

**SSN College of Engineering,
Department of Computer Science and Engineering
CS6711 Security Laboratory**

Exercise 1:

To implement the substitution techniques: Caesar Cipher and Playfair Cipher

Programming Language: Java

Hints:

Encryption Procedure for Caesar Cipher:

1. Read the plain text message
2. Read the key value (displacement)
3. To generate the cipher text , replace each letter of plaintext by a letter at the position specified by the key value down the alphabetical stream.
4. Display the cipher text.

Decryption Procedure for Caesar Cipher:

1. Use the cipher text as input
2. Use the same key value as displacement
3. To retrieve the plaintext text from cipher text, replace a letter of cipher text by the letter at the position specified by the key value in the reverse alphabetical stream.
4. Display the plain text.

Encryption Procedure for Playfair Cipher:

1. Read the plain text message
2. Read the key value (a string without any repetition letters)
3. Construct a 5 X 5 matrix and fill in the key text in row wise manner.
4. Fill in the remaining cells of the matrix with the rest of the alphabets sans the letters of the key.
5. Split the plain text into two letter words without repetition.
6. If a pair has a repeated letter, insert filler like 'X'
7. If both letters fall in the same row, replace each with letter to right (wrapping back to start from end)
8. If both letters fall in the same column, replace each with the letter below it (wrapping to top from bottom)
9. Otherwise each letter is replaced by the letter in the same row and in the column of the other letter of the pair
10. Display the cipher text.

Decryption Procedure for Playfair Cipher:

1. Use the cipher text as input
2. Use the same key value
3. To retrieve the plaintext text from cipher text, split the plain text into two letter words without repetition.
4. Repeat the steps 7 to 9 of encryption to generate the plaintext
5. Display the plain text.

**SSN College of Engineering,
Department of Computer Science and Engineering
CS6711 Security Laboratory**

Exercise 2a:

To implement the substitution technique: Hill Cipher

Hints:

Encryption Procedure for Hill Cipher:

$$E: C = KP \bmod 26$$

1. Read the plain text message
2. Read the key (a square matrix, say order of $n \times n$)
3. Split the plain text into chunks of size n , convert them into their numerical equivalent and form a columnar matrix.
4. Perform matrix multiplication on K and plain text vector mod 26.
5. Generate cipher text by decoding the outcome of step 4 into equivalent alphabets.
6. Display the cipher text.

Decryption Procedure for Hill Cipher:

$$D: P = K^{-1}C \bmod 26$$

1. Use the cipher text as input
2. Compute the inverse matrix of K, that is K^{-1}
3. Encode the cipher text into their numerical equivalent and form a columnar matrix with respect to the order of K^{-1} .
4. Perform matrix multiplication on K^{-1} and cipher text vector mod 26.
5. Retrieve plain text by decoding the outcome of step 4 into equivalent alphabets.
6. Display the plain text.

**SSN College of Engineering,
Department of Computer Science and Engineering
CS6711 Security Laboratory**

Exercise 2b:

To implement the substitution technique: Vigenère Cipher

Programming Language: Java

Hints:

Encryption Procedure for Vigenère Cipher:

1. Read the plain text message
2. Read the key phrase
3. Construct a reference Vigenère table, where each row of table consists of all letters of the English alphabet.
 - a. The first row starts with the letter a, and each following row is shifted by one letter (second row starts with b, third with c...).
4. To encrypt, pick a letter in the plaintext and its corresponding letter in the keyword, use the keyword letter and the plaintext letter as the row index and column index, respectively, and the entry at the row-column intersection is the letter in the cipher text.
5. Display the cipher text.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Vigenère table

Decryption Procedure for Vigenère Cipher:

1. Use the cipher text as input
2. Use the same key phrase
3. To decrypt the cipher text, find in the row corresponding to the n-th letter of the key phrase a cell in which the n-th letter of the cipher text resides.
 - Its column is denoted by the n-th letter of the plain text.
4. Display the plain text.

**SSN College of Engineering,
Department of Computer Science and Engineering
CS6711 Security Laboratory**

Exercise 3a:

To implement the transposition techniques: Rail fence Cipher

Programming Language: Java

Hints:

Encryption Procedure for Rail fence Cipher:

1. Read the plain text message
2. Read the key value (The key for the rail fence cipher is just the number of rails / levels)
3. Write the letters of the plaintext diagonally down to the right until you reach the number of rows specified by the key.
 - Then bounce back up diagonally until you hit the first row again. This continues until the end of the plaintext.
4. To generate the cipher text, read off along the rows one by one.
5. Display the cipher text.

Decryption Procedure for Rail fence Cipher:

1. Use the cipher text as input and number of rails as key value
2. Break up the cipher text letters into equal groups for each rail.
3. To retrieve the plaintext text from cipher text, read off the message vertically.
4. Display the plain text.

Exercise 3b:

To implement the transposition techniques: Row & Column Ciphers

Programming Language: Java

Hints:

Encryption Procedure for Row & Column Cipher:

1. Read the plain text message
2. Write the plaintext in row by row forming a matrix / grid of $m \times n$.
3. Read the key value (a number with permuted digits ranging from 1 to number of columns in the plaintext matrix)
4. Use the key as column headers
5. Generate the cipher text with respect to the order of the column headers.
6. Display the cipher text.

Decryption Procedure for Row & Column Cipher:

1. Use the cipher text as input
2. The number of rows and columns of the cipher text matrix be now $n \times m$
3. Write the cipher text row by row in n rows
4. Let the row headers be the permuted key values of 1 to n
 - If encryption key is 4 3 1 2 5 6 7
 - Decryption key will be 3 4 2 1 5 6 7
5. Take the letters in the order of key from the first column to retrieve the plain text. Repeat extracting letters in the other columns according to the key.
6. Display the plain text.

**SSN College of Engineering,
Department of Computer Science and Engineering
CS6711 Security Laboratory**

Exercise 4:

To implement the Data Encryption Standard (DES) Algorithm.

Programming Language: Java

Hints:

Key Generation

1. Initialize the permutation tables, left shift schedules.
2. Read the 64 bit key.
3. 64 bits goes through a permutation called PC-1(permuted choice) resulting 56 bits.
4. 56 bits are divided into two halves
5. Each half will be rotated left by 1 or 2 bits depending on the round
6. Both sides go through permute choice 2 (PC-2) which selects 24 bits from left and right resulting a 48 bit round key.

Encryption Procedure for DES:

1. Initialize the permutation tables, S boxes, expansion tables, left shift schedules.
2. A block of 64 bits is permuted by an initial permutation called IP.
3. Resulting 64 bits are divided in two halves of 32 bits, left and right.
4. Right half goes through a function F (Feistel function)
5. Left half is XOR-ed with output from F function above.
6. Left and right are swapped(except last round).
7. If last round, apply an inverse permutation IP-1 on both halves and that's the output else, goto step 3.
8. Display the cipher text.

Feistel function F:

1. Expansion – 32 bits to 48 bits based on an expansion table.
2. Key mixing – round key combined with 48 bits from previous step by XOR operation.
3. Substitution – previous result divided into 8x6bits blocks before processed by s-boxes(substitution boxes)
4. Permutation based on a fixed permutation table.

Decryption Procedure for DES:

1. Use the cipher text as input.
2. Apply the same set of operations from step 2 to 7 of encryption procedure.
3. Use the keys K_i in reverse order (use K_{16} on the first iteration, K_{15} on the second until K_1 on last iteration).
4. Display the plain text.

**SSN College of Engineering,
Department of Computer Science and Engineering
CS6711 Security Laboratory**

Exercise 5:

To implement the Rivest-Shamir-Adleman (RSA) Algorithm.

Programming Language: Java

Hints:

Key Generation

1. Generate two large random primes, p and q , of approximately equal size such that their product $n = pq$.
2. Compute $n = pq$ and $(\phi) \phi = (p-1)(q-1)$.
3. Choose an integer e , $1 < e < \phi$, such that $\gcd(e, \phi) = 1$.
4. Compute the secret exponent d , $1 < d < \phi$, such that $ed \equiv 1 \pmod{\phi}$.
5. The public key is (n, e) and the private key (d, p, q) .
- 6.

Encryption Procedure for RSA:

Sender does the following:-

1. Obtains the recipient B's public key (n, e) .
2. Represents the plaintext message as a positive integer m , $1 < m < n$.
3. Computes the cipher text $C = m^e \bmod n$.
4. Display the cipher text C .

Decryption Procedure for RSA:

1. Use the cipher text as input.
2. Uses his private key (n, d) to compute $m = C^d \bmod n$.
3. Extracts the plaintext from the message representative m .
4. Display the plain text.

**SSN College of Engineering,
Department of Computer Science and Engineering
CS 6711 Security Laboratory**

Exercise 6:

To implement the Diffie Hellman Key Exchange algorithm

Programming Language: Java

Hints:

1. Choose a prime number p and g is a primitive root modulo p .
2. Check for the primality of the number p (using Miller Rabin Method)
3. Read X_A , the secret key of A, such that $X_A < p$.
4. Compute the public key of A, $Y_A = g^{X_A} \mod p$
5. Read X_B , the secret key of B, , such that $X_B < p$..
6. Compute the public key of B, $Y_B = g^{X_B} \mod p$
7. Compute A's shared secret key, $K = Y_B^{X_A} \mod p$
8. Compute B's shared secret key, $K = Y_A^{X_B} \mod p$
9. Display A and B's shared secret keys.

**SSN College of Engineering,
Department of Computer Science and Engineering
CS 6711 Security Laboratory**

Exercise 7:

To implement the message digest MD5

Programming Language: Java

Hints:

1. Read the message
2. Divide the message into 512 bit blocks.
3. Append padding bits,
 - A single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message equals to $448 \bmod 512$.
4. Append length
 - A 64-bit representation of the length of the message is appended
5. Initialize MD buffers, A, B, C, D
 - word A: 01 23 45 67
 - word B: 89 ab cd ef
 - word C: fe dc ba 98
 - word D: 76 54 32 10
6. Invoke the compress function for four times
7. Display the message digest from the buffers.

MD5 Compression function:

1. Perform $a = b + ((a + g(b, c, d) + X[k] + T[i]) \lll s)$
 - a, b, c, d refer to the 4 words of the buffers
 - $g(b, c, d)$ is a different nonlinear function in each round (F, G, H, I)
 - $F(X, Y, Z) = XY \text{ or not } (X) Z$
 - $G(X, Y, Z) = XZ \text{ or } Y \text{ not } (Z)$
 - $H(X, Y, Z) = X \text{ xor } Y \text{ xor } Z$
 - $I(X, Y, Z) = Y \text{ xor } (X \text{ or not } (Z))$
 - $T[i]$ is a constant value
 - $\lll s$ circular left shift of 32 bit argument by s bits
 - $X[k] : M[q \times 16 + K] = K^{\text{th}}$ 32 bit word in q^{th} 512 block of the message
 - $+$ is addition modulo 2^{32}
2. Perform a 32 bit circular right shift such that;
 - $a=d; \quad b=a; \quad c=b; \quad d=c;$

T[1] = D76AA478	T[17] = F61E2562	T[33] = FFFA3942	T[49] = F4292244
T[2] = E8C7B756	T[18] = C040B340	T[34] = 8771F681	T[50] = 432AFF97
T[3] = 242070DB	T[19] = 265E5A51	T[35] = 699D6122	T[51] = AB9423A7
T[4] = C1BDCEEE	T[20] = E9B6C7AA	T[36] = FDE5380C	T[52] = FC93A039
T[5] = F57C0FAF	T[21] = D62F105D	T[37] = A4BEEA44	T[53] = 655B59C3
T[6] = 4787C62A	T[22] = 02441453	T[38] = 4BDECFA9	T[54] = 8F0CCC92
T[7] = A8304613	T[23] = D8A1E681	T[39] = F6BB4B60	T[55] = FFEFF47D
T[8] = FD469501	T[24] = E7D3FBC8	T[40] = BEBFB7C0	T[56] = 85845DD1
T[9] = 698098D8	T[25] = 21E1CDE6	T[41] = 289B7EC6	T[57] = 6FA87E4F
T[10] = 8B44F7AF	T[26] = C33707D6	T[42] = EAA127FA	T[58] = FE2CE6E0
T[11] = FFFF5BB1	T[27] = F4D50D87	T[43] = D4EF3085	T[59] = A3014314
T[12] = 895CD7BE	T[28] = 455A14ED	T[44] = 04881D05	T[60] = 4E0811A1
T[13] = 6B901122	T[29] = A9E3E905	T[45] = D9D4D039	T[61] = F7537E82
T[14] = FD987193	T[30] = FCEFA3F8	T[46] = E6DB99E5	T[62] = BD3AF235
T[15] = A679438E	T[31] = 676F02D9	T[47] = 1FA27CF8	T[63] = 2AD7D2BB
T[16] = 49B40821	T[32] = 8D2A4C8A	T[48] = C4AC5665	T[64] = EB86D391

T table

**SSN College of Engineering,
Department of Computer Science and Engineering
CS 6711 Security Laboratory**

Exercise 7:

To implement the message digest SHA1

Programming Language: Java

Hints:

1. Read the message
2. Divide the message into 512 bit blocks.
3. Append padding bits,
 - A single “1” bit is appended to the message, and then “0” bits are appended so that the length in bits of the padded message equals to $448 \bmod 512$.
4. Append length
 - A 64-bit representation of the length of the message is appended
5. Initialize MD buffers, A, B, C, D,E
 - word A: 67452301
 - word B: efc dab89
 - word C: 98badcfe
 - word D: 10325476
 - word E: c3d2e1f0
6. Invoke the compress function for four times
7. Display the message digest from the buffers.

SHA1 Compression function:

1. $(A, B, C, D, E) \leftarrow (E + f(t, B, C, D) + (A \ll 5) + W_t + K_t), A, (B \ll 30), C, D)$

- t is the step number
- W_t is derived from the message block
- K_t is a constant value derived from sin table.
 - $K(t) = 5A827999 \quad (0 \leq t \leq 19)$
 - $K(t) = 6ED9EBA1 \quad (20 \leq t \leq 39)$
 - $K(t) = 8F1BBCDC \quad (40 \leq t \leq 59)$
 - $K(t) = CA62C1D6 \quad (60 \leq t \leq 79)$
- Each $f(t)$, $0 \leq t \leq 79$, operates on three 32-bit words B, C, D and produces a 32-bit word as output.
 - $f(t; B, C, D)$ is defined as follows: for words B, C, D ,
 - $f(t; B, C, D) = (B \text{ AND } C) \text{ OR } ((\text{NOT } B) \text{ AND } D) \quad (0 \leq t \leq 19)$
 - $f(t; B, C, D) = B \text{ XOR } C \text{ XOR } D \quad (20 \leq t \leq 39)$
 - $f(t; B, C, D) = (B \text{ AND } C) \text{ OR } (B \text{ AND } D) \text{ OR } (C \text{ AND } D) \quad (40 \leq t \leq 59)$
 - $f(t; B, C, D) = B \text{ XOR } C \text{ XOR } D \quad (60 \leq t \leq 79)$.
- $\ll s$ circular left shift of 32 bit argument by s bits
- $+$ is addition modulo 2^{32}

2. Perform a 32 bit circular right shift such that;

- $a=e; \quad b=a; \quad c=b; \quad d=c; \quad e=d;$

For further clarification, read the associated pdf

T[1] = D76AA478	T[17] = F61E2562	T[33] = FFFA3942	T[49] = F4292244
T[2] = E8C7B756	T[18] = C040B340	T[34] = 8771F681	T[50] = 432AFF97
T[3] = 242070DB	T[19] = 265E5A51	T[35] = 699D6122	T[51] = AB9423A7
T[4] = C1BDCEEE	T[20] = E9B6C7AA	T[36] = FDE5380C	T[52] = FC93A039
T[5] = F57COFAF	T[21] = D62F105D	T[37] = A4BEEA44	T[53] = 655B59C3
T[6] = 4787C62A	T[22] = 02441453	T[38] = 4BDECFA9	T[54] = 8F0CCC92
T[7] = A8304613	T[23] = D8A1E681	T[39] = F6BB4B60	T[55] = FFEFF47D
T[8] = FD469501	T[24] = E7D3FBC8	T[40] = BEBFBC70	T[56] = 85845DD1
T[9] = 698098D8	T[25] = 21E1CDE6	T[41] = 289B7EC6	T[57] = 6FA87E4F
T[10] = 8B44F7AF	T[26] = C33707D6	T[42] = EAA127FA	T[58] = FE2CE6E0
T[11] = FFFF5BB1	T[27] = F4D50D87	T[43] = D4EF3085	T[59] = A3014314
T[12] = 895CD7BE	T[28] = 455A14ED	T[44] = 04881D05	T[60] = 4E0811A1
T[13] = 6B901122	T[29] = A9E3E905	T[45] = D9D4D039	T[61] = F7537E82
T[14] = FD987193	T[30] = FCEFA3F8	T[46] = E6DB99E5	T[62] = BD3AF235
T[15] = A679438E	T[31] = 676F02D9	T[47] = 1FA27CF8	T[63] = 2AD7D2BB
T[16] = 49B40821	T[32] = 8D2A4C8A	T[48] = C4AC5665	T[64] = EB86D391

T table

PGP Tutorial For Windows (Kleopatra – Gpg4Win)

Basically, each individual has a unique PGP key. In the program GPA, you import people's unique key to your list of keys. When you go to write a PGP message, you type it normally in the clipboard { you'll learn about the clipboard later, it's your friend } and then press an encrypt button, which then lets you pick from your unique list of keys to encrypt to, where ONLY that person can read it. [this is why people give their public keys out, so anyone can encrypt them a message] === THE STEPS ===

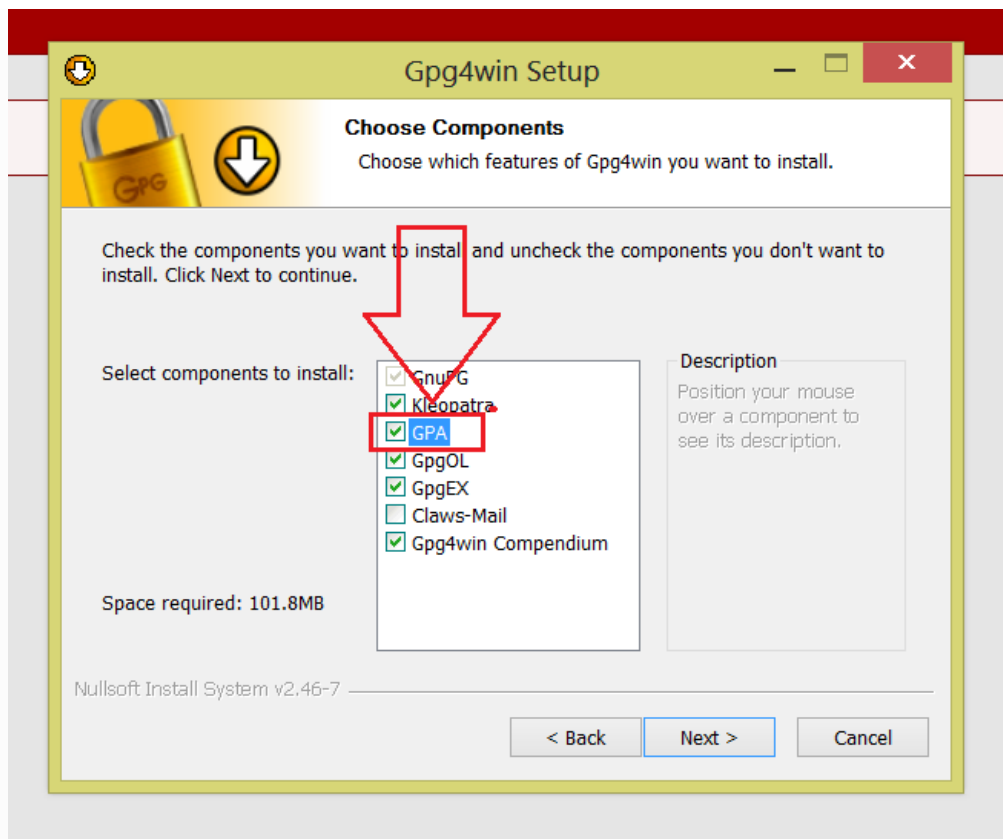
– Step One –

Okay, so first things first, let's get a PGP program. One of the most popular is GPA. Head over to this link to download gpg4win which includes GPA {you can see a list of the programs gpg4win contains to the left of the download page, GPA is one of them}

Download: <http://gpg4win.org/download.html>

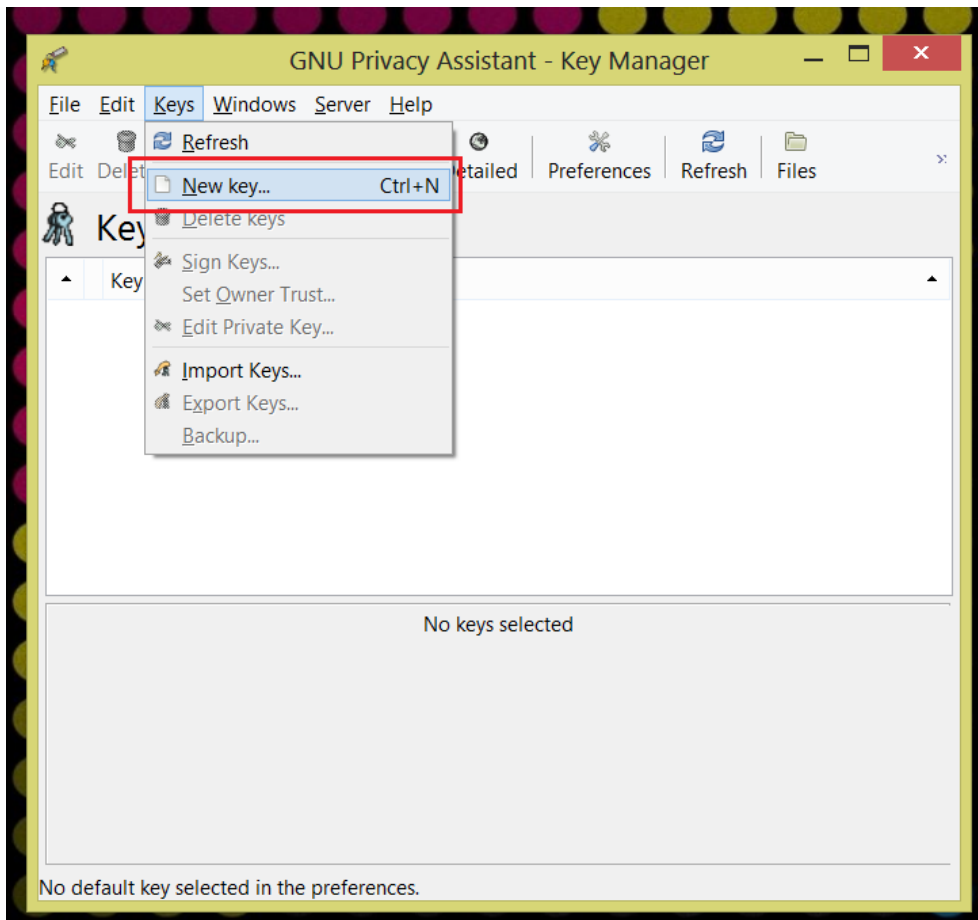
IMPORTANT !!!!!!! *****

When installing gpg4win you get the option to install which programs you want from the package. By default, GPA is not checked. MAKE SURE YOU CHECK GPA! You need it in order to easily encrypt and decrypt messages. This is what it looks like during the installation:



Next, you want to make a PGP key. Remember, none of the details need to be valid. I'd use your online name or a different alias when making your key. Something that isn't your gamertag for online games, or anything that may tie to you. A completely new alias. The e-mail doesn't need to be valid at all. Here are some pictures to help you through the process. Also make a backup of your key!!!

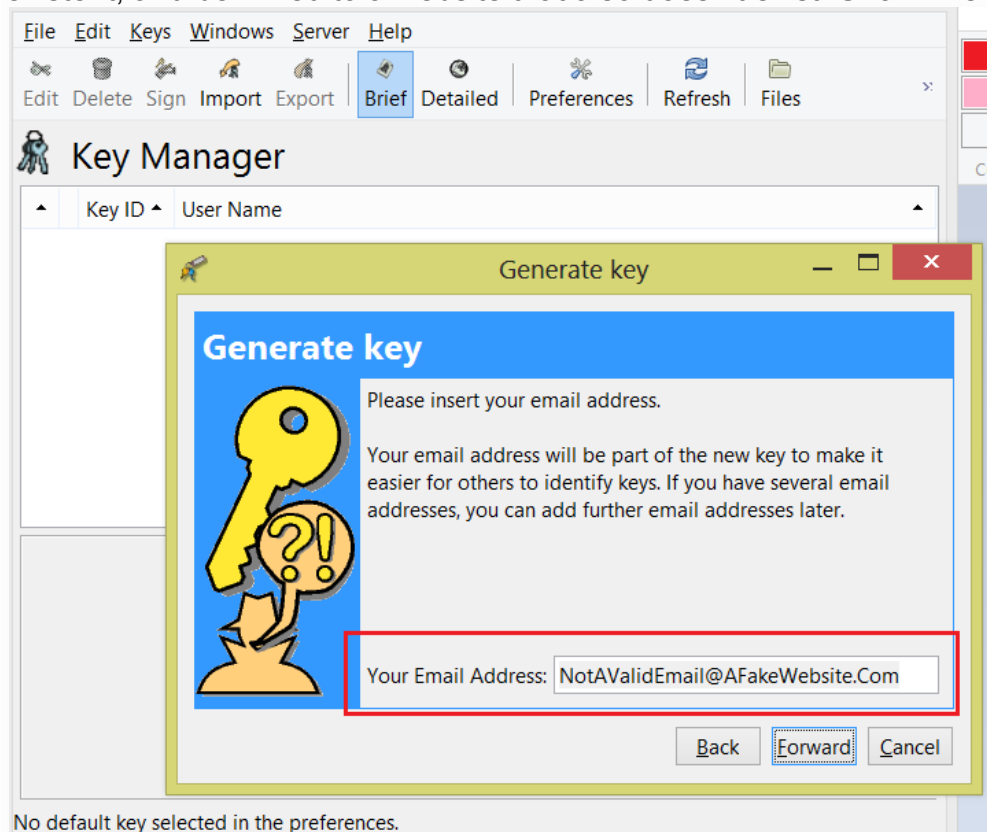
First, click the keys in the menu at the top. Alternatively, you can click CTRL+N to begin the process of creating a key. Shown here:



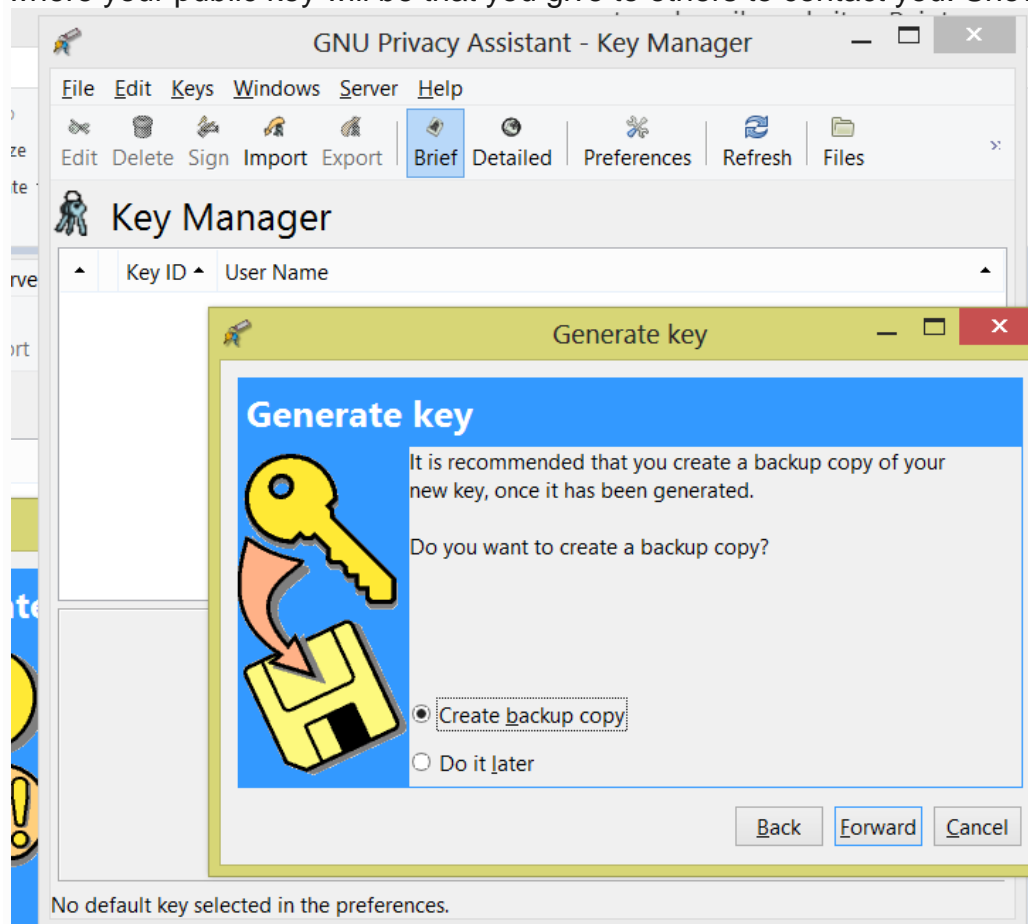
You will go through a set up, where you make a name for your key, which I suggest you use an alias. Shown here:



After selecting your alias it asks for an e-mail address. This e-mail should be non-existent, and be linked to a website that also doesn't exist. Shown here:



Then you're asked to make a backup of your key. I highly suggest you do this! Although you can make a back up at any time, you should just do it now. This is where your public key will be that you give to others to contact you. Shown here:



– Step 2 – Find Your Key –

Find where you put the back up of your key. It will be an .asc file but no worries, when asked to open the file just tell windows or whatever OS to open it using Notepad. Here you will find a public key similar to this.

```
[g]ley - Notepad
File Edit Format View Help
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (Mingw32)

mQENBFBKJzIBCAC7LKgvc4CbkgUSPyImk8Hl6whVnTZuUw0VrUveNvGFT3ZzLEXN
M5vn51zkKfug+AkX15/FZ0hYZkBdC40g7p5tVbb39p01qeJ5qNDN+ewD88a8fjwB
I0gqkkMN6D9IMachUghkg1TXdtMUD4p1iuc9bU5HNMSc8aqJLfgnJ8qcSEd5BGwV
WP6ZRXJARTrzvpK7/sOm0Idme4GPaD5hyaQa24vHnyjGYXgxGB8kUMI/FV1QhgZ4
jukUZ+06nQqjBgJLCBTEbyqTM6+2MxeaK6xCOP1pYIPgai8ncdAuSCCFJtOKr1Ah
evzS8TwxFLpMQ0+d6DqKH4pPIG6sXkQZS0KrABEBAAg0MU5vdFlvdXJSZWfStmFt
ZSA8Tm90QVZhbg1kRW1hawXAQUZha2VXZWJzaXRlLnVbT6JATkEEwECACMFA1KB
JzICGwMHCwkIBwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRADugESIEl1pcCACf
k4RjMrxr45K1TLmU/R1J8ISYZFksLEdR//RgC5JAG6nLrWUcjmxjP1+cmYis+MZD
2krNud6wAuhYKz6QNaAlGMB8tSwH38348HCwaue1VVNXBxv21HeGQ57Peh5Qy3Jg
/I/tEwaSmmL+b290jTvc6ALesYUKpOcfx0iGyfyksie3eHDSioV3qIWJdsdfqheQ
oZyadezM3L64Kmi6syxqEHYOru+613jzJgD0+qVupThw80v1q/E/N0Yt91eIUhhc
+2p8m8SjZSruqrZBP4xwFuhDHPvKyCL1QW9Llirx84wjr4vjvY639oewQzB09wC
hAPBzH3MeTLe7AI1eqweuQENBFBKJzIBCADSYYNprCHlwBX//1nd0fMH5o/1bk6P
e9aEvfrte7DH0H5mQ4xYAPndDNBOUCH82Uvwxzd9+yvfiQgNS3hEwVyzQa9SYYQa
CbDmIHDACHUbynAN2LCgpFj/yjs1r3ZkvDY011p1rnyckWxUOMqtirTM8FdGtYJSN
QPnwi/YMQkZc2cnWM3e/64TW5bh30Tvj69QFK3fytM4LCZdNyqA9ccYcfvX1zDkd
fsroyiKzwmK2vE8hf/2I1gLSBzL6NY26L1RgzHyNveUVBhQKZ9GcPWNeBsEyEQQ
8JQS71JRPHB5afUKNcWknm6BPLlJkzi+cgyxp+YJ8bBgL7anyB3/7yDrABEBAAgJ
AR8EGAECaAKFAlKBzICGwwACgkQJAA7oBEiBJYpoAgAsLYLs+WctnQasNnz8APT
Vfm08bYSJ9hpnLVXuwlCh47/EgxvaSWLU0HUil8iQIED13AYT1zfIwM5vMbxae0U
7UHyTng075kghUD2wr4G1BAXt2eDv7rcUuqEMYA0EH535L31t03pI1cz88vG/uwW
Ya/E4fbQh/OfU5Qov80wOcZrBM1ay2UMA7XJCQANeryOpdcSRuPDlHsaJfuKMV/f
BVSD4cgySSRQb5h75ljHZLGoCqmwZf0UQ9JSxSxFrM/0LcbIpnCS5kcGKDrSIEft
bVgm0GsQH8KvCewwXesJVNc5C6jVMK9iSw99Ltm2BRTGvmD5TUIpQgA5GcbY/0g
YQ==
=p7FU
-----END PGP PUBLIC KEY BLOCK-----
```

When sharing your key with others, you want to copy and paste from the beginning dashes to the end dashes. Exactly how I have copied and pasted above.

— HOW TO IMPORT SOMEONE ELSE'S PGP KEY TO YOUR GPA PROGRAMS

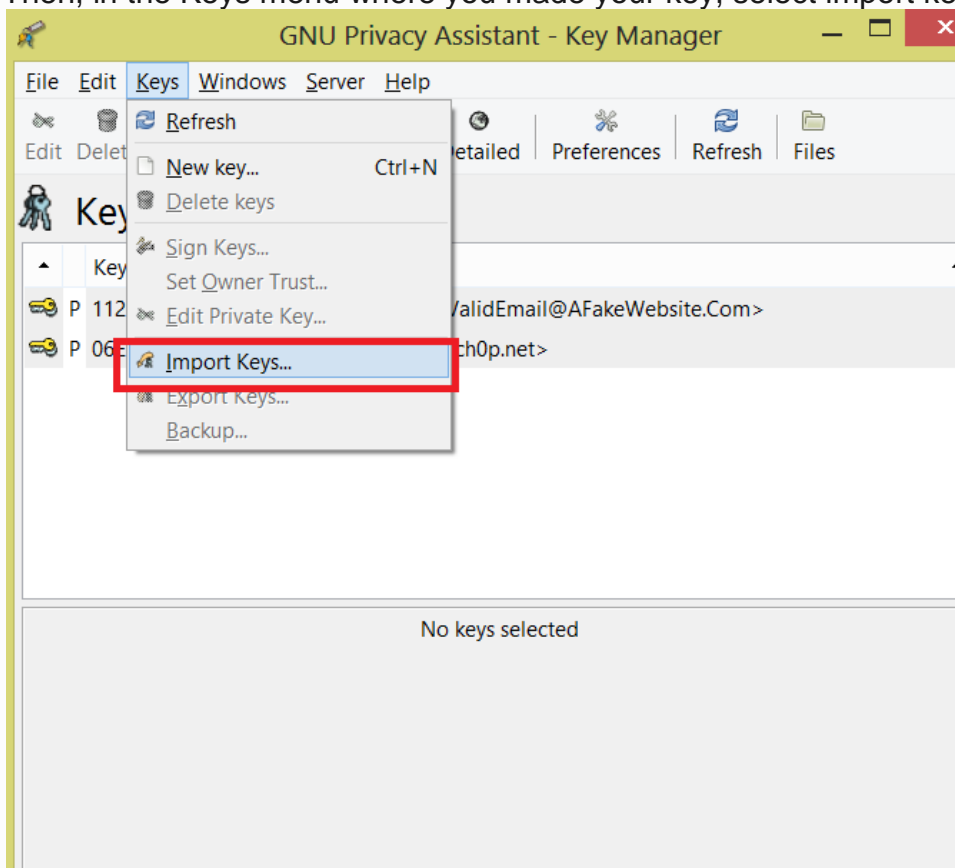
You see people giving their public keys away so others can contact them. Simply open a notepad file, copy and paste their key and import it using the GPA program. I will show you how to do this.

First make a blank text file and copy the user's public key to it. Shown here:

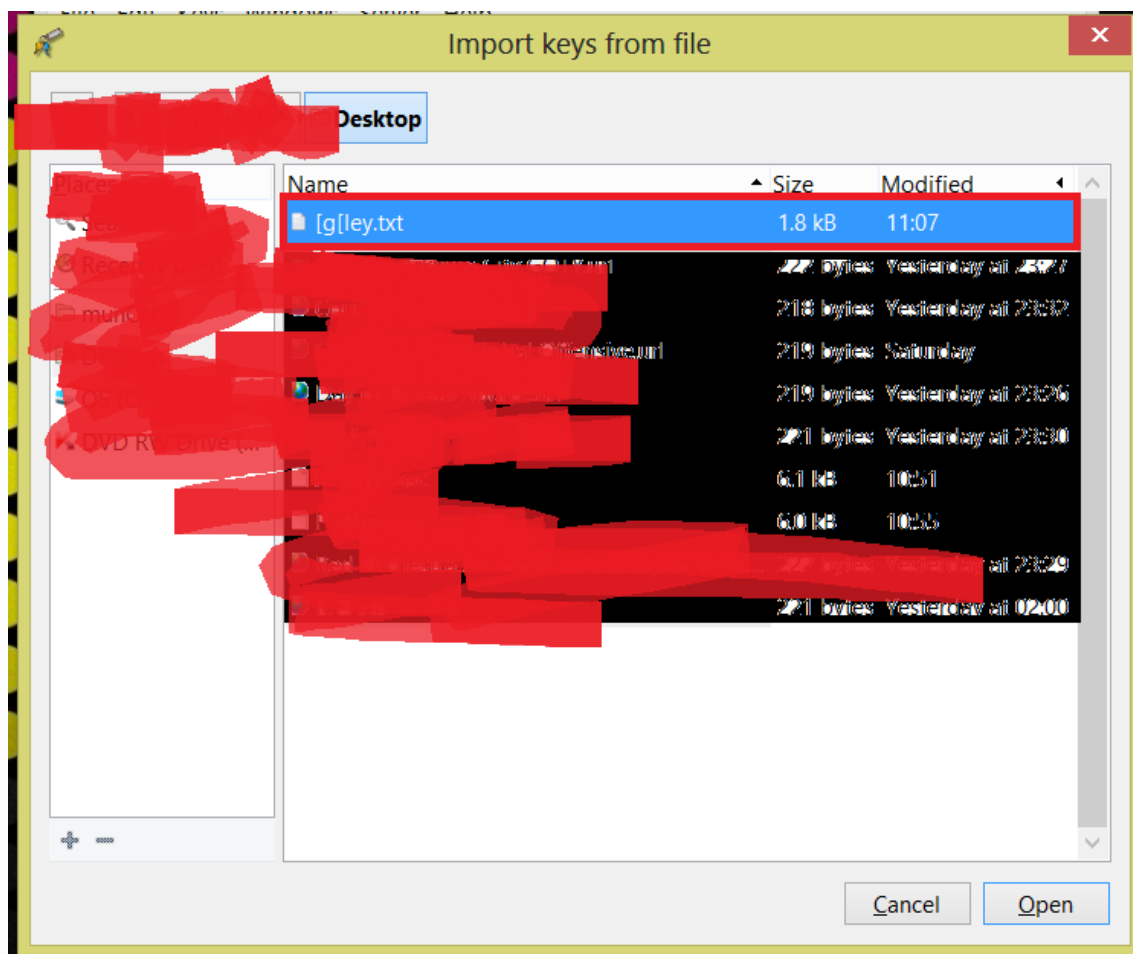
```
gitley - Notepad
File Edit Format View Help
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (MingW32)

mQENBFKBjZIBCAC7LKgvc4CbkgUSPYimk8Hl6whVnTzuUw0VrUveNvGFT3Zz1ExN
M5vn51zkKfUg+AkXi5/FZ0hYZkBDc40g7p5tvbb39po1qeJ5qNDN+eWD88a8fjwB
I0gqkkMN6D9IMachUghkg1TXdtMUD4p1iuc9BU5HNMSc8aqJLfgnJ8qcSEd5BGwV
WP6ZRXJARTrzvpK7/sOm0Idme4GPAd5hyaQ24vHnyjGYXgxB8kUMI/FV1QhGz4
jukUZ+06nQqjBgJLCBTEbyqTM6+2MxeaK6xCOP1pYIPgai8ncdAuSCCFJtOKr1Ah
evzS8TwxFLpMQ0+d6DqKH4pPIG6sXkQZS0KrABEBAAg0MU5vdf1vdXJ5ZWFSNmFt
ZSA8Tm90QVZhbG1kRW1hawxAQUZha2VXZWJzaXR1LkNvbT6JATKEEwECACMFAlKB
JzICGwMHCwkIBWMCAYVCAIJCgsEFgIDAQIeAQIXgAAKCRAKADugESIelipcCACf
k4RjMrxr45K1TLMU/R1J8ISYZFksLEdR//RgC5JAG6nLrWUcjmXjP1+cmYis+MzD
2krNud6wAuhYKz6QNaAlGMB8tSwH38348HCwaue1VVNXBxv21HeGQ57Peh5Qy3Jg
/I/tEwaSmML+b290jTVc6ALesYUKpOcfX0iGyfyksie3eHDSioV3qIWJdsdfqheQ
oZyadezM3L64Kmi6syxqEHYOru+6l3jZjgD0+qVupThW80v1q/E/N0Yt9leIUhhc
+2p8m85JZSruqrZBP4xwFuhDHPvKyCL1QWq9LlIRx84wjr4vjVY639oewQZb09wC
hAPBzH3MeTLe7AI1eqweuQENBFKBjZIBCADSYNprCHLWBX//1nD0fMH50/1bk6P
e9aEvfrte7DH0H5mQ4xYAPndDNBOUCH82Uvwxzd9+yvfiQgNS3hEwVYzQa9SQYQa
CbDmIHDACHubynAN2LCgpfj/yJ3s1r3ZkvDY011p1rnycwXu0MqtirTM8fDgtYJ3N
QPnwi/YMQkZc2cnwM3e/64TWsBh30TvJ69QFK3fytm4LCzdNyqA9cCycFvX1zDkd
fsroyiKzwmK2ve8hf/2I1gLSBzL6NY26L1RgzHyNveUVBhQKKz9GcPWNeBsEYEQQ
8Q57L1RPHB5afUKNcwknm6BPL1jKzi+cgyxp+YJ8bBgL7anyB3/7yDrABEBAAgJ
AR8EGAECaAKFA1KBjZICGwACQKQAA7oBEiBJYpoAgAsLYLS+WCTnQasNnz8APT
Vfm08bYSJ9hpnLVXuWLC47/EgxvaSWLU0HUil8iQIE013AYT1zFIwM5vMbxae0U
7UHytnG075kghUD2wr4G1BAXt2edv7rcUuqEMYA0EH535L31t03pI1cz88vG/uwW
Ya/E4fbQh/OFUSQoV80wOczrBM1ay2UMA7XJCQANeryOpdcsRuPD1HsaJfuKMV/f
BVsd4cgySsRQb5h751jHZLGoCqmWZf0UQ9J5sXsFrM/0LcbIpnCS5kcKDrSIEft
bVgm0G5QH8KvCewwXesJVnc5C6jVmK9iSw99Ltm2BRTGvmOd5TUIpQgA5Gcby/0g
YQ==
-----END PGP PUBLIC KEY BLOCK-----
```

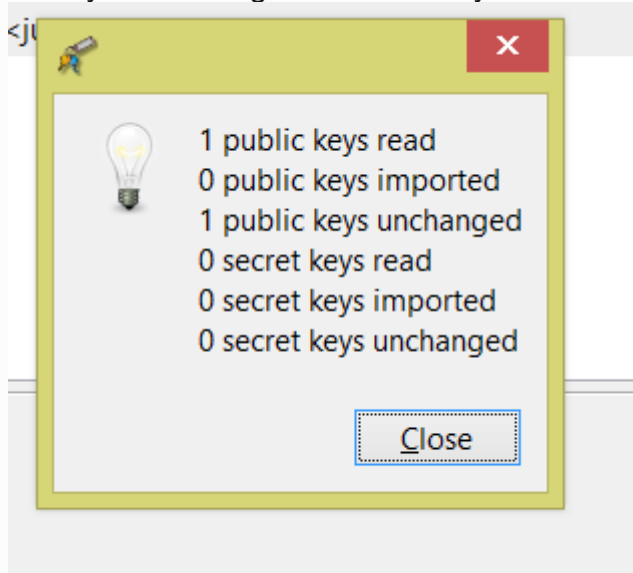
Then, in the Keys menu where you made your key, select import keys. Shown here:



Select the Text file you saved with the public key in it. Shown here:

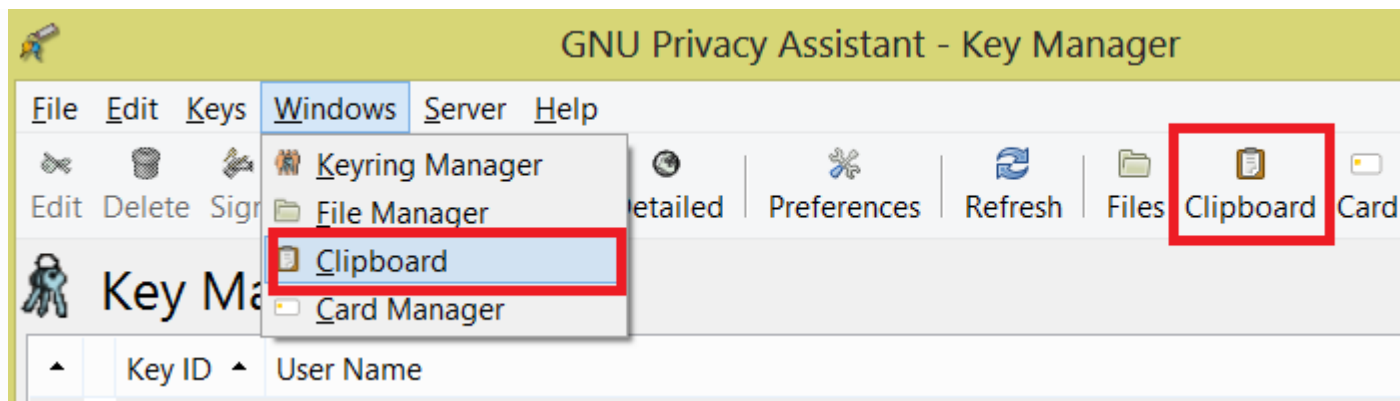


Then you should get this if the key was successfully imported:

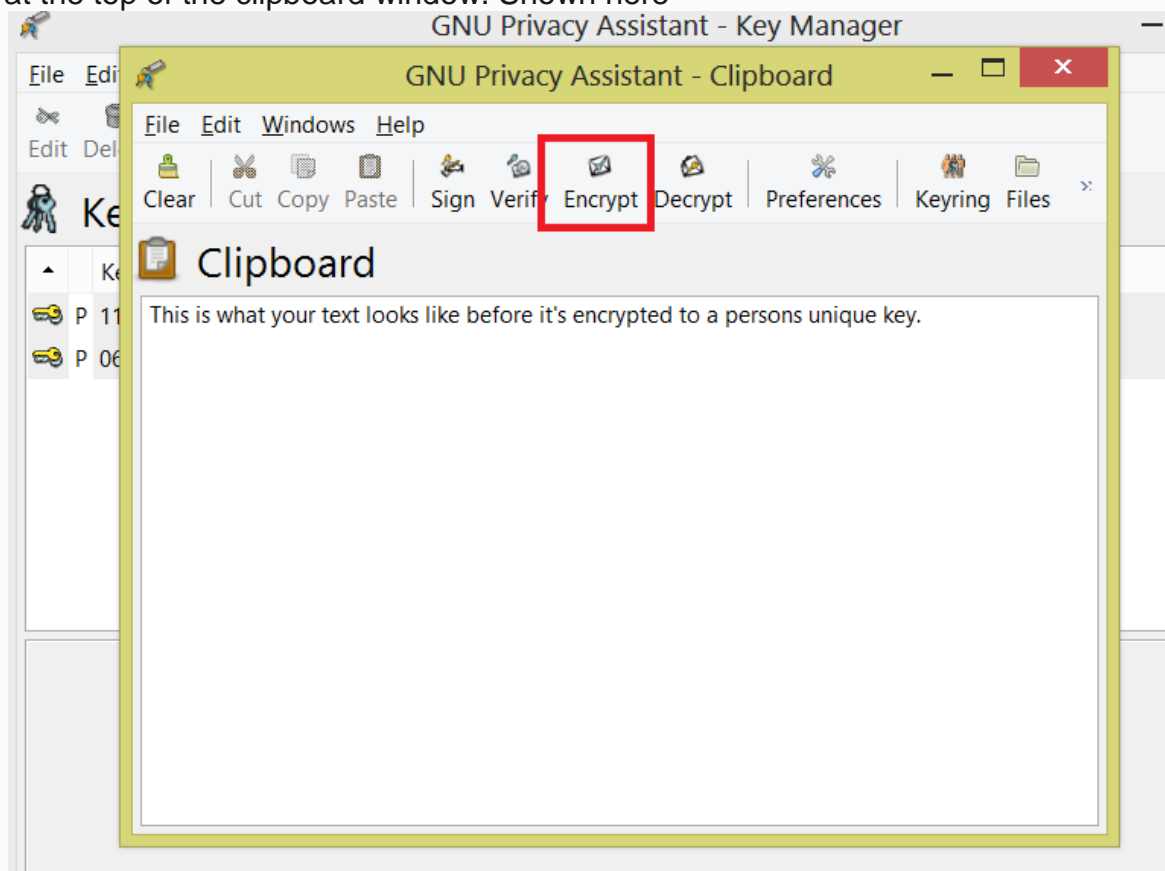


Now, let's send an encrypted message.

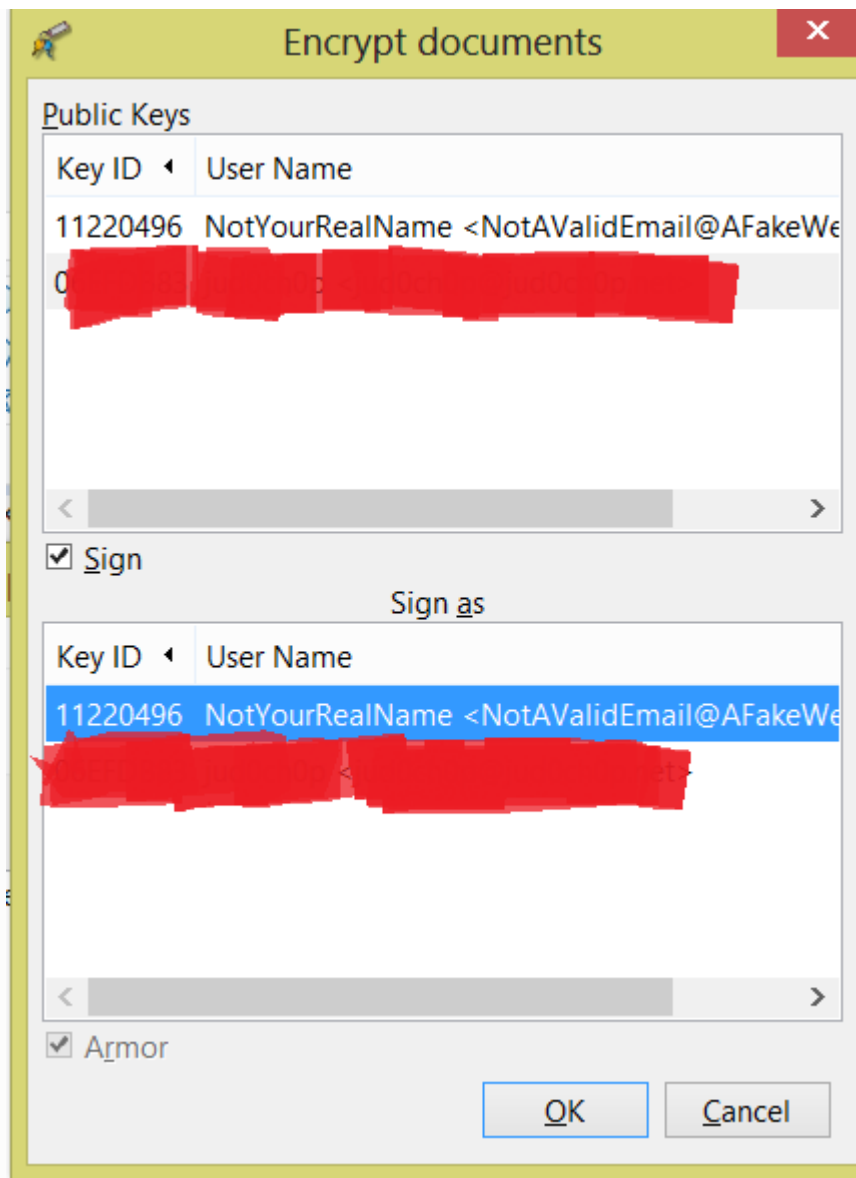
First, open the clipboard. You can get there through the Windows menu or through the clipboard icon on the quickbar. Shown here:



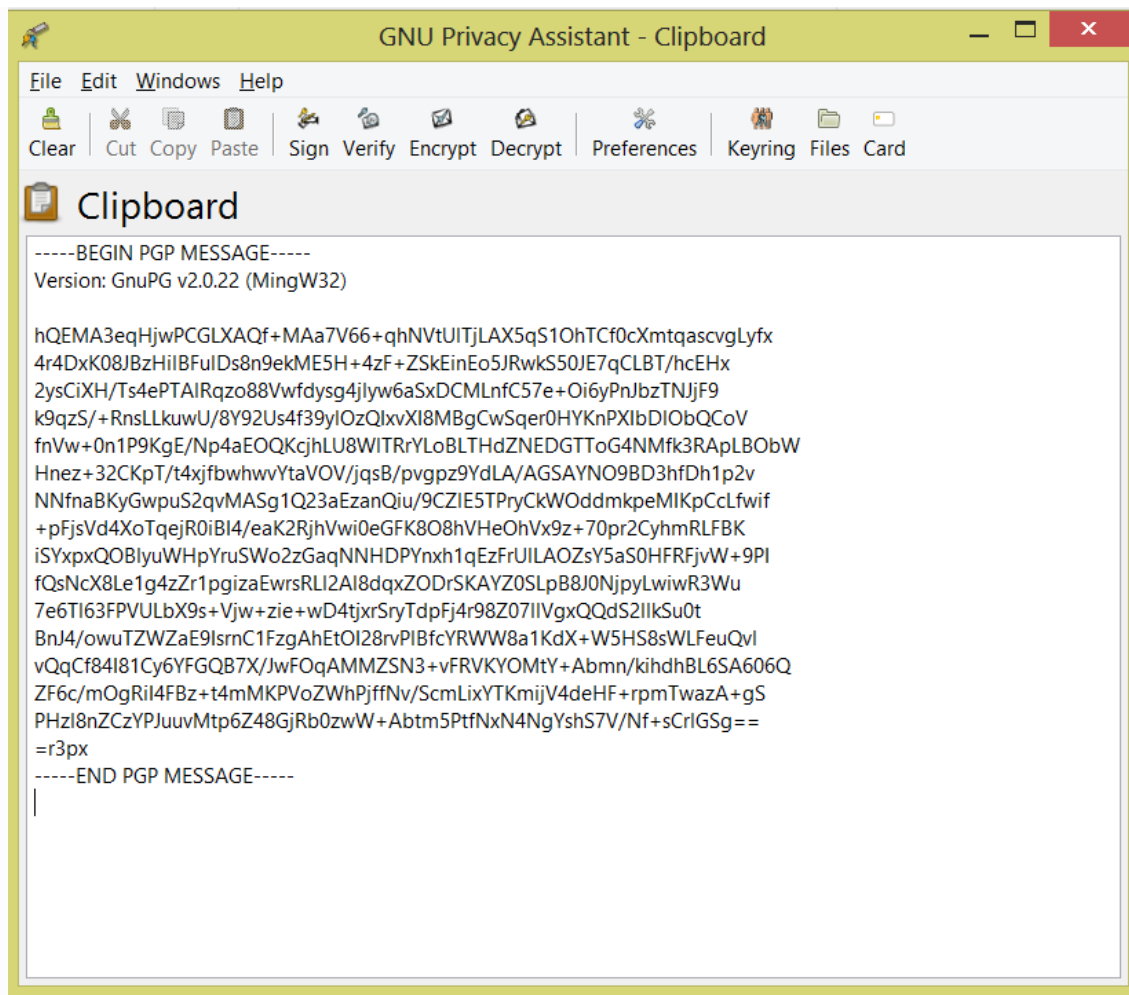
Then after opening clipboard type the message you'd like to send and select encrypt at the top of the clipboard window. Shown here



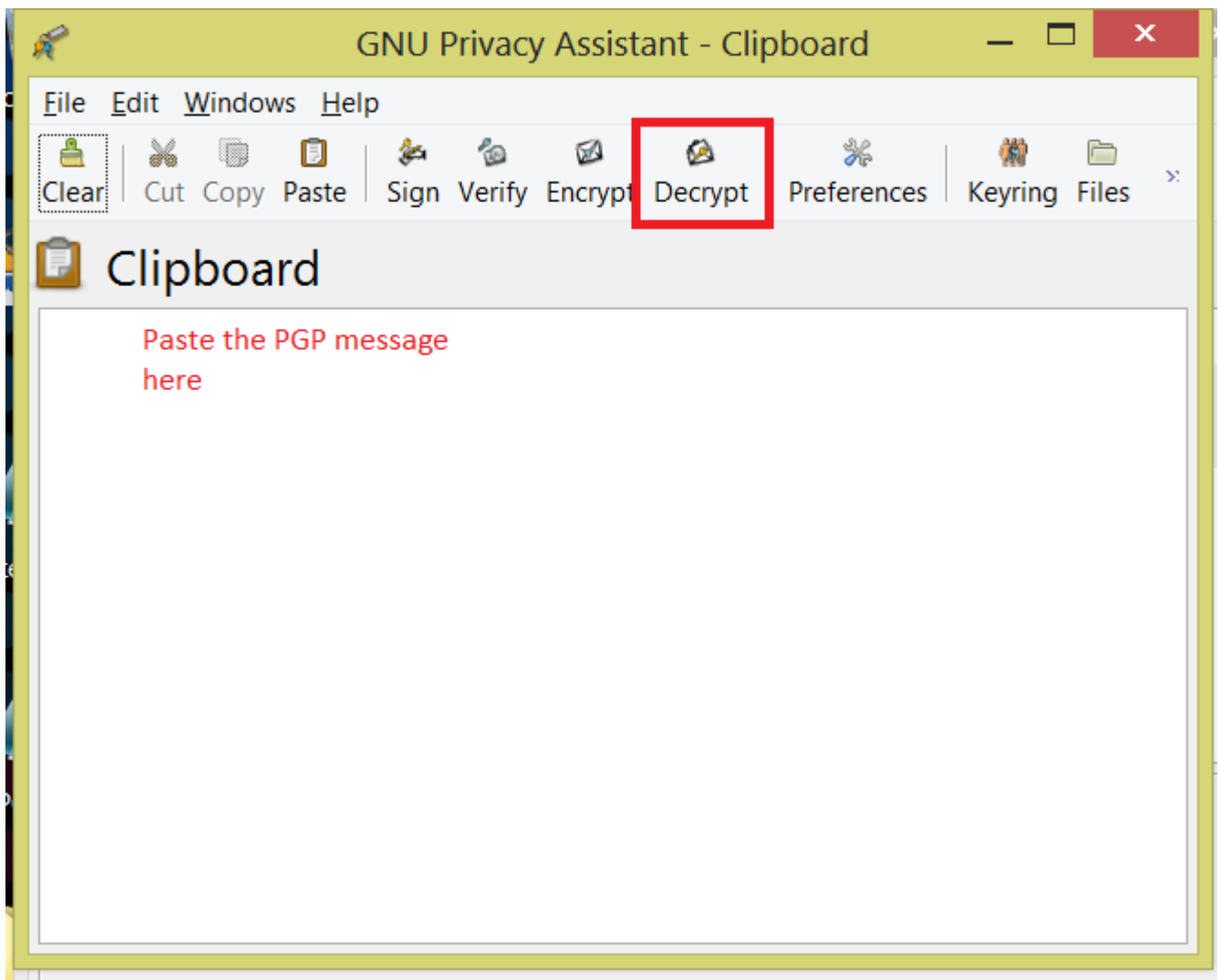
When you press encrypt, you are given a menu shown below. In this menu you select what key you're using to send the message, and what key is going to be receiving the message. I chose to send the fake account used to make this tutorial a message with my personal account. Here's what that menu looks like:



After you select who's sending and who's receiving you should get an encrypted message that looks like this:



This encrypted message is what you send instead of cleartext. So when messaging on websites, simply paste the PGP message. If you receive a PGP message, you can also use the clipboard to decrypt the message you have received by opening the clipboard, pasting the PGP message you got, and then pressing the decrypt button, shown here:



That about sums it up. I hope that people with questions on PGP and how it's used can be solved here, as I tried to make the tutorial as noob as possible. Please be safe when communicating confidential or sensitive information on websites. Always PGP. Never FE. Be safe people. If you have questions, comment, and I'll try my best to answer them.

**SSN College of Engineering,
Department of Computer Science and Engineering
CS6711 Security Laboratory**

Exercise 10:

To setup a honey pot and monitor the honeypot on network

Hints:

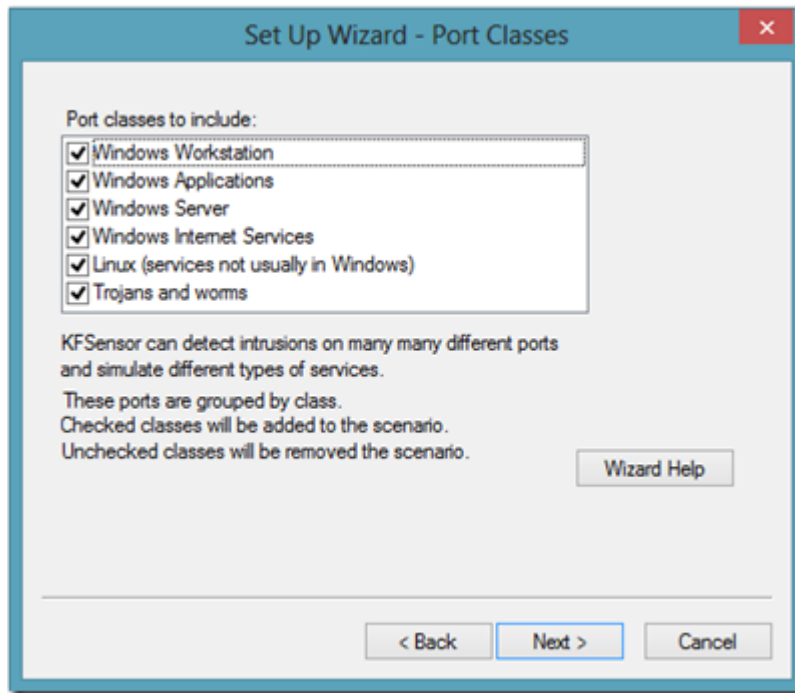
- Honey Pot is a device placed on Computer Network specifically designed to capture malicious network traffic.
- KF Sensor is the tool to setup as honeypot when KF Sensor is running it places a siren icon in the windows system tray in the bottom right of the screen.
- It performs by opening ports on the machine it is installed on and waiting for connections to be made to those ports.
- By doing this it sets up a target, or a honeypot server, that will record the actions of a hacker.
- KFSensor_Server- Performs core functionality by listening to both TCP and UDP ports on the server machine and interacts with visitors and generates events. It runs as a daemon at the background.
- **KFSensor Monitor :**
 - Interprets all the data and alerts captured by server in graphical form.
 - Using it you can configure the KFSensor Server and monitor the events generated by the KFSensor Server.
- **Sim server** is short for simulated server.
- It is a definition of how KFSensor should emulate real server software.
- A visitor is an entity that connects to KFSensor.
- Visitors could be hackers, worms, viruses or even legitimate users that have stumbled onto KFSensor by mistake.
- Visitors can also be referred to as the clients of the services provided by KFSensor.
- An event is a record of an incident detected by the KFSensor Service.
- For example if a visitor attempts to connect to the simulated web server then an event detailing the connection is generated.
- Events are recorded in the log file and displayed in the KFSensor monitor.
- KFSensor is rules based. All of the data that was produced was the result of KFSensor detecting certain types of activity and then using a rule to determine what type of action should be taken.
- We can easily modify the existing rules or add your own

Setting Up a KF Sensor HoneyPot:

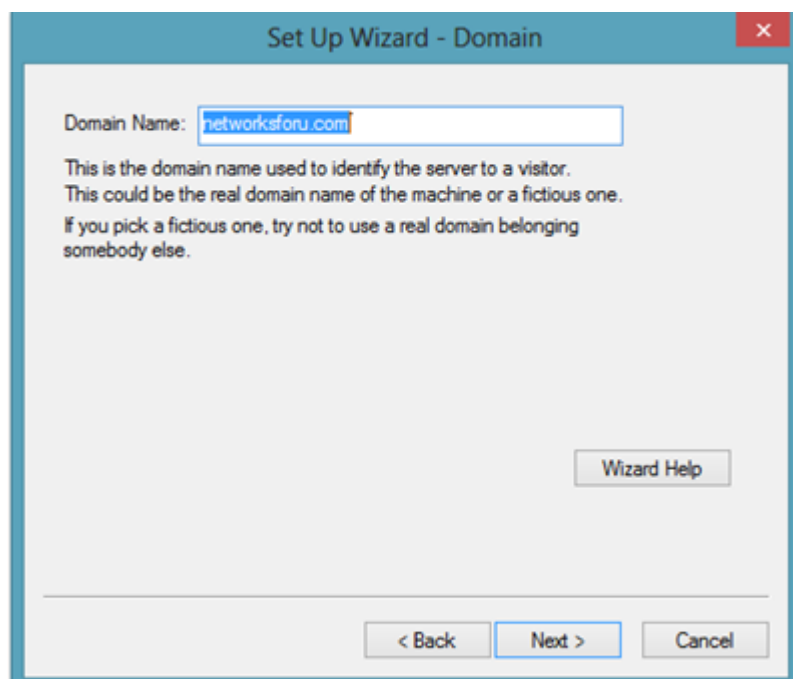
1. Download and install winpcap
2. Download KF Sensor Evaluation Set File from KF Sensor Website.
3. Install with License Agreement and appropriate directory path.
4. Reboot the Computer now.
5. The KF Sensor automatically starts during windows boot Click Next to setup wizard.
6. Select all port classes to include and Click Next.
7. Send the email and Send from email enter the ID and Click Next.
8. Select the options such as Denial of Service[DOS], Port Activity, Proxy Emulsion, Network Port Analyzer, Click Next.
9. Select Install as System service and Click Next.
10. Click finish.

Monitor the honeypot on network

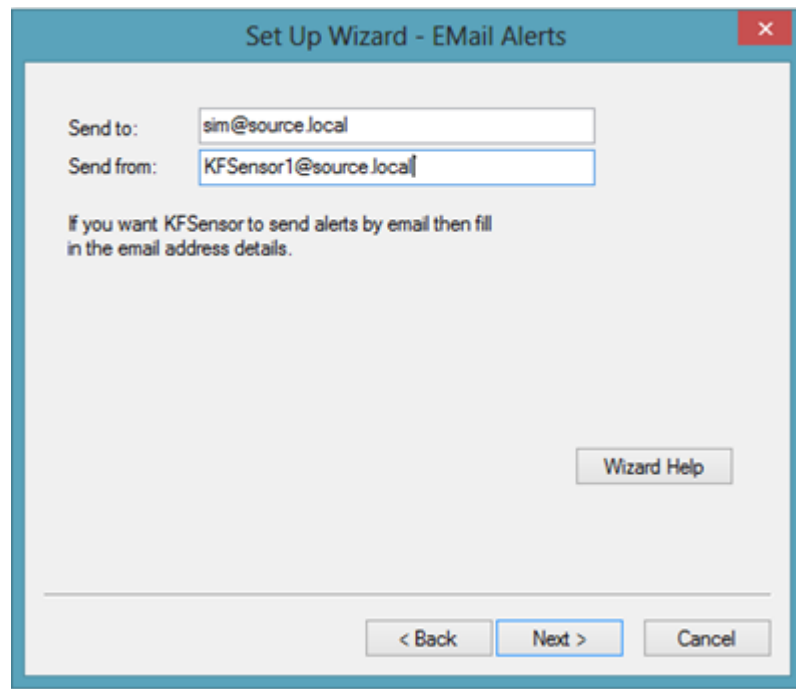
1. Select Settings > Set Up Wizard. The Set Up Wizard guides you through the configurations of:
 - Port Classes
 - Domain Name Selection
 - Email Alerts
 - Systems Service
2. Click the Next button to begin configuring KF Sensor. By default all the port classes will be selected.



3. Now you need to give your system a name. Use a fictitious name that may be attractive to someone who is doing discovery for “juicy” targets. For example, using the following words somewhere in your domain name may get you more hits: - credit - bank- financial- investment- accounting- private- internal. Enter your domain name (don’t forget to include the .com, .org, .net or whatever extension you are going to use). Click Next.



4. If you would like to receive email alerts of events, enter your target email address and the source email address in this window.



The image shows a Windows-style dialog box titled "Set Up Wizard - EMail Alerts". It has a blue title bar with a red close button. The main area is light gray. There are two text input fields: "Send to:" with the value "sim@source.local" and "Send from:" with the value "KFSensor1@source.local". Below these fields is a small text instruction: "If you want KFSensor to send alerts by email then fill in the email address details." At the bottom right is a button labeled "Wizard Help". At the bottom center are three buttons: "< Back", "Next >", and "Cancel".

5. Now you can configure the system services. Click the Wizard Help button for more details on each option.
- Denial of Service
 - Normal/Cautious
 - Port Activity
 - 1-12 Hours
 - Proxy Emulation
 - Allow banner grabs and loop backs
 - No external connections
 - Network Protocol Analyzer
 - Disable packet dump files
 - Enable packet dump files
6. Use the following settings and Click Next.

Denial Of Service Options

Normal

Controls how many events are recorded before the server locks up

Port Activity

1 Hour

How long a port should indicate activity after after an event

Proxy Emulation

Allow banner grabs and loop backs

Controls if KFSensor is allowed to make limited external connections

Network Protocol Analyzer

Enable packet dump files

Dump files are useful for detailed analysis but take up a lot of disk space

7. Now you are on the system service set up window. A system service allows KFSensor to run like a daemon on your system regardless of who is logged into it. You can change between users without affecting the system service. You must be logged in as the administrator to install the system service.
 "Install as a system service" should be selected.

Set Up Wizard - Systems Service

☒ Install as systems service

A systems service is a special type of application that Windows runs in the background and is similar in concept to a UNIX daemon.

The KFSensor Server becomes independent of the logged on user, so you can log off and another person can log on without affecting the server.

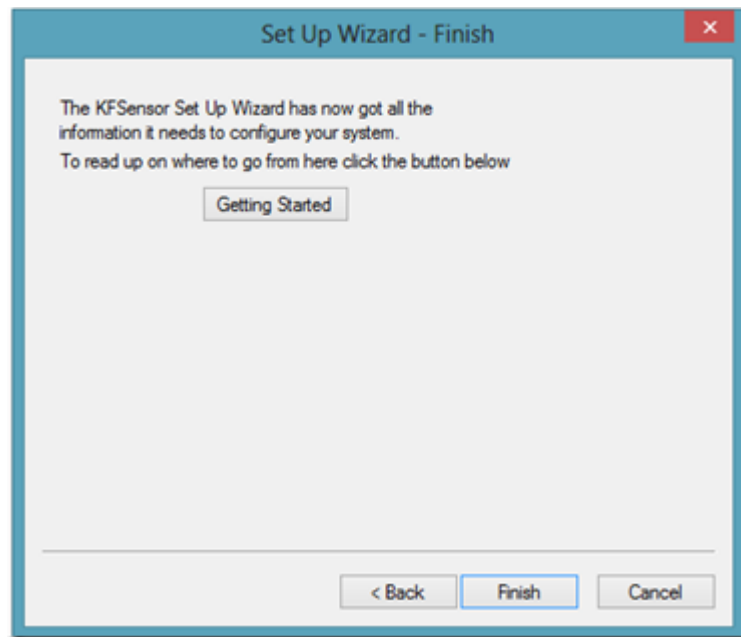
The KFSensor Server can be configured to start automatically when the systems starts, even before you log on.

You must be logged in a the Administrator to install a systems service.

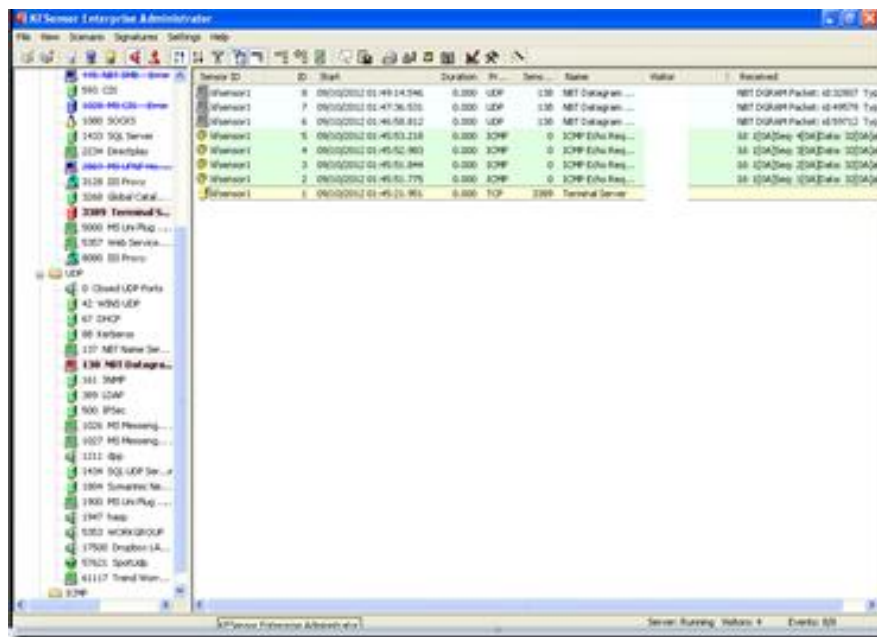
Wizard Help

< Back Next > Cancel

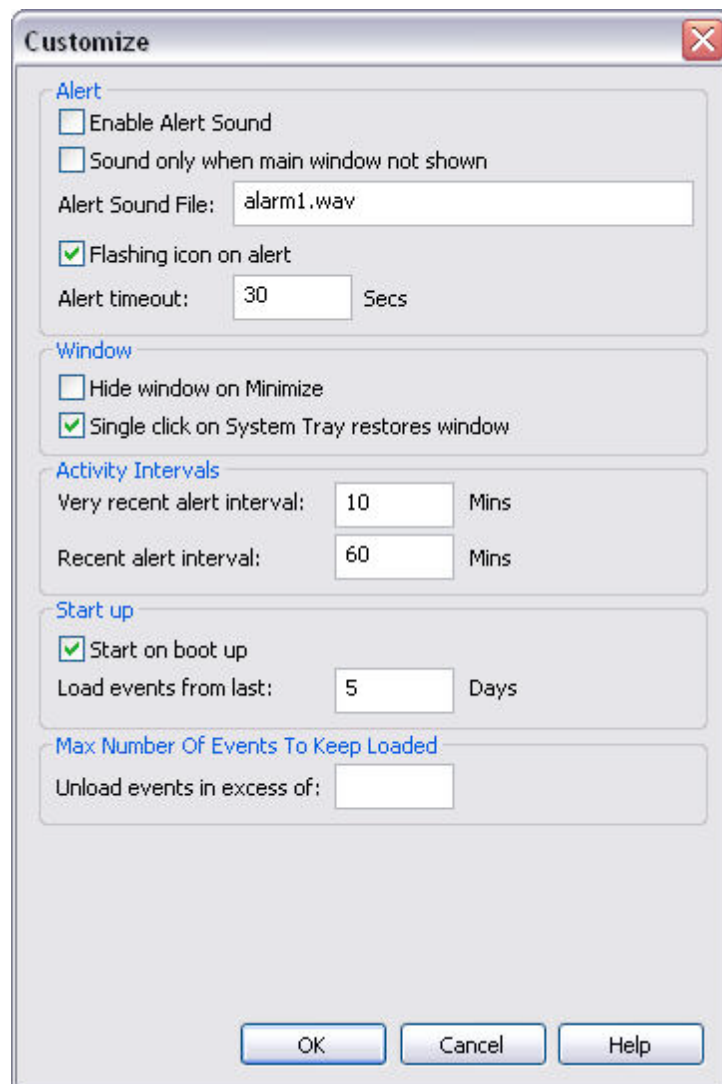
8. KFSensor should now be ready to configure your system. Click Finish.



After completing the setup wizard, you can see the KFSensor output by using the interactive GUI. Note: The “Visitor” filed contain the source IP/Host Name.



- Now we are going to customize KF Sensor. Select *Settings > Customize*. In this area you define the alert behavior, KFSensor window behavior, recent activity intervals, startup behavior and the maximum number of events to keep loaded. We definitely want to disable the audible alarm and we want to increase the number of events that are displayed when KFSensor starts up. Configure your KFSensor as shown next.



Click OK when you have set these configurations.

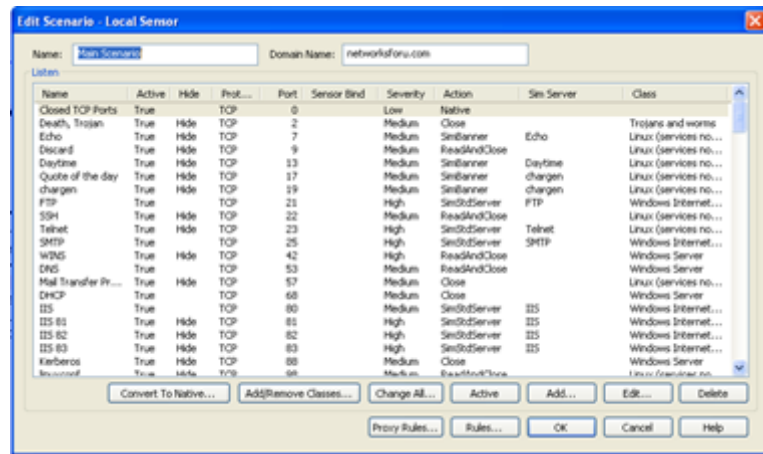
10. Edit Active Scenario: To create or modify rules, Scenario menu ->select the Edit Active Scenario command ->you will see a dialog box which contains a summary of all of the existing rules. Either select a rule and click the Edit button to edit a rule, or you can click the Add button to create a new rule.

11. Adding a rule :

- Click the Add button and you will see the Add Listen dialog box.
- The first thing that this dialog box asks for is a name. This is just a name for the rule.
- Pick something descriptive though, because the name that you enter is what will show up in the logs whenever the rule is triggered.

12. Convert to Native service:

- Convert the stroked off services as native services. *Select Scenario ->Edit Active Scenario.
- Choose the respective service listed in the dialog box opened and press convert to native button and ok.



13. Setting up Server :

- To start the server, Settings-> Set Up Wizard, Go through the wizard, give fictitious mail ids when they are asked and start the server running by pressing the finish button.
- Kfsensor now start showing the captured information in its window.

14. FTP Emulation:

- Open command prompt and type
 - Ftp ipaddress
 - Enter user name anonymous
 - Enter any password
 - Get any file name with path
- Monitor this ftp access in KFSensor monitor
- Right click KFSensor entry, select Event details, see the details captured by the server
- Create visitor rule by right clicking the FTP entry and check either ignore / close under actions in the dialog box that opened.

- Now redo the above said operations at the command prompt and see how the emulation behaves.
- You can see/ modify the created rules in Scenario->edit active visitor rules.

15. SMTP Emulation:

- open command prompt and type
 - telnet ipaddress 25
 - Helo
 - Mail from:<mail-id>
 - Rcpt to:<mail-id>
 - Data
 - type contents of mail end that with . in new line
- Check the kfsensor for the captured information.

16. IIS emulation:

- Enable Telnet client, server, Internet Information server in Control Panel-> Programs-> Turn windows features on/off
 - Check Telnet client, Telnet server, IIS-> FTP (both options),
- Create an index.html, store it in c:\keyfocus\kfsensor\files\iis7\wwwroot
- Select scenario->edit simserver
 - Choose IIS and edit
 - Make sure index.html is in first place in the listed htm files in the dialog box
- Check the kfsensor for the captured information.

17. DOS attack:

- Settings-> DOS attack settings modify (reduce) values in general tab, ICMP and other tabs. Press ok.
- Open command prompt and type
- Ping ipaddress -t or
- Ping -l 65000 ipaddress -t
- Check the kfsensor for the DOS attack alerts, open event details in right click menu for further details.

**SSN College of Engineering,
Department of Computer Science and Engineering
CS6711 Security Laboratory**

Exercise 11:

To install rootkit and to study about the variety of options.

Introduction :

- 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).
- 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

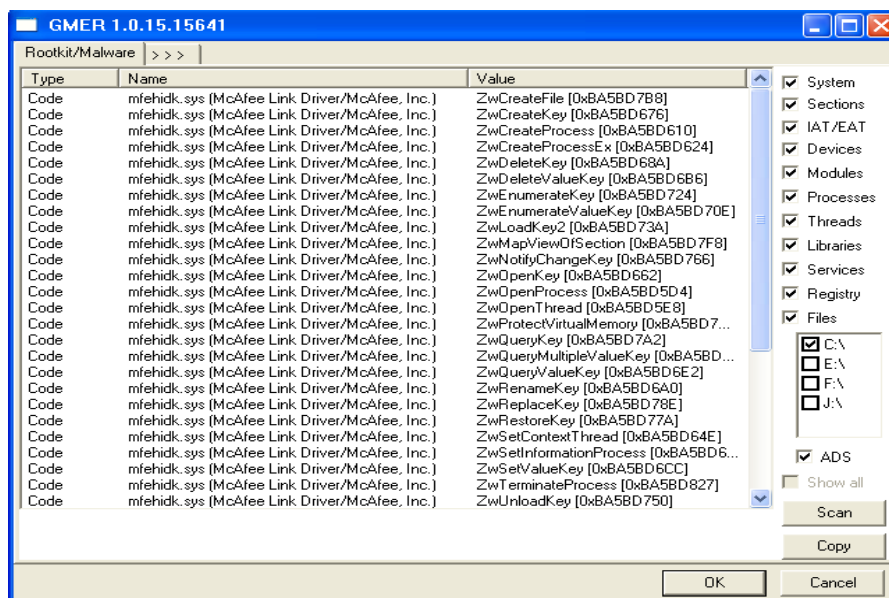
GMER :

- GMER is a free rootkit detector developed by Przemyslaw Gmerek, a Polish security researcher.
- Some of the features that GMER provides include detection of hidden processes, threads, services, files. It also provides removal and restoration options if a rootkit is detected.

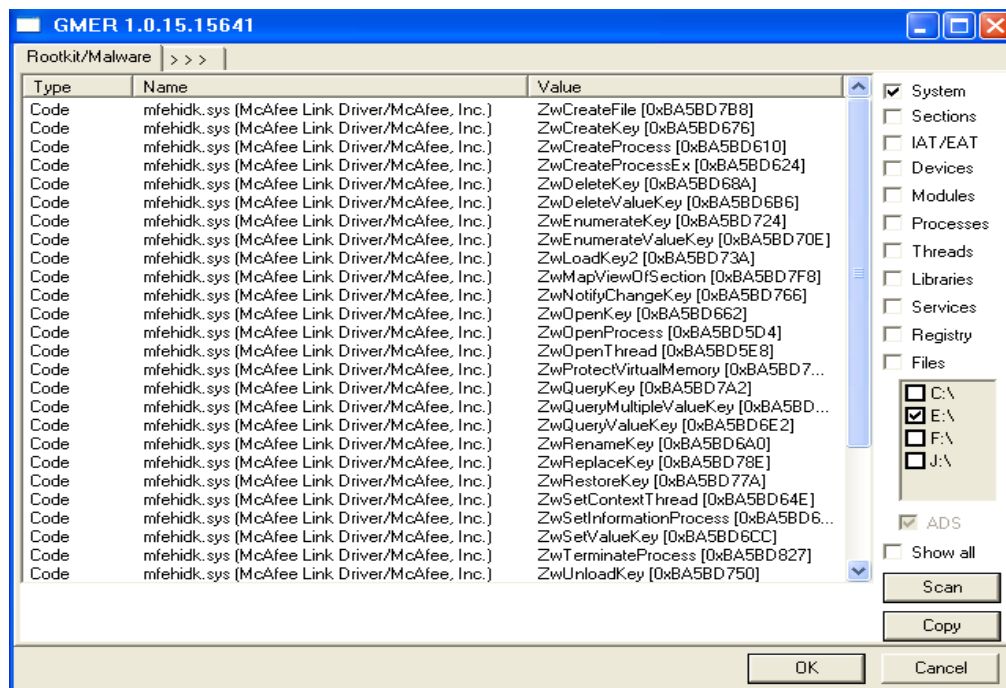
- GMER has built-in protection by hooking various Windows OS services to prevent malware from interfering with its operation.
- Additionally, GMER randomly changes the name of its running process as another method of self-protection.
- GMER has the ability to scan and display all currently loaded drivers and tell you whether they are hidden and whether the drivers file is visible on disk.
- It scans for hidden, locked or falsified files on the system
- It scans and displays the currently running processes (similar to Process Explorer) but shows if the process is hidden or locked.
- It scans for Stealth objects which looks for rootkit symptoms in general.
- It scans for Hidden services and displays them.
- Once you have found something malicious, you can right click on the driver/file/service and either copy, wipe or force delete it.

Installation of GMER:

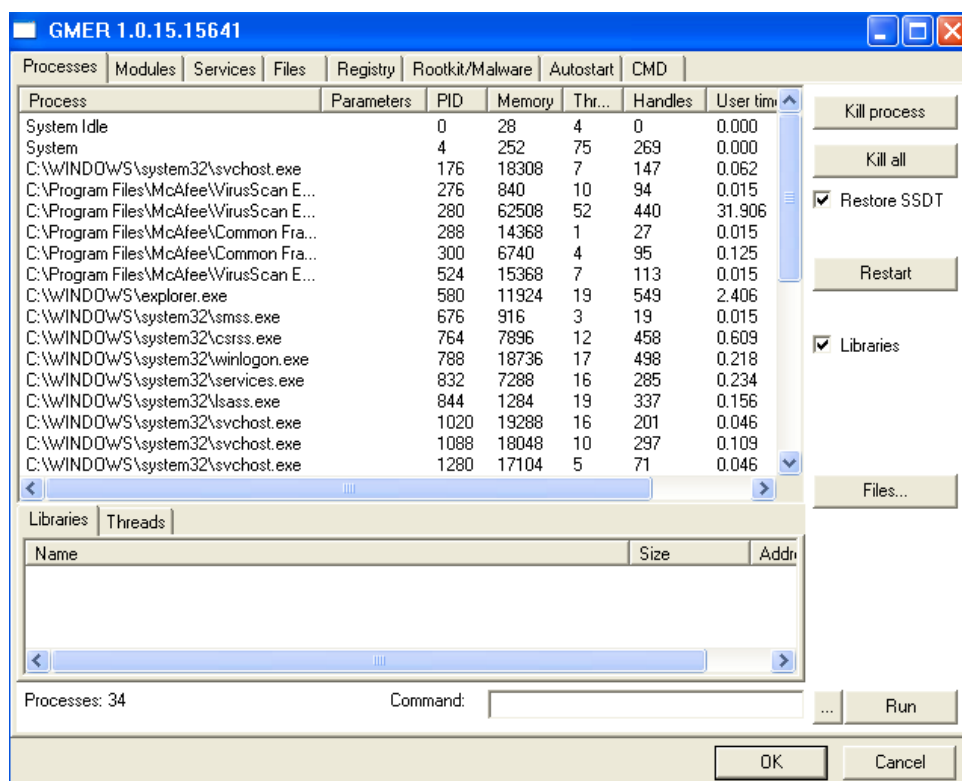
1. Download and install the Rootkit Tool from GMER website. www.gmer.net
2. Now the rootkit screen will be displayed



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



5. Click on the options



- This displays the Processes, Modules, Services, Files, Registry, RootKit/Malwares, Autostart, CMD of local host.
- Select Processes menu and kill any unwanted process if any. (Read the details listed, Malware affected processes, services if any will be shown in red. Before selecting the kill option, care must be taken, since the Rootkit tool's suspects may be systems core services.)

8. Modules menu displays the various system files like .sys, .dll
9. Services menu displays the complete services running with Autostart, Enable, Disable, System, Boot.
10. Files menu displays full files on Hard-Disk volumes.
11. Registry displays Hkey_Current_user and Hkey_Local_Machine.
12. Rootkits/Malawares scans the local drives selected.
13. Autostart displays the registry base Autostart applications.
14. CMD allows the user to interact with command line utilities or Registry

**SSN College of Engineering,
Department of Computer Science and Engineering
CS6711 Security Laboratory**

Exercise 12:

To demonstrate intrusion detection system (ids) using any snort.

Installing Snort in Windows:.

1. Double-click the WinPcap_4_1_3.exe installer file and follow the on-screen prompts. Typically no customization or configuration is required for this install, although on many systems a restart may be required to make sure the WinPcap netgroup packet filter (NPF) driver is running.
2. Double-click the Snort_2_9_8_2_Installer.exe file and follow the on-screen prompts.
 - a. Accept the license agreement
 - b. Choose the components (Snort, dynamic modules, documentation) you want to install. All are selected by default. Documentation is not strictly required for our purposes if space is at a premium (the space required to install is reduced by about 50% if documentation is unchecked).
 - c. By default the installer creates a root directory for Snort at c:\Snort, although you can specify a different directory if desired. When you select "Next" the installation executes.
 - d. At the end of the installation, the program displays a message that Snort has successfully been installed. The message includes a note that WinPcap is required (it refers to 4.1.1 although 4.1.3 is the current version), recommends tightening security on Snort, and directs you to edit the snort.conf file.
3. Open the Snort rules package. Depending on your operating system, Windows may be able to open the zipped archive automatically, or you can use a utility such as WinZip, 7Zip, or WinRAR to open it.
 - a. Create a subfolder under c:\Snort called rules, and another called preproc_rules.
 - b. Extract the contents of the rules folder in the archive to c:\Snort\rules
 - c. Extract the contents of the preproc_rules folder in the archive to c:\Snort\preproc_rules
 - d. Ignore the so_rules folder; while Sourcefire offers pre-compiled versions of the shared object rules for many Linux distributions, no such option exists for Windows. Compiling the Snort shared object rules to run on Windows is well beyond the technical scope of this course.
 - e. Also ignore the contents of the etc folder in the archive.

Once you have completed installing these components, you can check to see if the program responds:

Let's begin with retrieving files from www.snort.org. There are two things we want to download: the Snort installer package and the rules files.

1. Get the latest version of Snort by browsing to <https://www.snort.org/downloads> and clicking on the link for the Windows installer:
[Snort_2_9_8_2_Installer.exe](#)
2. Get the latest version of the rules by browsing to <https://www.snort.org/downloads/#rule-downloads> and clicking on the link for the current Registered User release: [snortrules-snapshot-2982.tar.gz](#)

Note that you must create an account (which is free) and log in to Snort.org in order to download the "registered" rules file or purchase an annual subscription to download the "subscriber" rules file. The "community" version of the rules is free and requires no user registration.

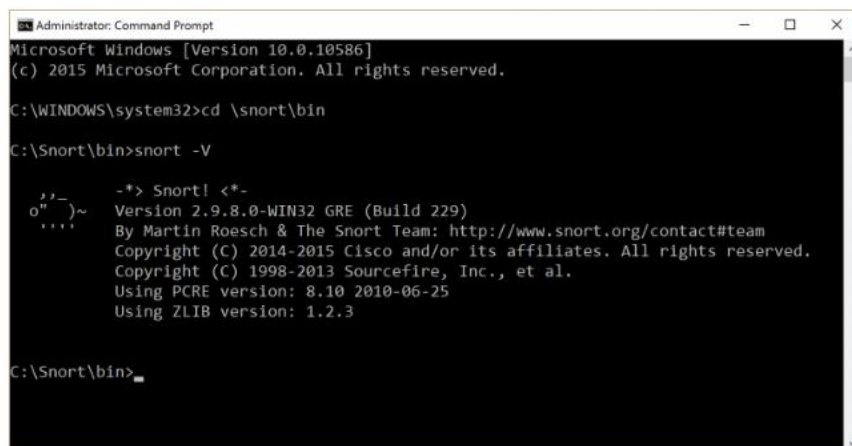
3. Get the WinPcap installer by browsing to <http://www.winpcap.org/install/default.htm> and clicking on the link for the Version 4.1.3 installer for windows (http://www.winpcap.org/install/bin/WinPcap_4_1_3.exe).

Now install the programs (in the case of WinPcap and Snort) and extract the rules files (in the case of the Snort rules package). It is recommended that WinPcap is installed before Snort, but it is not required; at the end of the Snort installation process the program will prompt that you need to install WinPcap, whether or not the utility is already installed. If you have installed any other programs that rely on packet capture, such as Wireshark, then you will already have WinPcap installed and you can skip the first step below.

1. Double-click the WinPcap_4_1_3.exe installer file and follow the on-screen prompts. Typically no customization or configuration is required for this install, although on many systems a restart may be required to make sure the WinPcap netgroup packet filter (NPF) driver is running.
2. Double-click the Snort_2_9_8_2_Installer.exe file and follow the on-screen prompts.
 - a. Accept the license agreement
 - b. Choose the components (Snort, dynamic modules, documentation) you want to install. All are selected by default. Documentation is not strictly required for our purposes if space is at a premium (the space required to install is reduced by about 50% if documentation is unchecked).
 - c. By default the installer creates a root directory for Snort at c:\Snort, although you can specify a different directory if desired. When you select "Next" the installation executes.
 - d. At the end of the installation, the program displays a message that Snort has successfully been installed. The message includes a note that WinPcap is required (it refers to 4.1.1 although 4.1.3 is the current version), recommends tightening security on Snort, and directs you to edit the snort.conf file.
3. Open the Snort rules package. Depending on your operating system, Windows may be able to open the zipped archive automatically, or you can use a utility such as WinZip, 7Zip, or WinRAR to open it.
 - a. Create a subfolder under c:\Snort called rules, and another called preproc_rules.
 - b. Extract the contents of the rules folder in the archive to c:\Snort\rules
 - c. Extract the contents of the preproc_rules folder in the archive to c:\Snort\preproc_rules
 - d. Ignore the so_rules folder; while Sourcefire offers pre-compiled versions of the shared object rules for many Linux distributions, no such option exists for Windows. Compiling the Snort shared object rules to run on Windows is well beyond the technical scope of this course.
 - e. Also ignore the contents of the etc folder in the archive.

Once you have completed installing these components, you can check to see if the program responds:

1. Change to the Snort program directory: `c:\>cd \snort\bin`
2. Check the installed version for Snort: `c:\snort\bin>snort -V`
3. The -V option (it must be a capital V) simply returns the current installed version of the program. If Snort is installed on the system, you should see something similar to the screenshot below (which shows an installed version 2.9.8.0):



```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd \snort\bin

C:\Snort\bin>snort -V

  _ _ _ _ _
 o"  _ _ _ _ _
  '    '

-*> Snort! <*-
Version 2.9.8.0-WIN32 GRE (Build 229)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.3

C:\Snort\bin>

```

4. You should also check to see what network adapters are on your system, so you can tell Snort to listen on the appropriate interface when it runs. To see a list of interfaces, run the command: `c:\snort\bin>snort -W`

On current Windows systems there will be at least two (Ethernet and wireless), three if there is a modem in the computer, and four or more depending on what additional software is installed on the computer. If both wired and wireless network interfaces are active, you should disable one before you try to run Snort, since Windows offers no way to direct a program to use a specific interface when multiple connections are available. Record the number of the interface you will use (the instructions below assume the interface number is 2; substitute the appropriate number for your computer when using the -i option in Snort start-up commands).

The next thing to do is to edit the snort.conf file to make it reflect the environment where your computer is running (see [Configuring Snort with snort.conf](#)). You should make sure that when you edit the file, you are working on the one in **c:\Snort\etc** (and not any other versions that may exist in temporary or download directories).

CONFIGURING SNORT

Getting Snort installed successfully can be a challenge, but it is also only the first step in setting the tool up so you can launch it to start monitoring traffic and generating alerts. To get Snort ready to run, you need to change the default configuration settings file (which is created as part of the Snort installation) to match your local environment and operational preferences. If you accepted the default locations proposed during the Windows installer execution, then the `snort.conf` file will be located in the directory `C:\Snort\etc`. The configuration file is plain text, so you can use any text editor to edit it, but Wordpad (or even better, the free [Notepad++](#)) is recommended at least for the first time to ensure the proper formatting is maintained (when opening the baseline `snort.conf` file in Notepad all the text runs together).

When you open the file for viewing or editing, you will see it is organized into nine parts or steps:

1. Set the network variables
2. Configure the decoder
3. Configure the base detection engine
4. Configure dynamic loaded libraries
5. Configure preprocessors
6. Configure output plugins
7. Customize your rule set
8. Customize preprocessor and decoder rule set
9. Customize shared object rule set

As you can see, there are a lot of ways to customize Snort, and making sense of the entire `snort.conf` file can be a little daunting. To get running for the first time, many of the defaults can be left alone. The following edits are recommended:

1. Step 1

- a. Change the declaration for **HOME_NET** to your actual home network IP address range, rather than leaving the default “any”. The simplest way to do this is to use a CIDR format expression, to cover the entire range of relevant addresses (particularly when using Network Address Translation such as in environments protected by gateways or routers).
 - i. For a typical home network, the expression will be `192.168.0.1/24` or `192.168.1.1/24` (if you’re not sure whether your third number is a 0 or 1, check your gateway/router documentation or just ping it. If you want to cover all IP addresses beginning with 192.168, then use the expression `192.168.0.0/16`
 - ii. In a typical large office network using network address translation, the expression will be `10.0.0.0/8`
 - iii. In some environments (including home environments connecting to the Internet via cable modem without the use of a gateway or router) the appropriate IP address range to use may be dictated by the ISP from which you get your Internet service.
 - iv. If you are unsure which IP address range to specify for your home network, you can quickly check to see the IP address assigned to your computer by opening a command shell window and typing `ipconfig` at the prompt.
 - v. Finally, you can leave the **HOME_NET** declaration as “any” if you are unable to accurately determine a specific IP range to use.
- b. Change the declaration for **EXTERNAL_NET** to `!$HOME_NET` – this expression means the external network will be defined as any IP address that is not part of the home network. **Important!** If you leave **HOME_NET** declared as “any” you **cannot** use `!$HOME_NET`, as the expression will translate to “not any” and throw an error when you try to start Snort.
- c. Generally speaking, you can leave unchanged all the other server declarations, although if you want you can reduce the list of web server ports declared for **HTTP_PORTS**.
- d. Change the var **RULE_PATH** declaration to match the actual location of your rules files. Typically the rules will be stored in `c:\Snort\rules`, so you can use that full path name or whatever the right location is on your system.
- e. Similarly, change the **PREPROC_RULE_PATH** to match the appropriate directory location on your system, such as `c:\Snort\preproc_rules`.
- f. Comment out (meaning put a # character in the first position in the line) the **SO_RULE_PATH** declaration, as the Windows implementation of Snort doesn’t use shared object rules.

g. The reputation preprocessor is a relatively recent addition to Snort that allows you to configure trusted or untrusted IP addresses using separately referenced files that list the addresses (whitelist for trusted, blacklist for untrusted). If you intend to enable the reputation preprocessor then the path to the whitelist and blacklist files needs to be provided at the end of step 1. **Please note:** if you leave the reputation preprocessor enabled, you *must* create the whitelist and blacklist rules files referenced in the preprocessor configuration, or Snort will generate an error and fail to start. If you want to work with the reputation preprocessor later, be sure to comment it out in step 5.

2. Step 2

- a. For most users, there are no changes needed to the decoder configurations.
- b. At the end of this section, there is a configuration setting to indicate the default directory where Snort logs should be written. Uncomment this line by deleting the # character in the first position and edit the line to include the **c:\Snort\log** default directory path.

3. Step 3

- a. For most users, there are no changes needed to the base detection engine settings, so move on to step 4. These settings are used for performance tuning and reflect memory and processing capabilities.

4. Step 4

- a. Change the dynamic loaded library path references to reflect their location in Windows, and in the case of the dynamic engine to replace the default Linux filename with the Windows equivalent. Snort references these locations and loads the libraries at start-up.
 - i. **dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor**
 - ii. **dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll**
 - iii. Comment out (put a # in the first position in the line) the **dynamicdetection directory** declaration.
- b. Note that the dynamic engine is actually pointing to a file, while the other two declarations point to directories. It's always a good idea to double-check the accuracy of these locations by browsing to them with the file browser or performing directory listings from the command line. Be sure there is no trailing character on the dynamic preprocessor directory.
- c. One point to be aware of when configuration is done and you move on to running Snort: loading the dynamic libraries requires Snort to write to the Windows registry, an action typically requiring administrator privileges. For this reason the command shell should be launched with the "Run as administrator" option from the Windows start menu when preparing to start Snort.

5. Step 5

- a. Be aware that there are many, many preprocessors for use with Snort, and you very likely will not want or need to have all of them running. Each preprocessor has a separate readme file with configuration options and settings documented in it, so if you want to use a particular preprocessor, you should consult those files or the Snort manual to make sure you set them up properly.
- b. Comment out (put a # in the first position on the line) all the rows in the Inline packet normalization preprocessor. This preprocessor is only used when Snort is implemented in in-line IPS mode, and Snort should ignore it otherwise, but on Windows it will cause an error if left uncommented.
- c. For general-purpose Snort usage, it usually makes sense to disable (comment out) some of the preprocessors, particularly ones like those for normalization listed first in Step 5 that only apply to Snort in in-line mode. Of the others, it is fine to leave default preprocessors active, but at a minimum it is a good idea to keep at least the following preprocessors active (using default configuration settings):
 - i. **frag3**
 - ii. **stream5**
 - iii. **http_inspect**
 - iv. **ftp_telnet**
 - v. **smtp**
 - vi. **dns**
 - vii. **ssl**
 - viii. **sensitive_data**
- d. The most recent releases of Snort include some very interesting new preprocessors, some of which are not included in snort.conf by default. You can learn more about these preprocessors and the configuration syntax used to add them to the file in Step 5 by consulting the Snort documentation or the "readme" file for each preprocessor.
- e. As noted in Step #1 above, if you choose to keep the **reputation preprocessor** enabled you must create whitelist and blacklist files corresponding to the references in the configuration settings for the reputation preprocessor, which is at the very end of Step #5. You can opt to comment it out for initial setup and come back to it later. Snort by default includes a set of rules in a file called "blacklist.rules" that is *not* used by the reputation preprocessor. For this reason it is strongly recommended to avoid later confusion that you choose names for the whitelist and blacklist files that do not include "rules" in the names (for example, "white.list" and "black.list").

6. Step 6

a. Typically, only one of the output plugins is used with Snort at any one time. The default in recent releases of Snort is unified2, but as noted above this is not well supported on Windows platforms. If you intend to use syslog, then uncomment that line to activate the syslog output plugin. If you intend to use screen output only, leave all the output plugins commented out.

i. Uncomment and edit the syslog output line in snort.conf, so it reads like this:

```
output alert_syslog: host=127.0.0.1:514, LOG_AUTH LOG_ALERT
```

ii. **Note:** If you choose to use syslog output, then you also need to install and run a syslog server; see [Installing a Syslog Server](#).

b. If you have used previous versions of Snort, you may notice that there are no database output configuration options in the snort.conf file. As of the 2.9.3 version of Snort direct logging to database is no longer supported.

c. Leave the metadata reference lines at the end of step 6 **uncommented**: `include classification.config` and `include reference.config`

7. Step 7

a. If you have installed the Snort VRT ruleset, then you can tailor the series of include statements in step 7 to match whatever environment characteristics and types of rules you want. For initial testing, sometimes it can be helpful to reduce the number of rules loaded at start-up, but make sure that the line for **"local.rules"** remains uncommented, as that is where you will place the rules that you write yourself.

b. For first-time users, you may want to comment out most of the include statements listed in step 7 until you verify your configuration.

c. If you create your own rules in separate rules files (instead of adding them to local.rules), add an include statement for your custom files following the same syntax you see for all the other statements in step 7.

8. Step 8

a. There are not very many settings in step 8, so in general you just want to make sure that you uncomment any rules here that correspond to preprocessors you configured to load in step 5. By default, if you kept the standard settings in step 2 and enabled at least some preprocessors, the uncomment the first two lines in step 8

i. `include $PREPROC_RULE_PATH\preprocessor.rules`

ii. `include $PREPROC_RULE_PATH\decoder.rules`

b. If you enabled the sensitive_data preprocessor (in step 5), then uncomment the third line in step 8: `include $PREPROC_RULE_PATH\sensitive-data.rules`

c. Make sure the rules you declare in these statements are actually present in the appropriate directory (such as **c:\Snort\preproc_rules**)

9. Step 9

a. The rules referenced in Step #9 are shared object rules, which are different from (although similarly named) the rules listed in Step #7. Because shared object rules are not well supported on Windows, leave all the shared object rules commented out in step 9.

b. Leave the event thresholding line at the end of step 9 **un-commented**: `include threshold.conf`

GENERATING ALERTS

To see if Snort is working, beyond just getting it to load without errors (not a trivial feat in itself), it is helpful to generate some alerts. The easiest way to do this to validate setup and configuration is to create a couple of testing rules, load them in Snort, and trigger them so you can check to see if they generate alerts as expected. Put your testing rules in the **local.rules** file that is located in the **c:\Snort\rules** directory.

1. Open **local.rules** with a text editor such as Notepad++ or Wordpad.
2. Move down beyond the commented header information to the first blank line. Start with some generic rules to test network traffic detection. Enter the following, all on one line: `alert icmp any any -> any any (msg:"ICMP Testing Rule"; sid:1000001; rev:1;)`
3. Press Enter to move to a new line, and create another rule to check TCP traffic detection: `alert tcp any any -> any 80 (msg:"TCP Testing Rule"; sid:1000002; rev:1;)`
4. Press Enter to move to a new line, and create another rule to check UDP traffic detection: `alert udp any any -> any any (msg:"UDP Testing Rule"; sid:1000003; rev:1;)`
5. You can create any number of additional rules you like; just be sure to start each one on a new line.
6. Save the file and exit the editor. **Note:** If you use Notepad, it is important to save the file as type "All Files" rather than the text documents default. The default will add ".txt" to the rule file name (so it will become local.rules.txt) and Snort will generate an error when it tries to load the file.

If you load these rules by starting Snort with the **-A console** option, when you test the rules by performing the steps listed below, you can see the output on the screen as it happens. Note that the startup command shown below uses interface #2, which is often the correct choice, but many systems have multiple network interfaces so it is a good idea to determine which one you want Snort to monitor by running the command **snort -w** to see the available interfaces.

1. Open a command shell by locating Command Prompt in the Accessories of the Windows start menu.
2. Right-click on Command Prompt and select "Run as administrator"
3. Navigate to the directory where Snort is installed: `c:\Windows\system32> cd \Snort\bin`
4. Start Snort: `c:\Snort\bin> snort -i 2 -c c:\Snort\etc\snort.conf -A console`
5. Open *another* Command Prompt window, leaving Snort running in the first (you do not need to run the second one as administrator).
6. Send a ping command to your local gateway (or any other host): `c:\> ping 192.168.1.1`
7. Open a web browser and browse to any web page.
8. You should see the alerts Snort produces in the first terminal shell where Snort is running.

Ordinarily, you won't need to do anything special to generate UDP alerts, because the operating system already generates plenty of UDP activity (such as ARP requests and responses or SSDP traffic) when it is connected to a network. If you are running standalone and don't see any UDP alerts, you can open a browser and enter a URL in the address bar; DNS lookups typically use UDP by default.