–

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# CS6513 – Security Laboratory

VII SEMESTER - R 2013

**LABORATORY MANUAL**

Name : ________________________________________

Register No. : ________________________________________

Section : ________________________________________

## VISION

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

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

## MISSION

- To educate the students with the state-of-art technologies to meet the growing challenges of the electronics industry
- To carry out research through continuous interaction with research institutes and industry, on advances in communication systems
- To provide the students with strong ground rules to facilitate them for systematic learning, innovation and ethical practices

# PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

## 1. Fundamentals

To impart students with fundamental knowledge in Mathematics, Science and fundamentals of engineering that will would 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 solution for them

## 3. Breadth

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

## 4. Professionalism 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 develop 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

# PROGRAMME OUTCOMES (POs)

a) To apply 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 problems and develop an efficient practical solutions

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

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

g) To develop a system that will meet expected need with 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 engage lifelong learning and exhibit their technical skills

j) To develop and manage projects in multidisciplinary environments

# 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

## Expt. No. 1(a)

# IMPLEMENTATION OF SUBSTITUTION AND TRANSPOSITION TECHNIQUES CAESAR CIPHER

## Aim:

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

## Software requirements:

C / C++ / Java or equivalent compiler

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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 Ms in the message with the corresponding letter from the cipher alphabet
6. So M is replaced by P and we would replace all the E's 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

## Outcome:

Thus the outcome of implementing caesar cipher has been attained.

## Application:

Communicating the message in between the users with privacy.

## Expt. No. 1(b)

# IMPLEMENTATION OF SUBSTITUTION AND TRANSPOSITION TECHNIQUES PLAYFAIR CIPHER

## Aim:

To write a program to implement playfair cipher algorithm

## Software requirements:

C / C++ / Java or equivalent compiler

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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

## Outcome:

Thus the outcome of playfair cipher has been attained.

## Application:

Communicating the message in between the users.

## Expt. No. 1(c)

# IMPLEMENTATION OF SUBSTITUTION AND TRANSPOSITION TECHNIQUES HILL CIPHER

## Aim:

To write a program to implement hill cipher algorithm

## Software requirements:

C / C++ / Java or equivalent compiler.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## Algorithm:

1. In a Hill cipher encryption, the plaintext message is broken up into blocks of length n, according to the 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 cipher text message is produced.
4. For decryption of the cipher text message, the inverse of the encryption matrix must be found once found, the decryption matrix is then dotted with each $n$-block of cipher text, 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

## Outcome:

Thus the outcome of hill cipher has been attained.

## Application:

Communicating the messages in between the users with trust using above algorithm.

## Expt. No. 1(d)

# IMPLEMENTATION OF SUBSTITUTION AND TRANSPOSITION TECHNIQUES VIGNERE CIPHER

## Aim:

To write a java program to implement vignere cipher

## Software requirements:

C / C++ / Java or equivalent compiler.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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

## Outcome:

Thus the outcome of vignere cipher has been attained.

## Application:

Communicating the messages in between the users with privacy.

## Expt. No. 1(e)

# IMPLEMENTATION RAIL FENCE TRANSFORMATION TECHNIQUES

## Aim:

To write a java program to implement rail fence algorithm

## Software requirements:

C / C++ / Java or equivalent compiler.

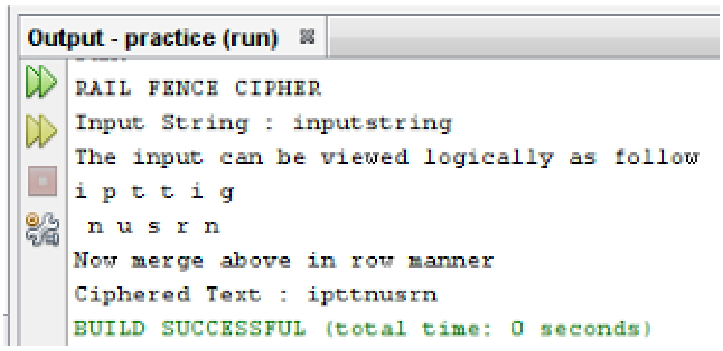
## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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.

## Sample Output:



```
RAIL FENCE CIPHER
Input String : inputstring
The input can be viewed logically as follow
i p t t i g
 n u s r n
Now merge above in row manner
Ciphered Text : ipttnusrn
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Result:

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

## Outcome:

Thus the outcome of rail fence has been attained.

## Application:

Communicating the messages in between the users with privacy.

## Viva-voce

1. What is public-key cryptography?

2. What is block cipher?

3. What is stream cipher?

4. Name a most widely used stream cipher.

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

6. What are Brute Force Attacks?

7. What is the length of key in playfair cipher?

8. What is the length of matrix in playfair cipher?

9. What is the length of key in hill cipher?

10. What is the length of key in caesar cipher?

11. What is the length of key in monoalphabetic cipher?

12. What is the length of key in one time pad cipher?

13. What is the length of key in polyalphabetic cipher?

14. What is the length of key in railfence cipher?

## Expt. No. 2(a)

# IMPLEMENTATION OF DES

## Aim:

To write a program to implement DES algorithm

## Software requirements:

C / C++ / Java or equivalent compiler.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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 called Initial Permutation (IP).

## Sample Output:

**Input.txt**

```
JavaCode
```

**encrypted.txt**

```
—w~Z5-ó&Ïɛ E
```

**decrypted.txt**

```
JavaCode
```

**Result:**

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

## Outcome:

Thus the outcome of DES has been attained.

## Application:

Developing any private application with high security.

## Expt. No. 2(b)

# IMPLEMENTATION OF RSA ALGORITHM

## Aim:

To write a program to implement RSA algorithm

## Software requirements:

C / C++ / Java or equivalent compiler.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## Algorithm:

1. Generate two large random primes, P and Q, of approximately equal size.
2. Compute $N = P \times Q$.
3. Compute $Z = (P\text{-}1) \times (Q\text{-}1)$.
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. $N = P \times Q = 11 \times 3 = 33$
   $Z = (P\text{-}1) \times (Q\text{-}1) = 10 \times 2 = 20$
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 $E$ x $D \equiv 1$ (mod $Z$)
   Compute $D = E^{\wedge -1}$ mod $Z = 3^{\wedge -1}$ mod 20
   Find a value for $D$ such that $Z$ divides (($E$ x $D$)-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 = $M^{\wedge E}$ mod $N$
= $7^{\wedge 3}$ mod 33
= 343 mod 33
= 13

Hence the ciphertext c = 13

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

## Outcome:

Thus the outcome of RSA algorithm has been attained.

## Application:

It's a public key cryptography used in banking application.

## Expt. No. 2(c)

# IMPLEMENTATION OF DIFFIEE-HELLMAN

## Aim:

To write a program to implement Diffiee- hellman key exchange algorithm

## Software requirements:

C / C++ / Java or equivalent compiler.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## Algorithm:

1. Diffiee-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 f6cb9b55 f26660b7 6b9950a5 a49f9fe8 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  ba0e6979  f335ee25  2e26762c  f7df3af9  d7ea612e
7540f071 f50051ae 7d061113 d0cc4ae1 03406d44 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  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 b7cf0932 8cc8a6e1 c167a8b 547c8d28 e0a3ae1e 2bb3a675
916ea37f 0bfa2135 62f1fb62 ca4f1be a883dfe1 5ae59f06 928b665e 807b5525 64014c3b fecf492a
l:   512
```

## Result:

Thus the program to implement cryptographic algorithm using diffiee-hellman algorithm was executed successfully.

## Outcome:

Thus the outcome of diffiee-hellman has been attained.

## Application:

Communicating the message in between two parties.

## Expt. No. 2(d)

# IMPLEMENTATION OF MD5

## Aim:

To write a program to implement MD5 algorithm

## Software requirements:

C / C++ / Java or equivalent compiler.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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.

## Outcome:

Thus the outcome of MD5 has been attained.

## Application:

Developing any application with security.

## Expt. No. 2(e)

# IMPLEMENTATION OF SHA-1

## Aim:

To write a program to implement Secure Hash Algorithm-1

## Software requirements:

C / C++ / Java or equivalent compiler.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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 was executed successfully.

## Outcome:

Thus the outcome of SHA-1 has been attained.

## Application:

Presently used in banking application.

## Viva-voce

1. What is DES?

2. What is the input key length in DES?

3. What is length plain text length in DES?

4. What is Diffie-Hellman?

5. What is RSA algorithm?

6. What is SHA-1?

7. What is the output length ofplain text in DES?

8. Abbreviate RSA.

9. Abbreviate SHA-1.

## Expt. No. 3

# IMPLEMENT THE SIGNATURE SCHEME – DIGITAL SIGNATURE STANDARD

## Aim:

To write a program to implement digital signature algorithm

## Software requirements:

C / C++ / Java or equivalent compiler.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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/YfN8OsNSNLtucrrZT
ah0SrdWSzdGCOfYLdUZmPQ72j1SkLhYspsTsUb/U6FPSYT4QebNSYobDtjKujkHdRimHI9TO4lLuqVQRdW
U= true
```

## Result:

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

## Outcome:

Thus the outcome of digital signature algorithm has been attained.

## Application:

Communicating any application between the users with trust.

## Viva-voce

1. What is the difference between DSA and RSA?
2. What is Digital Signatures?
3. What is message authentication?
4. What are the services provided by digital certificates?
5. What are the types of DSA?
6. What are notations used in DSA?
7. Define – Digital certificate
8. Define – Signature

## Expt. No. 4

# DEMONSTRATE HOW TO PROVIDE SECURE DATA STORAGE, SECURE DATA TRANSMISSION AND FOR CREATING DIGITAL SIGNATURES (GnuPG)

## Aim:

To implement secure data storage, transmission and for create digital signatures

## Software requirements:

C, C++ compilers, Java or equivalent compiler GnuPG.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## Algorithm:

1. The first part of the DSA algorithm is the public key and private key generation, which can be described.
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.

## Outcome:

Thus the outcome of secure data storage, transmission has been attained.

## Application:

It can be used in defense for communicating securely.

## Viva-voce

1. What are Digital certificates?
2. What is Secure Sockets Layer (SSL)?
3. What is the technology available to ensure data privacy and integrity during transmission?
4. What are the services provided by digital certificates?

## Expt. No. 5

# SETUP A HONEYPOT AND MONITOR THE HONEYPOT ON NETWORK

## Aim:

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

## Software requirements:

C / C++ / Java or equivalent compiler GnuPG, KF Sensor or Equivalent, Snort, Net Stumbler or Equivalent.

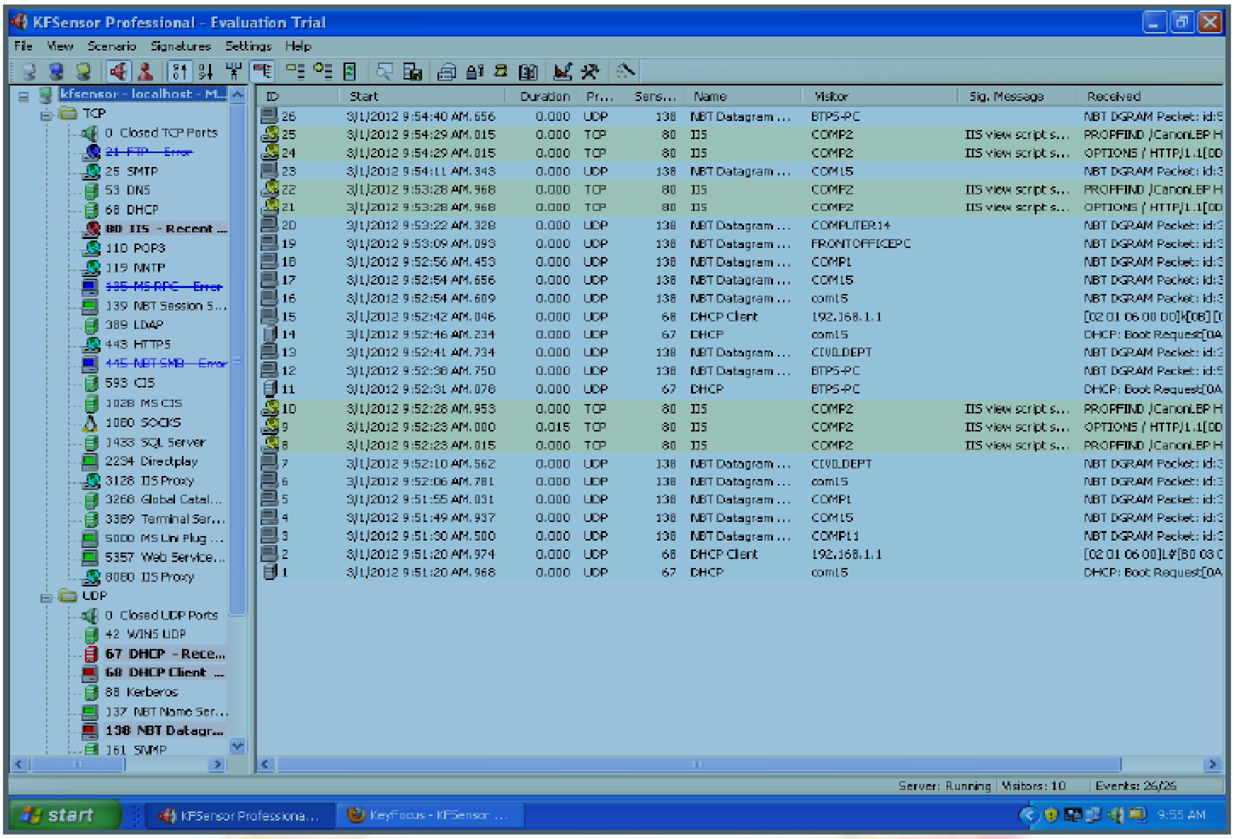
## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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:

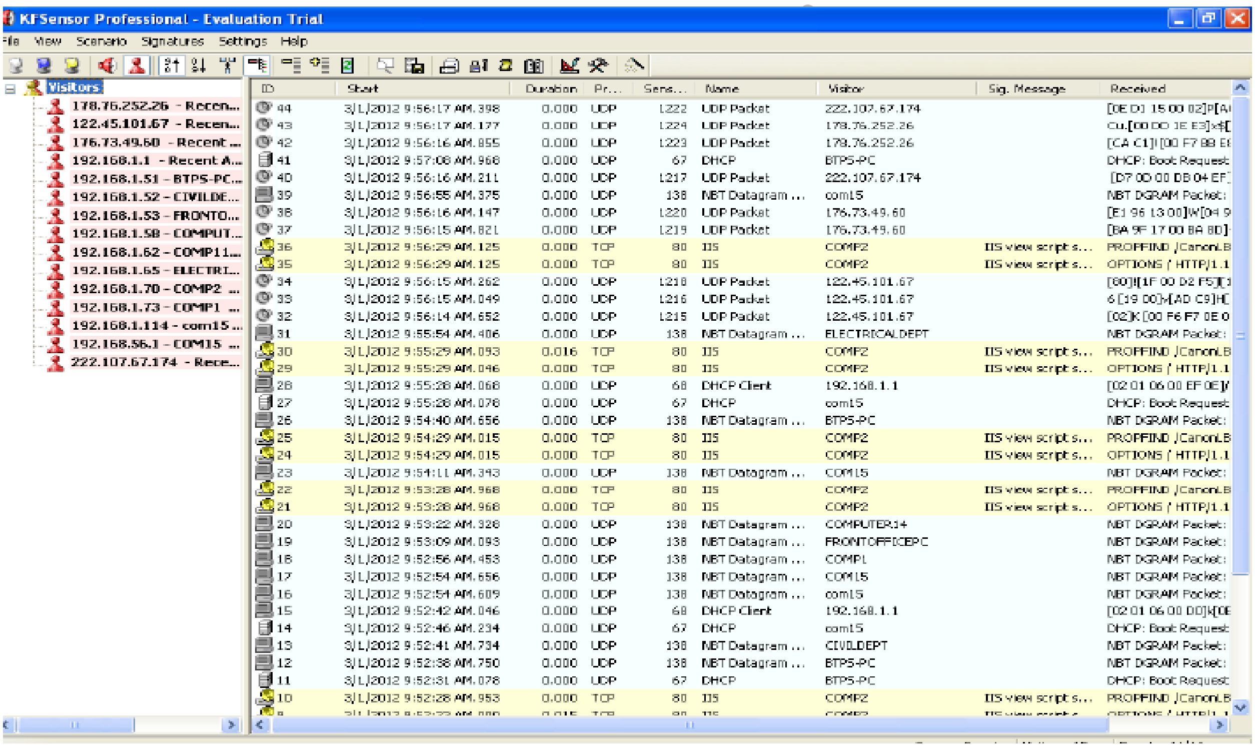

KFSensor Professional - Evaluation Trial
File View Scenario Signatures Settings Help

kfsensor - localhost - M...
- TCP
  - 0 Closed TCP Ports
  - ~~21 FTP - Error~~
  - 25 SMTP
  - 53 DNS
  - 68 DHCP
  - **80 IIS - Recent ...**
  - 110 POP3
  - 119 NNTP
  - ~~135 MS RPC - Error~~
  - 139 NBT Session S...
  - 389 LDAP
  - 443 HTTPS
  - ~~445 NBT SMB - Error~~
  - 593 CIS
  - 1028 MS CIS
  - 1080 SOCKS
  - 1433 SQL Server
  - 2234 Directplay
  - 3128 IIS Proxy
  - 3268 Global Catal...
  - 3389 Terminal Ser...
  - 5000 MS Uni Plug ...
  - 5357 Web Service...
  - 8080 IIS Proxy
- UDP
  - 0 Closed UDP Ports
  - 42 WINS UDP
  - **67 DHCP - Rece...**
  - **68 DHCP Client ...**
  - 88 Kerberos
  - 137 NBT Name Ser...
  - **138 NBT Datagr...**
  - 161 SNMP

| ID | Start | Duration | Pr... | Sens... | Name | Visitor | Sig. Message | Received |
|---|---|---|---|---|---|---|---|---|
| 26 | 3/1/2012 9:54:40 AM.656 | 0.000 | UDP | 138 | NBT Datagram ... | BTPS-PC | | NBT DGRAM Packet: id:5 |
| 25 | 3/1/2012 9:54:29 AM.015 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLBP H |
| 24 | 3/1/2012 9:54:29 AM.015 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | OPTIONS / HTTP/1.1[0D |
| 23 | 3/1/2012 9:54:11 AM.343 | 0.000 | UDP | 138 | NBT Datagram ... | COM15 | | NBT DGRAM Packet: id:3 |
| 22 | 3/1/2012 9:53:28 AM.968 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLBP H |
| 21 | 3/1/2012 9:53:28 AM.968 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | OPTIONS / HTTP/1.1[0D |
| 20 | 3/1/2012 9:53:22 AM.328 | 0.000 | UDP | 138 | NBT Datagram ... | COMPUTER14 | | NBT DGRAM Packet: id:3 |
| 19 | 3/1/2012 9:53:09 AM.093 | 0.000 | UDP | 138 | NBT Datagram ... | FRONTOFFICEPC | | NBT DGRAM Packet: id:3 |
| 18 | 3/1/2012 9:52:56 AM.453 | 0.000 | UDP | 138 | NBT Datagram ... | COMP1 | | NBT DGRAM Packet: id:3 |
| 17 | 3/1/2012 9:52:54 AM.656 | 0.000 | UDP | 138 | NBT Datagram ... | COM15 | | NBT DGRAM Packet: id:3 |
| 16 | 3/1/2012 9:52:54 AM.609 | 0.000 | UDP | 138 | NBT Datagram ... | com15 | | NBT DGRAM Packet: id:3 |
| 15 | 3/1/2012 9:52:42 AM.046 | 0.000 | UDP | 68 | DHCP Client | 192.168.1.1 | | [02 01 06 00 00]k[0B] [0 |
| 14 | 3/1/2012 9:52:46 AM.234 | 0.000 | UDP | 67 | DHCP | com15 | | DHCP: Boot Request[0A |
| 13 | 3/1/2012 9:52:41 AM.734 | 0.000 | UDP | 138 | NBT Datagram ... | CIVILDEPT | | NBT DGRAM Packet: id:3 |
| 12 | 3/1/2012 9:52:38 AM.750 | 0.000 | UDP | 138 | NBT Datagram ... | BTPS-PC | | NBT DGRAM Packet: id:5 |
| 11 | 3/1/2012 9:52:31 AM.078 | 0.000 | UDP | 67 | DHCP | BTPS-PC | | DHCP: Boot Request[0A |
| 10 | 3/1/2012 9:52:28 AM.953 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLBP H |
| 9 | 3/1/2012 9:52:23 AM.000 | 0.015 | TCP | 80 | IIS | COMP2 | IIS view script s... | OPTIONS / HTTP/1.1[0D |
| 8 | 3/1/2012 9:52:23 AM.015 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLBP H |
| 7 | 3/1/2012 9:52:10 AM.562 | 0.000 | UDP | 138 | NBT Datagram ... | CIVILDEPT | | NBT DGRAM Packet: id:3 |
| 6 | 3/1/2012 9:52:06 AM.781 | 0.000 | UDP | 138 | NBT Datagram ... | com15 | | NBT DGRAM Packet: id:3 |
| 5 | 3/1/2012 9:51:55 AM.031 | 0.000 | UDP | 138 | NBT Datagram ... | COMP1 | | NBT DGRAM Packet: id:3 |
| 4 | 3/1/2012 9:51:49 AM.937 | 0.000 | UDP | 138 | NBT Datagram ... | COM15 | | NBT DGRAM Packet: id:3 |
| 3 | 3/1/2012 9:51:30 AM.500 | 0.000 | UDP | 138 | NBT Datagram ... | COMP11 | | NBT DGRAM Packet: id:3 |
| 2 | 3/1/2012 9:51:20 AM.974 | 0.000 | UDP | 68 | DHCP Client | 192.168.1.1 | | [02 01 06 00]L#[80 03 C |
| 1 | 3/1/2012 9:51:20 AM.968 | 0.000 | UDP | 67 | DHCP | com15 | | DHCP: Boot Request[0A |

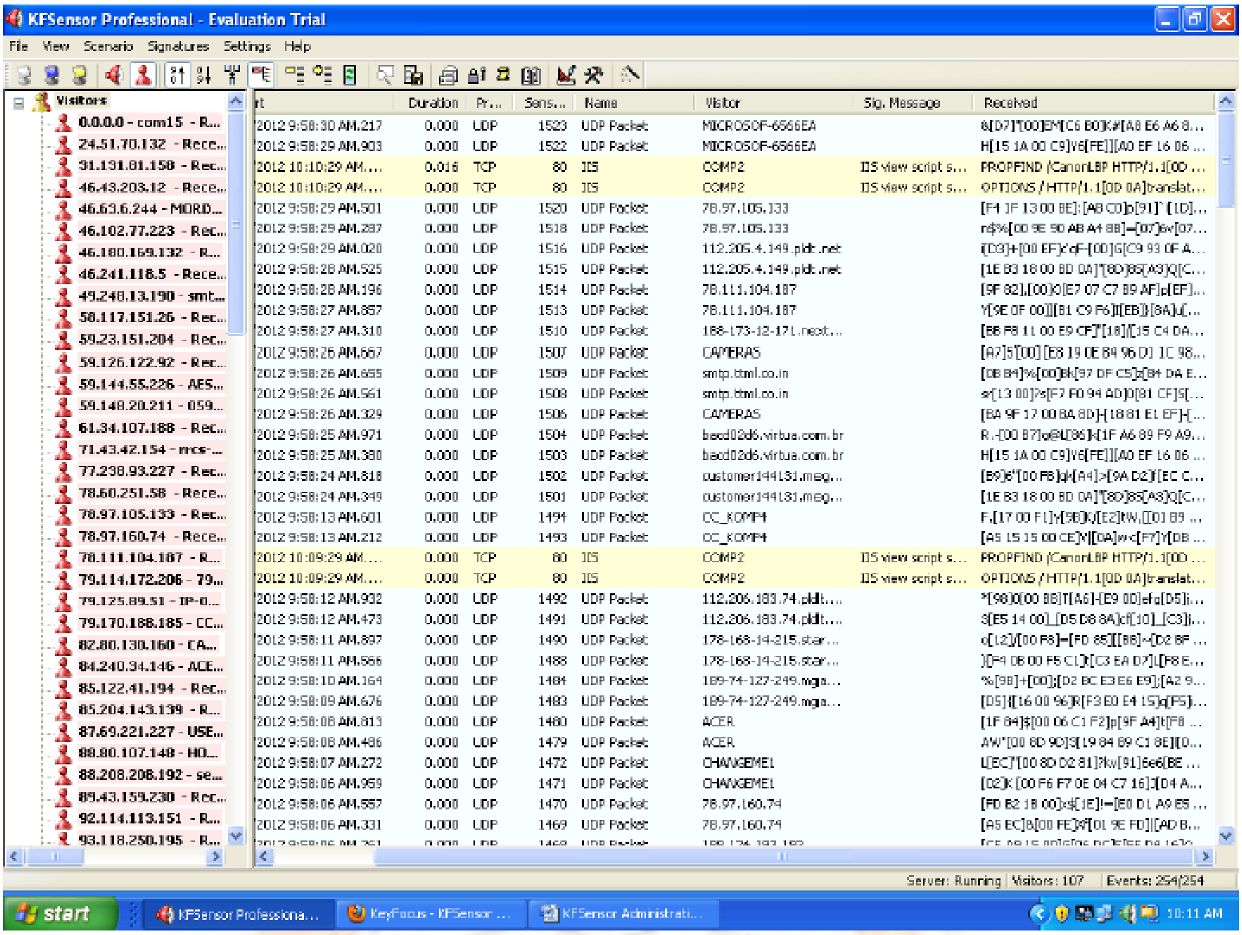
Server: Running | Visitors: 10 | Events: 26/26

start
KFSensor Professiona...
KeyFocus - KFSensor ...
9:55 AM


KFSensor Professional - Evaluation Trial
File View Scenario Signatures Settings Help

Visitors
- 178.76.252.26 - Recen...
- 122.45.101.67 - Recen...
- 176.73.49.60 - Recent ...
- 192.168.1.1 - Recent A...
- 192.168.1.51 - BTPS-PC...
- 192.168.1.52 - CIVILDE...
- 192.168.1.53 - FRONTO...
- 192.168.1.58 - COMPUT...
- 192.168.1.62 - COMP11...
- 192.168.1.65 - ELECTRI...
- 192.168.1.70 - COMP2 ...
- 192.168.1.73 - COMP1 ...
- 192.168.1.114 - com15 ...
- 192.168.56.1 - COM15 ...
- 222.107.67.174 - Rece...

| ID | Start | Duration | Pr... | Sens... | Name | Visitor | Sig. Message | Received |
|---|---|---|---|---|---|---|---|---|
| 44 | 3/1/2012 9:56:17 AM.398 | 0.000 | UDP | 1222 | UDP Packet | 222.107.67.174 | | [0E D1 15 00 02]P[A( |
| 43 | 3/1/2012 9:56:17 AM.177 | 0.000 | UDP | 1224 | UDP Packet | 178.76.252.26 | | Cu.[00 DD 1E E3]x$[ |
| 42 | 3/1/2012 9:56:16 AM.855 | 0.000 | UDP | 1223 | UDP Packet | 178.76.252.26 | | [CA C1]![00 F7 88 E( |
| 41 | 3/1/2012 9:57:08 AM.968 | 0.000 | UDP | 67 | DHCP | BTPS-PC | | DHCP: Boot Request |
| 40 | 3/1/2012 9:56:16 AM.211 | 0.000 | UDP | 1217 | UDP Packet | 222.107.67.174 | | [D7 0D 00 DB 04 EF] |
| 39 | 3/1/2012 9:56:55 AM.375 | 0.000 | UDP | 138 | NBT Datagram ... | com15 | | NBT DGRAM Packet: |
| 38 | 3/1/2012 9:56:16 AM.147 | 0.000 | UDP | 1220 | UDP Packet | 176.73.49.60 | | [E1 96 13 00]W[04 9 |
| 37 | 3/1/2012 9:56:15 AM.821 | 0.000 | UDP | 1219 | UDP Packet | 176.73.49.60 | | [BA 9F 17 00 8A 8D] |
| 36 | 3/1/2012 9:56:29 AM.125 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLB |
| 35 | 3/1/2012 9:56:29 AM.125 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | OPTIONS / HTTP/1.1 |
| 34 | 3/1/2012 9:56:15 AM.262 | 0.000 | UDP | 1218 | UDP Packet | 122.45.101.67 | | [80]![1F 00 D2 F5][1 |
| 33 | 3/1/2012 9:56:15 AM.049 | 0.000 | UDP | 1216 | UDP Packet | 122.45.101.67 | | 6 [19 00]v[AD C9]H[ |
| 32 | 3/1/2012 9:56:14 AM.652 | 0.000 | UDP | 1215 | UDP Packet | 122.45.101.67 | | [02]K [00 F6 F7 0E 0 |
| 31 | 3/1/2012 9:55:54 AM.406 | 0.000 | UDP | 138 | NBT Datagram ... | ELECTRICALDEPT | | NBT DGRAM Packet: |
| 30 | 3/1/2012 9:55:29 AM.093 | 0.016 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLB |
| 29 | 3/1/2012 9:55:29 AM.046 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | OPTIONS / HTTP/1.1 |
| 28 | 3/1/2012 9:55:28 AM.068 | 0.000 | UDP | 68 | DHCP Client | 192.168.1.1 | | [02 01 06 00 EF 0E]/ |
| 27 | 3/1/2012 9:55:28 AM.078 | 0.000 | UDP | 67 | DHCP | com15 | | DHCP: Boot Request |
| 26 | 3/1/2012 9:54:40 AM.656 | 0.000 | UDP | 138 | NBT Datagram ... | BTPS-PC | | NBT DGRAM Packet: |
| 25 | 3/1/2012 9:54:29 AM.015 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLB |
| 24 | 3/1/2012 9:54:29 AM.015 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | OPTIONS / HTTP/1.1 |
| 23 | 3/1/2012 9:54:11 AM.343 | 0.000 | UDP | 138 | NBT Datagram ... | COM15 | | NBT DGRAM Packet: |
| 22 | 3/1/2012 9:53:28 AM.968 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLB |
| 21 | 3/1/2012 9:53:28 AM.968 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | OPTIONS / HTTP/1.1 |
| 20 | 3/1/2012 9:53:22 AM.328 | 0.000 | UDP | 138 | NBT Datagram ... | COMPUTER14 | | NBT DGRAM Packet: |
| 19 | 3/1/2012 9:53:09 AM.093 | 0.000 | UDP | 138 | NBT Datagram ... | FRONTOFFICEPC | | NBT DGRAM Packet: |
| 18 | 3/1/2012 9:52:56 AM.453 | 0.000 | UDP | 138 | NBT Datagram ... | COMP1 | | NBT DGRAM Packet: |
| 17 | 3/1/2012 9:52:54 AM.656 | 0.000 | UDP | 138 | NBT Datagram ... | COM15 | | NBT DGRAM Packet: |
| 16 | 3/1/2012 9:52:54 AM.609 | 0.000 | UDP | 138 | NBT Datagram ... | com15 | | NBT DGRAM Packet: |
| 15 | 3/1/2012 9:52:42 AM.046 | 0.000 | UDP | 68 | DHCP Client | 192.168.1.1 | | [02 01 06 00 D0]k[0E |
| 14 | 3/1/2012 9:52:46 AM.234 | 0.000 | UDP | 67 | DHCP | com15 | | DHCP: Boot Request |
| 13 | 3/1/2012 9:52:41 AM.734 | 0.000 | UDP | 138 | NBT Datagram ... | CIVILDEPT | | NBT DGRAM Packet: |
| 12 | 3/1/2012 9:52:38 AM.750 | 0.000 | UDP | 138 | NBT Datagram ... | BTPS-PC | | NBT DGRAM Packet: |
| 11 | 3/1/2012 9:52:31 AM.078 | 0.000 | UDP | 67 | DHCP | BTPS-PC | | DHCP: Boot Request |
| 10 | 3/1/2012 9:52:28 AM.953 | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLB |



KFSensor Professional - Evaluation Trial
File View Scenario Signatures Settings Help

Visitors
- 0.0.0.0 - com15 - R...
- 24.51.70.132 - Rece...
- 31.131.81.158 - Rec...
- 46.43.203.12 - Rece...
- 46.63.6.244 - MORD...
- 46.102.77.223 - Rec...
- 46.180.169.132 - R...
- 46.241.118.5 - Rece...
- 49.248.13.190 - smt...
- 58.117.151.26 - Rec...
- 59.23.151.204 - Rec...
- 59.126.122.92 - Rec...
- 59.144.55.226 - AES...
- 59.148.20.211 - 059...
- 61.34.107.188 - Rec...
- 71.43.42.154 - mcs-...
- 77.238.93.227 - Rec...
- 78.60.251.58 - Rece...
- 78.97.105.133 - Rec...
- 78.97.160.74 - Rece...
- 78.111.104.187 - R...
- 79.114.172.206 - 79...
- 79.125.89.51 - IP-0...
- 79.170.188.185 - CC...
- 82.80.130.160 - CA...
- 84.240.34.146 - ACE...
- 85.122.41.194 - Rec...
- 85.204.143.139 - R...
- 87.69.221.227 - USE...
- 88.80.107.148 - HO...
- 88.208.208.192 - se...
- 89.43.159.230 - Rec...
- 92.114.113.151 - R...
- 93.118.250.195 - R...

| rt | Duration | Pr... | Sens... | Name | Visitor | Sig. Message | Received |
|---|---|---|---|---|---|---|---|
| 2012 9:58:30 AM.217 | 0.000 | UDP | 1523 | UDP Packet | MICROSOF-6566EA | | 8[D7]"[00]EM[C6 B0]K#[A8 E6 A6 8... |
| 2012 9:58:29 AM.903 | 0.000 | UDP | 1522 | UDP Packet | MICROSOF-6566EA | | H[15 1A 00 C9]Y6[FE]][A0 EF 16 06 ... |
| 2012 10:10:29 AM.... | 0.016 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLBP HTTP/1.1[0D ... |
| 2012 10:10:29 AM.... | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | OPTIONS / HTTP/1.1[0D 0A]translat... |
| 2012 9:58:29 AM.501 | 0.000 | UDP | 1520 | UDP Packet | 78.97.105.133 | | [F4 1F 13 00 8E]:[AB C0]p[91]`{[1D]... |
| 2012 9:58:29 AM.287 | 0.000 | UDP | 1518 | UDP Packet | 78.97.105.133 | | n$%[00 9E 90 AB A4 8B]=[07]6v[07... |
| 2012 9:58:29 AM.020 | 0.000 | UDP | 1516 | UDP Packet | 112.205.4.149.pldt.net | | i[D3]+[00 EF]c'qF-[0D]G[C9 93 0F A... |
| 2012 9:58:28 AM.525 | 0.000 | UDP | 1515 | UDP Packet | 112.205.4.149.pldt.net | | [1E 83 18 00 8D 0A]"[8D]85[A3]Q[C... |
| 2012 9:58:28 AM.196 | 0.000 | UDP | 1514 | UDP Packet | 78.111.104.187 | | [9F 82],[00]O[E7 07 C7 89 AF]p[EF]... |
| 2012 9:58:27 AM.857 | 0.000 | UDP | 1513 | UDP Packet | 78.111.104.187 | | Y[9E 0F 00]][B1 C9 F6]I[EB]}[8A]u[... |
| 2012 9:58:27 AM.310 | 0.000 | UDP | 1510 | UDP Packet | 188-173-12-171.next... | | [B8 F8 11 00 E9 CF]"[18]/[15 C4 DA... |
| 2012 9:58:26 AM.667 | 0.000 | UDP | 1507 | UDP Packet | CAMERAS | | [A7]5"[00] [E8 19 0E 84 96 D1 1C 98... |
| 2012 9:58:26 AM.655 | 0.000 | UDP | 1509 | UDP Packet | smtp.ttml.co.in | | [DB 84]%[00]Bk[97 DF C5]z[B4 DA E... |
| 2012 9:58:26 AM.561 | 0.000 | UDP | 1508 | UDP Packet | smtp.ttml.co.in | | ≈[13 00]?s[F7 F0 94 AD]0[81 CF]5[... |
| 2012 9:58:26 AM.329 | 0.000 | UDP | 1506 | UDP Packet | CAMERAS | | [BA 9F 17 00 BA 8D]-[18 81 E1 EF]H[... |
| 2012 9:58:25 AM.971 | 0.000 | UDP | 1504 | UDP Packet | bacd02d6.virtua.com.br | | R.-[00 87]g@L[86]k[1F A6 89 F9 A9... |
| 2012 9:58:25 AM.380 | 0.000 | UDP | 1503 | UDP Packet | bacd02d6.virtua.com.br | | H[15 1A 00 C9]Y6[FE]][A0 EF 16 06 ... |
| 2012 9:58:24 AM.818 | 0.000 | UDP | 1502 | UDP Packet | customer144131.meg... | | [B9]6"[00 FB]qk[A4]>[9A D2][EC C... |
| 2012 9:58:24 AM.349 | 0.000 | UDP | 1501 | UDP Packet | customer144131.meg... | | [1E 83 18 00 8D 0A]"[8D]85[A3]Q[C... |
| 2012 9:58:13 AM.601 | 0.000 | UDP | 1494 | UDP Packet | CC_KOMP4 | | F.[17 00 F1]y[9B]K/[E2]tW,[[01 B9 ... |
| 2012 9:58:13 AM.212 | 0.000 | UDP | 1493 | UDP Packet | CC_KOMP4 | | [A5 15 15 00 CE]V[[0A]w<[F7]Y[DB ... |
| 2012 10:09:29 AM.... | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | PROPFIND /CanonLBP HTTP/1.1[0D ... |
| 2012 10:09:29 AM.... | 0.000 | TCP | 80 | IIS | COMP2 | IIS view script s... | OPTIONS / HTTP/1.1[0D 0A]translat... |
| 2012 9:58:12 AM.932 | 0.000 | UDP | 1492 | UDP Packet | 112.206.183.74.pldt.... | | *[98]0[00 8B]T[A6]-[E9 00]efg[D5]j... |
| 2012 9:58:12 AM.473 | 0.000 | UDP | 1491 | UDP Packet | 112.206.183.74.pldt.... | | 3[E5 14 00]_[D5 D8 8A]cf[10]_[C3]j... |
| 2012 9:58:11 AM.897 | 0.000 | UDP | 1490 | UDP Packet | 178-168-14-215.star... | | o[12]/[00 F8]=[FD 85][[B8]~[D2 8F ... |
| 2012 9:58:11 AM.566 | 0.000 | UDP | 1488 | UDP Packet | 178-168-14-215.star... | | )[F4 0B 00 F5 C1]t[C3 EA D7]L[F8 E... |
| 2012 9:58:10 AM.164 | 0.000 | UDP | 1484 | UDP Packet | 189-74-127-249.mga... | | %[98]+[00];[D2 BC E3 E6 E9];[A2 9... |
| 2012 9:58:09 AM.676 | 0.000 | UDP | 1483 | UDP Packet | 189-74-127-249.mga... | | [D5]{[16 00 96]R[F3 E0 E4 15]q[F5]... |
| 2012 9:58:08 AM.813 | 0.000 | UDP | 1480 | UDP Packet | ACER | | [1F 84]$[00 06 C1 F2]p[9F A4]t[F8 ... |
| 2012 9:58:08 AM.486 | 0.000 | UDP | 1479 | UDP Packet | ACER | | AW"[00 6D 9D]3[19 84 B9 C1 8E][0... |
| 2012 9:58:07 AM.272 | 0.000 | UDP | 1472 | UDP Packet | CHANGEME1 | | L[EC]"[00 8D D2 81]?kv[91]6e6[BE ... |
| 2012 9:58:06 AM.959 | 0.000 | UDP | 1471 | UDP Packet | CHANGEME1 | | [D2]K [00 F6 F7 0E 04 C7 16]J[D4 A... |
| 2012 9:58:06 AM.557 | 0.000 | UDP | 1470 | UDP Packet | 78.97.160.74 | | [FD B2 18 00]x$[1E]!=[E0 D1 A9 E5 ... |
| 2012 9:58:06 AM.331 | 0.000 | UDP | 1469 | UDP Packet | 78.97.160.74 | | [A5 EC]B[00 FE]Xf[01 9E FD]|[AD B... |

Server: Running | Visitors: 107 | Events: 254/254

Result:

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

Outcome:

Thus the outcome to setup a honeypot and monitor the honeypot on network has been attained.

Application:

Tool to capture the malicious node traffic.

## Viva-voce

1. What do you understand by honeypot on network?
2. How is honeypot works?
3. How will you install honey pots?
4. What are log files?
5. How will you uninstall honeypots?
6. List any tool which is similar to honeypot.

Expt. No. 6

# INSTALLATION OF ROOTKITS AND STUDY THE VARIETY OF OPTIONS

## Aim:

To install the rootkits and study the variety of options

## Software requirements:

C / C++ / Java or equivalent compiler GnuPG, KF Sensor or Equivalent, Snort, Net Stumbler or Equivalent.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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.

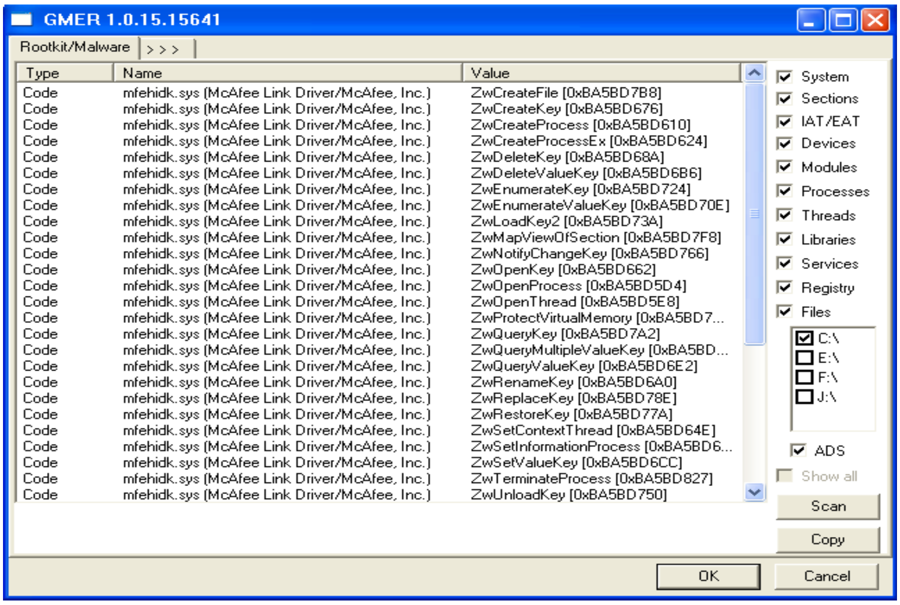
A rootkit is a collection of tools (programs) that enable administrator-level access to a computer or computer network. Typically, a cracker installs a rootkit on a computer after first obtaining user-level access, either by exploiting a known vulnerability or cracking a password. Once the rootkit is 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.



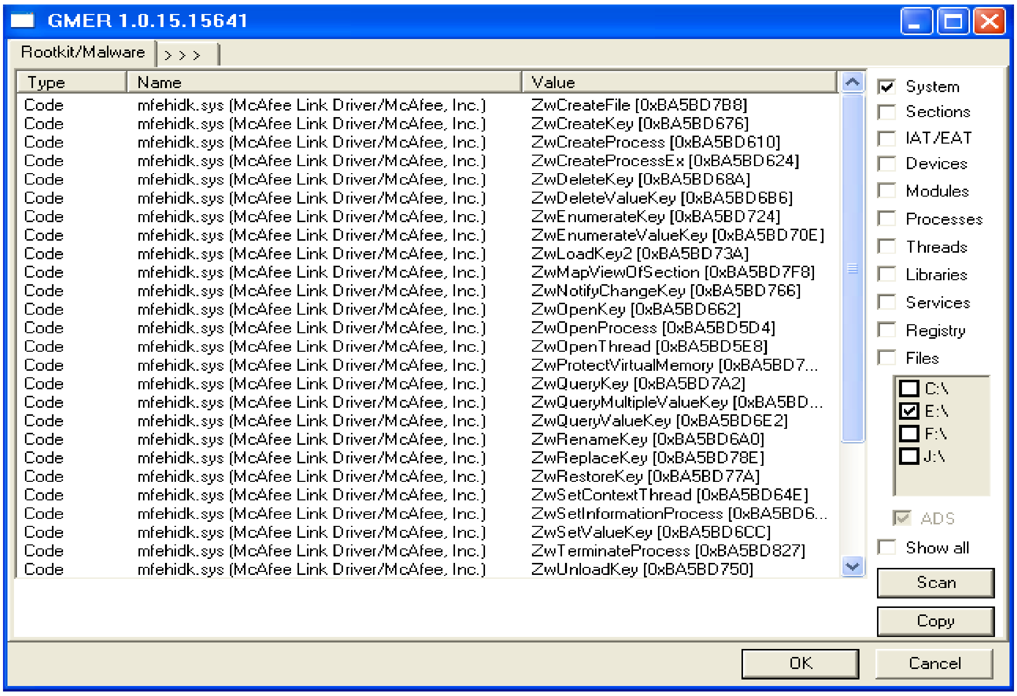
GMER 1.0.15.15641

Rootkit/Malware | > > >

| Type | Name | Value |
|---|---|---|
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwCreateFile [0xBA5BD7B8] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwCreateKey [0xBA5BD676] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwCreateProcess [0xBA5BD610] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwCreateProcessEx [0xBA5BD624] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwDeleteKey [0xBA5BD68A] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwDeleteValueKey [0xBA5BD6B6] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwEnumerateKey [0xBA5BD724] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwEnumerateValueKey [0xBA5BD70E] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwLoadKey2 [0xBA5BD73A] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwMapViewOfSection [0xBA5BD7F8] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwNotifyChangeKey [0xBA5BD766] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwOpenKey [0xBA5BD662] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwOpenProcess [0xBA5BD5D4] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwOpenThread [0xBA5BD5E8] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwProtectVirtualMemory [0xBA5BD7... |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwQueryKey [0xBA5BD7A2] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwQueryMultipleValueKey [0xBA5BD... |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwQueryValueKey [0xBA5BD6E2] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwRenameKey [0xBA5BD6A0] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwReplaceKey [0xBA5BD78E] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwRestoreKey [0xBA5BD77A] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwSetContextThread [0xBA5BD64E] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwSetInformationProcess [0xBA5BD6... |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwSetValueKey [0xBA5BD6CC] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwTerminateProcess [0xBA5BD827] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwUnloadKey [0xBA5BD750] |

☑ System
☑ Sections
☑ IAT/EAT
☑ Devices
☑ Modules
☑ Processes
☑ Threads
☑ Libraries
☑ Services
☑ Registry
☑ Files
☑ C:\
□ E:\
□ F:\
□ J:\
☑ ADS
□ Show all

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



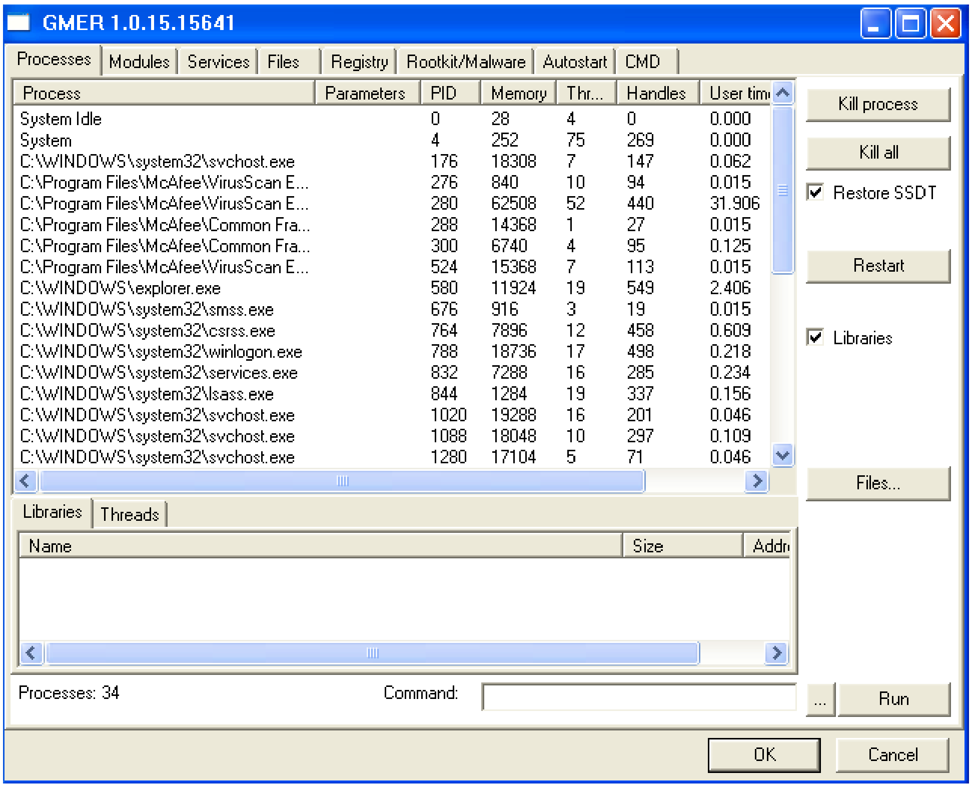
GMER 1.0.15.15641

Rootkit/Malware | > > >

| Type | Name | Value |
|---|---|---|
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwCreateFile [0xBA5BD7B8] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwCreateKey [0xBA5BD676] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwCreateProcess [0xBA5BD610] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwCreateProcessEx [0xBA5BD624] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwDeleteKey [0xBA5BD68A] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwDeleteValueKey [0xBA5BD6B6] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwEnumerateKey [0xBA5BD724] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwEnumerateValueKey [0xBA5BD70E] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwLoadKey2 [0xBA5BD73A] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwMapViewOfSection [0xBA5BD7F8] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwNotifyChangeKey [0xBA5BD766] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwOpenKey [0xBA5BD662] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwOpenProcess [0xBA5BD5D4] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwOpenThread [0xBA5BD5E8] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwProtectVirtualMemory [0xBA5BD7... |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwQueryKey [0xBA5BD7A2] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwQueryMultipleValueKey [0xBA5BD... |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwQueryValueKey [0xBA5BD6E2] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwRenameKey [0xBA5BD6A0] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwReplaceKey [0xBA5BD78E] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwRestoreKey [0xBA5BD77A] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwSetContextThread [0xBA5BD64E] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwSetInformationProcess [0xBA5BD6... |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwSetValueKey [0xBA5BD6CC] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwTerminateProcess [0xBA5BD827] |
| Code | mfehidk.sys (McAfee Link Driver/McAfee, Inc.) | ZwUnloadKey [0xBA5BD750] |

☑ System
□ Sections
□ IAT/EAT
□ Devices
□ Modules
□ Processes
□ Threads
□ Libraries
□ Services
□ Registry
□ Files
□ C:\
☑ E:\
□ F:\
□ J:\
☑ ADS
□ Show all

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



GMER 1.0.15.15641

Processes | Modules | Services | Files | Registry | Rootkit/Malware | Autostart | CMD
Kill process
Kill all
☑ Restore SSDT
Restart
☑ Libraries
Files...
Run
OK
Cancel

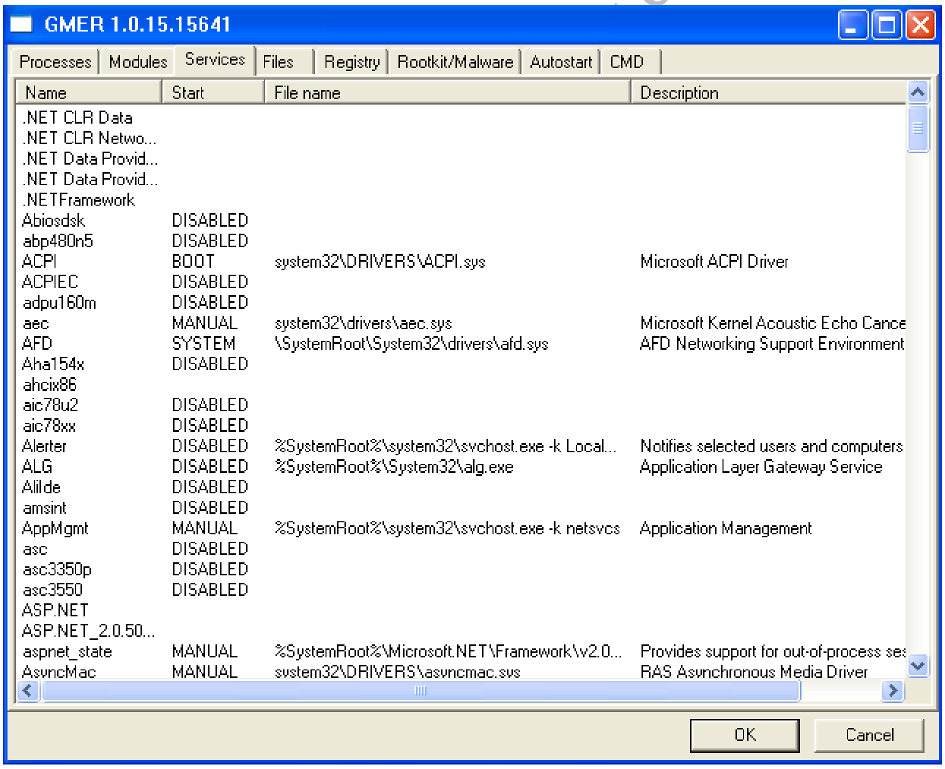
| Process | Parameters | PID | Memory | Thr... | Handles | User tim |
|---|---|---|---|---|---|---|
| System Idle | | 0 | 28 | 4 | 0 | 0.000 |
| System | | 4 | 252 | 75 | 269 | 0.000 |
| C:\WINDOWS\system32\svchost.exe | | 176 | 18308 | 7 | 147 | 0.062 |
| C:\Program Files\McAfee\VirusScan E... | | 276 | 840 | 10 | 94 | 0.015 |
| C:\Program Files\McAfee\VirusScan E... | | 280 | 62508 | 52 | 440 | 31.906 |
| C:\Program Files\McAfee\Common Fra... | | 288 | 14368 | 1 | 27 | 0.015 |
| C:\Program Files\McAfee\Common Fra... | | 300 | 6740 | 4 | 95 | 0.125 |
| C:\Program Files\McAfee\VirusScan E... | | 524 | 15368 | 7 | 113 | 0.015 |
| C:\WINDOWS\explorer.exe | | 580 | 11924 | 19 | 549 | 2.406 |
| C:\WINDOWS\system32\smss.exe | | 676 | 916 | 3 | 19 | 0.015 |
| C:\WINDOWS\system32\csrss.exe | | 764 | 7896 | 12 | 458 | 0.609 |
| C:\WINDOWS\system32\winlogon.exe | | 788 | 18736 | 17 | 498 | 0.218 |
| C:\WINDOWS\system32\services.exe | | 832 | 7288 | 16 | 285 | 0.234 |
| C:\WINDOWS\system32\lsass.exe | | 844 | 1284 | 19 | 337 | 0.156 |
| C:\WINDOWS\system32\svchost.exe | | 1020 | 19288 | 16 | 201 | 0.046 |
| C:\WINDOWS\system32\svchost.exe | | 1088 | 18048 | 10 | 297 | 0.109 |
| C:\WINDOWS\system32\svchost.exe | | 1280 | 17104 | 5 | 71 | 0.046 |

Libraries | Threads

| Name | Size | Addr |
|---|---|---|

Processes: 34 Command:

7. Click on the option services.



GMER 1.0.15.15641

Processes | Modules | Services | Files | Registry | Rootkit/Malware | Autostart | CMD
OK
Cancel

| Name | Start | File name | Description |
|---|---|---|---|
| .NET CLR Data | | | |
| .NET CLR Netwo... | | | |
| .NET Data Provid... | | | |
| .NET Data Provid... | | | |
| .NETFramework | | | |
| Abiosdsk | DISABLED | | |
| abp480n5 | DISABLED | | |
| ACPI | BOOT | system32\DRIVERS\ACPI.sys | Microsoft ACPI Driver |
| ACPIEC | DISABLED | | |
| adpu160m | DISABLED | | |
| aec | MANUAL | system32\drivers\aec.sys | Microsoft Kernel Acoustic Echo Cance |
| AFD | SYSTEM | \SystemRoot\System32\drivers\afd.sys | AFD Networking Support Environment |
| Aha154x | DISABLED | | |
| ahcix86 | | | |
| aic78u2 | DISABLED | | |
| aic78xx | DISABLED | | |
| Alerter | DISABLED | %SystemRoot%\system32\svchost.exe -k Local... | Notifies selected users and computers |
| ALG | DISABLED | %SystemRoot%\System32\alg.exe | Application Layer Gateway Service |
| AliIde | DISABLED | | |
| amsint | DISABLED | | |
| AppMgmt | MANUAL | %SystemRoot%\system32\svchost.exe -k netsvcs | Application Management |
| asc | DISABLED | | |
| asc3350p | DISABLED | | |
| asc3550 | DISABLED | | |
| ASP.NET | | | |
| ASP.NET_2.0.50... | | | |
| aspnet_state | MANUAL | %SystemRoot%\Microsoft.NET\Framework\v2.0... | Provides support for out-of-process ses |
| AsyncMac | MANUAL | system32\DRIVERS\asyncmac.sys | RAS Asynchronous Media Driver |

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.

## Outcome:

Thus the outcome of installing the rootkits and the variety of options has been attained.

## Application:

Tool that enable administrator-level access to a computer or computer network.

# Viva-voce

1. What is Rootkit?
2. What are called Rootkits in Windows?
3. What are the basic classes of Rootkits?
4. Why Rootkits are used?
5. How Rootkits stay undetected?
6. What are the Rootkit capabilities?

## Expt. No. 7

# 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

## Software requirements:

C / C++ / Java or equivalent compiler GnuPG, KF Sensor or Equivalent, Snort, Net Stumbler or Equivalent.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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.

## Outcome:

Thus the outcome of to perform wireless audit on an access point or a router and decrypt WEP and WPA (Net Stumbler) has been attained.

## Application:

Tool to find the adjacent nodes.

## Viva-voce

1. What is WAP?
2. What is BSSID?
3. What is SSID?
4. What is Latitude, Longitude Distance?
5. What is WEP?
6. What is Net Stumbler?
7. Define – Audit
8. Define – Router

Expt. No. 8

# DEMONSTRATE INTRUSION DETECTION SYSTEM (IDS) USING ANY TOOL (SNORT OR ANY OTHER S/W)

## Aim:

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

## Software requirements:

C / C++ / Java or equivalent compiler GnuPG, KF Sensor or Equivalent, Snort, Net Stumbler or Equivalent.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## 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.

Outcome:

Thus the outcome of intrusion detection system has been attained.

Application:

Tools to act as hacker or ethical hacker.

## Viva-voce

1. What is Intrusion Detection System (IDS)?
2. What are the activities of IDS?
3. What is intrusion or intruder?
4. What is Snort?
5. What are the uses of Snort?
6. What are the three modes of Snort?
7. What are different types of IDS?
8. Abbreviate HIDS.
9. Abbreviate NIDS.
10. Define – IPS

Expt. No. 9

## DEMONSTRATE NETWORK PROTOCOL ANALYZER USING ANY TOOL

### Aim:

To demonstrate network protocol analyzer using the tool like snort or any software

### Software requirements:

C / C++ / Java or equivalent compiler GnuPG, KF Sensor or Equivalent, Snort, Net Stumbler or Equivalent.

### Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

### Algorithm:

1. Download the wireshark file, and then execute the executable file (usually Wireshark-winxx-version.exe), and click Next.
2. On the next page there will be a license agreement, you can read it if you want. Click Next to continue to next process.
3. On step 3, there is a window where you can choose component to be installed on your computer. Click next when you finished choose your packet.
4. The next installation process is selecting additional tasks.
5. Click next to go to the next step.
6. In this step you will be asked to choose install location. If you don't know about this, then just leave it to default and click next.
7. This step will ask you whether want to install WinPcap or not. If you didn't have WinPcap installed on your system, you can check the Install WinPcap checkbox.

## Result:

Thus the experiment of demonstrating network protocol analyzer was executed successfully.

## Outcome:

Thus the outcome of network protocol analyzer has been attained.

## Application:

Using the above tool we can find adjacent nodes IP address.

## Viva-voce

1. What is wireshark?
2. Is any other tool, which is similar to wireshark?
3. Can I use Wireshark commercially?
4. Does Wireshark work on Windows Vista or Windows Server 2008?
5. What protocols are currently supported?

## Expt. No. 10

# DEMONSTRATE SQL INJECTION USING ANY TOOL

## Aim:

To demonstrate SQL Injection using any tool

## Software requirements:

C / C++ / Java or equivalent compiler GnuPG, KF Sensor or Equivalent, Snort, Net Stumbler or Equivalent.

## Hardware requirements:

Dual core processor, DDR2 1GB RAM, 250 GB HDD.

## Algorithm:

1. Finding Vulnerable Website.
2. Checking the Vulnerability.
3. Finding Number of columns.
4. Displaying the Vulnerable columns.
5. Finding version, database, user Now replace the 3 from the query with “version()”.
6. Finding the Table Name.
7. Finding the Column Name.
8. Finding the Admin Panel.

## Result:

Thus the experiment of demonstrating SQL Injection was executed successfully.

## Outcome:

Thus the outcome of SQL Injection has been attained.

## Application:

Its used to hack the any website using the abouve tool.

## Viva-voce

1. What is sql injection?
2. What an attacker can do?
3. Define – SQL
4. Is their any other tool for sql injection?
5. Is website or mobile hacking are similar?
6. Define – Hacking
7. What are types of hackers?
8. Which hack type you belong to?