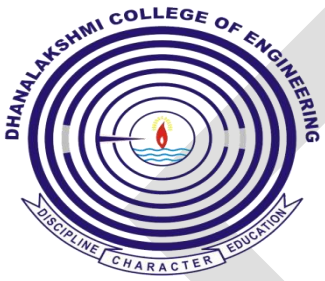
DHANALAKSHMI COLLEGE OF ENGINEERING
TAMBARAM, CHENNAI – 601 301



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CS6711 – Security Laboratory

VII SEMESTER
R 2013

LABORATORY MANUAL

NAME : …….

REG. NO. : …….

CLASS : ……..

DHANALAKSHMI COLLEGE OF ENGINEERING

VISION

Dhanalakshmi College of Engineering is committed to provide highly disciplined, conscientious and enterprising professionals conforming to global standards through value based quality education and training.

MISSION

- To provide competent technical manpower capable of meeting requirements of the industry
- To contribute to the promotion of Academic Excellence in pursuit of Technical Education at different levels
- To train the students to sell his brawn and brain to the highest bidder but to never put a price tag on heart and soul

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision of the Department:

To strive for acquiring, applying and imparting knowledge in Computer Science and Engineering through quality education and to provide enthusiastic professionals with commitment

Mission of the Department:

- To produce highly competent and globally employable engineers in the field of Computer Science and Engineering
- To inculcate human values among the student community and make them realize their commitment to the society
- To exhibit excellence in pursuit of research and innovative products with a zeal to serve the society

PROGRAM EDUCATIONAL OBJECTIVES

1. FUNDAMENTALS

To impart students with fundamental knowledge in Mathematics, Science and fundamentals of engineering that will mould them to be successful professionals

2. CORE COMPETENCE

To provide students with sound knowledge in engineering and experimental skills to identify complex software problems in industry and to develop practical solutions for them

3. BREADTH

To provide relevant training and experience to bridge the gap between theory and practice which enables them to find solutions for real time problems in industry and organization, and to design products requiring interdisciplinary skills

4. PROFESSIONAL SKILLS

To bestow students with adequate training and provide opportunities to work as team that will build up their communication skills, individual leadership and supportive qualities, and to enable them to adapt and work in ever changing technologies

5. LIFELONG LEARNING

To develop the ability of students to establish themselves as professionals in Computer Science and Engineering and to create awareness about the need for lifelong learning and pursuing advanced degrees

PROGRAM OUTCOMES

On completion of the B.E. (CSE) degree, the graduates will be able

a) To apply the basic knowledge of Mathematics, Science and engineering fundamentals in Computer Science and Engineering field

b) To design and conduct experiments as well as to analyze and interpret and apply the same in the career

c) To design and develop innovative and creative software applications

d) To understand a complex real world problem and develop an efficient practical solution

e) To create, select and apply appropriate techniques, resources, modern engineering and IT tools

f) To understand their roles as a professionals and give the best to the society

g) To develop a system that will meet expected needs within realistic constraints such as economical, environmental, social, political, ethical, safe and sustainable

h) To communicate effectively and make others understand exactly what they are trying to convey in both verbal and written forms

i) To work in a team as team member or a leader and make unique contributions and work with coordination

j) To engage in lifelong learning and exhibit their technical skills

k) To develop and manage projects in multidisciplinary environments

CS6711 - Security Laboratory
SYLLABUS

COURSE OBJECTIVES

- Be exposed to the different cipher techniques
- Learn to implement the algorithms like DES, RSA, MD5, SHA-1
- Understand the Digital Signature Standard
- Learn to use network security tools like GnuPG, KF sensor, Net Strumbler
- Be familiar with the intrusion detection system

LIST OF EXPERIMENTS

1. Implement the following Substitution & Transposition Techniques concepts:

a) Caesar Cipher
b) Playfair Cipher
c) Hill Cipher
d) Vignere Cipher
e) Rail fence – row & Column Transformation

2. Implement the following algorithms

a) DES
b) RSA Algorithm
c) Diffie-Hellman
d) MD5
e) SHA-1

3. Implement the SIGNATURE SCHEME - Digital Signature Standard

4. Demonstrate how to provide secure data storage, secure data transmission and for creating digital signatures (GnuPG).

5. Setup a honey pot and monitor the honeypot on network (KF Sensor)

6. Installation of rootkits and study about the variety of options

7. Perform wireless audit on an access point or a router and decrypt WEP and WPA.(Net Stumbler)

8. Demonstrate intrusion detection system (ids) using any tool (snort or any other s/w).

COURSE OUTCOMES

- Implement the cipher techniques
- Apply the mathematical foundation required for various cryptographic algorithms
- Develop the various security algorithms
- Design the signature scheme by applying Digital Signature Standard
- Use different open source tools for network security and analysis

- Demonstrate the intrusion detection system

INDEX

Ex.No. 1(a)

Date:

Caesar Cipher

Implementation of Substitution and Transposition Techniques

Aim:

To write a Java program to implement substitution and transposition techniques using caesar cipher algorithm

Algorithm:

1. Caesar cipher is an example of a substitution cipher in which plaintext letters in the original message are replaced (substituted for) by cipher text letters
2. The easiest way to understand this is to consider that there are two alphabets:

   PLAIN_ALPHABET: ABCDEFGHIJKLMNOPQRSTUVWXYZ

   CIPHER_ALPHABET: DEFGHIJKLMNOPQRSTUVWXYZABC
3. The cipher alphabet is a shifted version of the plain alphabet. In this case, each letter in the cipher alphabet has to be shifted by 3 places to the right
4. The shift -- (i.e., the number 3) is the secret key which must be shared by Alice and Bob if they want to send secret messages using this cipher
5. To *encrypt* the message MEET ME AT THE DOCK we would replace all the M*s* in the message with the corresponding letter from the cipher alphabet
6. So M is replaced by P. And we would replace all the Es by H and so on. Thus, the encryption of our message would be PHHW PH DW WLH GRFN

Sample Output:

Enter any String: Hello World

Enter the Key: 5

Encrypted String is: MjqqtBtwqi

Decrypted String is: Hello World

Result:

Thus the Java program to implement substitution and transposition techniques using caesar cipher algorithm was executed successfully

Ex.No. 1(b)

Date:

Playfair Cipher

Aim:

To write a Java program to implement playfair cipher algorithm

Algorithm:

1. The playfair cipher was the first practical digraph substitution cipher. The technique encrypts pairs of letters (digraphs), instead of single letters as in the simple substitution cipher

2. The 'key' for a playfair cipher is generally a word, for the sake of example we will choose 'monarchy'. This is then used to generate a 'key square', e.g.

```
m o n a r
c h y b d
e f g i k
l p q s t
u v w x z
```

3. Any sequence of 25 letters can be used as a key, so long as all letters are in it and there are no repeats. Note that there is no 'j', it is combined with 'i'. We now apply the encryption rules to encrypt the plaintext

i) Remove any punctuation or characters that are not present in the key square (this may mean spelling out numbers, punctuation etc.)

ii) Identify any double letters in the plaintext and replace the second occurrence with an 'x' e.g. 'hammer' -> 'hamxer'

iii) If the plaintext has an odd number of characters, append an 'x' to the end to make it even

iv) Break the plaintext into pairs of letters, e.g. 'hamxer' -> 'ha mx er'

v) The algorithm now works on each of the letter pairs

vi) Locate the letters in the key square, (the examples given are using the key square above)

a. If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. The order is important – the first encrypted letter of the pair is the one that lies on the same row as the first plaintext letter. 'ha' -> 'bo', 'es' -> 'il'
b. If the letters appear on the same row of the table, replace them with the letters to their immediate right respectively (wrapping around to the left side of the row if a letter in the original pair was on the right side of the row). 'ma' -> 'or', 'lp' -> 'pq'
c. If the letters appear on the same column of the table, replace them with the letters immediately below respectively (wrapping around to the top side of the column if a letter in the original pair was on the bottom side of the column). 'rk' -> 'dt', 'pv' -> 'vo'

Sample Output:

Enter the text to be encrypted: OR

```
m * * a *
* * * * *
* * * * *
l * * s *
* * * * *
```

Hence, al -> ms

```
* * * * *
* h y b d
* * * * *
* * * * *
* * * * *
```

Hence, hb -> yd

```
* * n * *
* * y * *
* * * * *
* * q * *
* * w * *
```

Hence, nq -> yw

```
plaintext:  wearediscoveredsaveyourselfx
ciphertext: ugrmkcsxhmufmkbtoxgcmvatluiv
```

Result:

Thus the Java program to implement substitution and transposition techniques using playfair cipher algorithm was executed successfully

Ex.No. 1(c)

Date:

Hill Cipher

Aim:

To write a program to implement hill cipher algorithm

Algorithm:

1. In a Hill cipher encryption, the plaintext message is broken up into blocks of length n according to the $m \times n$ matrix chosen
2. Each block of plaintext letters is then converted into a vector of numbers and is dotted with the matrix
3. The results are then converted back to letters and the ciphertext message is produced
4. For decryption of the ciphertext message, the inverse of the encryption matrix must be found Once found, the decryption matrix is then dotted with each n-block of ciphertext, producing the plaintext message.

Sample Output:

Enter a 3 letter string: hai

Encrypted string is :fdx

Inverse Matrix is :

0.083333336 0.41666666 -0.33333334

-0.41666666 -0.083333336 0.6666667

0.5833333 -0.083333336 -0.33333334

Decrypted string is :hai

Result:

Thus the Java program to implement substitution and transposition techniques using hill cipher algorithm was executed successfully

Ex.No. 1(d)

Date:

Vignere Cipher

Aim:

To write a Java program to implement Vignere cipher

Algorithm :

1. A Vignere Square or Vignere table consists of the alphabet written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar ciphers
2. At different points in the encryption process, the cipher uses a different alphabet from one of the rows. The alphabet used at each point depends on a repeating keyword
3. The person sending the message to be encrypted (eg. attackatdawn) chooses a keyword and repeats it until it matches the length of the plaintext, for example, the keyword lemon, the cipher key will be lemonlemonle
4. Using a VignereSquare and a CipherKey each row starts with a key letter. The remainder of the row holds the letters A to Z (in shifted order)
5. Although there are 26 key rows shown, you will only use as many keys (different alphabets) as there are unique letters in the key string, here just 5 keys, {L, E, M, O, N}
6. For successive letters of the message, we are going to take successive letters of the key string, and encipher each message letter using its corresponding key row. Choose the next letter of the key, go along that row to find the column heading that matches the message character; the letter at the intersection of [key-row, msg-col] is the enciphered letter
7. The first letter of the plaintext, A, is paired with L, the first letter of the key. So use row L and column A of the Vignere square, namely L. Similarly, for the second letter of the plaintext, the second letter of the key is used; the letter at row E and column T is X. The rest of the plaintext is enciphered in a similar fashion

Sample Output:

Plaintext: ATTACKATDAWN

Key: LEMONLEMONLE

Ciphertext: LXFOPVEFRNHR

Result:

Thus the Java program to implement substitution and transposition techniques using vignere cipher algorithm was executed successfully

Ex.No. 1(e)

Date:

Rail Fence – Row & Column Transformation

Aim:

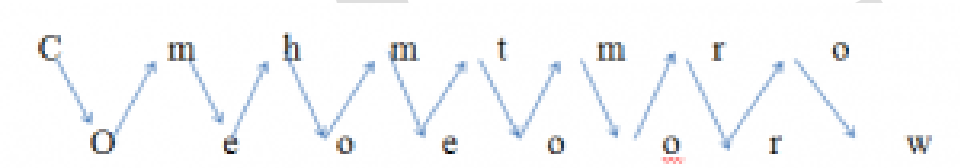
To write a Java program to implement rail fence algorithm

Algorithm:

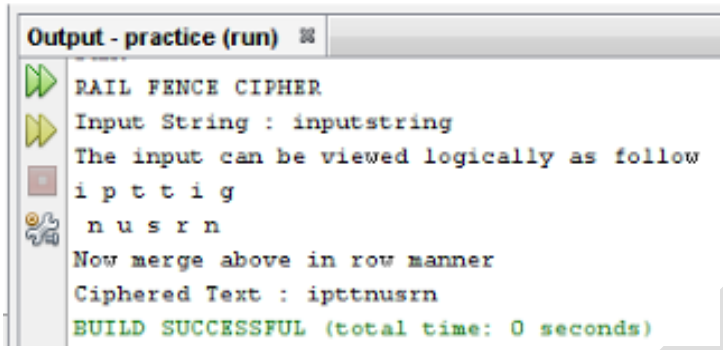
1. In the rail fence cipher, the plaintext is written downwards and diagonally on successive "rails" of an imaginary fence, then moving up when we reach the bottom rail
2. When we reach the top rail, the message is written downwards again until the whole plaintext is written out. The message is then read off in rows
3. Write down the plain text message as a sequence of diagonals
4. Read the plain text written in Step 1 as a sequence of rows

Example: Original plain text message: Come home tomorrow

Arrange the plain text message as sequence of diagonals



Sample Output:



Result:

Thus the Java program to implement substitution and transposition techniques using rail fence algorithm was executed successfully

Viva-Voce:

1. What is public-key cryptography?

It is also known as asymmetric cryptography. In this type of cryptography a key pair consists of public key and private key is used to securely exchange the message between two parties. One is used to encrypt the message and other key is used to decrypt the message. The advantage of public-key cryptography is that you only need to keep your private key secure and distribute the public key.

2. What is block cipher?

Block cipher is an algorithm that converts a block of plaintext into ciphertext. It takes two inputs, n bits of fixed length of blocks and a secret key and output is n bits of ciphertext.

3. What is stream cipher? Name a most widely used stream cipher.

Stream cipher is symmetric encryption algorithm with takes one bit or one byte as input and encrypted with a secret key called keystream. The keystream generator function produces keys K1, K2, K3 … which are XORed with plaintext P1, P2, P3 … to produce cipher text C1, C2, C3 …

4. What are the differences among encoding, encryption and hashing?

Encoding: Basically encoding is used to protect the integrity of data as it crosses through communication network to keep its original message upon arriving. It is primarily an insecure function because it is easily reversible.

Encryption: Encryption is basically designed for confidentiality and data integrity and reversible only if you have the appropriate key.

Hashing: With hashing the operation is one-way i.e. non-reversible. It takes an input (or message) and returns a fixed-size string, which is called the hash value.

5. What are Brute Force Attacks?

Brute forcing is a mechanism which is used by an attacker to break the encryption of data by applying a set of various key. Cryptanalyst has a set of number of keys and apply them one by one to the encryption algorithm until he get the right key.

Implementation of Cryptographic Algorithms

Ex.No. 2(a)

Date:

DES

Aim:

To write a Java program to implement DES algorithm

Algorithm:

1. Firstly, we need to process the key

2. Get a 64-bit key from the user. (Every 8th bit is considered a parity bit. For a key to have correct parity, each byte should contain an odd number of "1" bits.)

3. Calculate the key schedule

4. Perform the following permutation on the 64-bit key

5. Split the permuted key into two halves. The first 28 bits are called C[0] and the last 28 bits are called D[0]

6. Calculate the 16 subkeys. Start with i = 1. Perform one or two circular left shifts on both C[i-1] and D[i-1] to get C[i] and D[i], respectively. The number of shifts per iteration are given below:

Iteration # 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Left Shifts 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1

7. Permute the concatenation C[i]D[i] as indicated below. This will yield K[i], which is 48 bits long. Permuted Choice 2 (PC-2)

8. Loop back to 1.2.3.1 until K[16] has been calculated. Process a 64-bit data block

9. Get a 64-bit data block. If the block is shorter than 64 bits, it should be padded as appropriate for the application

10. Perform the following permutation on the data block. Initial Permutation (IP)

Sample Output:

Input.txt

JavaCode

encrypted.txt

—w~Z5-ó&ÏεE

decrypted.txt

JavaCode

Result:

Thus the Java program to implement cryptographic algorithm using DES algorithm was executed successfully

Ex.No. 2(b)

Date:

RSA Algorithm

Aim:

To write a Java program to implement RSA algorithm

Algorithm:

1. Generate two large random primes, P and Q, of approximately equal size
2. Compute $N = P \times Q$
3. Compute $Z = (P-1) \times (Q-1)$
4. Choose an integer E, $1 < E < Z$, such that GCD $(E, Z) = 1$
5. Compute the secret exponent D, $1 < D < Z$, such that $E \times D \equiv 1 \pmod{Z}$
6. The public key is (N, E) and the private key is (N, D)

An example of RSA encryption :

1. Select primes P=11, Q=3

2. $N = P \times Q = 11 \times 3 = 33$
$Z = (P-1) \times (Q-1) = 10 \times 2 = 20$

3. Lets choose E=3
Check GCD$(E, P-1)$ = GCD(3, 10) = 1 (i.e. 3 and 10 have no common factors except 1), and check GCD$(E, Q-1)$ = GCD(3, 2) = 1, therefore GCD(E, Z) = GCD(3, 20) = 1

4. Compute D such that $E \times D \equiv 1 \pmod{Z}$
Compute $D = E^{\wedge -1} \bmod Z = 3^{\wedge -1} \bmod 20$
Find a value for D such that Z divides $((E \times D)-1)$
Find D such that 20 divides 3D-1
Simple testing (D = 1, 2, ...) gives D = 7
Check: $(E \times D)-1 = 3.7 - 1 = 20$, which is divisible by Z

5. Public key = (N, E) = (33, 3) and Private key = (N, D) = (33, 7)

Now say we want to encrypt the message m = 7,
Cipher code $= M^{\wedge E} \bmod N$
$= 7^{\wedge 3} \bmod 33$
$= 343 \bmod 33$
$= 13$

Hence the ciphertext c = 13

To check decryption we compute Message' $= C^{\wedge D} \bmod N$
$= 13^{\wedge 7} \bmod 33$
$= 7$

Sample Output:

Enter a Prime number: 5

Enter another prime number: 11

Encryption keys are: 33, 55

Decryption keys are: 17, 55

Result:

Thus the Java program to implement cryptographic algorithm using RSA algorithm was executed successfully

Ex.No. 2(c)

Date:

Diffie-Hellman Key Exchange

Aim:

To write a Java program to implement Diffie hellman key exchange algorithm

Algorithm:

1. Diffie-Hellman key exchange (DH) is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key
2. The algorithm generates a public key and a private key for the client
3. Create a KeyPairGenerator Object that generates private/public keys for the DH algorithm, using the *getInstance(String algortihm)* API method
4. Initialize the KeyGenerator so as to generate keys with a 1024-bit length, using the *initialize(int keysize)* API method
5. Create a KeyPair Object , with the *genKeyPair()* API method, that generates the key pair.
6. Create the PrivateKey and PublicKey Objects of the key pair, with the *getPrivate()* and *getPublic()* API methods of the KeyPair
7. Return for both keys the names of their primary encoded formats, using for both their *getformat()* API methods

Sample Output:

Private key format :PKCS#8

Diffie-Helman Private key parameters are:SunJCE Diffie-Hellman Private Key:

x: a391eed7 d10d95d3 3952005c 117c56ad a3d686c5 8a60d504 2fde2db6 11686543
0025c0b7 e038f63f cb82151b a7cb24fb f6c2ab69 9c517155 67818cec 782cf977

p: fd7f5381 1d751229 52df4a9c 2eece4e7 f611b752 3cef4400 c31e3f80 b6512669
455d4022 51fb593d 8d58fabf c5f5ba30 f6cb9b55 6cd7813b 801d346f f26660b7
6b9950a5 a49f9fe8 047b1022 c24fbba9 d7feb7c6 1bf83b57 e7c6a8a6 150f04fb
83f6d3c5 1ec30235 54135a16 9132f675 f3ae2b61 d72aeff2 2203199d d14801c7

g: f7e1a085 d69b3dde cbbcab5c 36b857b9 7994afbb fa3aea82 f9574c0b 3d078267
5159578e bad4594f e6710710 8180b449 167123e8 4c281613 b7cf0932 8cc8a6e1
3c167a8b 547c8d28 e0a3ae1e 2bb3a675 916ea37f 0bfa2135 62f1fb62 7a01243b
cca4f1be a8519089 a883dfe1 5ae59f06 928b665e 807b5525 64014c3b fecf492a

l: 512

Public key format :X.509

Diffie-Helman Public key parameters are:SunJCE Diffie-Hellman Public Key:

y: d3fabd76 139865f1 63507aa2 6a9480a9 f180692b ba0e6979 f335ee25 2e26762c
f7df3af9 d7ea612e 7540f071 f50051ae 7d061113 fd0d2d0c d0cc4ae1 03406d44
84398cc6 59a93dcd 6ec827d1 06edb2f0 02d48ee5 f2c9cb94 785f39df cc88ec65
5a224a1c 318b51fe 9c40445b fedb5f14 3fe83f51 82d0357c 1004652e 93c9ad81

p: fd7f5381 1d751229 52df4a9c 2eece4e7 f611b752 3cef4400 c31e3f80 b6512669
455d4022 51fb593d 8d58fabf c5f5ba30 f6cb9b55 6cd7813b 801d346f f26660b7
6b9950a5 a49f9fe8 047b1022 c24fbba9 d7feb7c6 1bf83b57 e7c6a8a6 150f04fb
83f6d3c5 1ec30235 54135a16 9132f675 f3ae2b61 d72aeff2 2203199d d14801c7

g: f7e1a085 d69b3dde cbbcab5c 36b857b9 7994afbb fa3aea82 f9574c0b 3d078267
5159578e bad4594f e6710710 8180b449 167123e8 4c281613 b7cf0932 8cc8a6e1
3c167a8b 547c8d28 e0a3ae1e 2bb3a675 916ea37f 0bfa2135 62f1fb62 7a01243b
cca4f1be a8519089 a883dfe1 5ae59f06 928b665e 807b5525 64014c3b fecf492a

l: 512

Result:

Thus the Java program to implement cryptographic algorithm using diffie hellman algorithm was executed successfully

Ex.No. 2(d)

Date:

MD5

Aim:

To write a Java program to implement message digest 5 algorithm

Algorithm:

1. Append padded bits - The message is padded so that its length is congruent to 448, modulo 512. Means extended to just 64 bits shy of being of 512 bits long. – A single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits equals 448 modulo 512
2. Append length - A 64 bit representation of b is appended to the result of the previous step. The resulting message has a length that is an exact multiple of 512 bits
3. Initialize MD Buffer - A four-word buffer (A,B,C,D) is used to compute the message digest. – Here each of A,B,C,D, is a 32 bit register. These registers are initialized to the following values in hexadecimal:
   word A: 01 23 45 67 word B: 89 ab cd ef word C: fe dc ba 98 word D: 76 54 32 10
4. Process message in 16-word blocks – Four auxiliary functions that take as input three 32-bit words and produce as output one 32-bit word. F(X,Y,Z) = XY v not(X) Z G(X,Y,Z) = XZ v Y not(Z) H(X,Y,Z) = X xor Y xor Z I(X,Y,Z) = Y xor (X v not(Z))
5. Process message in 16-word blocks cont – if the bits of X, Y, and Z are independent and unbiased, the each bit of F(X,Y,Z), G(X,Y,Z), H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased

Sample Output:

Message digest object info:

Algorithm = MD5

Provider = SUN version 1.6

ToString = MD5 Message Digest from SUN, <initialized>

MD5("") = D41D8CD98F00B204E9800998ECF8427E

MD5("abc") = 900150983CD24FB0D6963F7D28E17F72

MD5("abcdefghijklmnopqrstuvwxyz") = C3FCD3D76192E4007DFB496CCA67E13Be) SHA 1

Result:

Thus the Java program to implement MD5 algorithm was executed successfully

Ex.No. 2(e)

Date:

SHA-1

Aim:

To write a Java program to implement Secure Hash Algorithm 1

Algorithm:

1. Append Padding Bits Message is "padded" with a 1 and as many 0's as necessary to bring the message length to 64 bits fewer than an even multiple of 512
2. Append Length 64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating the length of the original message
3. Prepare Processing Function:

   SHA1 requires 80 processing functions defined as

   f(t;B,C,D) = (B AND C) OR ((NOT B) AND D) (0 <= t <= 19)

   f(t;B,C,D) = B XOR C XOR D (20 <= t <= 39)

   f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <=59)

   f(t;B,C,D) = B XOR C XOR D (60 <= t <= 79)
4. Prepare Processing Constants:

   SHA1 requires 80 processing constant words defined as

   K(t) = 0x5A827999 (0 <= t <= 19)

   K(t) = 0x6ED9EBA1 (20 <= t <= 39)

   K(t) = 0x8F1BBCDC (40 <= t <= 59)

   K(t) = 0xCA62C1D6 (60 <= t <= 79)

Sample Output:

Message digest object info:

Algorithm = SHA1

Provider = SUN version 1.6

ToString = SHA1 Message Digest from SUN, <initialized>

SHA1("") = DA39A3EE5E6B4B0D3255BFEF95601890AFD80709

SHA1("abc") = A9993E364706816ABA3E25717850C26C9CD0D89D

SHA1("abcdefghijklmnopqrstuvwxyz")=32D10C7B8CF96570CA04CE37F2A19D8424

0D3A89

Result:

Thus the Java program to implement using SHA-1 algorithm was executed successfully

Viva-Voce:

1. What is DES?

DES is the Data Encryption Standard, an encryption block cipher defined and endorsed by the U.S. government in 1977 as an official standard; the details can be found in the latest official FIPS (Federal Information Processing Standards) publication concerning DES.

2. What is Diffie-Hellman?

It is a method by which a key can be securely shared by two users without any actual exchange.

3. What is RSA algorithm?

RSA is short for Rivest-Shamir-Adleman algorithm. It is the most commonly used public key encryption algorithm in use today.

4. How do you use RSA for both authentication and secrecy?

RSA is a public key encryption algorithm. The RSA algorithms are based on the mathematical part that it is easy to find and multiply large prime numbers together, but it is extremely difficult to factor their product.
For authentication: One can encrypt the hash (MD4/SHA) of the data with a private key. This is known as digital signature.
For Secrecy: Secrecy/confidentiality is achieved by encrypting the data with public key and decrypting with private key.

5. What is SHA-1?

SHA is a secure hash algorithm developed by National Institute of Standard technology SHA-1 is a type of SHA message digest size of SHA 1 Is 160 bits and message size is < 2^64 it uses block cipher.

Ex.No. 3

Date:

Implement the SIGNATURE SCHEME - Digital Signature Standard

Aim:

To write a Java program to implement digital signature algorithm

Algorithm:

1. The first part of the DSA algorithm is the public key and private key generation, which can be described as:
   - Choose a prime number q, which is called the prime divisor
   - Choose another primer number p, such that p-1 mod q = 0. p is called the prime modulus
   - Choose an integer g, such that 1 < g < p, g**q mod p = 1 and g = h**((p–1)/q) mod p. q is also called g's multiplicative order modulo p
   - Choose an integer, such that 0 < x < q
   - Compute y as g**x mod p
   - Package the public key as {p,q,g,y}
   - Package the private key as {p,q,g,x}
2. The second part of the DSA algorithm are the signature generation and signature verification. To generate a message signature, the sender can follow these steps:
   - Generate the message digest h, using a hash algorithm like SHA1
   - Generate a random number k, such that 0 < k < q
   - Compute r as (g**k mod p) mod q. If r = 0, select a different k
   - Compute i, such that k*i mod q = 1. i is called modular multiplicative inverse of k modulo q
   - Compute s = i*(h+r*x) mod q. If s = 0, select a different k
   - Package the digital signature as {r,s}
3. To verify a message signature, receiver of the message and digital signature can follow these steps:
   - Generate the message digest h, using the same hash algorithm
   - Compute w, such that s*w mod q = 1. w is called the modular multiplicative inverse of s modulo q
   - Compute u1 = h*w mod q
   - Compute u2 = r*w mod q
   - Compute v = (((g**u1)*(y**u2)) mod p) mod q

- If v == r, the digital signature is valid

Sample Output:

Signature:

imwaKe99tkM6H6hiiP0rubmb/MrYJZLiwLdRSjslF2KlA5B23az5M2LKftQFCB+NHCe5F5/YfN8Os
NSNLtucrrZTah0SrdWSzdGCOfYLdUZmPQ72j1SkLhYspsTsUb/U6FPSYT4QebNSYobDtjKujkHdR
imHI9TO4lLuqVQRdWU= true

Result:

Thus the Java programs to implement digital signature algorithm was executed successfully

Viva-Voce:

1. What is the difference between DSA and RSA?

DSA means Digital Signature Algorithm. The basic difference between DAS and RSA is RSA can encrypt and sign, however DSA is only used for digital signature.

2. What is Digital Signatures?

Digital signature is an attachment to an electronic message used for security purpose. It is used to verify the authenticity of the sender.

3. What is message authentication?

When the verifier validates the digital signature using public key of a sender, he is assured that signature has been created only by sender who possess the corresponding secret private key and no one else.

4. What are the services provided by digital certificates?

Digital certificate provide authentication, nonrepudiation and integrity services.

Ex.No. 4

Date:

Providing secure data storage, secure data transmission and creating digital signatures

Aim:

To implement secure data storage, transmission and create digital signatures

Algorithm:

1. The first part of the DSA algorithm is the public key and private key generation, which can be described as:
2. Choose a prime number q, which is called the prime divisor
3. Choose another primer number p, such that p-1 mod q = 0. p is called the prime modulus
4. The second part of the DSA algorithm is the signature generation and signature verification
5. To generate a message signature
6. Generate the message digest h, using a hash algorithm and compute it

Result:

Thus the experiment to perform secure data storage, transmission were done and digital signature was created successfully

Viva-Voce:

1. What are Digital certificates?

Digital certificates are digital documents attesting to the binding of a public key to an individual or other entity. They allow verification of the claim that a given public key does in fact belong to a given individual. Certificates help prevent someone from using a phony key to impersonate someone else.

2. What is Secure Sockets Layer (SSL)?

The Secure Sockets Layer (SSL) is a computer networking protocol that manages server authentication, client authentication and encrypted communication between servers and clients.

3. What is the technology available to ensure data privacy and integrity during transmission?

1. Controlling Access within the Network
2. Encrypting Data for Network Transmission
3. Secure Sockets Layer (SSL) Protocol
4. Firewall

4. What are the services provided by digital certificates?

Digital certificate provide authentication, nonrepudiation and integrity services.

Ex.No. 5

Date:

Setup a Honeypot and Monitor the Honeypot on Network

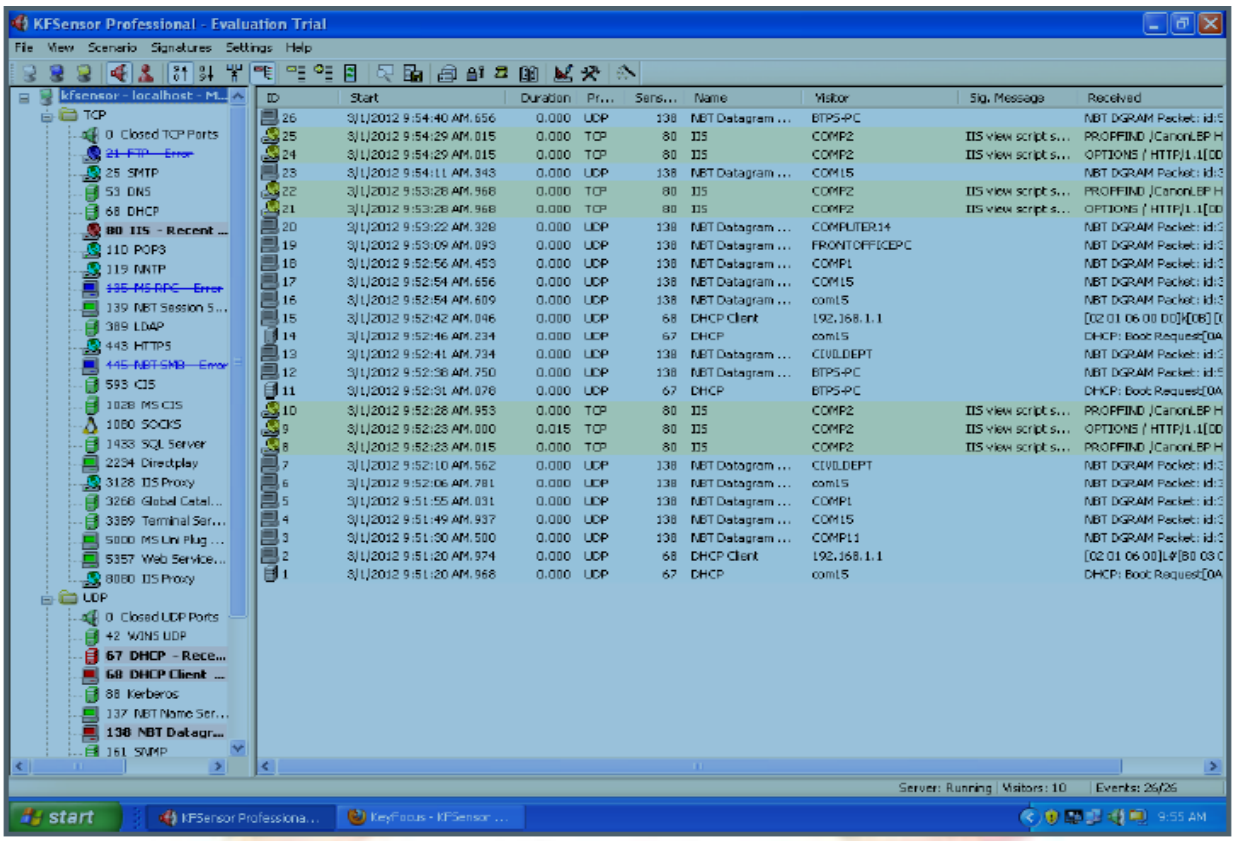
Aim:

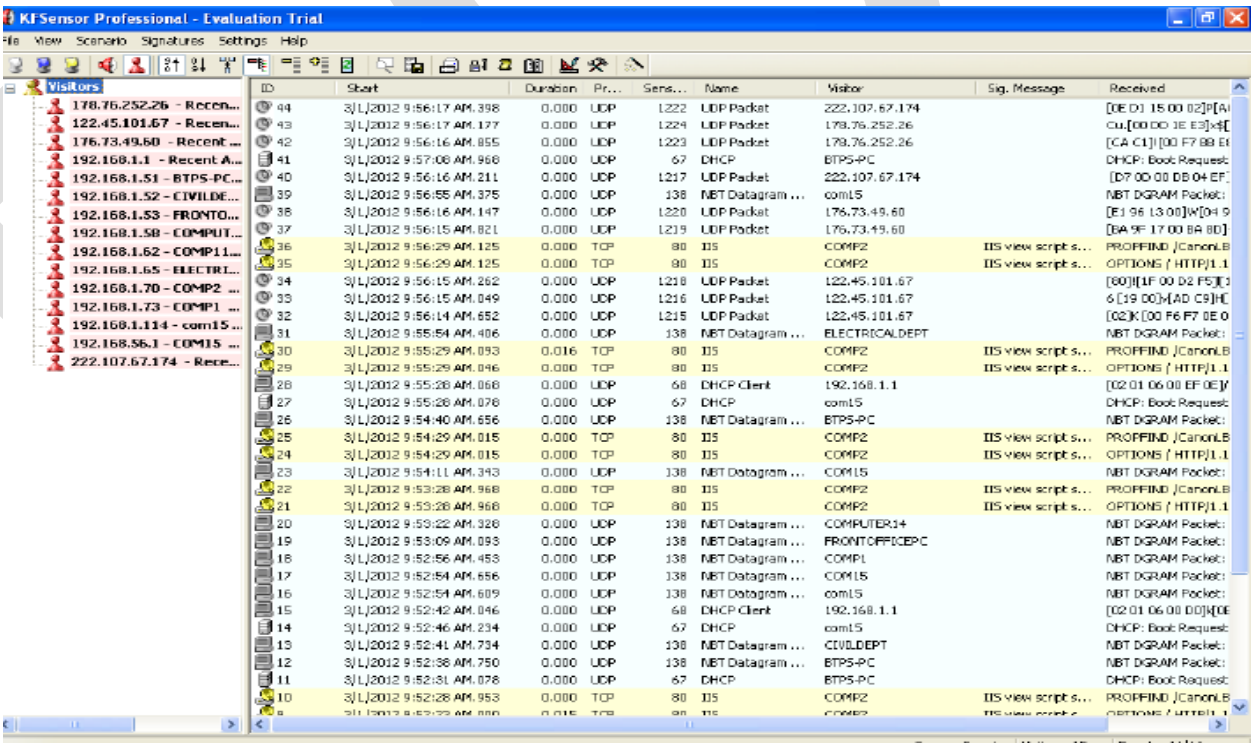
To set up a honeypot and monitor the honeypot on a given network

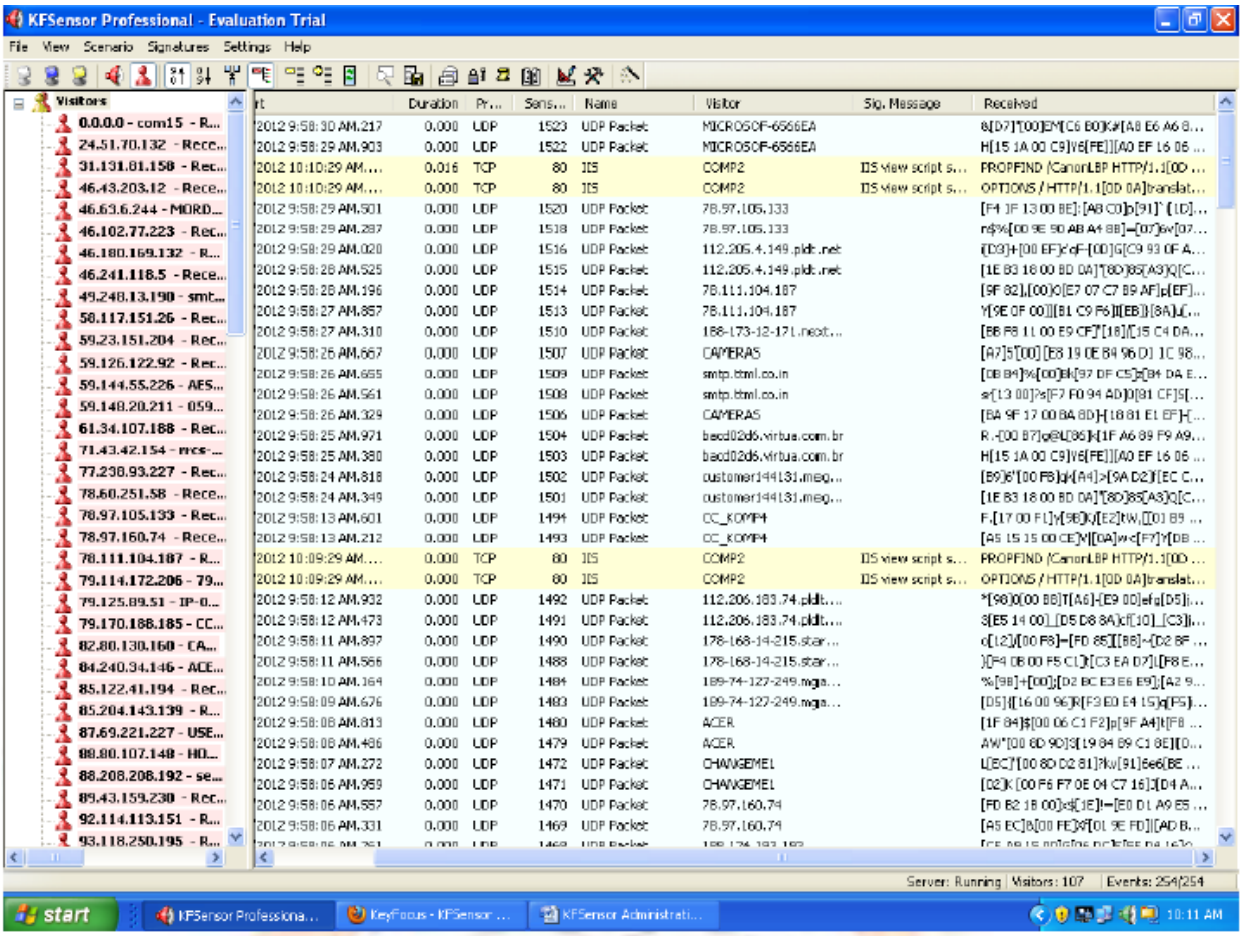
Algorithm:

1. Honeypot is a device placed on computer network specifically designed to capture malicious network traffic
2. KF Sensor is the tool to setup as honeypot when KF Sensor is running it places a siren icon in the windows system tray in the bottom right of the screen. If there are no alerts then green icon is displayed
3. Download KF Sensor Evaluation Setup File from KF Sensor Website
4. Install with License Agreement and appropriate directory path
5. Reboot the computer now
6. The KF Sensor automatically starts during windows boot Click Next to setup wizard
7. Select all port classes to include and Click Next
8. Send the email and Send from email enter the ID and Click Next
9. Select the options such as Denial of Service[DOS], Port Activity, Proxy Emulsion, Network Port Analyzer, Click Next
10. Select Install as system service and Click Next
11. Click finish

Sample Output:







Result:

Thus the experiment to setup a Honeypot and monitor the Honeypot on network was done successfully

Viva-Voce:

1. **What do you understand by honeypot on network?**

   A honeypot is a device placed on a computer network specifically designed to capture malicious network traffic. The logging capability of a honeypot is far greater than any other network security tool and captures raw packet level data even including the keystrokes and mistakes made by hackers. The captured information is highly valuable as it contains only malicious traffic with little to no false positives. Honeypots are becoming one of the leading security tools used to monitor the latest tricks and exploits of hackers by recording their every move so that the security community can more quickly respond to new exploits.

2. **How is honeypot works?**

   Honeypot works by opening over 1000 UDP and TCP listening sockets on your computer and these sockets are designed to mimic vulnerable services. When an attacker connects to these services they are fooled into thinking they are attacking a real server. The honeypot safely captures all communications with the attacker and logs these results for future analysis. Should an attacker attempt an exploit or upload a rootkit or trojan to the server the honeypot environment will safely store these files on your computer for analysis and submission to antivirus vendors.

3. **How will you install honey pots?**

   (i)Honeypot can be downloaded from the web site at: http://www.atomicsoftwaresolutions.com/honeybot.php

   (ii) After clicking the download link save Honeypot_010.exe to a location on your hard drive

   (iii) Double click the Honeypot_010.exe installation file to begin the setup process.

   (iv) Follow the prompts in the setup process. The default installation folder for setup is c:\honeybot\

   (v) Setup will create a shortcut in the Start Menu folder and an option is available to create a desktop icon

   (vi) Now you can launch Honeypot using the programs shortcut icon

   (vii) Click on the blue play button to start the Honeypot listening engine

   (viii) Using a Web Browser try to access various network systems by providing their IP addresses

4. **What are log files?**

   All log files are saved by default to c:\honeybot\logs folder. Log files store information relating to the hits on the system and also store all data received and sent to the attacking computer.

5. **How will you uninstall honeypots?**

   Click on the red stop button to shut down all listening services and terminate all existing open sockets. Uninstalling Honeypot Click the Uninstall Honeypot icon in the programs start menu to uninstall Honeypot and follow the prompts.

Ex.No. 6

Date:

Installation of rootkits and study the variety of options

Aim:

To install the rootkits and study the variety of options

Procedure:

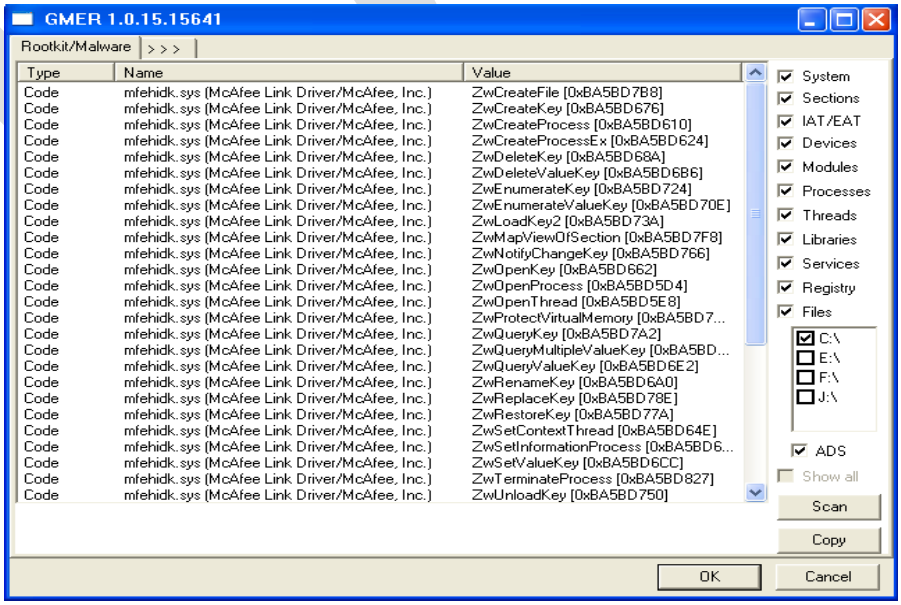
A rootkit is a stealthy type of malicious software (malware) designed to hide the existence of certain processes or programs from normal methods of detection and enables continued privileged access to a computer. The term rootkit is a concatenation of "root" (the traditional name of the privileged account on Unix operating systems) and the word "kit" (which refers to the software components that implement the tool). The term "rootkit" has negative connotations through its association with malware.

A rootkit is a collection of tools (programs) that enable administrator-level access to a computer or computer network. Typically, a cracker installs a rootkit on a computer after first obtaining user-level access, either by exploiting a known vulnerability or cracking a password. Once the rootkit is installed, it allows the attacker to mask intrusion and gain root or privileged access to the computer and, possibly, other machines on the network.

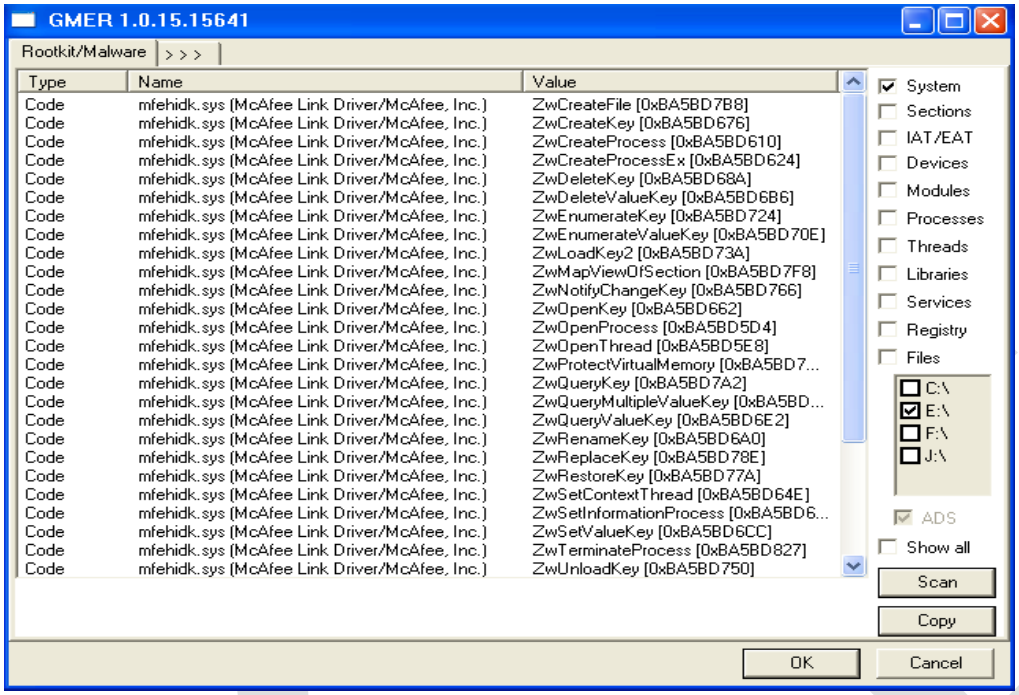
A rootkit may consist of spyware and other programs that: monitor traffic and keystrokes; create a "backdoor" into the system for the hacker's use; alter log files; attack other machines on the network; and alter existing system tools to escape detection.

Steps:

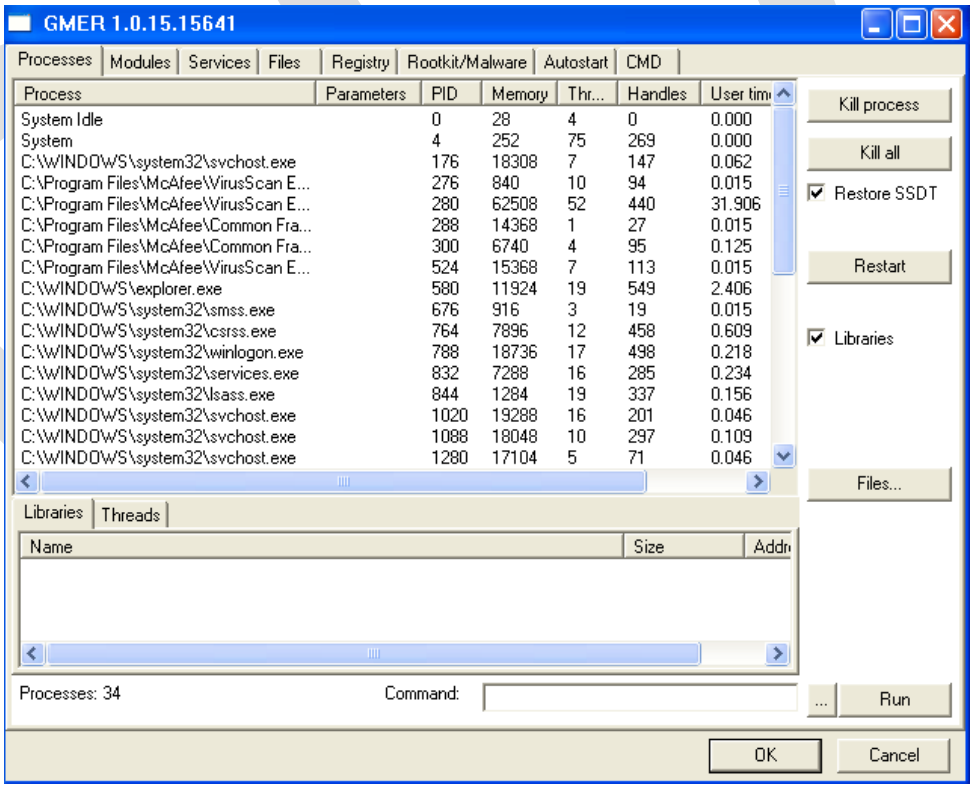
1. Double click on rootkit folder
2. Double click on the GMER rootkit application
3. Now the rootkit screen will be displayed



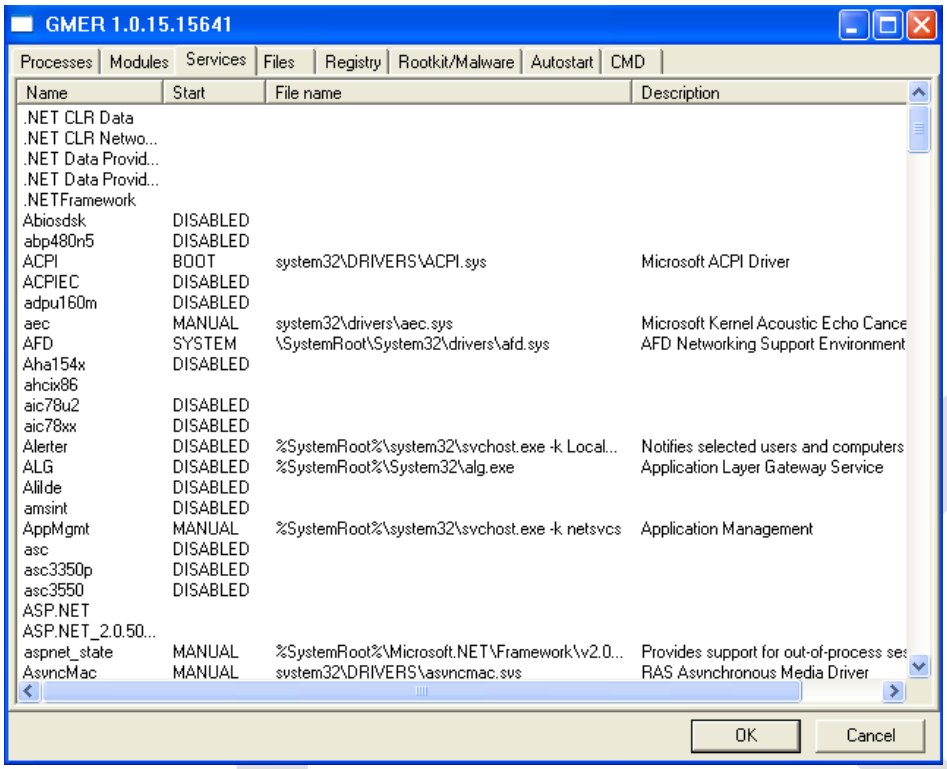
4. Select anyone of the drive which is shown at right side of the screen
5. After selecting the drive click on scan button



6. Click on the option processes the screen will be displayed



7. Click on the option services



8. Now click on different options to perform different actions

Result:

Thus the experiment of installing the rootkits and the variety of options were studied

Viva-Voce

1. **What is Rootkit?**

   The term Rootkit originally referred to a collection of tools used to gain administrative access on UNIX operating systems. The collection of tools often included well-known system monitoring tools that were modified to hide the actions of an unauthorized user. An unauthorized user would replace the existing tools on the system with the modified versions preventing authorized users from discovering the security breach.
   Rootkit - "A tool used to protect backdoors and other tools from detection by administrators"

2. **What are called Rootkits in Windows?**

   They refers to programs that use system hooking or modification to hide files, processes, registry keys, and other objects in order to hide programs and behaviors. In particular, Windows rootkits do not necessarily include any functionality to gain administrative privileges. In fact, many Windows rootkits require administrative privileges to even function.

3. **What are the basic classes of Rootkits?**

   Two basic classes of Windows rootkits: kernel mode rootkits & user mode rootkits.

4. **Why Rootkits are used?**

   Rootkits are used by criminals for a variety of purposes, usually to turn a computer into part of a botnet, which can then, in turn, go on to infect other computers or send spam email messages. The rootkit owner can install key loggers to capture user-entered passwords for online banking and similar activities, or steal the user's personal details to use for identity fraud.

5. **How Rootkits stay undetected?**

   Many rootkits infect the boot sectors of the computer's hard disk, allowing them to load before the computers operating system. The rootkit then patches the operating system and changes common functions to hide its existence. For example, the root kit could intercept calls for a list of files in a directory, removing its own file names before showing the results to the user, so it would appear as if the directory is clean.

6. **What are the Rootkit capabilities?**

   Current Rootkit Capabilities: Rootkits Hide processes, Hide files, Hide registry entries, Hide services, Completely bypass personal firewalls, Undetectable by antivirus, Remotely undetectable, Covert channels - undetectable on the network, Defeat cryptographic hash checking, Install silently, All capabilities ever used by viruses or worms.

Ex.No. 7

Date:

Perform wireless audit on an access point or a router and decrypt WEP and WPA (Net Stumbler)

Aim:

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

Algorithm:

A packet sniffer, sometimes referred to as a network monitor or network analyzer, can be used by a network or system administrator to monitor and troubleshoot network traffic. Using the information captured by the packet sniffer an administrator can identify erroneous packets and use the data to pinpoint bottlenecks and help maintain efficient network data transmission. In its simple form a packet sniffer simply captures all of the packets of data that pass through a given network interface. By placing a packet sniffer on a network in promiscuous mode, a Malicious intruder can capture and analyze all of the network traffic. Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. Download and install wireshark network analyzer. Steps to capture traffic:

1. Open Wireshark network analyzer
2. Select interface: Go to capture option in menu bar and select interface
3. Start Capturing

Result:

Thus the experiment to perform wireless audit on an access point or a router and decrypt WEP and WPA (Net Stumbler) was done successfully

Viva-Voce:

1. **What is WAP?**

   The Net Stumbler provides following details of WLAN (wireless LAN):
   MAC : The Media Access Control or MAC address is a unique code assigned to networking hardware, in this case the MAC address is referring to the address assigned to the Wireless AP (WAP). So beside the green bubble we see the 12 character MAC address for that AP.

2. **What is BSSID?**

   The text contains the BSSID (Basic Service Set Identifier) for wireless devices. The icon shows the signal strength as reported in the last scan: Gray means the item was not detected, or a colored icon ranging from red to green reports the signal strength. A lock appears in the icon if encryption is enabled on the network. For devices on a wired network segment, the icon shows a T-shaped network cable and the MAC address is displayed.

3. **What is SSID?**

   SSID (Service Set Identifier): The reported SSID. This may be blank for access points that report their existence but not their SSID. For wired network items, the SSID is assumed to be the SSID that was associated when the item was discovered.

4. **What is Latitude, Longitude Distance?**

   If you are using a GPS receiver, this indicates the estimated position of the object. This position is currently the location where the strongest signal was seen, which is never the actual location. Distance is measured from your current position to the object's estimated position.

5. **What is WEP?**

   Access Points (APs) that do have encryption enabled. One of the flaws with the latest version of Net Stumbler is that all enabled encryption is displayed as WEP. Decrypt 802.11 Wireshark can decrypt WEP and WPA/WPA2 in pre-shared (or personal) mode. WPA/WPA2 enterprise mode decryption is not yet supported. You can add decryption keys using Wireshark's 802.11 preferences or by using the wireless toolbar. Up to 64 keys are supported.

Ex.No. 8

Date:

Demonstrate intrusion detection system

Aim:

To demonstrate intrusion detection system using the tool like snort or any software

Algorithm:

1. Start one of the tool, clear all history captures.
2. As new capture file captures all the communication with the network, hence, stop all other communications with the network
3. Now open internet explorer and go to Gmail and sign in with your account
4. Compose a new mail which includes a model attachment file (this file is common for all tools)
5. Send the mail to yourself and sign out
6. Stop capture procedure in the tool
7. Continue the same procedure with same model attachment file for the rest of the tools

Result:

Thus the experiment of demonstrating intrusion detection system was executed successfully

Viva-Voce:

1. **What is Intrusion Detection System (IDS)?**

   With the development of network technologies and applications, network attacks are greatly increasing both in number and severity. As a key technique in network security domain, Intrusion Detection System (IDS) plays vital role of detecting various kinds of attacks and secures the networks. Main purpose of IDS is to find out intrusions among normal audit data and this can be considered as classification problem.

2. **What are the activities of IDS?**

   Intrusion detection systems (IDS) are an effective security technology, which can detect, prevent and possibly react to the attack. It performs monitoring of target sources of activities, such as audit and network traffic data in computer or network systems, requiring security measures, and employs various techniques for providing security services.

3. **What is intrusion or intruder?**

   Intrusion : Attempting to break into or misuse your system. Intruders may be from outside the network or legitimate users of the network. Intrusion can be a physical, system or remote intrusion. Intrusion Detection Systems look for attack signatures, which are specific patterns that usually indicate malicious or suspicious intent.

4. **What is Snort?**

   Snort is an open source network intrusion prevention system, capable of performing real time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching, and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more.

5. **What are the uses of Snort?**

   Snort has three primary uses: It can be used as a straight packet sniffer like tcpdump, a packet logger (useful for network traffic debugging, etc.), or as a full blown network intrusion prevention system. The privacy of the Snort community is very important to source fire.

6. **What are the three modes of Snort?**

   Snort can be configured to run in three modes: 1. Sniffer mode 2. Packet Logger mode 3. Network Intrusion Detection System mode Sniffer mode: snort –v Print out the TCP/IP packets header on the screen. Packet Logger mode : snort –dev –l c:\log [create this directory in the C drive] and snort will automatically know to go into packet logger mode, it collects every packet it sees and places it in log directory. Network Intrusion Detection System mode : snort –d c:\log –h ipaddress/24 –c nort conf. This is a configuration file applies rule to each packet to decide it an action based upon the rule type in the file.