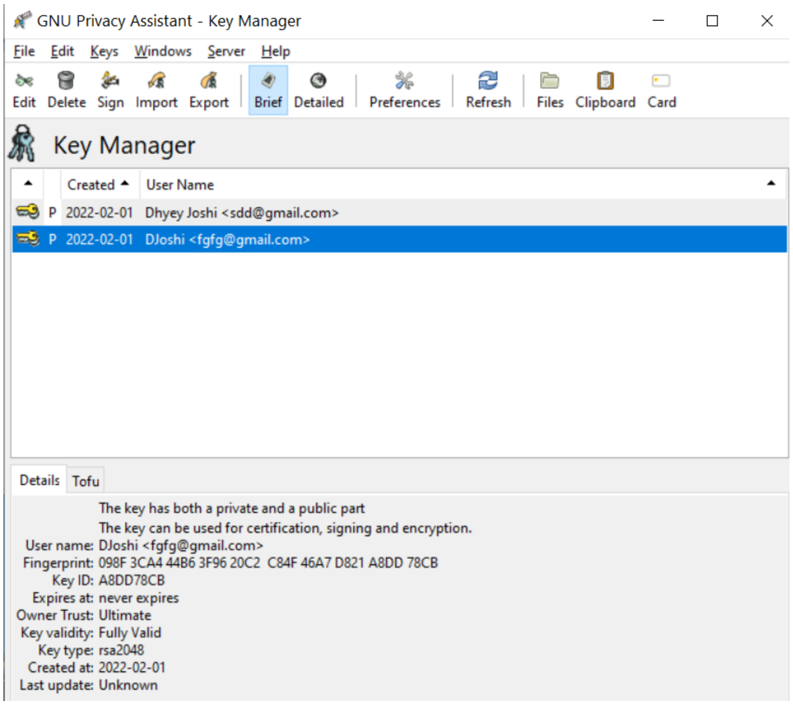
Even on fast computers this may take a while. Please be patient.

Please enter the passphrase to protect your new key

Passphrase: Shanshankjoshi25

Repeat: Shanshankjoshi25

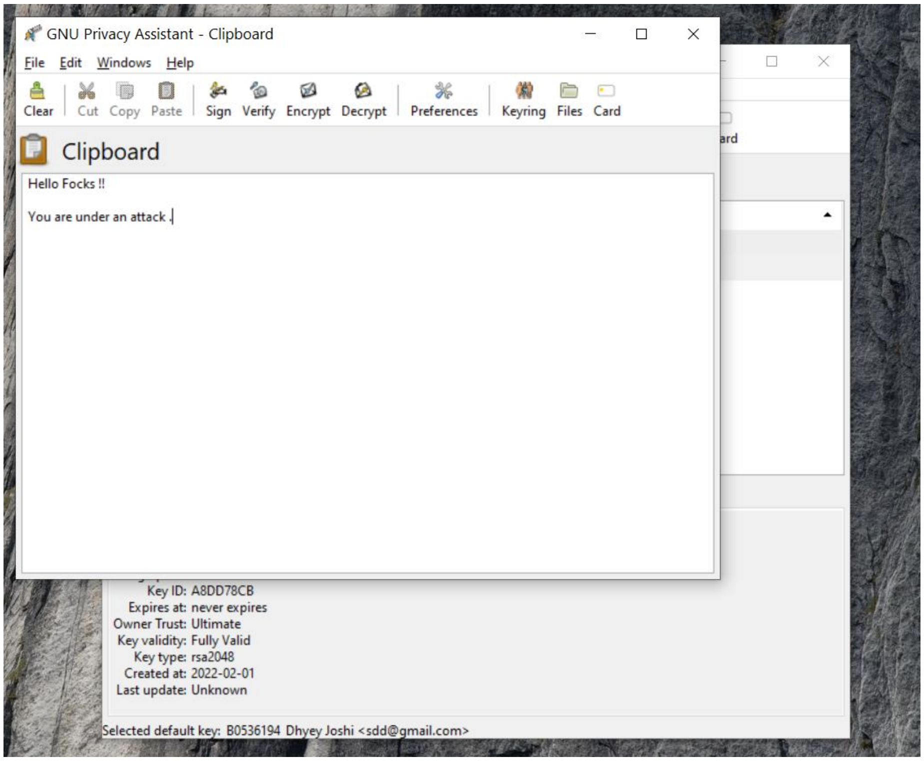
The key h
The key c
User name: Dhyey Jos
Fingerprint: CF07 59B2
Key ID: B0536194
Expires at: never exp
Owner Trust: Ultimate
Key validity: Fully Valid
Key type: rsa2048
Created at: 2022-02-01
Last update: Unknown


GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Edit Delete Sign Import Export Brief Detailed Preferences Refresh Files Clipboard Card
Key Manager
Created User Name
Details Tofu

P 2022-02-01 Dhyey Joshi <sdd@gmail.com>
P 2022-02-01 DJoshi <fgfg@gmail.com>

The key has both a private and a public part
The key can be used for certification, signing and encryption.
User name: DJoshi <fgfg@gmail.com>
Fingerprint: 098F 3CA4 44B6 3F96 20C2 C84F 46A7 D821 A8DD 78CB
Key ID: A8DD78CB
Expires at: never expires
Owner Trust: Ultimate
Key validity: Fully Valid
Key type: rsa2048
Created at: 2022-02-01
Last update: Unknown

Step 3: Then go to clipboard to encrypt the written message:

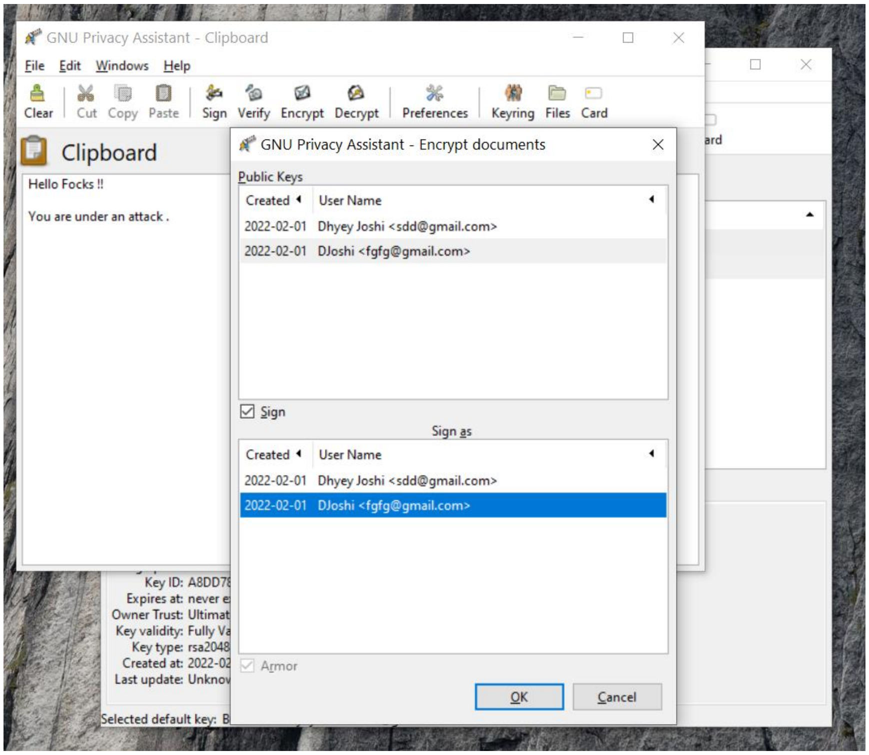

GNU Privacy Assistant - Clipboard
File Edit Windows Help
Clear Cut Copy Paste Sign Verify Encrypt Decrypt Preferences Keyring Files Card
Clipboard

Hello Focks !!

You are under an attack .

Key ID: A8DD78CB
Expires at: never expires
Owner Trust: Ultimate
Key validity: Fully Valid
Key type: rsa2048
Created at: 2022-02-01
Last update: Unknown

Selected default key: B0536194 Dhyey Joshi <sdd@gmail.com>


GNU Privacy Assistant - Clipboard
File Edit Windows Help
Clear Cut Copy Paste Sign Verify Encrypt Decrypt Preferences Keyring Files Card
Clipboard
GNU Privacy Assistant - Encrypt documents
OK
Cancel

Hello Focks !!

You are under an attack .

Public Keys

Created	User Name
2022-02-01	Dhyey Joshi <sdd@gmail.com>
2022-02-01	DJoshi <fgfg@gmail.com>

☑ Sign

Sign as

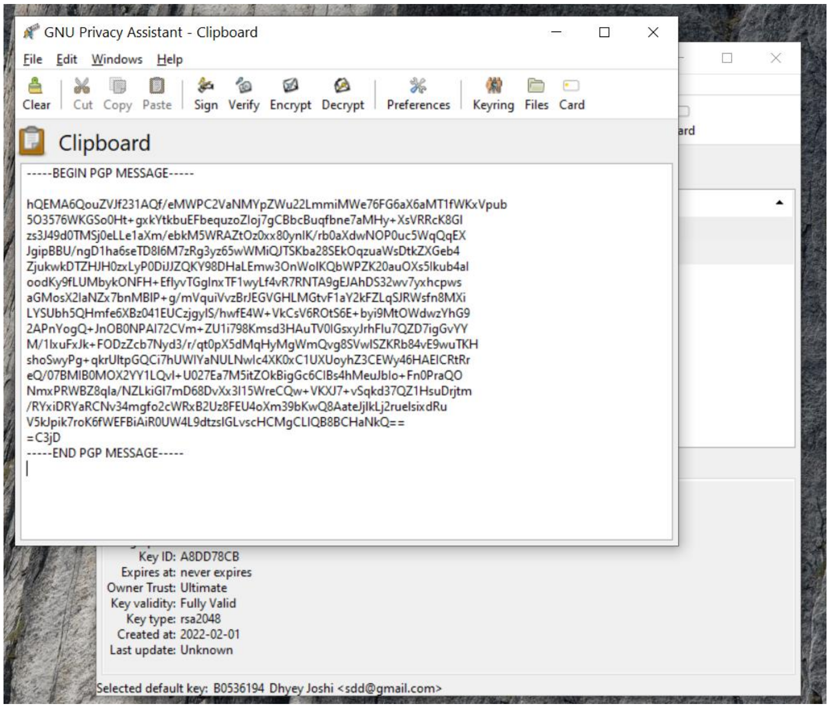
Created	User Name
2022-02-01	Dhyey Joshi <sdd@gmail.com>
2022-02-01	DJoshi <fgfg@gmail.com>

☑ Armor

Key ID: A8DD7
Expires at: never e
Owner Trust: Ultimat
Key validity: Fully Va
Key type: rsa2048
Created at: 2022-02
Last update: Unknov

Selected default key: B

Now enter the public key of the receiver and encrypt the message:

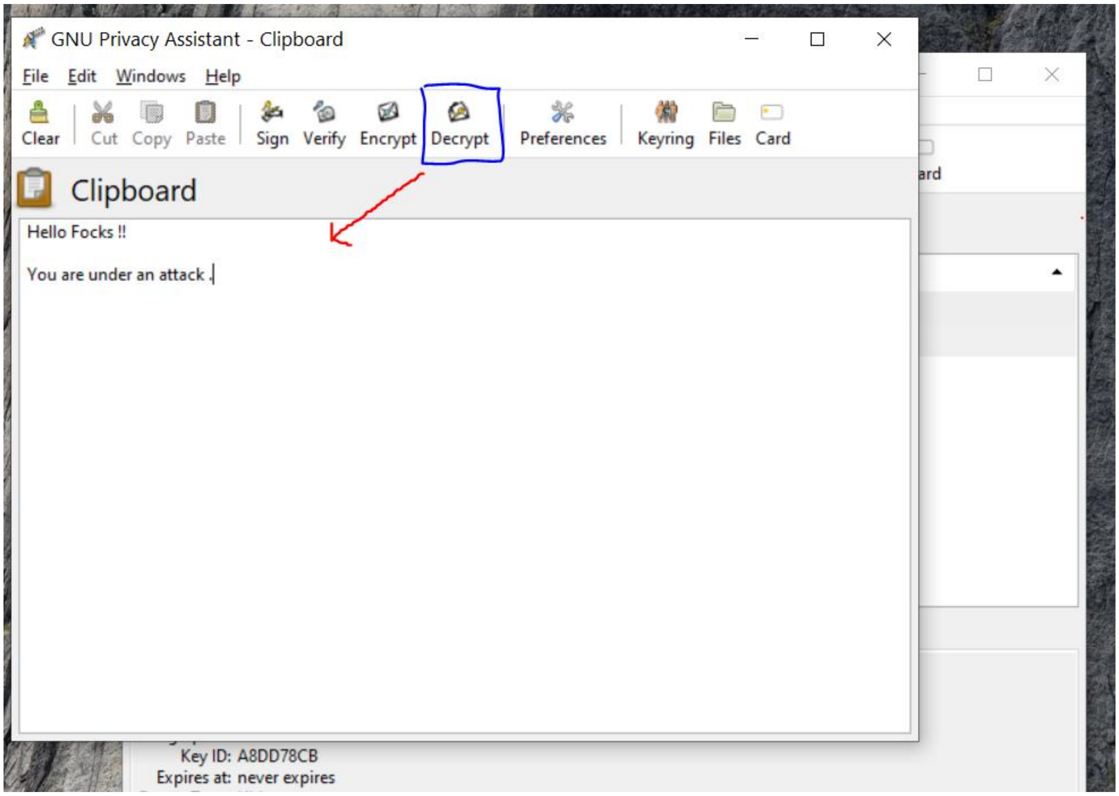

GNU Privacy Assistant - Clipboard
File Edit Windows Help
Clear Cut Copy Paste Sign Verify Encrypt Decrypt Preferences Keyring Files Card
Clipboard

```
-----BEGIN PGP MESSAGE-----

hQEMA6QouZVJf231AQf/eMWPC2VaNMYpZWu22LmmiMWe76FG6aX6aMT1fWKxVpub
5O3576WKGSo0Ht+gxkYtkbuEFbequzoZloj7gCBbcBuqfbne7aMHy+XsVRRcK8Gl
zs3J49d0TMSj0eLLe1aXm/ebkM5WRAZtOz0xx80ynlK/rb0aXdwNOP0uc5WqQqEX
JgipBBU/ngD1ha6seTD8l6M7zRg3yz65wWMiQJTSKba28SEkOqzuaWsDtkZXGeb4
ZjukwkDTZHJH0zxLyP0DiJJZQKY98DHaLEmw3OnWolKQbWPZK20auOXs5lkub4al
oodKy9fLUMbykONFH+EflyvTGglnxTF1wyLf4vR7RNTA9gEJAhDS32wv7yxhcpws
aGMosX2laNZx7bnMBIP+g/mVquiVvzBrJEGVGHLMGtvF1aY2kFZLqSJRWsfn8MXi
LYSUbh5QHmfe6XBz041EUCzjgylS/hwfE4W+VkCsV6ROtS6E+byi9MtOWdwzYhG9
2APnYogQ+JnOB0NPAI72CVm+ZU1i798Kmsd3HAuTV0lGsxyJrhFlu7QZD7igGvYY
M/1lxuFxJk+FODzZcb7Nyd3/r/qt0pX5dMqHyMgWmQvg8SVwlSZKRb84vE9wuTKH
shoSwyPg+qkrUltpGQCi7hUWlYaNULNwlc4XK0xC1UXUoyhZ3CEWy46HAElCRtRr
eQ/07BMIB0MOX2YY1LQvl+U027Ea7M5itZOkBigGc6ClBs4hMeuJblo+Fn0PraQO
NmxPRWBZ8qla/NZLkiGl7mD68DvXx3l15WreCQw+VKXJ7+vSqkd37QZ1HsuDrjtm
/RYxiDRYaRCNv34mgfo2cWRxB2Uz8FEU4oXm39bKwQ8AateJjlkLj2ruelsixdRu
V5kJpik7roK6fWEFBiAiR0UW4L9dtzslGLvscHCMgCLIQB8BCHaNkQ==
=C3jD
-----END PGP MESSAGE-----
```

Key ID: A8DD78CB
Expires at: never expires
Owner Trust: Ultimate
Key validity: Fully Valid
Key type: rsa2048
Created at: 2022-02-01
Last update: Unknown

Selected default key: B0536194 Dhyey Joshi <sdd@gmail.com>

Now to decrypt the message just click on decryption:



Hello Focks !!

You are under an attack .

Key ID: A8DD78CB
Expires at: never expires

CONCLUSION:

From this practical, we learnt the concept of GNU Privacy Guard and also learnt how to implement it with the help of Gpg4win software.

Aim: Perform port scanning using nmap on a single port and capture the packets using wireshark and analyze the output.

THEORY:

1. Nmap –

o Nmap can adapt to network conditions including latency and congestion during a scan.

o Nmap started as a Linux utility and was ported to other systems including Windows, macOS, and BSD. It is most popular on Linux, followed by Windows.

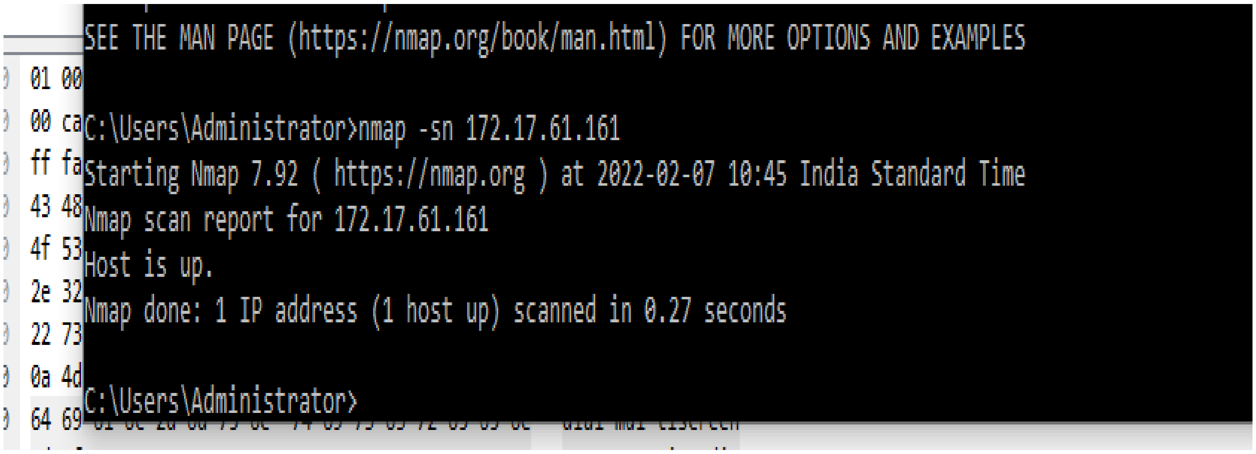
2. Wireshark –

o Wireshark is a free and open-source packet analyser.

o Wireshark is the world’s foremost and widely-used network protocol analyser. It lets you see what’s happening on your network at a microscopic level and is the de facto (and often de jure) standard across many commercial and non-profit enterprises, government agencies, and educational institutions.

PROCESS:

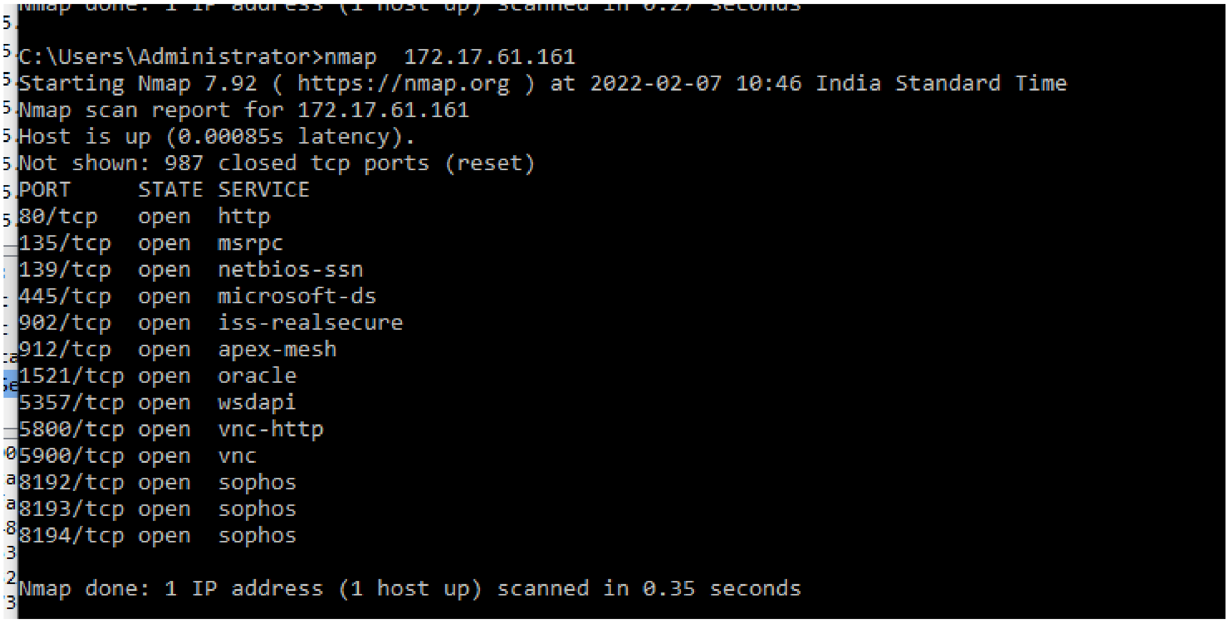
1. Host discovery / ping scan
   o Ping scan in nmap is done to check if the target host is alive or not.
   o Command – nmap –sn ip



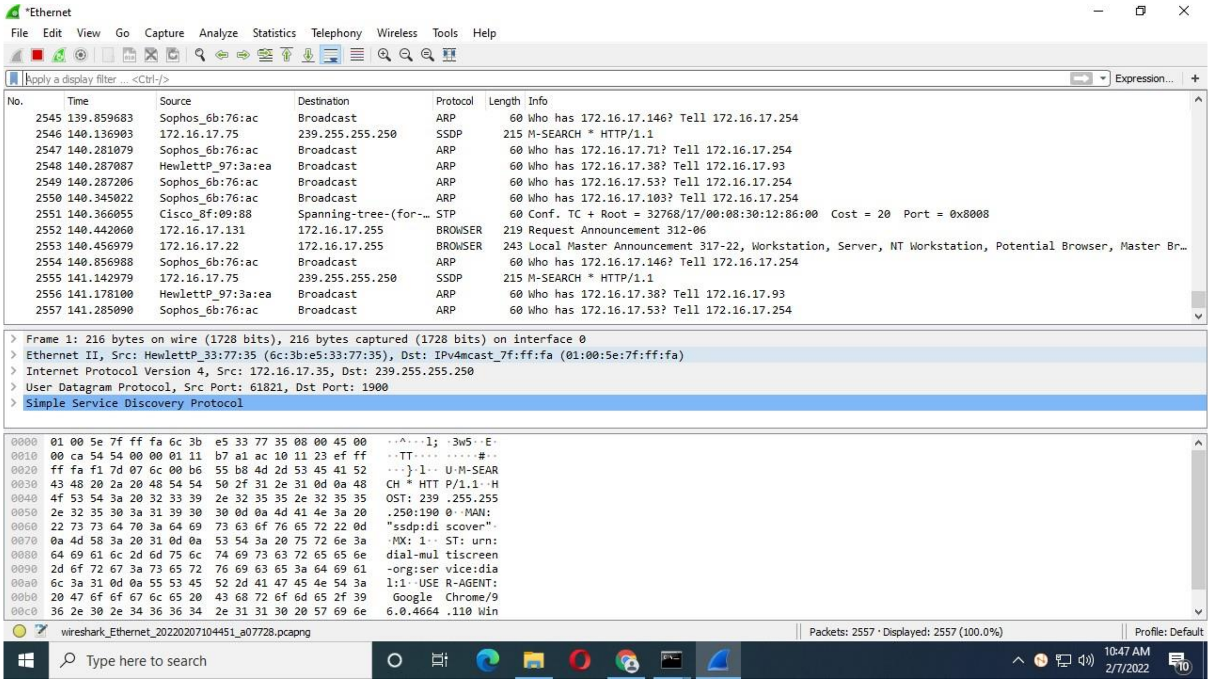
```
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES

C:\Users\Administrator>nmap -sn 172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:45 India Standard Time
Nmap scan report for 172.17.61.161
Host is up.
Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds

C:\Users\Administrator>
```

2. Default Scan -
   nmap ip o
   Command –

nmap ip



```
Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds

C:\Users\Administrator>nmap  172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:46 India Standard Time
Nmap scan report for 172.17.61.161
Host is up (0.00085s latency).
Not shown: 987 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
1521/tcp open  oracle
5357/tcp open  wsdapi
5800/tcp open  vnc-http
5900/tcp open  vnc
8192/tcp open  sophos
8193/tcp open  sophos
8194/tcp open  sophos

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

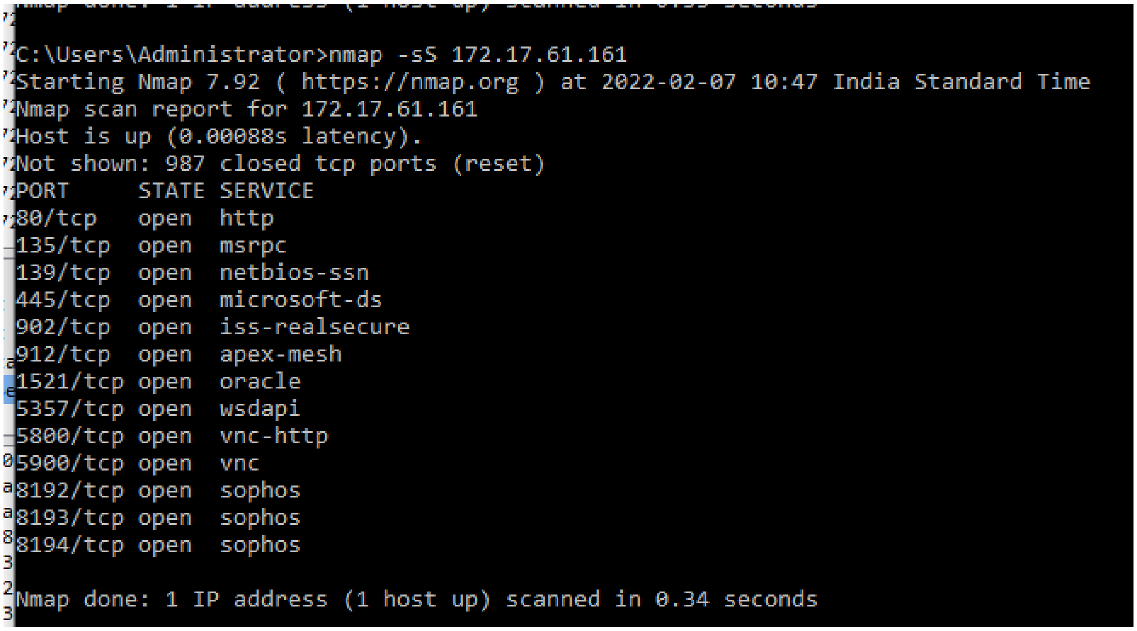


```
No.  Time        Source             Destination          Protocol Length Info
2545 139.859683  Sophos_6b:76:ac    Broadcast            ARP       60 Who has 172.16.17.146? Tell 172.16.17.254
2546 140.136903  172.16.17.75       239.255.255.250      SSDP     215 M-SEARCH * HTTP/1.1
2547 140.281079  Sophos_6b:76:ac    Broadcast            ARP       60 Who has 172.16.17.71? Tell 172.16.17.254
2548 140.287087  HewlettP_97:3a:ea  Broadcast            ARP       60 Who has 172.16.17.38? Tell 172.16.17.93
2549 140.287206  Sophos_6b:76:ac    Broadcast            ARP       60 Who has 172.16.17.53? Tell 172.16.17.254
2550 140.345022  Sophos_6b:76:ac    Broadcast            ARP       60 Who has 172.16.17.103? Tell 172.16.17.254
2551 140.366055  Cisco_8f:09:88     Spanning-tree-(for-… STP       60 Conf. TC + Root = 32768/17/00:08:30:12:86:00  Cost = 20  Port = 0x8008
2552 140.442060  172.16.17.131      172.16.17.255        BROWSER  219 Request Announcement 312-06
2553 140.456979  172.16.17.22       172.16.17.255        BROWSER  243 Local Master Announcement 317-22, Workstation, Server, NT Workstation, Potential Browser, Master Br…
2554 140.856988  Sophos_6b:76:ac    Broadcast            ARP       60 Who has 172.16.17.146? Tell 172.16.17.254
2555 141.142979  172.16.17.75       239.255.255.250      SSDP     215 M-SEARCH * HTTP/1.1
2556 141.178100  HewlettP_97:3a:ea  Broadcast            ARP       60 Who has 172.16.17.38? Tell 172.16.17.93
2557 141.285090  Sophos_6b:76:ac    Broadcast            ARP       60 Who has 172.16.17.53? Tell 172.16.17.254

> Frame 1: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits) on interface 0
> Ethernet II, Src: HewlettP_33:77:35 (6c:3b:e5:33:77:35), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
> Internet Protocol Version 4, Src: 172.16.17.35, Dst: 239.255.255.250
> User Datagram Protocol, Src Port: 61821, Dst Port: 1900
> Simple Service Discovery Protocol

0000  01 00 5e 7f ff fa 6c 3b  e5 33 77 35 08 00 45 00   ··^···l; ·3w5··E·
0010  00 ca 54 54 00 00 01 11  b7 a1 ac 10 11 23 ef ff   ··TT···· ·····#··
0020  ff fa f1 7d 07 6c 00 b6  55 b8 4d 2d 53 45 41 52   ···}·l·· U·M-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1··H
0040  4f 53 54 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   OST: 239 .255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 4d 41 4e 3a 20   .250:190 0··MAN:
0060  22 73 73 64 70 3a 64 69  73 63 6f 76 65 72 22 0d   "ssdp:di scover"·
0070  0a 4d 58 3a 20 31 0d 0a  53 54 3a 20 75 72 6e 3a   ·MX: 1·· ST: urn:
0080  64 69 61 6c 2d 6d 75 6c  74 69 73 63 72 65 65 6e   dial-mul tiscreen
0090  2d 6f 72 67 3a 73 65 72  76 69 63 65 3a 64 69 61   -org:ser vice:dia
00a0  6c 3a 31 0d 0a 55 53 45  52 2d 41 47 45 4e 54 3a   l:1··USE R-AGENT:
00b0  20 47 6f 6f 67 6c 65 20  43 68 72 6f 6d 65 2f 39    Google  Chrome/9
00c0  36 2e 30 2e 34 36 36 34  2e 31 31 30 20 57 69 6e   6.0.4664 .110 Win

wireshark_Ethernet_20220207104451_a07728.pcapng    Packets: 2557 · Displayed: 2557 (100.0%)    Profile: Default
```

3. TCP SYN Scan

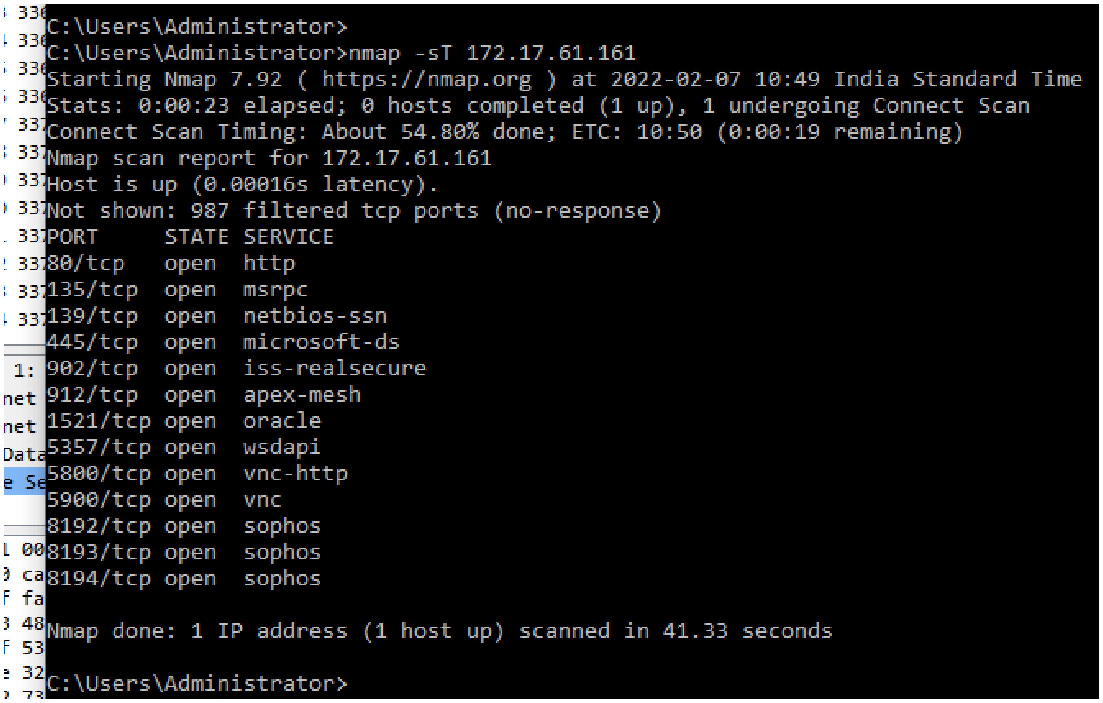
   o SYN scan is the default and most popular scan option for good reasons. It can be performed quickly, scanning thousands of ports

per second on a fast network not hampered by restrictive firewalls.
o Command: nmap -sS ip



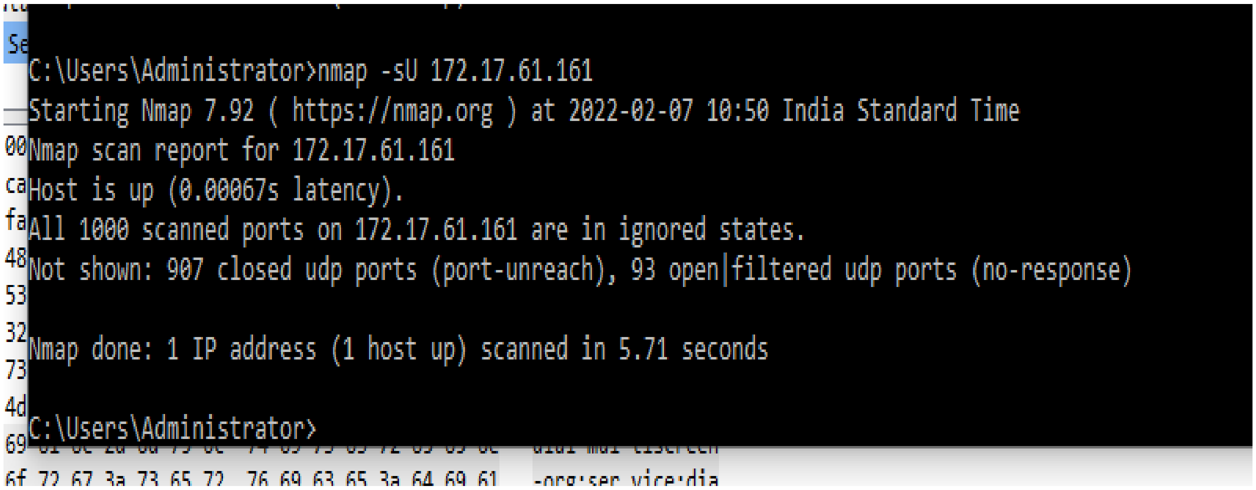
```
Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds

C:\Users\Administrator>nmap -sS 172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:47 India Standard Time
Nmap scan report for 172.17.61.161
Host is up (0.00088s latency).
Not shown: 987 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
1521/tcp open  oracle
5357/tcp open  wsdapi
5800/tcp open  vnc-http
5900/tcp open  vnc
8192/tcp open  sophos
8193/tcp open  sophos
8194/tcp open  sophos

Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
```

4. TCP Connect Scan o TCP connect scan is the default TCP scan type when SYN scan is not an option. This is the case when a user does not have raw packet privileges.
   o Instead of writing raw packets as most other scan types do, Nmap asks the underlying operating system to establish a connection with the target machine
   o Command: nmap -sT ip



```
C:\Users\Administrator>
C:\Users\Administrator>nmap -sT 172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:49 India Standard Time
Stats: 0:00:23 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 54.80% done; ETC: 10:50 (0:00:19 remaining)
Nmap scan report for 172.17.61.161
Host is up (0.00016s latency).
Not shown: 987 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
1521/tcp open  oracle
5357/tcp open  wsdapi
5800/tcp open  vnc-http
5900/tcp open  vnc
8192/tcp open  sophos
8193/tcp open  sophos
8194/tcp open  sophos

Nmap done: 1 IP address (1 host up) scanned in 41.33 seconds

C:\Users\Administrator>
```

5. UDP Scan o While most popular services on the Internet run over the TCP protocol, UDP services are widely deployed. o DNS, SNMP, and DHCP (registered ports 53, 161/162, and 67/68) are three of the most common.
   o Command: nmap -sU ip

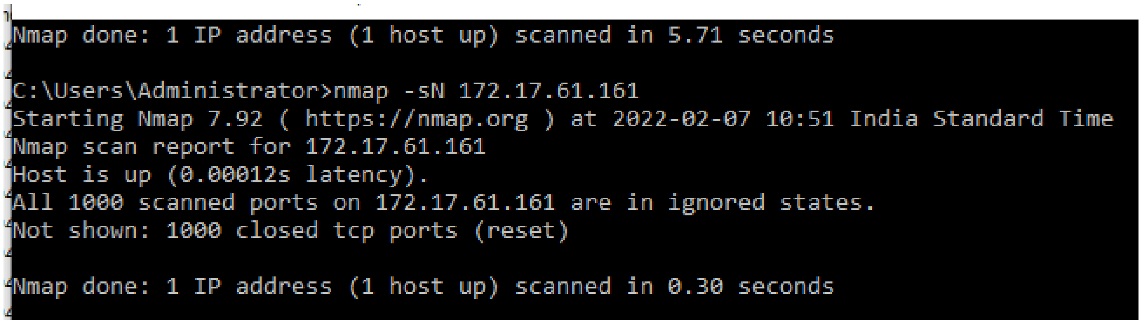


```
C:\Users\Administrator>nmap -sU 172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:50 India Standard Time
Nmap scan report for 172.17.61.161
Host is up (0.00067s latency).
All 1000 scanned ports on 172.17.61.161 are in ignored states.
Not shown: 907 closed udp ports (port-unreach), 93 open|filtered udp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 5.71 seconds

C:\Users\Administrator>
```

6. TCP null, TCP Xmas, TCP FIN o These three scan types (even more are possible with the --scanflags option described in the next section) exploit a subtle loophole in the TCP RFC to differentiate between open

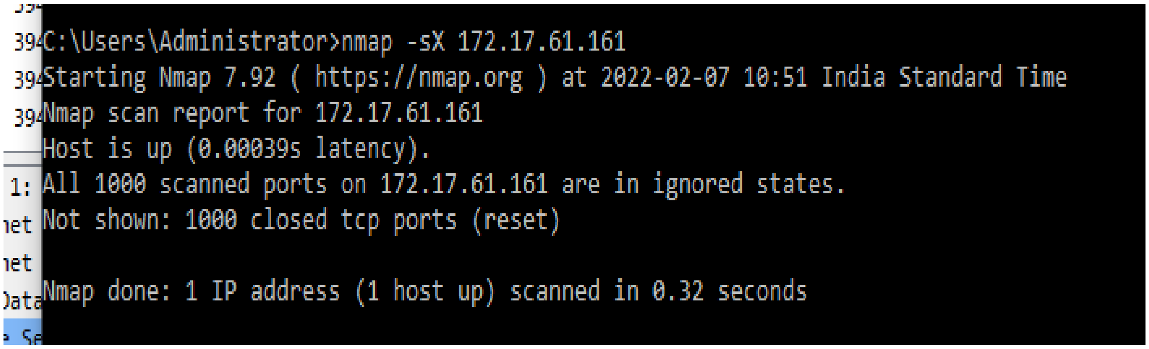
and closed ports.
o When scanning systems compliant with this RFC text, any packet not containing SYN, RST, or ACK bits will result in a returned RST if the port is closed and no response at all if the port is open. o As long as none of those three bits are included, any combination of the other three (FIN, PSH, and URG) are OK.
o Command:

i. TCP null: nmap -sN ip



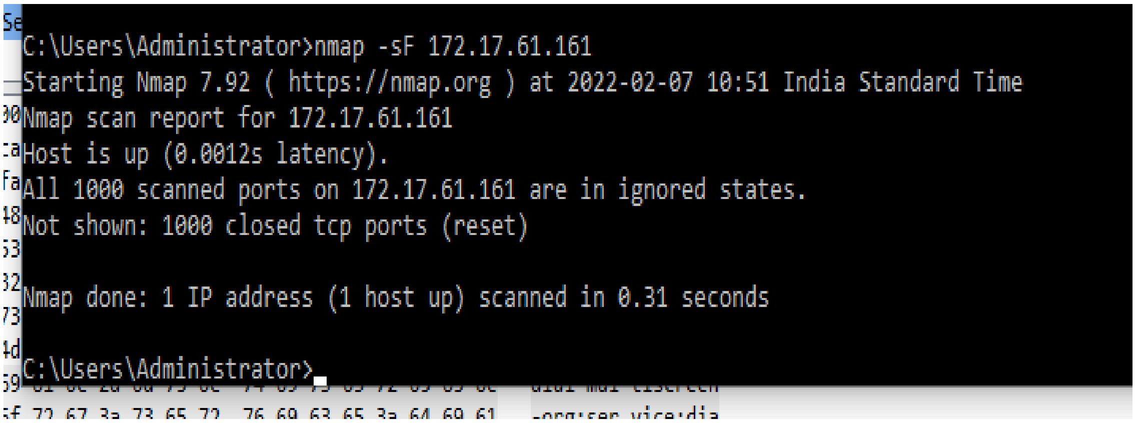
```
Nmap done: 1 IP address (1 host up) scanned in 5.71 seconds

C:\Users\Administrator>nmap -sN 172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:51 India Standard Time
Nmap scan report for 172.17.61.161
Host is up (0.00012s latency).
All 1000 scanned ports on 172.17.61.161 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds
```

ii. TCP Xmas: nmap -sX ip



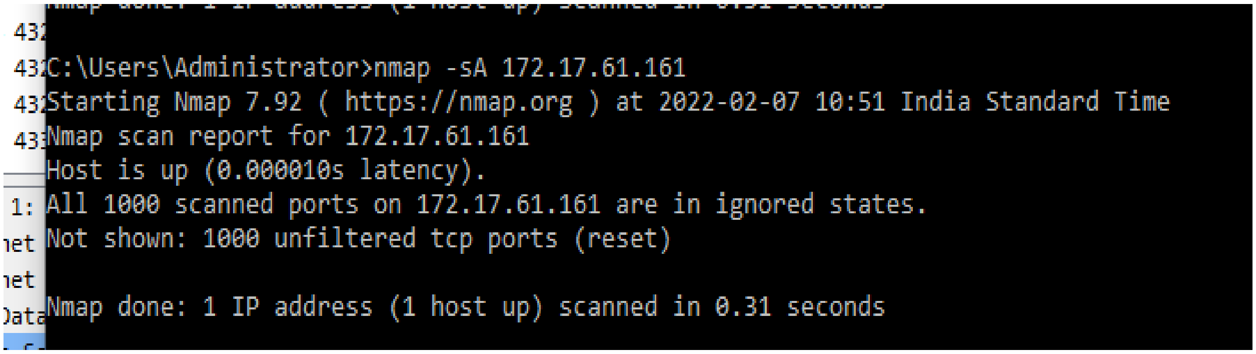
```
C:\Users\Administrator>nmap -sX 172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:51 India Standard Time
Nmap scan report for 172.17.61.161
Host is up (0.00039s latency).
All 1000 scanned ports on 172.17.61.161 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
```

iii. TCP FIN: nmap -sF ip



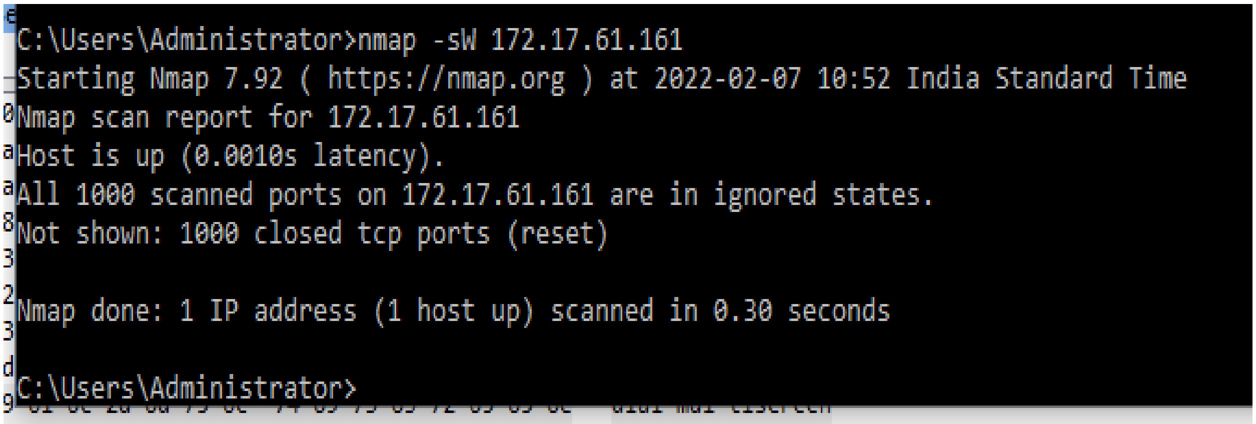
```
C:\Users\Administrator>nmap -sF 172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:51 India Standard Time
Nmap scan report for 172.17.61.161
Host is up (0.0012s latency).
All 1000 scanned ports on 172.17.61.161 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.31 seconds

C:\Users\Administrator>
```

7. TCP ACK
   o This scan is different than the others discussed so far in that it never determines open (or even open|filtered) ports.
   o It is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered.
   o Command: nmap -sA ip



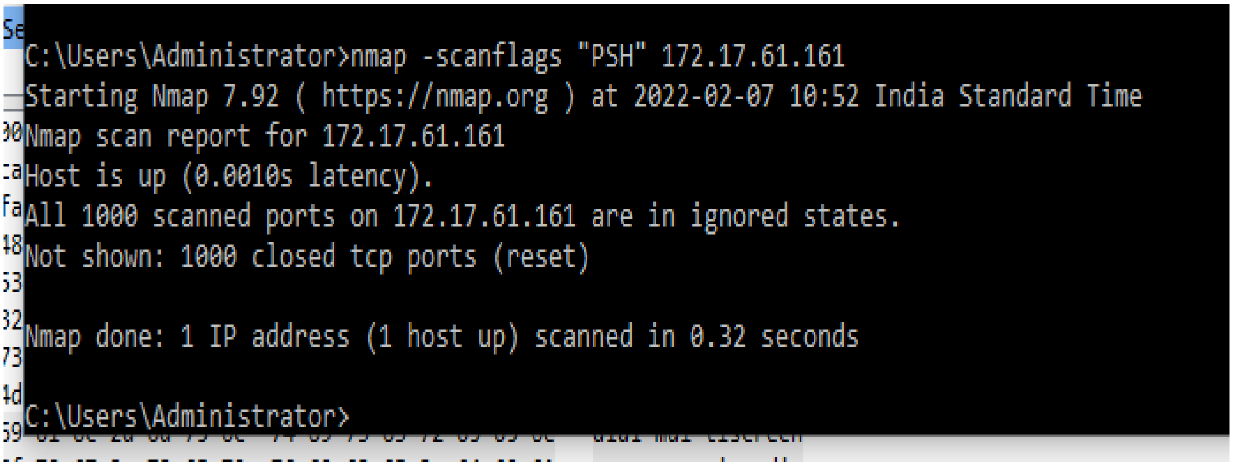
```
C:\Users\Administrator>nmap -sA 172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:51 India Standard Time
Nmap scan report for 172.17.61.161
Host is up (0.000010s latency).
All 1000 scanned ports on 172.17.61.161 are in ignored states.
Not shown: 1000 unfiltered tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.31 seconds
```

8. TCP window
   o Window scan is exactly the same as ACK scan except that it exploits an implementation detail of certain systems to differentiate open ports from closed ones, rather than always printing unfiltered when a RST is returned.
   o It does this by examining the TCP Window field of the RST packets returned.
   o Command: nmap -sW ip



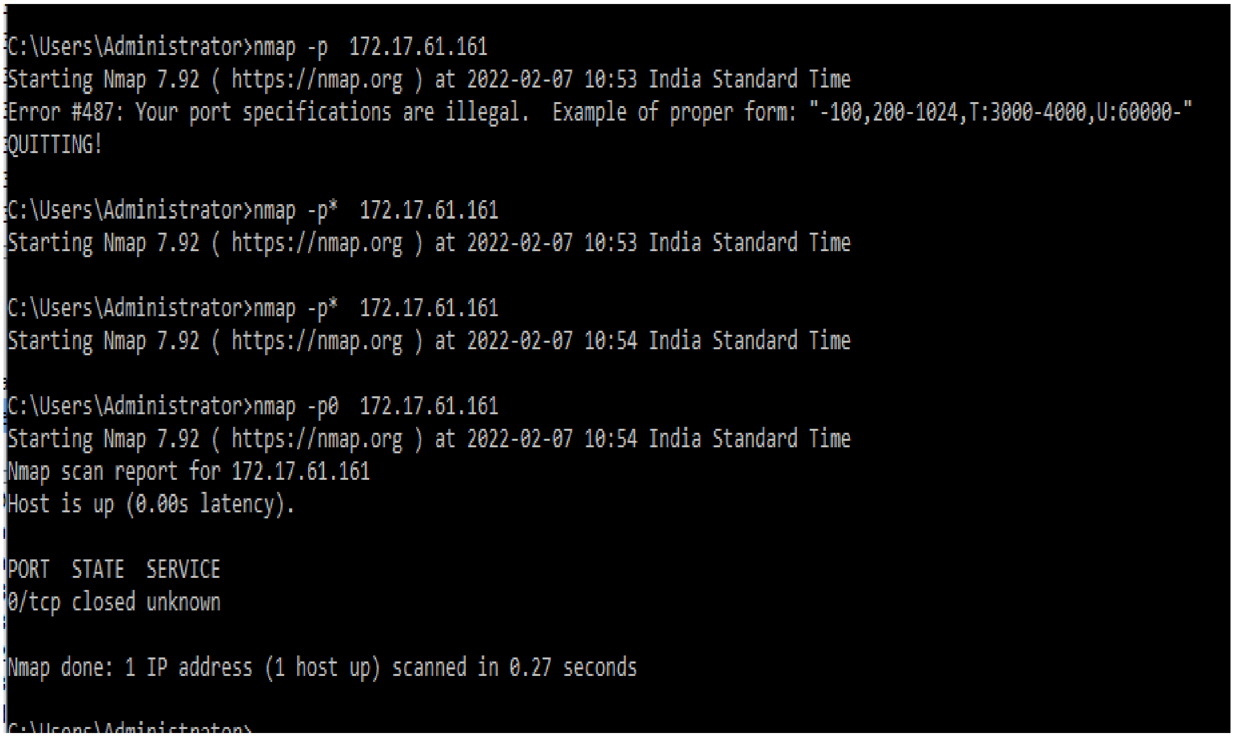
```
C:\Users\Administrator>nmap -sW 172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:52 India Standard Time
Nmap scan report for 172.17.61.161
Host is up (0.0010s latency).
All 1000 scanned ports on 172.17.61.161 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds

C:\Users\Administrator>
```

9. TCP PSH
   - o Truly advanced Nmap users need not limit themselves to the canned scan types offered. The --scanflags option allows you to design your own scan by specifying arbitrary TCP flags.
   - o Command: nmap -scanflags flag ip



```
C:\Users\Administrator>nmap -scanflags "PSH" 172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:52 India Standard Time
Nmap scan report for 172.17.61.161
Host is up (0.0010s latency).
All 1000 scanned ports on 172.17.61.161 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds

C:\Users\Administrator>
```

Other Options : -p, -p*, -p0- :



```
C:\Users\Administrator>nmap -p  172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:53 India Standard Time
Error #487: Your port specifications are illegal.  Example of proper form: "-100,200-1024,T:3000-4000,U:60000-"
QUITTING!

C:\Users\Administrator>nmap -p*  172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:53 India Standard Time

C:\Users\Administrator>nmap -p*  172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:54 India Standard Time

C:\Users\Administrator>nmap -p0  172.17.61.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-07 10:54 India Standard Time
Nmap scan report for 172.17.61.161
Host is up (0.00s latency).

PORT  STATE  SERVICE
0/tcp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds

C:\Users\Administrator>
```

CONCLUSION:

From this practical, we learnt the concept of port scanning using Nmap on a single port and also captured the packets using Wireshark.

Practical 8

Aim:

a) Perform Port Scanning, File Transfer, Client-server chat and Basic Webserver implementation using netcat.
b) Find the service running on the particular port using netcat.

THEORY:

NetCat:

- netcat (often abbreviated to nc) is a computer networking utility for reading from and writing to network connections using TCP or UDP.

- The command is designed to be a dependable back-end that can be used directly or easily driven by other programs and scripts.

- At the same time, it is a feature-rich network debugging and investigation tool, since it can produce almost any kind of connection its user could need and has a number of built- in capabilities.

- Its list of features includes port scanning, transferring files, and port listening, and it can be used as a backdoor.

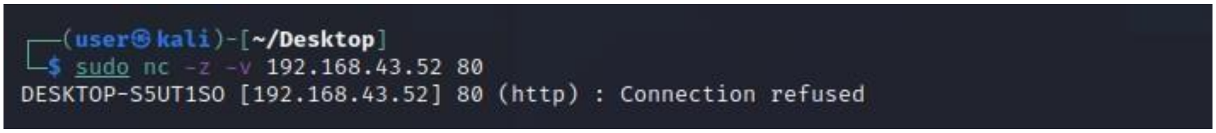
- The original netcat's features include:

i. Outbound or inbound connections, TCP or UDP, to or from any ports
ii. Full DNS forward/reverse checking, with appropriate warnings iii.
iii. Ability to use any local source port
iv. Ability to use any locally configured network source address
v. Built-in port-scanning capabilities, with randomization
vi. Built-in loose source-routing capability
vii. Can read command line arguments from standard input
viii. Slow-send mode, one line every N seconds
ix. Hex dump of transmitted and received data
x. Optional ability to let another program service establish connections
xi. Optional telnet-options responder.

Port Scanning:

- This may useful to know which ports are open and running services on atarget machine.
- Try the nc / netcat command as follow.
- The -z flag can be used to tell nc to report open ports, rather than initiate a connection.
- You need to specify hostname / ip along with the port range to limit and speedup operation:

o Command: nc -z -v hostname port-range
o Ex: nc -z -v localhost 80

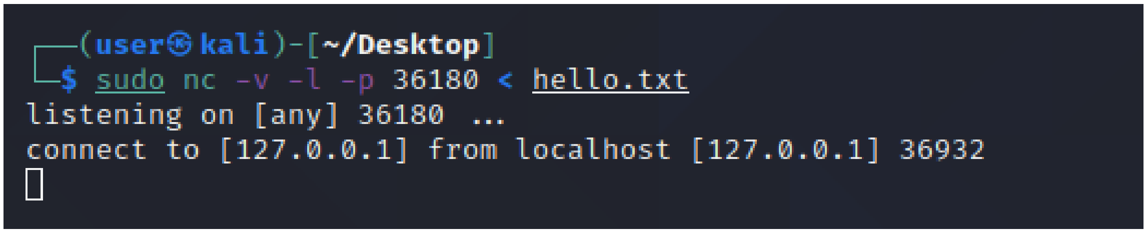


```
┌──(user㉿kali)-[~/Desktop]
└─$ sudo nc -z -v 192.168.43.52 80
DESKTOP-S5UT1SO [192.168.43.52] 80 (http) : Connection refused
```

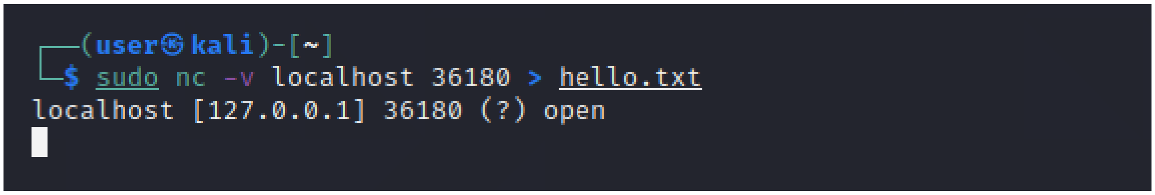
File Transfer:

- The nc (netcat) command can be used to transfer arbitrary data over the network.
- It represents a quick way for Linux administrators to transfer data withoutthe need for an additional data transfer services such as FTP, HTTP, SCPetc.
- This config will show you an example on how to transfer data between to network hosts.
- We will be transferring data myfile.txt file from a localhost to adestination host with an IP address 10.1.1.2.

o Command (Transmitter): nc -v -l -p port < filename
o Command (Reciever): nc -v hostname port > filename
o Ex (Transmitter): nc -v -l -p 36180 < hello.txt
o Ex (Receiver): nc -v localhost 36180 > hello.txt



```
┌──(user㉿kali)-[~/Desktop]
└─$ sudo nc -v -l -p 36180 < hello.txt
listening on [any] 36180 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 36932
```

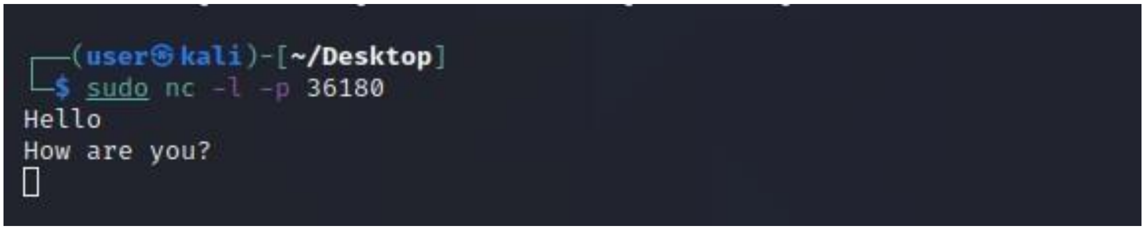


```
┌──(user㉿kali)-[~]
└─$ sudo nc -v localhost 36180 > hello.txt
localhost [127.0.0.1] 36180 (?) open
```

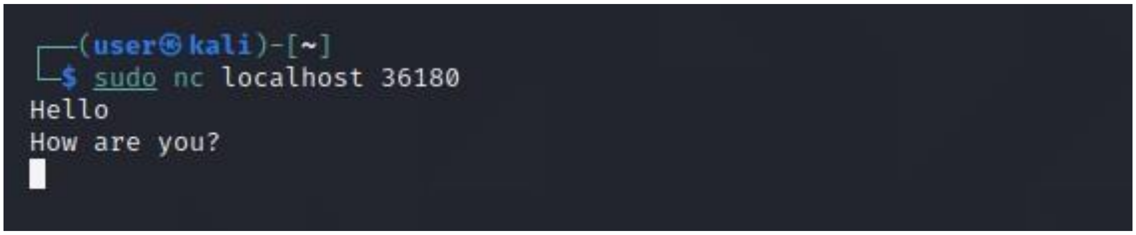
Client-Server Chat:

- To create a simple chat we need two instances of netcat, one to listen for incoming connections (the server) and another one to start the connection.

o Command (Server): nc -l -p port
o Command (Client): nc hostname port
o Ex (Server): nc -l -p 36180
o Ex (Client) : nc localhost 36180



```
┌──(user㉿kali)-[~/Desktop]
└─$ sudo nc -l -p 36180
Hello
How are you?
```

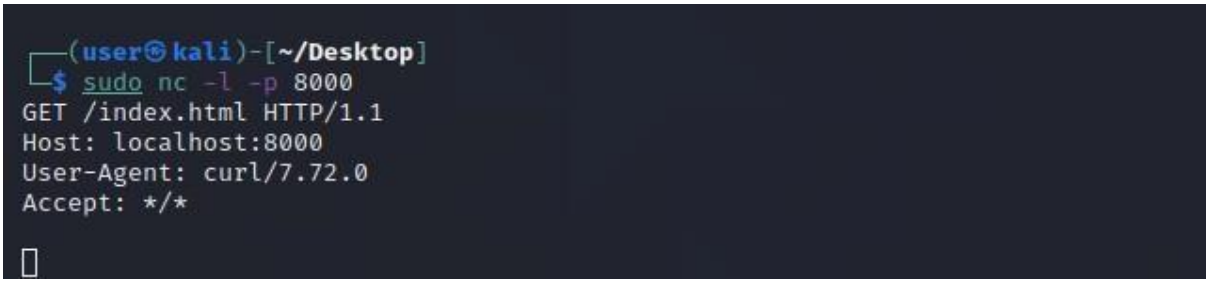


```
┌──(user㉿kali)-[~]
└─$ sudo nc localhost 36180
Hello
How are you?
```

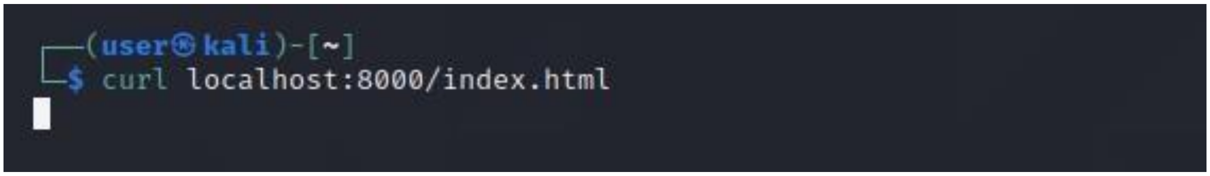
Basic Webserver Implementation:

- The netcat tool nc can operate as a TCP client. Because HTTP works overTCP, nc can be used as an HTTP server!
- Because nc is a UNIX tool, we can use it to make custom web servers: servers which return any HTTP headers you want, servers which return theresponse very slowly, servers which return invalid HTTP, etc.
- You can also use nc as a quick-and-dirty static file server.

o Command: nc -l -p 8000
o Ex: nc -l -p 8000



```
┌──(user㉿kali)-[~/Desktop]
└─$ sudo nc -l -p 8000
GET /index.html HTTP/1.1
Host: localhost:8000
User-Agent: curl/7.72.0
Accept: */*

```



```
┌──(user㉿kali)-[~]
└─$ curl localhost:8000/index.html
```

- After we start listening on port 8000, we can use command “curl localhost:8000/index.html” to send request on port 8000 onlocal server.
- We can confirm that by looking at response in server side where GET request will be reflected.

CONCLUSION:

In this practical, we learned about NetCat which has many functionality and it is easy to use tool. We first transferred a file using nc and then implemented a chat server. We also implemented basic web server with single command.

Practical 9

Aim: In computers, Foot printing is the process of accumulating data regarding a specific network environment, usually for the purpose of finding ways to intrude into the environment. Foot printing can reveal system vulnerabilities and improve the ease with which they can be exploited. Use the given approach to implement Foot printing: Gathering Target Information making use of following tools:

- Dmitry – Deep magic
- UA Tester
- What web

THEORY:

1. Dmitry - Deepmagic:

- DMitry (Deepmagic Information Gathering Tool) is a UNIX/(GNU)Linux Command Line Application coded in C. DMitry has the ability to gather as much information as possible about a host.
- Base functionality is able to gather possible subdomains, emailaddresses, uptime information, tcp port scan, whois lookups, and more.
- The following is a list of the current features:
  - An Open Source Project.
  - Perform an Internet Number whois lookup.
  - Retrieve possible uptime data, system and server data.
  - Perform a SubDomain search on a target host.
  - Perform an E-Mail address search on a target host.
  - Perform a TCP Portscan on the host target.
  - A Modular program allowing user specified modules

2. UA-Tester:

- This tool is designed to automatically check a given URL using a list of standard and non- standard User Agent strings provided by the user (1 per line).
- The results of these checks are then reported to the user for further manual analysis where required. Gathered data includes Response Codes, resulting

URL in the case of a 30x response,

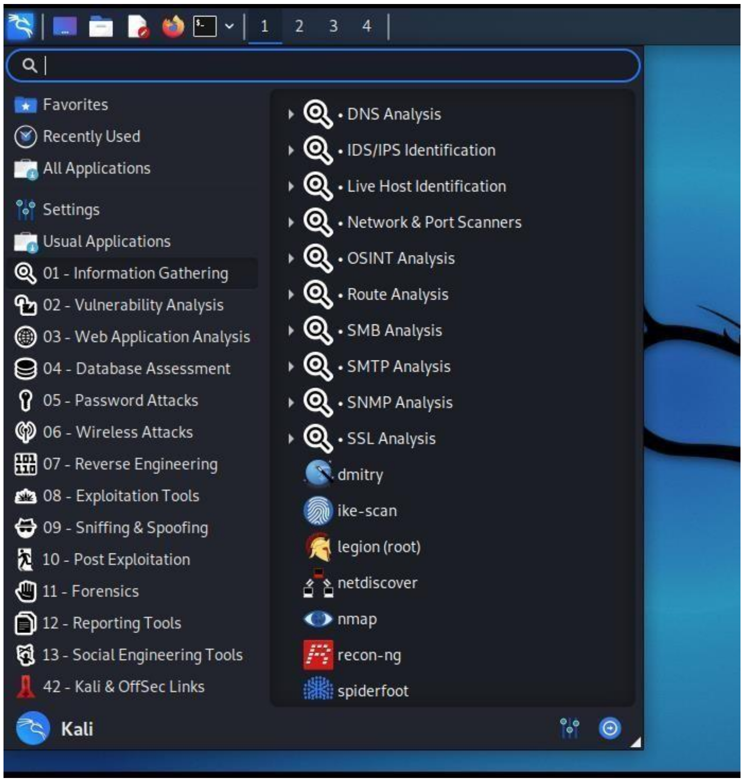
- MD5 and length of response body, and select Server headers.
- Results: When in non-verbose mode, only values that do not match theinitial reference connection are reported to the user.
- If no results are shown for a specific user agent then all results match theinitial reference connection.
- If you require a full output of all checks regardless of matches to the reference,please use the verbose setting.

3. Whatweb:

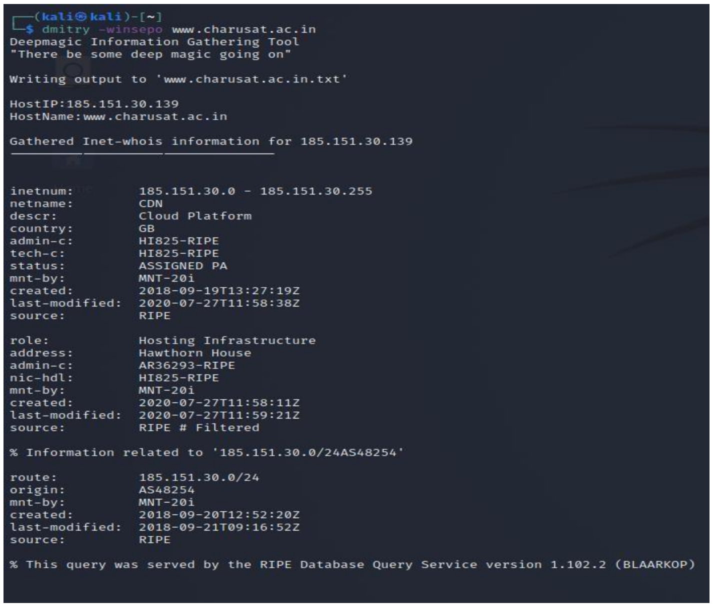
- WhatWeb identifies websites.
- Its goal is to answer the question, “What is that Website?”.
- WhatWeb recognizes web technologies including content management systems(CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices. WhatWeb has over 1700 plugins, each to recognise something different.
- WhatWeb also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more.
- WhatWeb can be stealthy and fast, or thorough but slow.
- WhatWeb supports an aggression level to control the trade off betweenspeed and reliability.
- When you visit a website in your browser, the transaction includes many hintsof what web technologies are powering that website.
- Sometimes a single webpage visit contains enough information to identify a website but when it does not, WhatWeb can interrogate the website further.
- The default level of aggression, called ‘stealthy’, is the fastest and requiresonly one HTTP request of a website.
- This is suitable for scanning public websites. More aggressive modes were developed for use in penetration tests.

Dmitry - Deepmagic:

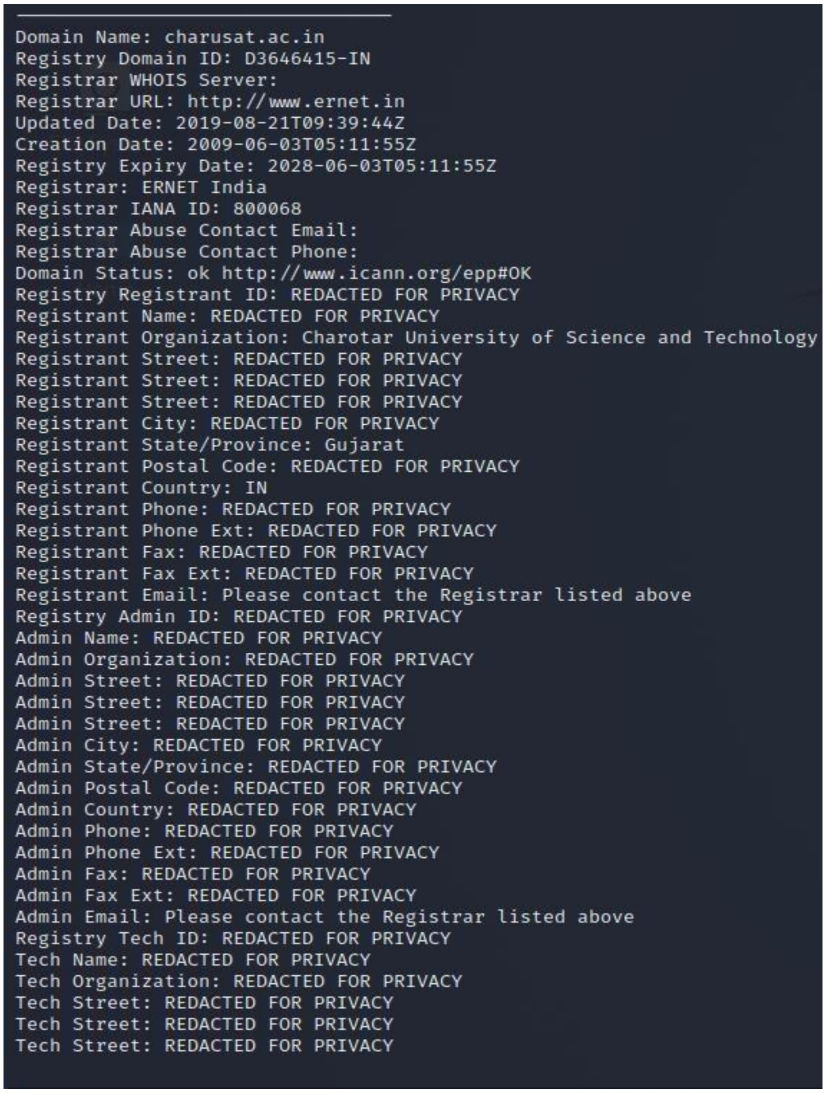
We can find Dmitry in Information Gathering section



Command: Dmitry -winsepo hackthissite.orgWe can use any website as target.

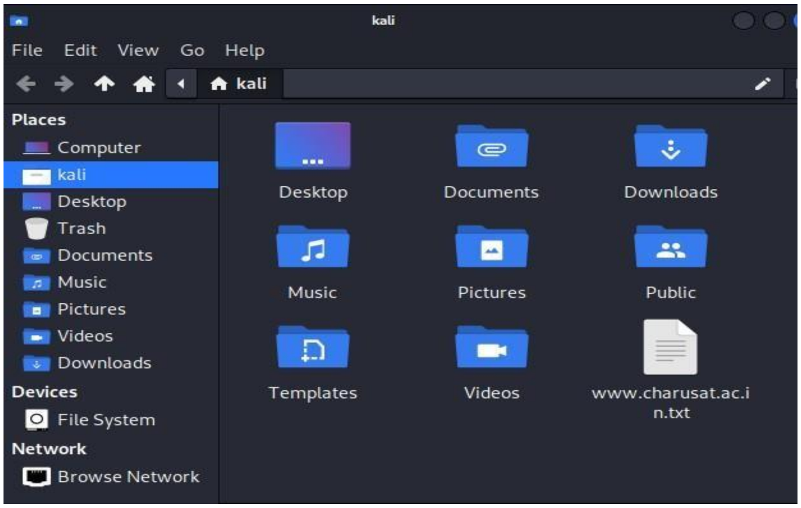


```
┌──(kali㉿kali)-[~]
└─$ dmitry -winsepo www.charusat.ac.in
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Writing output to 'www.charusat.ac.in.txt'

HostIP:185.151.30.139
HostName:www.charusat.ac.in

Gathered Inet-whois information for 185.151.30.139
---------------------------------


inetnum:        185.151.30.0 - 185.151.30.255
netname:        CDN
descr:          Cloud Platform
country:        GB
admin-c:        HI825-RIPE
tech-c:         HI825-RIPE
status:         ASSIGNED PA
mnt-by:         MNT-20i
created:        2018-09-19T13:27:19Z
last-modified:  2020-07-27T11:58:38Z
source:         RIPE

role:           Hosting Infrastructure
address:        Hawthorn House
admin-c:        AR36293-RIPE
nic-hdl:        HI825-RIPE
mnt-by:         MNT-20i
created:        2020-07-27T11:58:11Z
last-modified:  2020-07-27T11:59:21Z
source:         RIPE # Filtered

% Information related to '185.151.30.0/24AS48254'

route:          185.151.30.0/24
origin:         AS48254
mnt-by:         MNT-20i
created:        2018-09-20T12:52:20Z
last-modified:  2018-09-21T09:16:52Z
source:         RIPE

% This query was served by the RIPE Database Query Service version 1.102.2 (BLAARKOP)
```



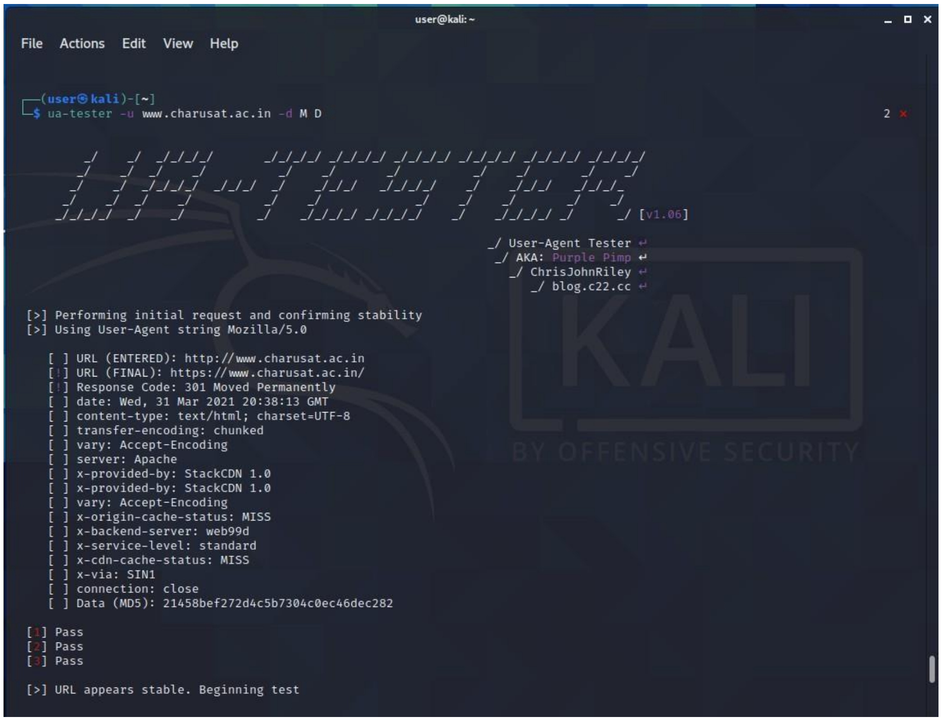
```
Domain Name: charusat.ac.in
Registry Domain ID: D3646415-IN
Registrar WHOIS Server:
Registrar URL: http://www.ernet.in
Updated Date: 2019-08-21T09:39:44Z
Creation Date: 2009-06-03T05:11:55Z
Registry Expiry Date: 2028-06-03T05:11:55Z
Registrar: ERNET India
Registrar IANA ID: 800068
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: ok http://www.icann.org/epp#OK
Registry Registrant ID: REDACTED FOR PRIVACY
Registrant Name: REDACTED FOR PRIVACY
Registrant Organization: Charotar University of Science and Technology
Registrant Street: REDACTED FOR PRIVACY
Registrant Street: REDACTED FOR PRIVACY
Registrant Street: REDACTED FOR PRIVACY
Registrant City: REDACTED FOR PRIVACY
Registrant State/Province: Gujarat
Registrant Postal Code: REDACTED FOR PRIVACY
Registrant Country: IN
Registrant Phone: REDACTED FOR PRIVACY
Registrant Phone Ext: REDACTED FOR PRIVACY
Registrant Fax: REDACTED FOR PRIVACY
Registrant Fax Ext: REDACTED FOR PRIVACY
Registrant Email: Please contact the Registrar listed above
Registry Admin ID: REDACTED FOR PRIVACY
Admin Name: REDACTED FOR PRIVACY
Admin Organization: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY
Admin City: REDACTED FOR PRIVACY
Admin State/Province: REDACTED FOR PRIVACY
Admin Postal Code: REDACTED FOR PRIVACY
Admin Country: REDACTED FOR PRIVACY
Admin Phone: REDACTED FOR PRIVACY
Admin Phone Ext: REDACTED FOR PRIVACY
Admin Fax: REDACTED FOR PRIVACY
Admin Fax Ext: REDACTED FOR PRIVACY
Admin Email: Please contact the Registrar listed above
Registry Tech ID: REDACTED FOR PRIVACY
Tech Name: REDACTED FOR PRIVACY
Tech Organization: REDACTED FOR PRIVACY
Tech Street: REDACTED FOR PRIVACY
Tech Street: REDACTED FOR PRIVACY
Tech Street: REDACTED FOR PRIVACY
```

The output is stored in “hackthissite.org.txt” file.



UA Tester:

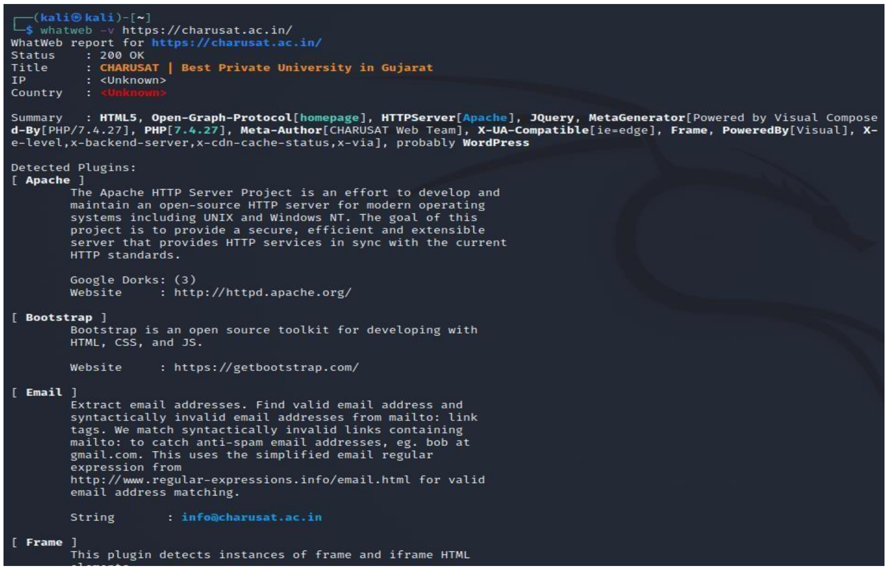
- We can use UA tester directly from terminal.
- Command: ua-tester -u www.charusat.ac.in -d M D



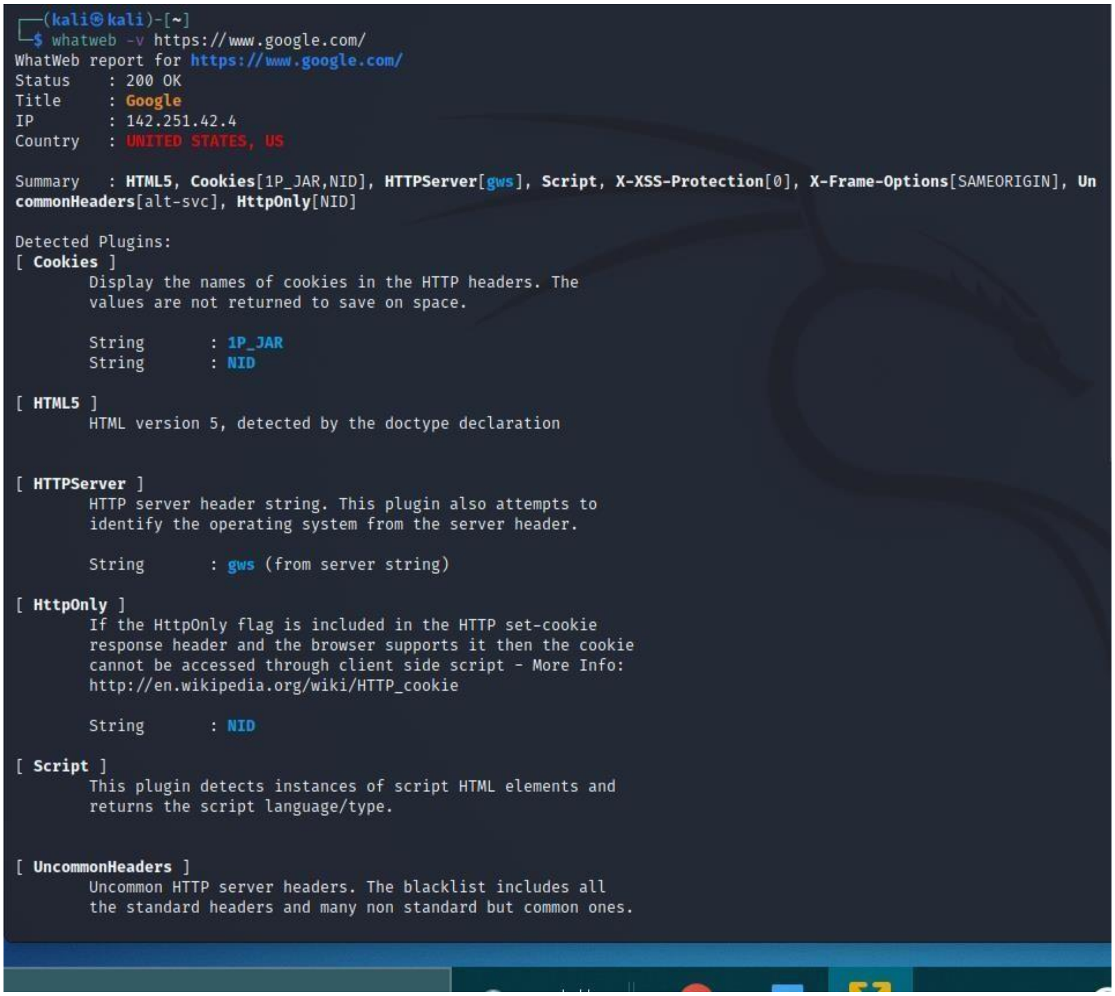
```
┌──(user㉿kali)-[~]
└─$ ua-tester -u www.charusat.ac.in -d M D                                   2 ✖

      _/    _/  _/_/_/_/     _/_/_/_/ _/_/_/_/ _/_/_/_/ _/_/_/_/ _/_/_/_/ _/_/_/_/
     _/    _/  _/    _/        _/     _/       _/          _/     _/       _/    _/
    _/    _/  _/_/_/_/ _/_/_/ _/     _/_/_/   _/_/_/_/    _/     _/_/_/   _/_/_/_/
   _/    _/  _/    _/        _/     _/             _/    _/     _/       _/    _/
  _/_/_/_/  _/    _/        _/     _/_/_/_/ _/_/_/_/    _/     _/_/_/_/ _/      _/ [v1.06]

                                             _/ User-Agent Tester ↵
                                              _/ AKA: Purple Pimp ↵
                                                _/ ChrisJohnRiley ↵
                                                   _/ blog.c22.cc ↵

[>] Performing initial request and confirming stability
[>] Using User-Agent string Mozilla/5.0

   [ ] URL (ENTERED): http://www.charusat.ac.in
   [!] URL (FINAL): https://www.charusat.ac.in/
   [!] Response Code: 301 Moved Permanently
   [ ] date: Wed, 31 Mar 2021 20:38:13 GMT
   [ ] content-type: text/html; charset=UTF-8
   [ ] transfer-encoding: chunked
   [ ] vary: Accept-Encoding
   [ ] server: Apache
   [ ] x-provided-by: StackCDN 1.0
   [ ] x-provided-by: StackCDN 1.0
   [ ] vary: Accept-Encoding
   [ ] x-origin-cache-status: MISS
   [ ] x-backend-server: web99d
   [ ] x-service-level: standard
   [ ] x-cdn-cache-status: MISS
   [ ] x-via: SIN1
   [ ] connection: close
   [ ] Data (MD5): 21458bef272d4c5b7304c0ec46dec282

[1] Pass
[2] Pass
[3] Pass

[>] URL appears stable. Beginning test
```

Whatweb:

- We can also use Whatweb directly from terminal.
- Command: whatweb -v https://charusat.ac.in/
- We can use any website we want as target.



```
┌──(kali㉿kali)-[~]
└─$ whatweb -v https://charusat.ac.in/
WhatWeb report for https://charusat.ac.in/
Status    : 200 OK
Title     : CHARUSAT | Best Private University in Gujarat
IP        : <Unknown>
Country   : <Unknown>

Summary   : HTML5, Open-Graph-Protocol[homepage], HTTPServer[Apache], JQuery, MetaGenerator[Powered by Visual Compose
d-By[PHP/7.4.27], PHP[7.4.27], Meta-Author[CHARUSAT Web Team], X-UA-Compatible[ie=edge], Frame, PoweredBy[Visual], X-
e-level,x-backend-server,x-cdn-cache-status,x-via], probably WordPress

Detected Plugins:
[ Apache ]
        The Apache HTTP Server Project is an effort to develop and
        maintain an open-source HTTP server for modern operating
        systems including UNIX and Windows NT. The goal of this
        project is to provide a secure, efficient and extensible
        server that provides HTTP services in sync with the current
        HTTP standards.

        Google Dorks: (3)
        Website     : http://httpd.apache.org/

[ Bootstrap ]
        Bootstrap is an open source toolkit for developing with
        HTML, CSS, and JS.

        Website     : https://getbootstrap.com/

[ Email ]
        Extract email addresses. Find valid email address and
        syntactically invalid email addresses from mailto: link
        tags. We match syntactically invalid links containing
        mailto: to catch anti-spam email addresses, eg. bob at
        gmail.com. This uses the simplified email regular
        expression from
        http://www.regular-expressions.info/email.html for valid
        email address matching.

        String       : info@charusat.ac.in

[ Frame ]
        This plugin detects instances of frame and iframe HTML
```



```
┌──(kali㉿kali)-[~]
└─$ whatweb -v https://www.google.com/
WhatWeb report for https://www.google.com/
Status    : 200 OK
Title     : Google
IP        : 142.251.42.4
Country   : UNITED STATES, US

Summary   : HTML5, Cookies[1P_JAR,NID], HTTPServer[gws], Script, X-XSS-Protection[0], X-Frame-Options[SAMEORIGIN], Un
commonHeaders[alt-svc], HttpOnly[NID]

Detected Plugins:
[ Cookies ]
        Display the names of cookies in the HTTP headers. The
        values are not returned to save on space.

        String       : 1P_JAR
        String       : NID

[ HTML5 ]
        HTML version 5, detected by the doctype declaration


[ HTTPServer ]
        HTTP server header string. This plugin also attempts to
        identify the operating system from the server header.

        String       : gws (from server string)

[ HttpOnly ]
        If the HttpOnly flag is included in the HTTP set-cookie
        response header and the browser supports it then the cookie
        cannot be accessed through client side script - More Info:
        http://en.wikipedia.org/wiki/HTTP_cookie

        String       : NID

[ Script ]
        This plugin detects instances of script HTML elements and
        returns the script language/type.


[ UncommonHeaders ]
        Uncommon HTTP server headers. The blacklist includes all
        the standard headers and many non standard but common ones.
```

CONCLUSION:

From this practical, we learnt tools like Dmitry, UA-tester and What web for information gathering and used that for Foot printing.

Practical 10

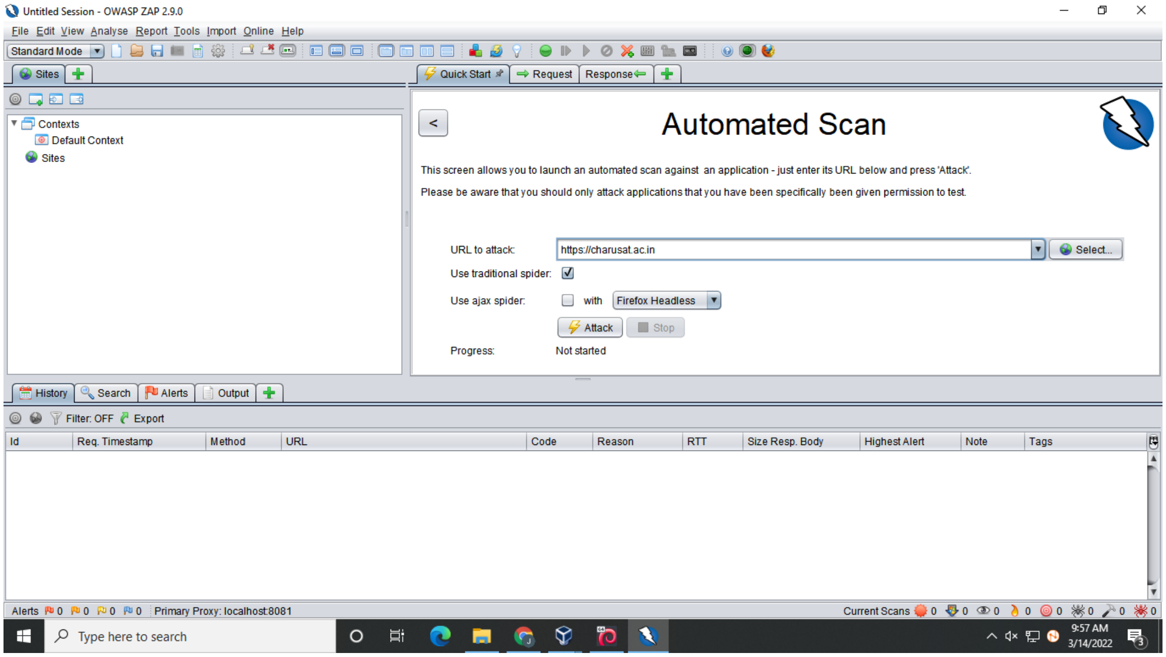
Aim: Find out Web Application Vulnerability using OWASP-ZAP tool.

THEORY:

- ❖ OWASP ZAP is a dynamic application security testing (DAST) tool for finding vulnerabilities in web applications. Like all OWASP projects, it's completely free and open source—and we believe it's the world's most popular web application scanner.
- ❖ ZAP will proceed to crawl the web application with its spider and passively scan each page it finds. Then ZAP will use the active scanner to attack all of the discovered pages, functionality, and parameters. ZAP provides 2 spiders for crawling web applications, you can use either or both of them from this screen.
- ❖ The spider is a tool that is used to automatically discover new resources (URLs) on a particular Site. It begins with a list of URLs to visit, called the seeds, which depends on how the Spider is started. ZAP will use its spider to crawl through the application, which will automatically scan all of the pages discovered. It will then use the active scanner to attack all of the pages. This is a useful way to perform an initial assessment of an application.

Procedure:

Step 1: Open the OWASP-ZAP Application.



Untitled Session - OWASP ZAP 2.9.0

Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: https://charusat.ac.in

Use traditional spider: ☑

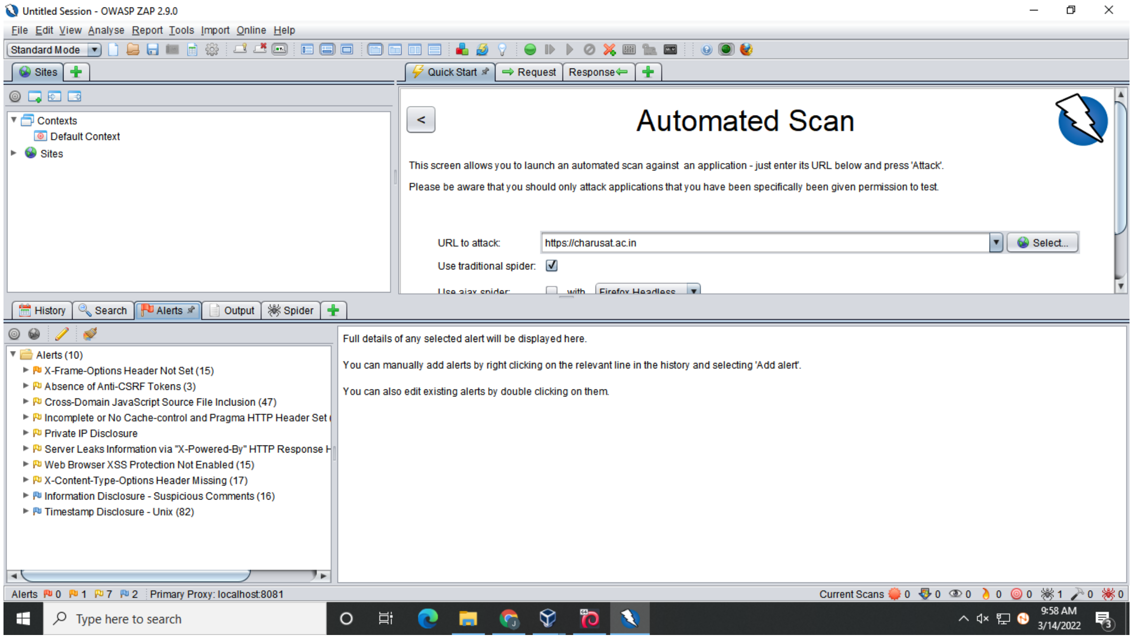
Use ajax spider: □ with Firefox Headless

Progress: Not started

Id	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags

Alerts 0 0 0 0 Primary Proxy: localhost:8081

Step 2: Enter the URL you wish to attack.



Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: https://charusat.ac.in

Use traditional spider: ☑

Use ajax spider: □ with Firefox Headless

- Alerts (10)
  - X-Frame-Options Header Not Set (15)
  - Absence of Anti-CSRF Tokens (3)
  - Cross-Domain JavaScript Source File Inclusion (47)
  - Incomplete or No Cache-control and Pragma HTTP Header Set
  - Private IP Disclosure
  - Server Leaks Information via "X-Powered-By" HTTP Response H
  - Web Browser XSS Protection Not Enabled (15)
  - X-Content-Type-Options Header Missing (17)
  - Information Disclosure - Suspicious Comments (16)
  - Timestamp Disclosure - Unix (82)

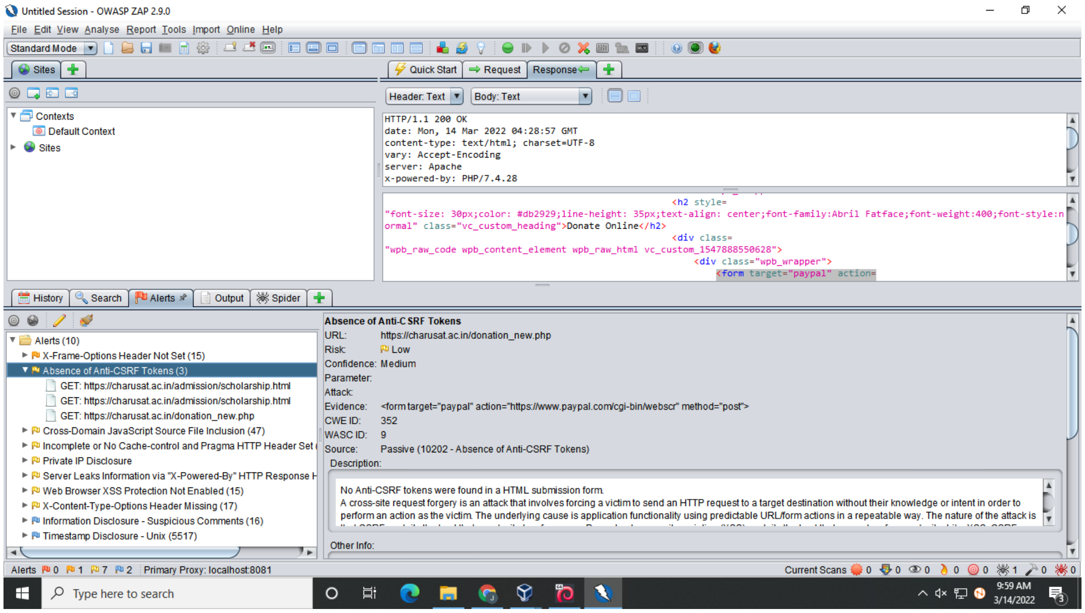
Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

Alerts 0 1 7 2 Primary Proxy: localhost:8081 Current Scans 0 0 0 0 0 1 0 0

Step 3: Alert section in Owasp-zap tool shows potential vulnerability which are associated with a specific request or domain & sub-domain of that particular URL.



```
HTTP/1.1 200 OK
date: Mon, 14 Mar 2022 04:28:57 GMT
content-type: text/html; charset=UTF-8
vary: Accept-Encoding
server: Apache
x-powered-by: PHP/7.4.28
```

```
                                        <h2 style=
"font-size: 30px;color: #db2929;line-height: 35px;text-align: center;font-family:Abril Fatface;font-weight:400;font-style:n
ormal" class="vc_custom_heading">Donate Online</h2>
                                        <div class=
"wpb_raw_code wpb_content_element wpb_raw_html vc_custom_1547888550628">
                                            <div class="wpb_wrapper">
                                                <form target="paypal" action=
```

- Alerts (10)
  - X-Frame-Options Header Not Set (15)
  - Absence of Anti-CSRF Tokens (3)
    - GET: https://charusat.ac.in/admission/scholarship.html
    - GET: https://charusat.ac.in/admission/scholarship.html
    - GET: https://charusat.ac.in/donation_new.php
  - Cross-Domain JavaScript Source File Inclusion (47)
  - Incomplete or No Cache-control and Pragma HTTP Header Set
  - Private IP Disclosure
  - Server Leaks Information via "X-Powered-By" HTTP Response H
  - Web Browser XSS Protection Not Enabled (15)
  - X-Content-Type-Options Header Missing (17)
  - Information Disclosure - Suspicious Comments (16)
  - Timestamp Disclosure - Unix (5517)

Absence of Anti-CSRF Tokens

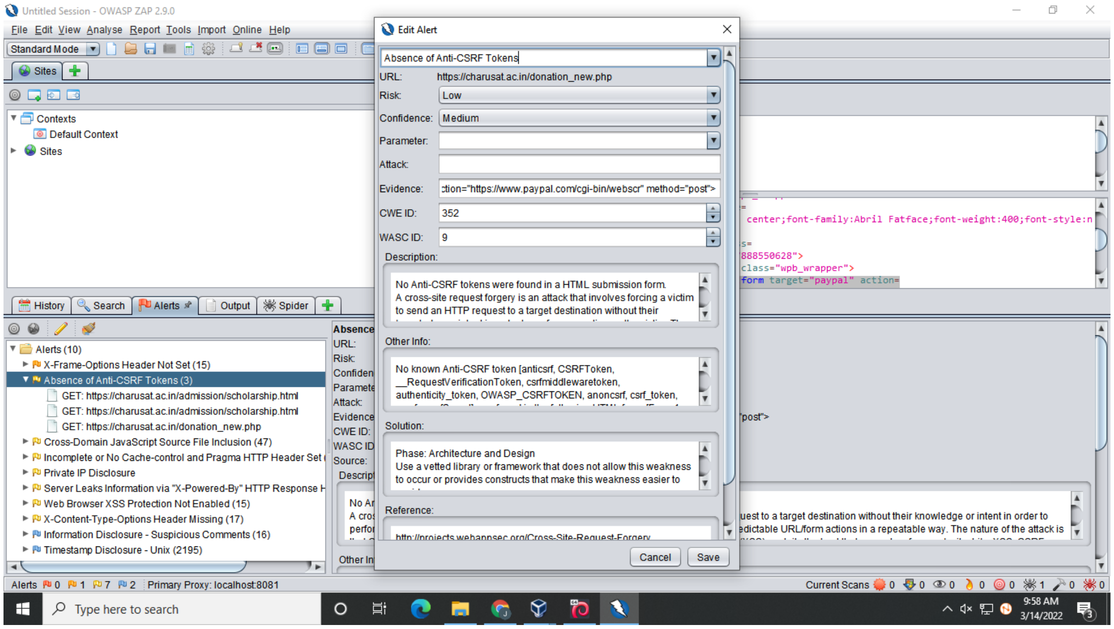
URL: https://charusat.ac.in/donation_new.php
Risk: Low
Confidence: Medium
Parameter:
Attack:
Evidence: <form target="paypal" action="https://www.paypal.com/cgi-bin/webscr" method="post">
CWE ID: 352
WASC ID: 9
Source: Passive (10202 - Absence of Anti-CSRF Tokens)
Description:

No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is

Other Info:

Alerts 0 1 7 2 Primary Proxy: localhost:8081 Current Scans 0 0 0 0 0 1 0 0

Step 4: On double clicking on the particular token in alert , it will also allows you to update or change alert details/information.



Edit Alert

Absence of Anti-CSRF Tokens

URL: https://charusat.ac.in/donation_new.php

Risk: Low

Confidence: Medium

Parameter:

Attack:

Evidence: ction="https://www.paypal.com/cgi-bin/webscr" method="post">

CWE ID: 352

WASC ID: 9

Description:

No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their

Other Info:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token,

Solution:

Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to

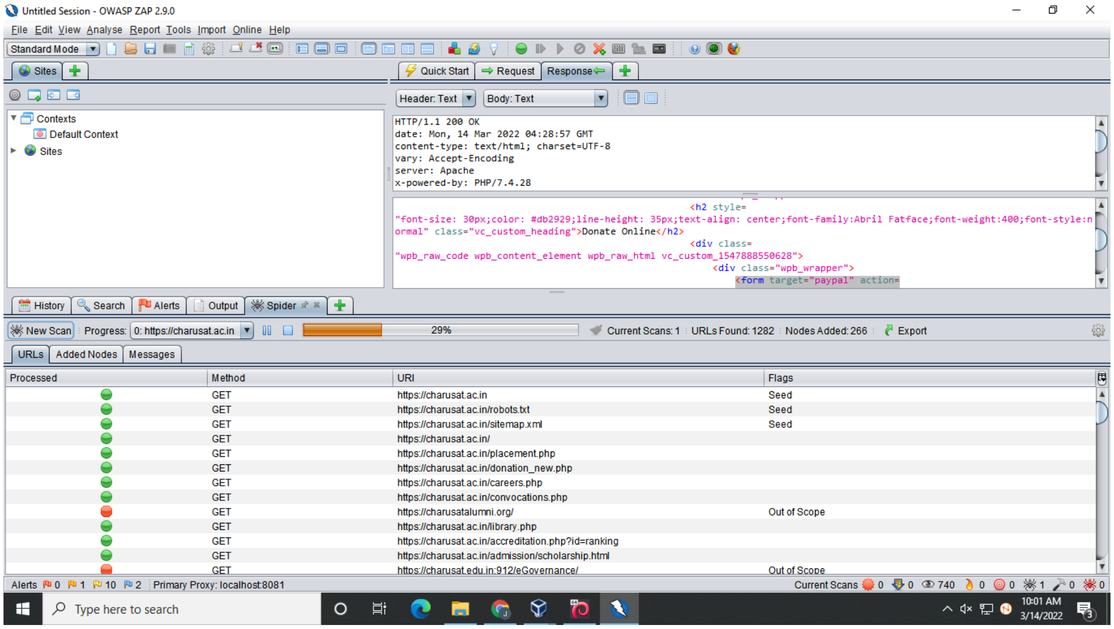
Reference:

http://projects.webappsec.org/Cross-Site-Request-Forgery

Alerts (10)
- X-Frame-Options Header Not Set (15)
- Absence of Anti-CSRF Tokens (3)
  - GET: https://charusat.ac.in/admission/scholarship.html
  - GET: https://charusat.ac.in/admission/scholarship.html
  - GET: https://charusat.ac.in/donation_new.php
- Cross-Domain JavaScript Source File Inclusion (47)
- Incomplete or No Cache-control and Pragma HTTP Header Set
- Private IP Disclosure
- Server Leaks Information via "X-Powered-By" HTTP Response H
- Web Browser XSS Protection Not Enabled (15)
- X-Content-Type-Options Header Missing (17)
- Information Disclosure - Suspicious Comments (16)
- Timestamp Disclosure - Unix (2195)

Untitled Session - OWASP ZAP 2.9.0
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
History Search Alerts Output Spider
Cancel Save
Alerts 0 1 7 2 Primary Proxy: localhost:8081
Current Scans 0 0 0 0 0 1 0 0
Type here to search
9:58 AM
3/14/2022

Step 5: In spider section it will automatically discovers new resources (URLs) on a particular Site. It begins with a list of URLs to visit, called the seeds, which depends on how the Spider is started.



```
HTTP/1.1 200 OK
date: Mon, 14 Mar 2022 04:28:57 GMT
content-type: text/html; charset=UTF-8
vary: Accept-Encoding
server: Apache
x-powered-by: PHP/7.4.28
```

```
                              <h2 style=
"font-size: 30px;color: #db2929;line-height: 35px;text-align: center;font-family:Abril Fatface;font-weight:400;font-style:n
ormal" class="vc_custom_heading">Donate Online</h2>
                              <div class=
"wpb_raw_code wpb_content_element wpb_raw_html vc_custom_1547888550628">
                                 <div class="wpb_wrapper">
                                   <form target="paypal" action=
```

New Scan Progress: 0: https://charusat.ac.in 29% Current Scans: 1 URLs Found: 1282 Nodes Added: 266 Export

Processed	Method	URI	Flags
●	GET	https://charusat.ac.in	Seed
●	GET	https://charusat.ac.in/robots.txt	Seed
●	GET	https://charusat.ac.in/sitemap.xml	Seed
●	GET	https://charusat.ac.in/	
●	GET	https://charusat.ac.in/placement.php	
●	GET	https://charusat.ac.in/donation_new.php	
●	GET	https://charusat.ac.in/careers.php	
●	GET	https://charusat.ac.in/convocations.php	
●	GET	https://charusatalumni.org/	Out of Scope
●	GET	https://charusat.ac.in/library.php	
●	GET	https://charusat.ac.in/accreditation.php?id=ranking	
●	GET	https://charusat.ac.in/admission/scholarship.html	
●	GET	https://charusat.edu.in:912/eGovernance/	Out of Scope

Untitled Session - OWASP ZAP 2.9.0
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start Request Response
Header: Text Body: Text
History Search Alerts Output Spider
URLs Added Nodes Messages
Alerts 0 1 10 2 Primary Proxy: localhost:8081
Current Scans 0 0 740 0 0 1 0 0
Type here to search
10:01 AM
3/14/2022

CONCLUSION:

From this practical, we learnt OWASP-ZAP tool for finding vulnerabilities in web applications.