**Techno India NJR Institute of Technology**

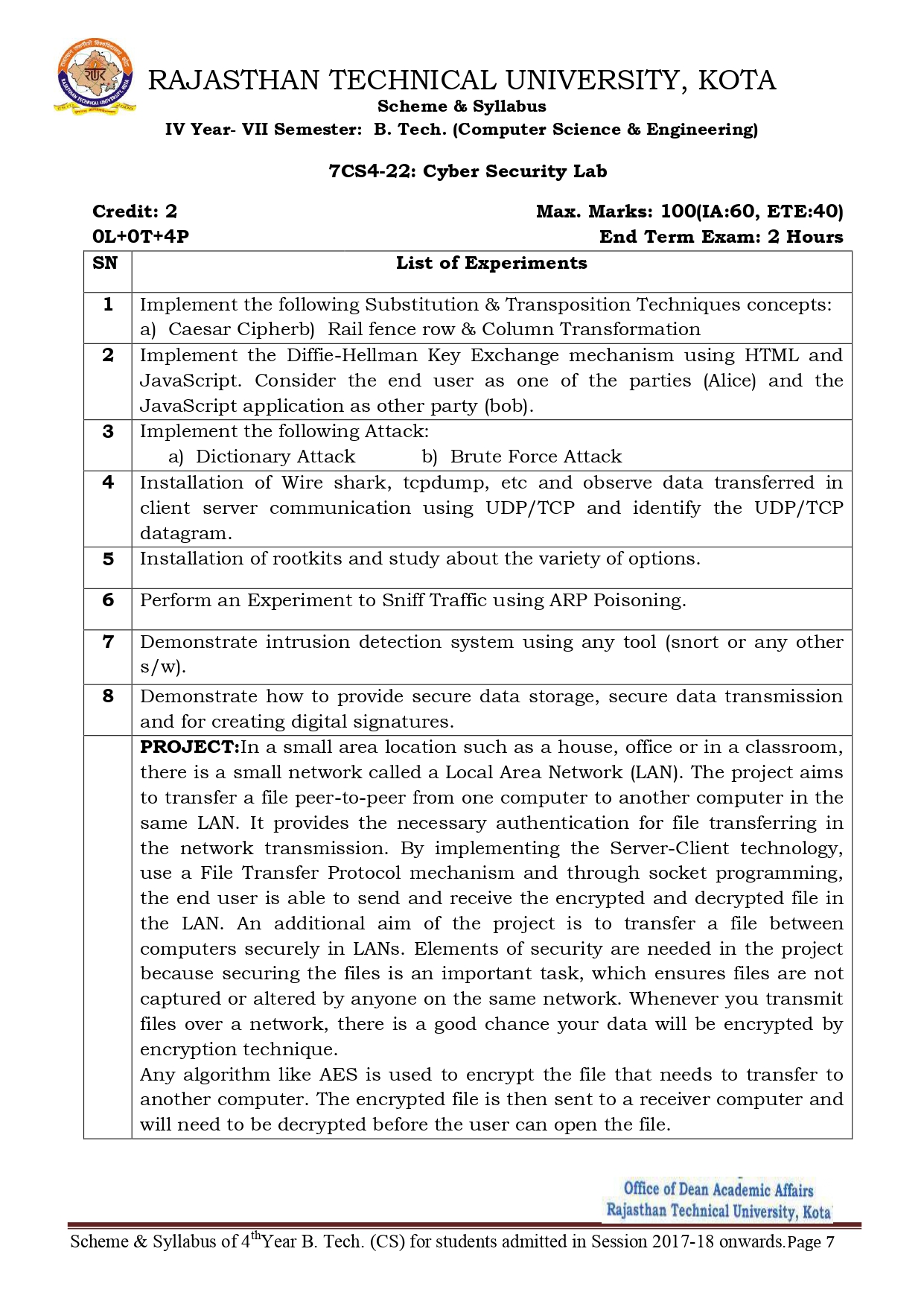


**Lab Manual**

**Cybersecurity Lab (7CS4-22)**

Prof P Chakrabarti

**Department of CSE**

****

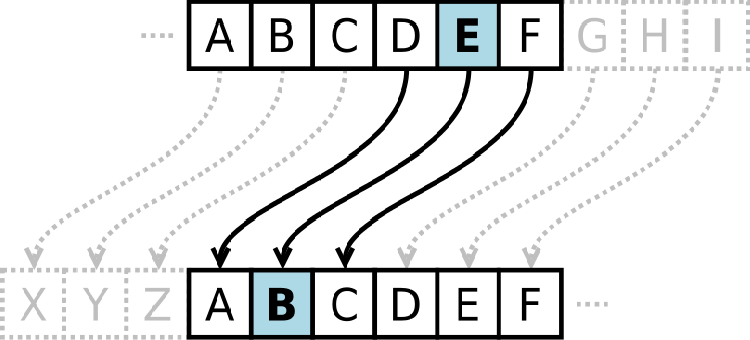
**Course Overview:** The course has certain outcomes by virtue of which the students will get an idea of the subject Cybersecurity.

|  |  |  |
| --- | --- | --- |
| **CO No** | **Cognitive Level** | **Course Outcome (LAB)** |
| 1 | Comprehension | Student will be able to work on substitution and transportation techniques |
| 2 | Application | Student will be able to design the knowledge of key exchange protocols and also analyze dictionary and brute force attacks |
| 3 | Application | Student will be able to install wire shark and rootkits. |
| 4 | Application | Student will be able to work and perform different operations on ARP poisoning. |
| 5 | Application | Student will be able to demonstrate intrusion detection system and secured data transmission using digital signature. |

# Course Outcome Mapping with Program Outcome (LAB):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Outcome** | **Program Outcome (LAB)** | | | | | | | | | | | | | | |
| CO No. | **Domain Specific** | | | | | **Domain Independent** | | | | | | | **PSO** | | |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO 1** | 2 | 1 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 2 | 0 |
| **CO 2** | 2 | 1 | 1 | 2 | 3 | 0 | 0 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 |
| **CO 3** | 2 | 2 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 0 |
| **CO 4** | 2 | 0 | 1 | 2 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 3 | 2 | 0 |
| **CO 5** | 2 | 0 | 1 | 2 | 3 | 0 | 0 | 0 | 2 | 0 | 2 | 2 | 3 | 2 | 0 |
| **1: Slight (Low), 2: Moderate (Medium), 3: Substantial (high)** | | | | | | | | | | | | | | | |

**EX. NO: 1(A)**



**AIM:**

To implement the simple substitution technique named Caesar cipher using C language.

**DESCRIPTION:**

To encrypt a message with a Caesar cipher, each letter in the message is changed using a simple rule: shift by three. Each letter is replaced by the letter three letters ahead in the alphabet. A becomes D, B becomes E, and so on. For the last letters, we can think of the

alphabet as a circle and "wrap around". W becomes Z, X becomes A, Y bec mes B, and Z

o

becomes C. To change a message back, each letter is replaced by the one three before it.

**EXAMPLE:**

**ALGORITHM:**

**STEP-1:** Read the plain text from the user.

**STEP-2:** Read the key value from the user.

**STEP-3:** If the key is positive then encrypt the text by adding the key with each character in the plain text.

**STEP-4:** Else subtract the key from the plain text.

**STEP-5:** Display the cipher text obtained above.

**PROGRAM: (Caesar Cipher)**

**#include <stdio.h> #include <string.h> #include<conio.h> #include <ctype.h> void main()**

**{**

**char plain[10], cipher[10]; int key,i,length;**

**int result; clrscr();**

**printf("\n Enter the plain text:"); scanf("%s", plain);**

**printf("\n Enter the key value:"); scanf("%d", &key);**

**printf("\n \n \t PLAIN TEXt: %s",plain); printf("\n \n \t ENCRYPTED TEXT: ");**

**for(i = 0, length = strlen(plain); i < length; i++)**

**{**

**cipher[i]=plain[i] + key;**

**if (isupper(plain[i]) && (cipher[i] > 'Z')) cipher[i] = cipher[i] - 26;**

**if (islower(plain[i]) && (cipher[i] > 'z')) cipher[i] = cipher[i] - 26;**

**printf("%c", cipher[i]);**

**}**

**printf("\n \n \t AFTER DECRYPTION : "); for(i=0;i<length;i++)**

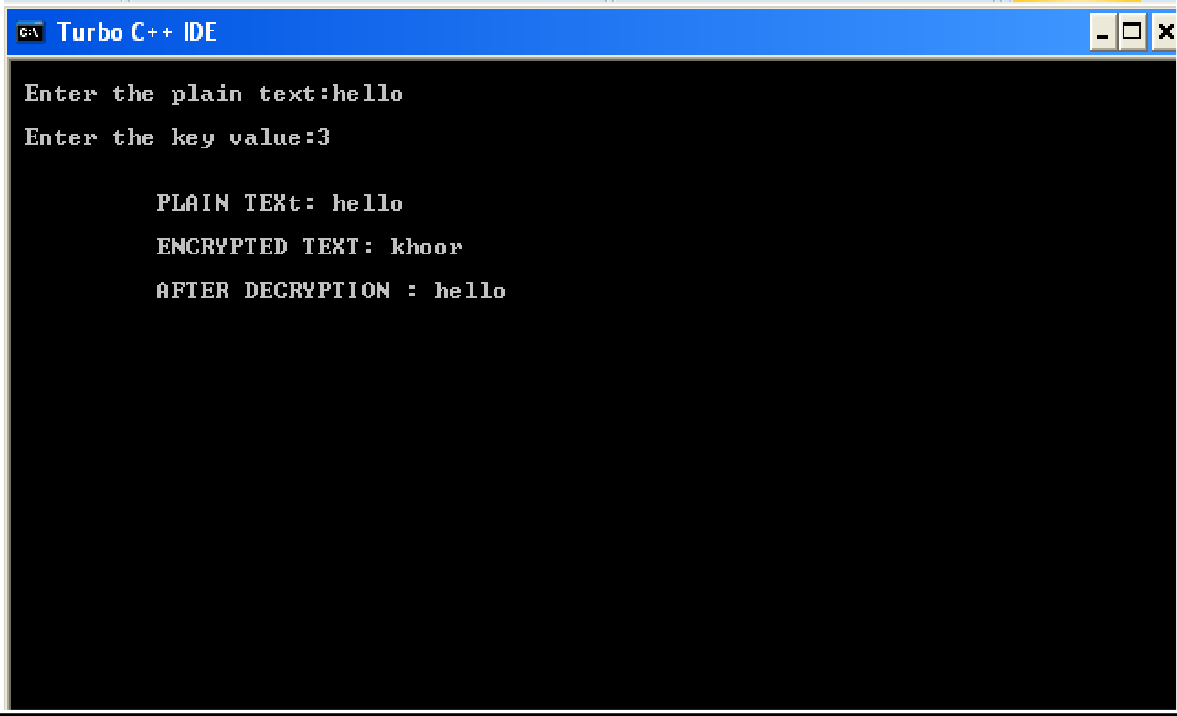
**{**

**plain[i]=cipher[i]-key; if(isupper(cipher[i])&&(plain[i]<'A')) plain[i]=plain[i]+26; if(islower(cipher[i])&&(plain[i]<'a')) plain[i]=plain[i]+26; printf("%c",plain[i]);**

**}**

**getch();**

**OUTPUT:**



**EX. NO: 1(B)**

## **IMPLEMENTATION OF RAIL FENCE – ROW & COLUMN**

**TRANSFORMATION TECHNIQUE**

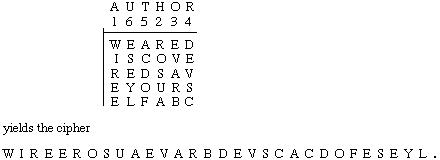
**AIM:**

To write a C program to implement the rail fence transposition technique.

**DESCRIPTION:**

In the rail fence cipher, the plain text is written downwards and diagonally on successive "rails" of an imaginary fence, then moving up when we reach the bottom rail. 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.

**EXAMPLE:**



**ALGORITHM:**

**STEP-1:** Read the Plain text.

**STEP-2:** Arrange the plain text in row columnar matrix format.

**STEP-3:** Now read the keyword depending on the number of columns of the plain text.

**STEP-4:** Arrange the characters of the keyword in sorted order and the corresponding columns of the plain text.

**STEP-5:** Read the characters row wise or column wise in the former order to get the cipher text.

**PROGRAM: (Rail Fence)**

**#include<stdio.h> #include<conio.h> #include<string.h> void main()**

**{**

**int i,j,k,l;**

**char a[20],c[20],d[20]; clrscr();**

**printf("\n\t\t RAIL FENCE TECHNIQUE"); printf("\n\nEnter the input string : "); gets(a);**

**l=strlen(a);**

**/\*Ciphering\*/ for(i=0,j=0;i<l;i++)**

**{**

**if(i%2==0) c[j++]=a[i];**

**}**

**for(i=0;i<l;i++)**

**{**

**if(i%2==1) c[j++]=a[i];**

**}**

**c[j]='\0';**

**printf("\nCipher text after applying rail fence :"); printf("\n%s",c);**

**/\*Deciphering\*/ if(l%2==0)**

**k=l/2;**

**else**

**k=(l/2)+1;**

**for(i=0,j=0;i<k;i++)**

**{**

**d[j]=c[i]; j=j+2;**

**}**

**for(i=k,j=1;i<l;i++)**

**{**

**d[j]=c[i]; j=j+2;**

**}**

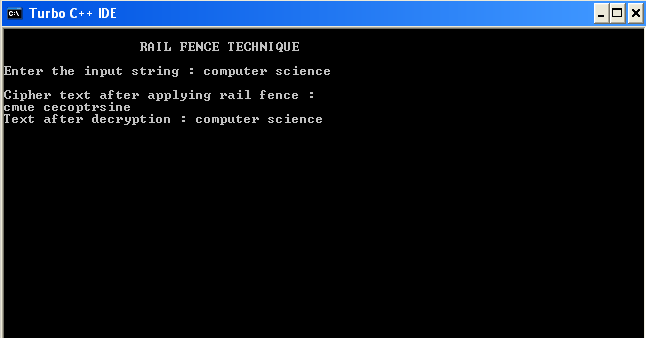
**d[l]='\0';**

**printf("\nText after decryption : "); printf("%s",d);**

**getch();**

**}**

## OUTPUT:



**EX. NO: 2**

**IMPLEMENTATION OF DIFFIE HELLMAN KEY EXCHANGE ALGORITHM**

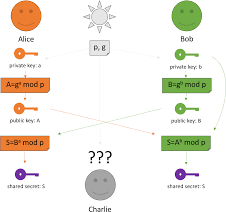
**AIM:**

To implement the Diffie-Hellman Key Exchange algorithm using C language.

**DESCRIPTION:**

Diffie–Hellman Key Exchange establishes a shared secret between two parties that can be used for secret communication for exchanging data over a public network. It is primarily used as a method of exchanging cryptography keys for use in symmetric encryption algorithms like AES. The algorithm in itself is very simple. The process begins by having the two parties, Alice and Bob. Let's assume that Alice wants to establish a shared secret with Bob.

**EXAMPLE:**



**ALGORITHM:**

**STEP-1:** Both Alice and Bob shares the same public keys g and p.

**STEP-2:** Alice selects a random public key a.

**STEP-3:** Alice computes his secret key A as ga mod p.

**STEP-4:** Then Alice sends A to Bob.

**STEP-5:** Similarly Bob also selects a public key b and computes his secret key as B and sends the same back to Alice.

**STEP-6:** Now both of them compute their common secret key as the other one’s secret key power of a mod p.

**PROGRAM: (Diffie Hellman Key Exchange)**

**#include<stdio.h> #include<conio.h>**

**long long int power(int a, int b, int mod)**

**{**

**long long int t; if(b==1)**

**return a; t=power(a,b/2,mod); if(b%2==0)**

**return (t\*t)%mod; else**

**return (((t\*t)%mod)\*a)%mod;**

**}**

**long int calculateKey(int a, int x, int n)**

**{**

**return power(a,x,n);**

**}**

**void main()**

**{**

**int n,g,x,a,y,b; clrscr();**

**printf("Enter the value of n and g : "); scanf("%d%d",&n,&g);**

**printf("Enter the value of x for the first person : "); scanf("%d",&x);**

**a=power(g,x,n);**

**printf("Enter the value of y for the second person : "); scanf("%d",&y);**

**b=power(g,y,n);**

**printf("key for the first person is :**

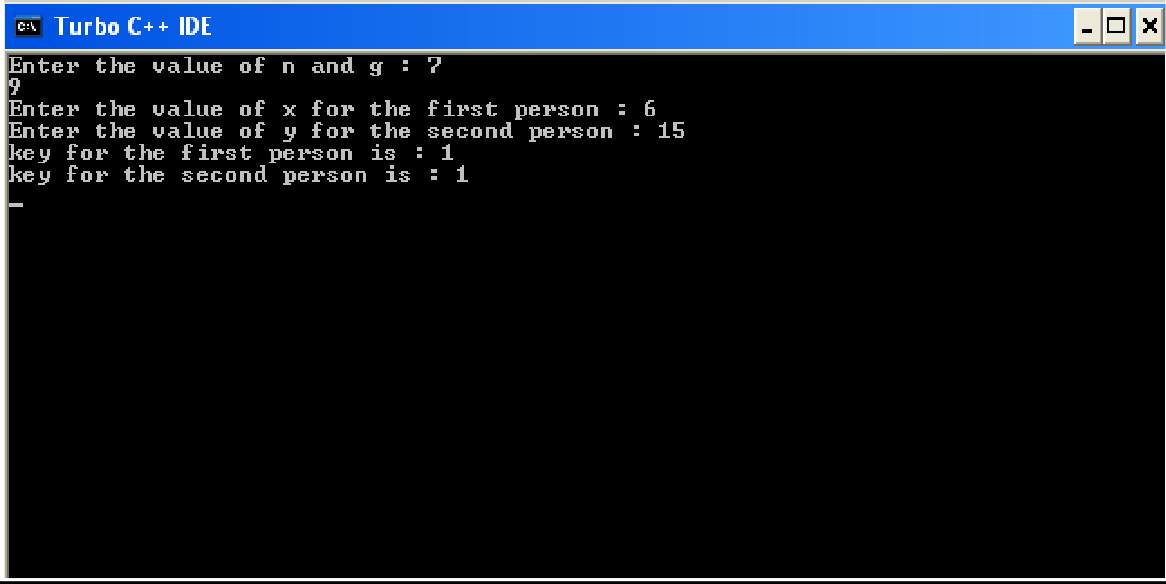
**%lld\n",power(b,x,n));**

**printf("key for the second person is :**

**%lld\n",power(a,y,n)); getch();**

**}**

**OUTPUT:**



**EX 3 BRUTE FORCE ATTACK USING C**

/\*  
 \* Basic string generation for brute-force attacks  
  
 \*/  
  
#include <string.h>  
#include <stdio.h>  
#include <stdlib.h>  
  
  
typedef struct charlist charlist\_t;  
struct charlist  
{  
 unsigned char character;  
 charlist\_t\* next;  
};  
  
/\* Return new initialized charlist\_t element.  
 \*  
 \* Elements are initialized  
 \* @return charlist\_t  
 \*/  
charlist\_t\* new\_charlist\_element()  
{  
 charlist\_t\* element;  
  
 if ((element = malloc(sizeof(charlist\_t))) != 0)  
 {  
 element->character = 0;  
 element->next = NULL;  
 }  
 else  
 {  
 perror("malloc() failed.");  
 }  
  
 return element;  
}  
  
/\* Free memory allocated by charlist.  
 \*  
 \* @param list Pointer at the first element.  
 \* @return void  
 \*/  
void free\_charlist(charlist\_t\* list)  
{  
 charlist\_t\* current = list;  
 charlist\_t\* next;  
  
 while (current != NULL)  
 {  
 next = current->next;  
 free(current);  
 current = next;  
 }  
}  
  
/\* Print the charlist\_t data structure.  
 \*  
 \* Iterates through the whole list and prints all characters  
 \* in the list including any '\0'.  
 \*  
 \* @param list Input list of characters.  
 \* @return void  
 \*/  
void print\_charlist(charlist\_t\* list)  
{  
 charlist\_t\* next = list;  
 while (next != NULL)  
 {  
 printf("%d ", next->character);  
 next = next->next;  
 }  
 printf("\n");  
}  
  
/\* Get next character sequence.  
 \*  
 \* It treats characters as numbers (0-255). Function tries to  
 \* increment character in the first position. If it fails,  
 \* new character is added to the back of the list.  
 \*  
 \* It's basicaly a number with base = 256.  
 \*  
 \* @param list A pointer to charlist\_t.  
 \* @return void  
 \*/  
void next(charlist\_t\* list)  
{  
 list->character++;  
 if (list->character == 0)  
 {  
 if (list->next == NULL)  
 {  
 list->next = new\_charlist\_element();  
 }  
 else  
 {  
 next(list->next);  
 }  
 }  
}  
  
int main()  
{  
 charlist\_t\* sequence;  
 sequence = new\_charlist\_element();  
  
 while (1)  
 {  
 next(sequence);  
 print\_charlist(sequence);  
 }  
  
 free\_charlist(sequence);  
}

**EX 4 INSTALLATION OF WIRESHARK**

Wireshark is **a network protocol analyzer**, or an application that captures packets from a network connection, such as from your computer to your home office or the internet. Packet is the name given to a discrete unit of data in a typical Ethernet network. Wireshark is the most often-used packet sniffer in the world.

Wireshark is the most often-used packet sniffer in the world. Like any other packet sniffer, Wireshark does three things:

1. **Packet Capture:** Wireshark listens to a network connection in real time and then grabs entire streams of traffic – quite possibly tens of thousands of packets at a time.
2. **Filtering:** Wireshark is capable of slicing and dicing all of this random live data using filters. By applying a filter, you can obtain just the information you need to see.
3. **Visualization:** Wireshark, like any good packet sniffer, allows you to dive right into the very middle of a network packet. It also allows you to visualize entire conversations and network streams.

### **How to Install Wireshark on Windows**

If you’re a Windows operating system user, download the version appropriate for your particular version. If you use Windows 10, for example, you’d grab the 64-bit Windows installer and follow the wizard to install. To install, you’ll need administrator permissions.

### **How to Install Wireshark on Linux**

If you have a [Linux system](https://www.comptia.org/blog/all-about-linux-and-linux), you’d install Wireshark using the following sequence (notice that you’ll need to have root permissions):

$ sudo apt-get install wireshark

$ sudo dpkg-reconfigure wireshark-common

$ sudo usermod -a -G wireshark $USER

$ newgrp wireshark

Once you have completed the above steps, you then log out and log back in, and then start Wireshark:

$ wireshark &

### **How to Capture Packets Using Wireshark**

Once you’ve installed Wireshark, you can start grabbing network traffic. But remember: To capture any packets, you need to have proper permissions on your computer to put Wireshark into promiscuous mode.

* + In a Windows system, this usually means you have administrator access.
  + In a Linux system, it usually means that you have root access.

As long as you have the right permissions, you have several options to actually start the capture. Perhaps the best is to select Capture >> Options from the main window. This will bring up the Capture Interfaces window,

This window will list all available interfaces. In this case, Wireshark provides several to choose from.

For this example, we’ll select the Ethernet 3 interface, which is the most active interface. Wireshark visualizes the traffic by showing a moving line, which represents the packets on the network.

Once the network interface is selected, you simply click the Start button to begin your capture. As the capture begins, it’s possible to view the packets that appear on the screen

Once you have captured all the packets that you want, simply click the red, square button at the top. Now you have a static packet capture to investigate.

**EX. NO: 05**

# INSTALLATION OF ROOTKITS

**AIM:**

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

**INTRODUCTION:**

Breaking the term rootkit into the two component words, root and kit, is a useful way to define it. Root is a UNIX/Linux term that's the equivalent ofAdministrator in Windows. The word kit denotes programs that allow someone to obtain root/admin-level access to the computer by executing the programs in the kit — all of which is done without end-user consent or knowledge.

A rootkit is a type of malicious software that is activated each time your system boots up. Rootkits are difficult to detect because they are activated before your system's Operating System has completely booted up. A rootkit often allows the installation of hidden files, processes, hidden user accounts, and more in the systems OS. Rootkits are able to intercept data from terminals,network connections, and the keyboard.

Rootkits have two primary functions: remote command/control (back door) and software eavesdropping. Rootkits allow someone, legitimate or otherwise, to administratively control a computer. This means executing files, accessing logs, monitoring user activity, and even changing the computer's configuration. Therefore, in the strictest sense, even versions of VNC are rootkits. This surprises most people, as they consider rootkits to be solely malware, but in of themselves they aren't malicious at all

The presence of a rootkit on a network was first documented in the early 1990s. At that time, Sun and Linux operating systems were the primary targets for a hacker looking to install a rootkit. Today, rootkits are available for a number of operating systems, including Windows, and are increasingly difficult to detect on any network.

**PROCEDURE:**

**STEP-1:** Download Rootkit Tool from GMER website [www.gmer.net.](http://www.gmer.net/)

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

**STEP-3:** Select Processes menu and kill any unwanted process if any.

**STEP-4:** Modules menu displays the various system files like .sys, .dll

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

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

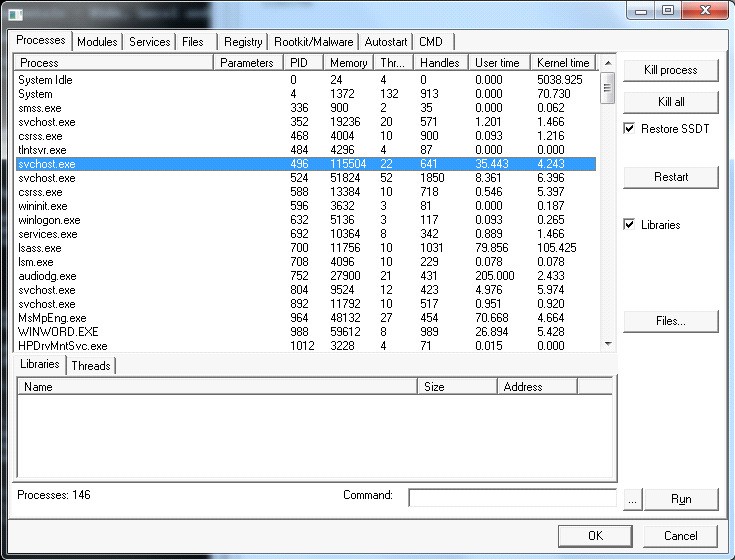
**STEP-7:** Registry displays Hkey\_Current\_user and Hkey\_Local\_Machine.

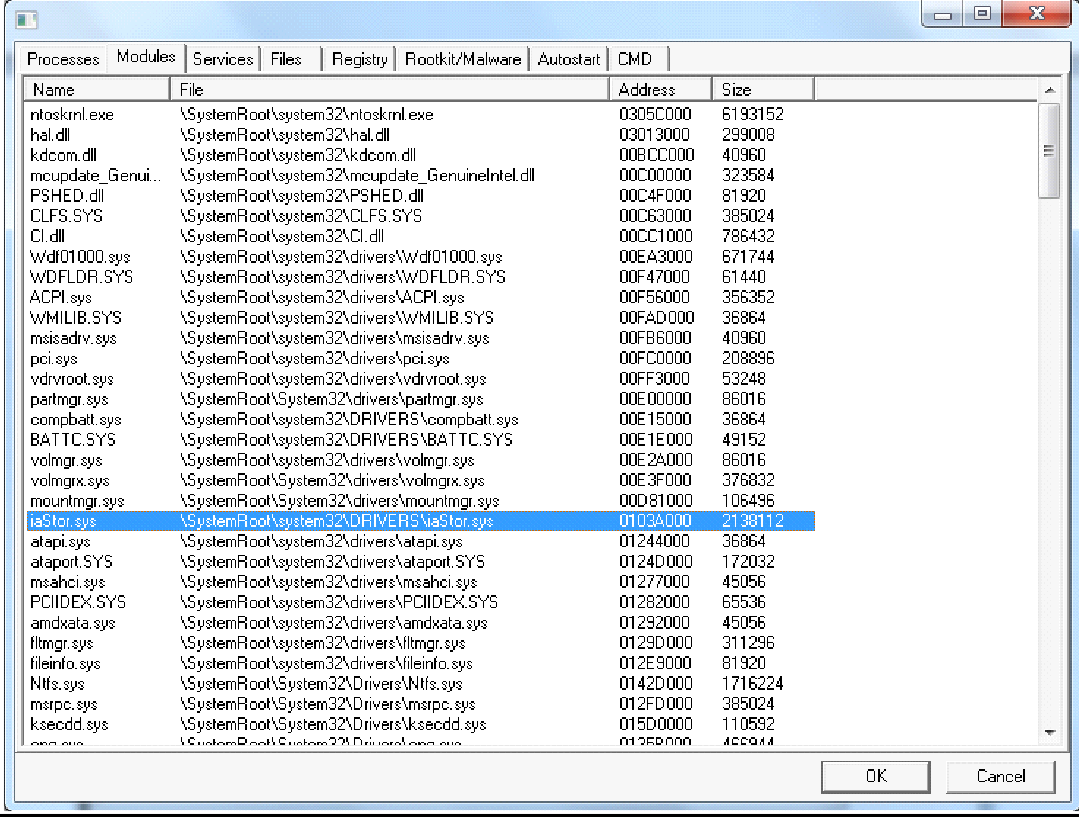
**STEP-8:** Rootkits / Malwares scans the local drives selected.

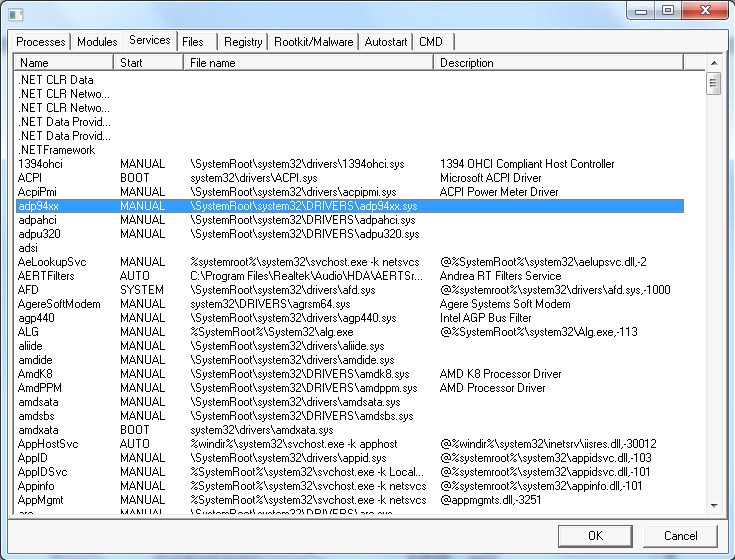
**STEP-9:** Autostart displays the registry base Autostart applications.

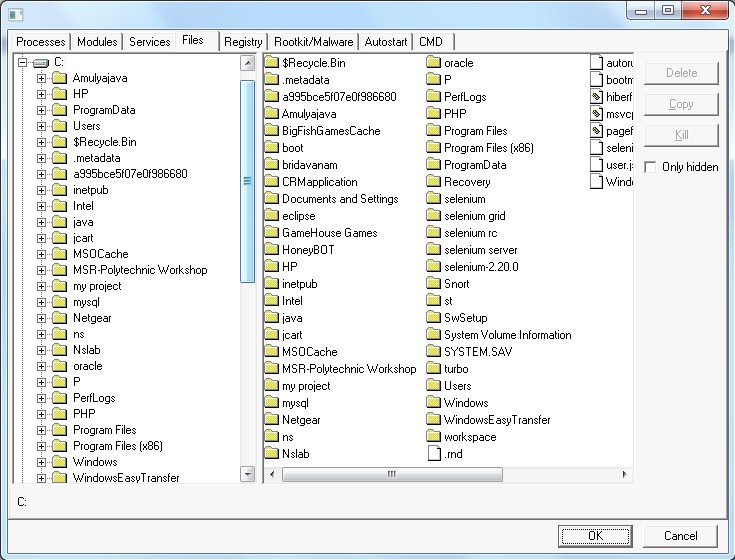
**STEP-10:**CMD allows the user to interact with command line utilities or Registry

**SCREENSHOTS:**



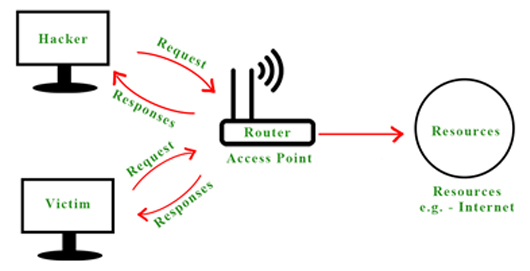






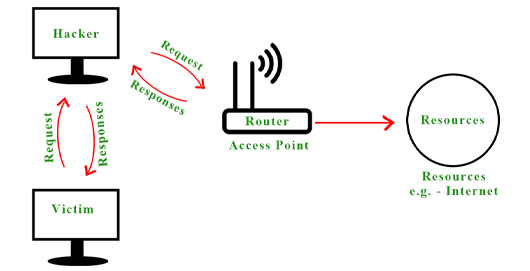
**EX6- ARP POISONING**

**Step-1:**  
**ARP spoofing** -It allows us to redirect the flow of packets in a computer network. Example of a typical Network as follows.



A Typical Computer Network

**Step-2 :**  
But when a hacker becomes Man-In-The-Middle by ARP Spoofing then all the requests and responses start flowing through the hacker’s system as shown below –



computer network after ARP spoofing

**Step-3 :**  
By doing this a hacker spoof’s the router by pretending to be the victim, and similarly, he spoofs the victim by pretending to be the router.

**How to do an ARP Spoof Attack :**  
We can do an ARP Spoof attack using the built-in tool called ARPSPOOF in Kali Linux, or we can also create an ARP Spoof attack using a [python program.](https://www.geeksforgeeks.org/python-how-to-create-an-arp-spoofer-using-scapy/?ref=rp)

**Execution steps :**  
Here, we will discuss the execution steps as follows.

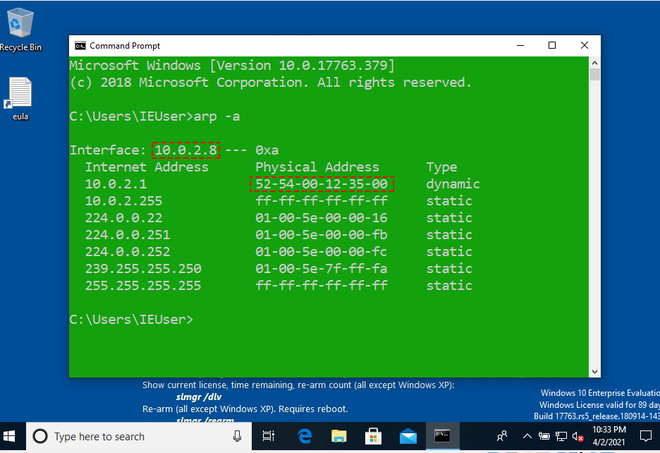
**Step-1:**  
We can run the built-in “ARPSPOOF’” tool in Kali Linux. In case the ARPSPOOF tool is not present, install the tool by running the following command as follows.

apt install dsniff

**Step-2 :**  
To run this attack we need two thingsVictim machine’s IP address & the IP of Gateway.  In this example, we are using a Windows Machine as our victim and Kali Machine to run the attack. To know the victim machines IP address and gateway IP by running the following command in both the Windows machine and Linux Machine as follows.

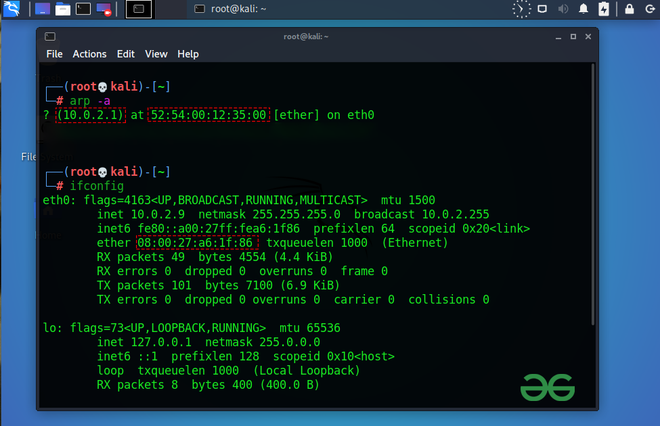
arp -a

**Output :**  
This will show us the following Outputs as follows.  
**Victim Machine (Windows Machine) –**



windows machine

**Attacker Machine (Kali Linux) –**  
From these, we can observe that the IP address of the Windows machine is 10.0.2.8 and the IP and MAC addresses of the gateway are 10.0.2.1 and 52:54:00:12:35:00, also the MAC address of our Kali Machine is 08:00:27:a6:1f:86.



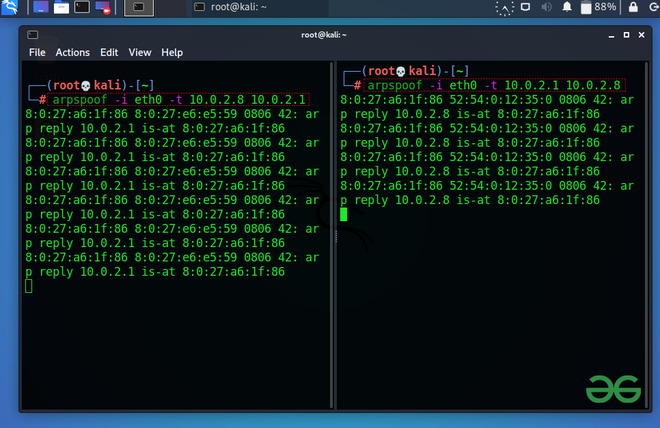
**Step-3 :**   
Now, write the following commands to perform the ARP Spoof attack.

arpspoof -i eth0 -t 10.0.2.8 10.0.2.1

Here eth0 is the name of the interface, 10.0.2.8 is the IP of the Windows machine and 10.0.2.1 is the IP of the gateway. This will fool the victim by pretending to be the router. So again we will run the above command one more time by switching its IP addresses as follows.

arpspoof -i eth0 -t 10.0.2.1 10.0.2.8

**Output :**  
**Attacker Machine (Kali Linux) –**  
This shows that our ARP Spoof attack is running, and we have successfully placed our system in the middle of the client and server.

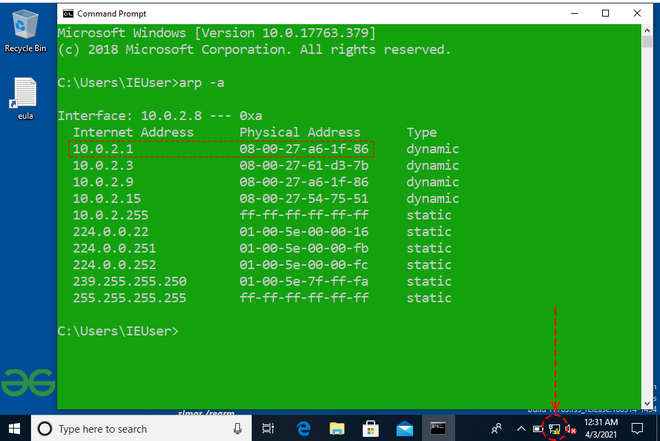


ARP Spoof attack running

 We can also check it by running the command as follows.

arp -a

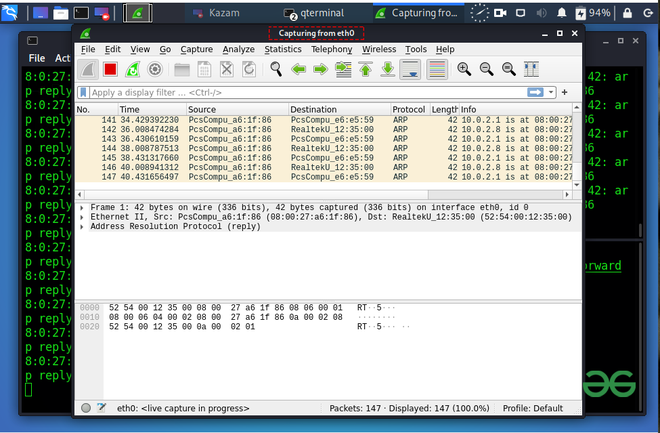
**Output :**  
In the output screen, you can observe that the MAC address of the gateway is changed to the MAC address of Kali Machine. Now all the data packets will flow through our Kali machine. Also, you can see that the internet connection of the victim machine is not working because it’s the security feature of Linux, which does not allow the flow of packets through it. So we need to enable Port Forwarding so that this computer will allow the packets to flow through it just like a router.



**Step-4 :**  
To enable Port Forwarding to run the command as follows.

echo 1 > /proc/sys/net/ipv4/ip\_forward

**Output :**  
This command will again establish the Internet connectivity of the victim computer. In this way, we can become the Man-In-The-Middle by using the ARP Spoof attack. So all the requests from the victim’s computer will not directly go to the router it will flow through the attacker’s machine and the attacker can sniff or extract useful information by using various tools like Wire Shark, etc. as shown below as follows.



Wire Shark – used to sniff useful information from the packets.

**EX. NO: 07**

## **WORKING WITH SNORT TOOL TO DEMONSTRATE INTRUSION DETECTION SYSTEM**

**AIM:**

Snort is an open source network intrusion detection system (NIDS) and it is a packet sniffer that monitors network traffic in real time.

**INTRODUCTION:**

**INTRUSION DETECTION SYSTEM :**

Intrusion detection is a set of techniques and methods that are used to detect suspicious activity both at the network and host level. Intrusion detection systems fall into two basic categories:

* + Signature-based intrusion detection systems
  + Anomaly detection systems.

Intruders have signatures, like computer viruses, that can be detected using software. You try to find data packets that contain any known intrusion-related signatures or anomalies related to Internet protocols. Based upon a set of signatures and rules, the detection system is able to find and log suspicious activity and generate alerts.

Anomaly-based intrusion detection usually depends on packet anomalies present in protocol header parts. In some cases these methods produce better results compared to signature-based IDS. Usually an intrusion detection system captures data from the network and applies its rules to that data or detects anomalies in it. Snort is primarily a rule-based IDS, however input plug-ins are present to detect anomalies in protocol headers.

**SNORT TOOL:**

Snort is based on libpcap (for library packet capture), a tool that is widely used in TCP/IPtraffic sniffers and analyzers. Through protocolanalysis and content searching and matching, Snort detects attack methods, including denial of service, buffer overflow, CGI attacks, stealthport scans, and SMB probes. When suspicious behavior is detected, Snort sends a real-time alert to syslog, a separate 'alerts' file, or to apop-up window.

Snort is currently the most popular free network intrusion detection software. The advantages of Snort are numerous. According to the snort web site, “It can perform protocol

analysis, content searching/matching, and can be used to detect a variety of attacks and probes, such as buffer overflow, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more” (Caswell).

One of the advantages of Snort is its ease of configuration. Rules are very flexible, easily written, and easily inserted into the rule base. If a new exploit or attack is found a rule for the attack can be added to the rule base in a matter of seconds. Another advantage of snort is that it allows for raw packet data analysis.

**SNORT can be configured to run in three modes:**

1. Sniffer mode
2. Packet Logger mode
3. Network Intrusion Detection System mode
4. **Sniffer mode**
   * **Snort –v** Print out the TCP/IP packets header on the screen
   * **Snort –vd** show the TCP/IP ICMP header with application data in transmit
5. **Packet Logger mode**
   * **snort –dev –l c:\log** [create this directory in the C drive] and snort will automatically know to go into packet logger mode, it collects every packet it sees and places it in log directory.
   * **snort –dev –l c:\log –h ipaddress/24**:This rule tells snort that you want to print out the data link and TCP/IP headers as well as application data into the log directory. snort –l c:\log –b This is binary mode logs everything into a single file.
6. **Network Intrusion Detection System mode**
   * **snort –d c:\log –h ipaddress/24 –c snort.conf** This is a configuration file applies rule to each packet to decide it an action based upon the rule type in the file.
   * **Snort –d –h ipaddress/24 –l c:\log –c snort.conf** This will cnfigure snort to run in its most basic NIDS form, logging packets that trigger rules specifies in the snort.conf.

**PROCEDURE:**

**STEP-1:** Sniffer mode🡪 snort –v 🡪 Print out the TCP/IP packets header on the screen.

**STEP-2:** Snort –vd 🡪 Show the TCP/IP ICMP header with application data in transit.

**STEP-3:** Packet Logger mode 🡪 snort –dev –l c:\log [create this directory in the C drive] and snort will automatically know to go into packet logger mode, it collects every packet it sees and places it in log directory.

**STEP-4:** snort –dev –l c:\log –h ipaddress/24 🡪 This rule tells snort that you want to print out the data link and TCP/IP headers as well as application data into the log directory.

**STEP-5:** snort –l c:\log –b 🡪 this binary mode logs everything into a single file.

**STEP-6:** Network Intrusion Detection System mode 🡪 snort –d c:\log –h ipaddress/24 –c snort.conf 🡪 This is a configuration file that applies rule to each packet to decide it an action based upon the rule type in the file.

**STEP-7:** snort –d –h ip address/24 –l c:\log –c snort.conf 🡪 This will configure snort to run in its most basic NIDS form, logging packets that trigger rules specifies in the snort.conf.

**STEP-8:** Download SNORT from snort.org. Install snort with or without database support.

**STEP-9:** Select all the components and Click Next. Install and Close.

**STEP-10:** Skip the WinPcap driver installation.

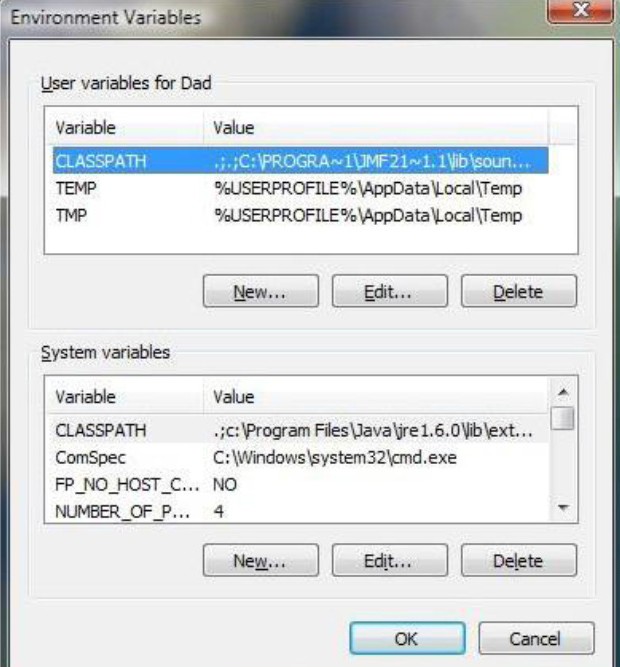
**STEP-11:** Add the path variable in windows environment variable by selecting new classpath.

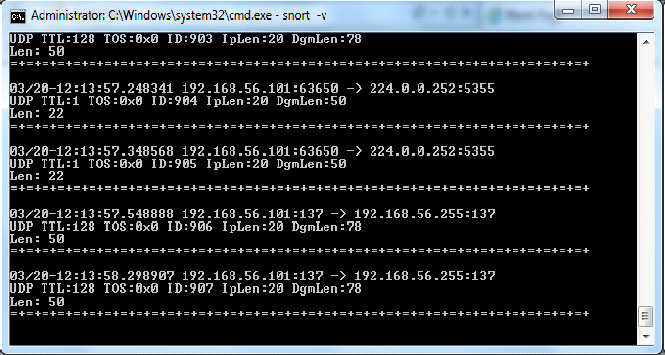
**STEP-12:** Create a path variable and point it at snort.exe variable name 🡪 path and variable value 🡪 c:\snort\bin.

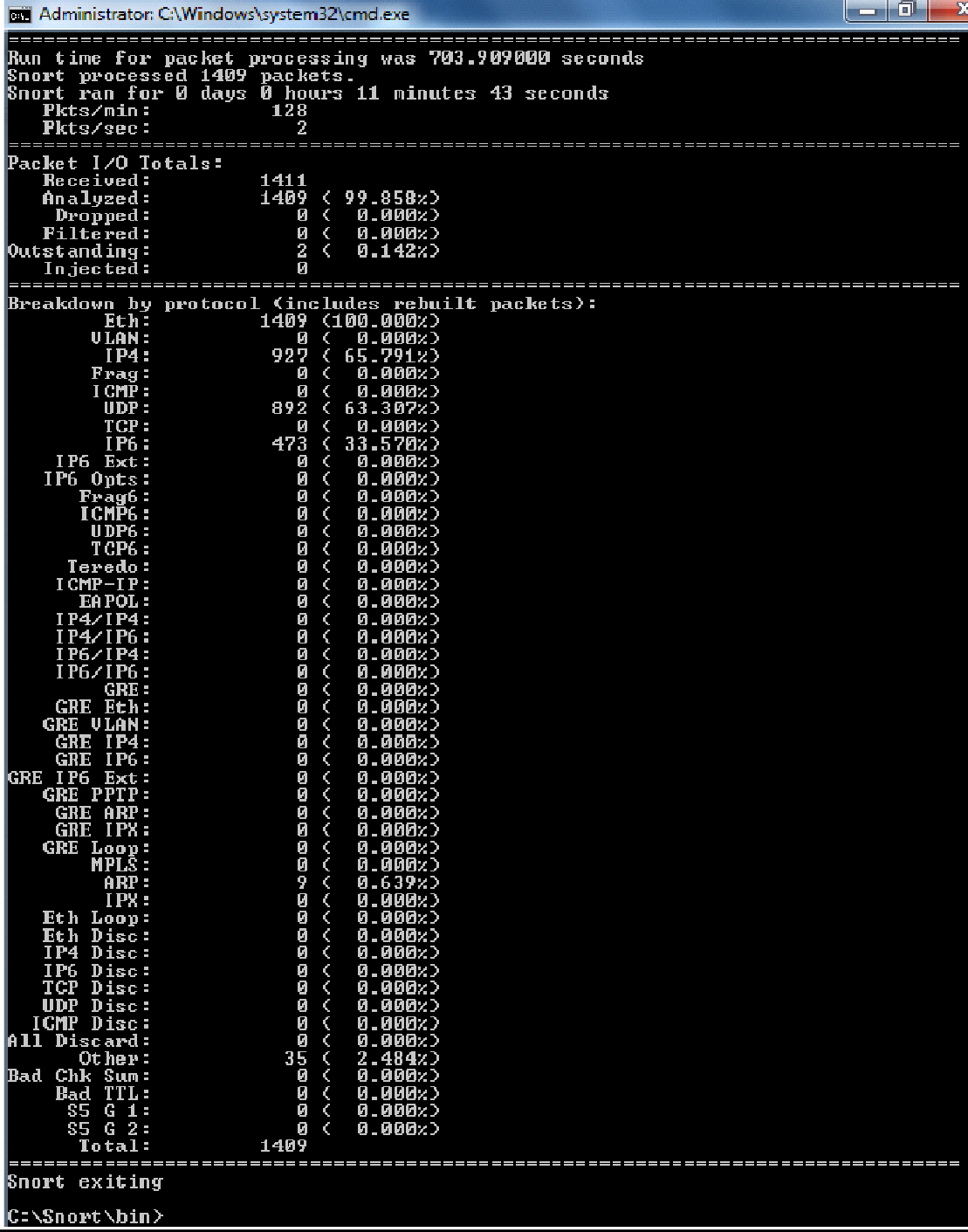
**STEP-13:** Click OK button and then close all dialog boxes. Open command prompt and type the following commands:

**INSTALLATION PROCESS :**









**EX. NO: 08**

**SECURE DATA STORAGE, SECURE DATA TRANSMISSION AND FOR CREATING DIGITAL SIGNATURES (GNUPG)**

**AIM:**

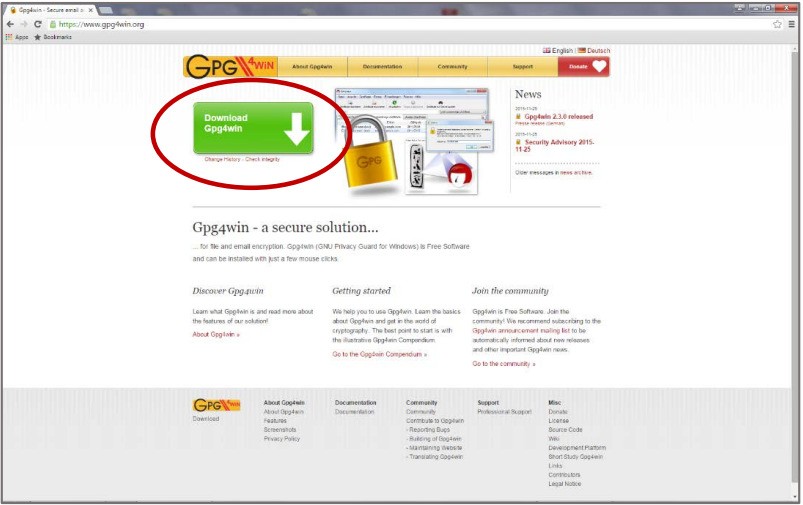
Demonstrate how to provide secure data storage, secure data transmission and for creating digital signatures (GnuPG).

**INTRODUCTION:**

* Here’s the final guide in my PGP basics series, this time focusing on Windows
* The OS in question will be Windows 7, but it should work for Win8 and Win8.1 as well
* Obviously it’s not recommended to be using Windows to access the DNM, but I won’t go into the reasons here.
* The tool well be using is GPG4Win

**INSTALLING THE SOFTWARE:**

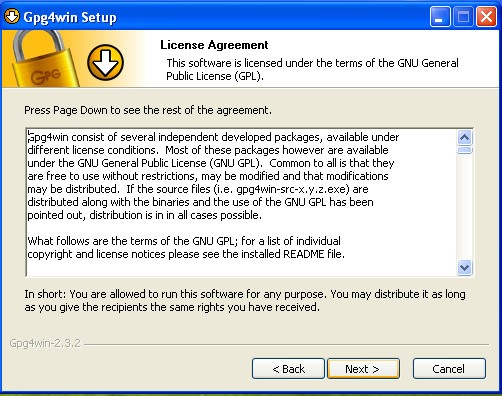
1. Visit [www.gpg4win.org.](http://www.gpg4win.org/) Click on the “Gpg4win 2.3.0” button
2. On the following screen, click the “Download Gpg4win” button.



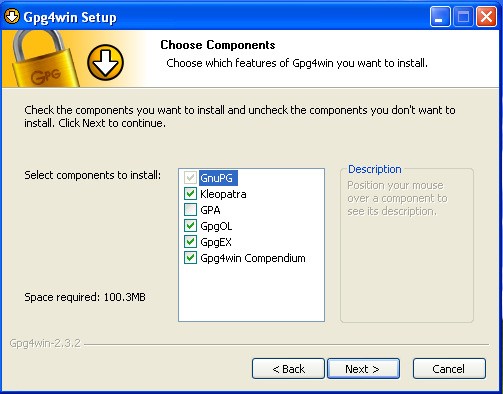
1. When the “Welcome” screen is displayed, click the “Next” button



1. When the “License Agreement” page is displayed, click the “Next” button



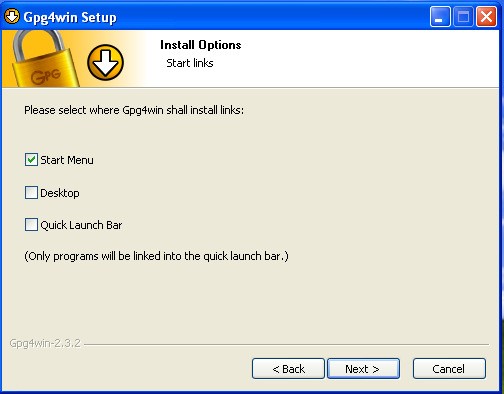
1. Set the check box values as specified below, then click the “Next” button



1. Set the location where you want the software to be installed. The default location is fine. Then, click the “Next” button.



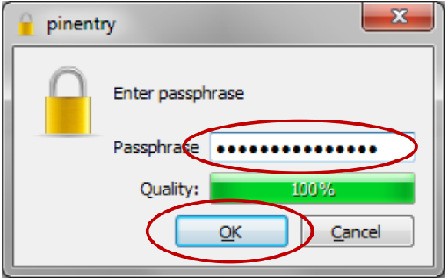
1. Specify where you want shortcuts to the software placed, then click the “Next” button.



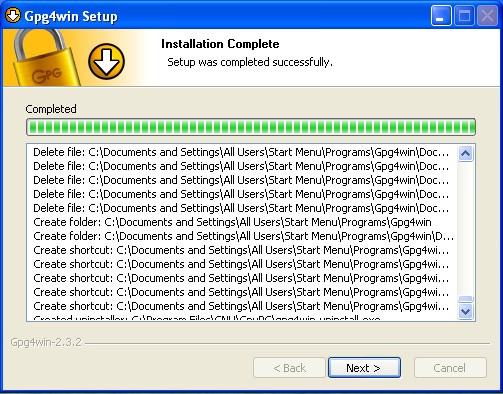
1. If you selected to have a GPG shortcut in your Start Menu, specify the folder in which it will be placed. The default “Gpg4win” is OK. Click the “Install” button to continue



1. A warning will be displayed if you have Outlook or Explorer opened. If this occurs, click the “OK” button.



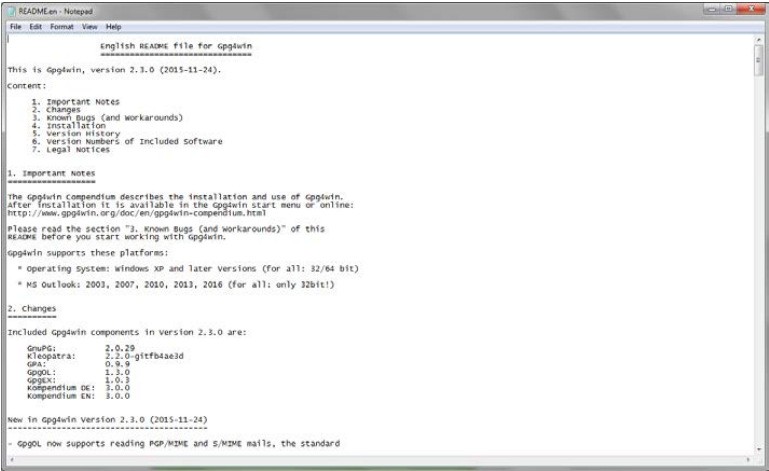
1. The installation process will tell you when it is complete. Click the “Next” button



1. Once the Gpg4win setup wizard is complete, the following screen will be displayed. Click the “Finish” button



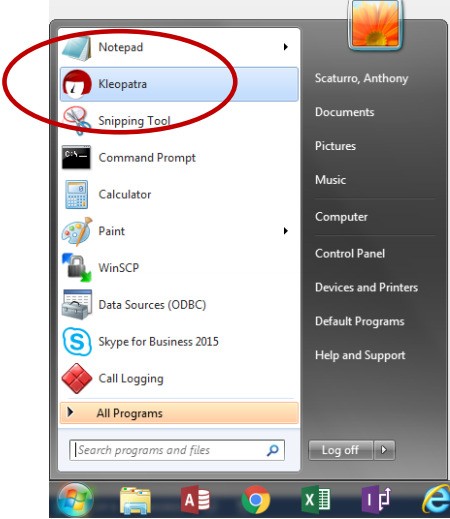
1. If you do not uncheck the “Show the README file” check box, the README file will be displayed. The window can be closed after you’ve reviewed it.



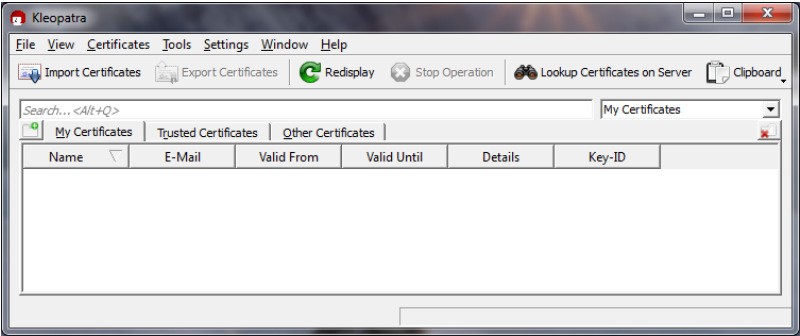
**CREATING YOUR PUBLIC AND PRIVATE KEYS**

GPG encryption and decryption is based upon the keys of the person who will be receiving the encrypted file or message. Any individual who wants to send the person an encrypted file or message must possess the recipient’s public key certificate to encrypt the message. The recipient must have the associated private key, which is different than the public key, to be able to decrypt the file. The public and private key pair for an individual is usually generated by the individual on his or her computer using the installed GPG program, called “Kleopatra” and the following procedure:

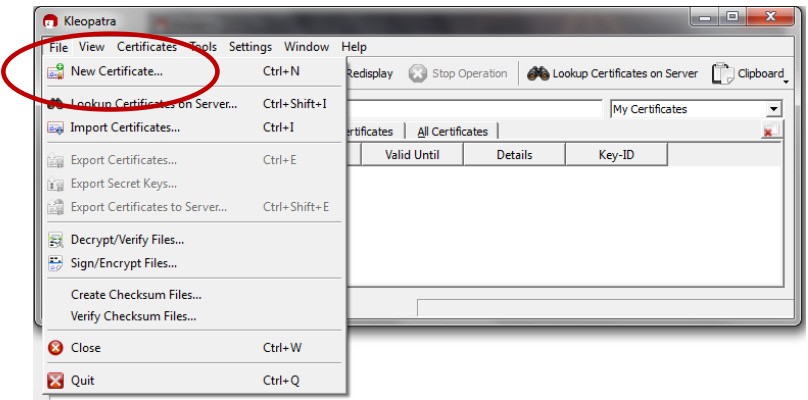
1. From your start bar, select the “Kleopatra” icon to start the Kleopatra certificate management software



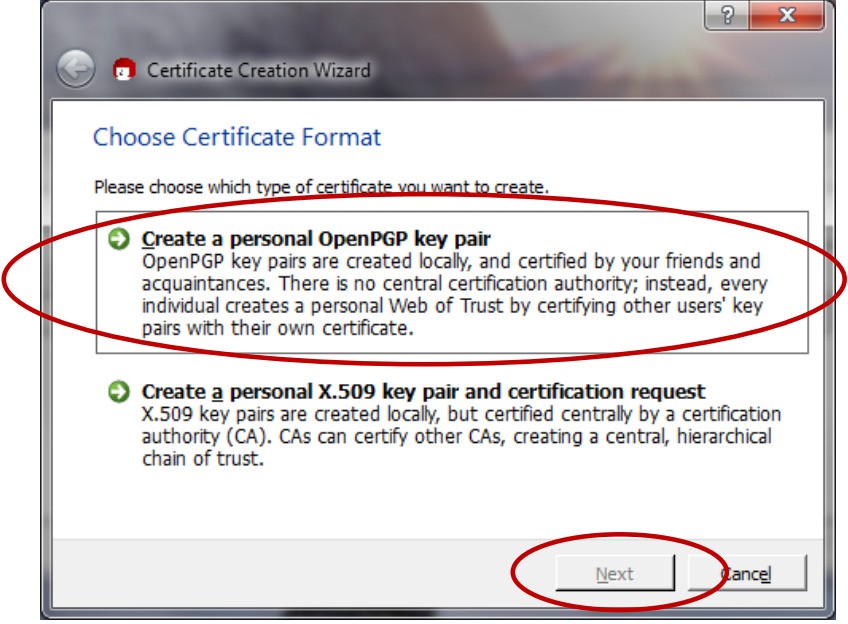
1. The following screen will be displayed



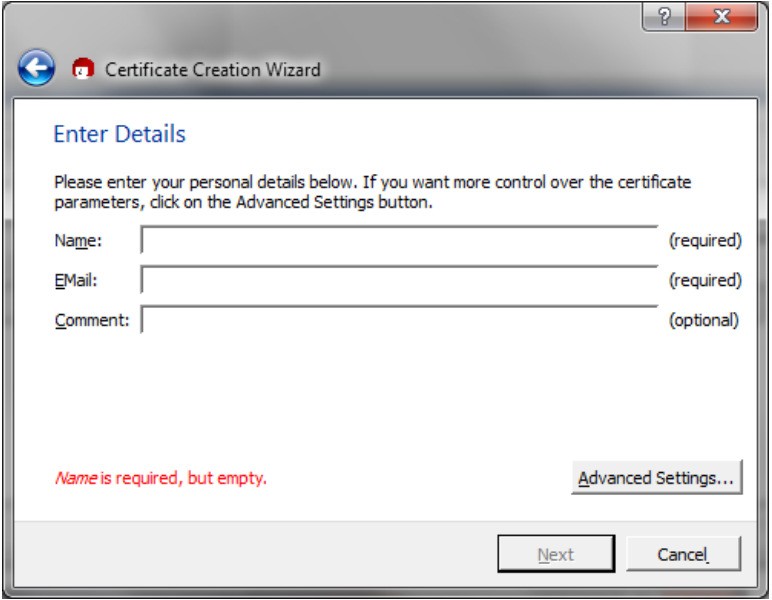
1. From the “File” dropdown, click on the “New Certificate” option



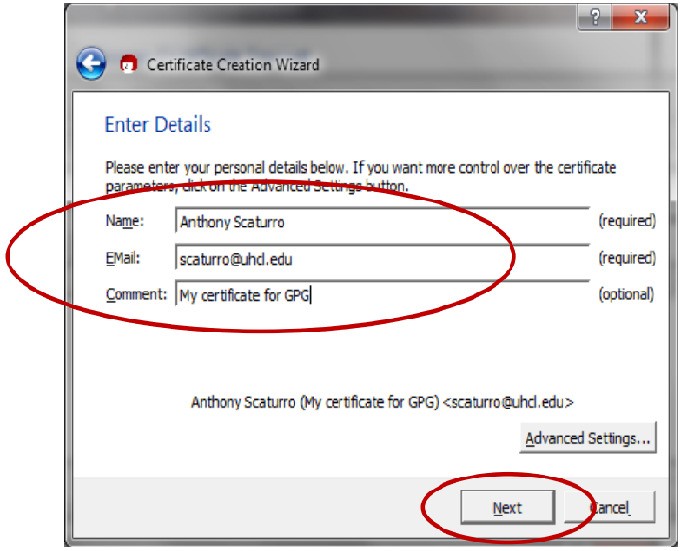
1. The following screen will be displayed. Click on “Create a personal OpenGPG key pair” and the “Next” button



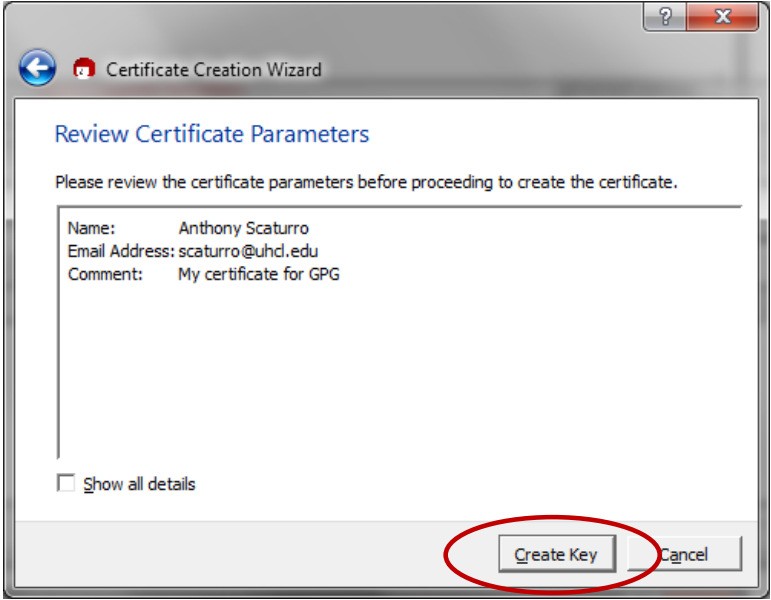
1. The Certificate Creation Wizard will start and display the following:



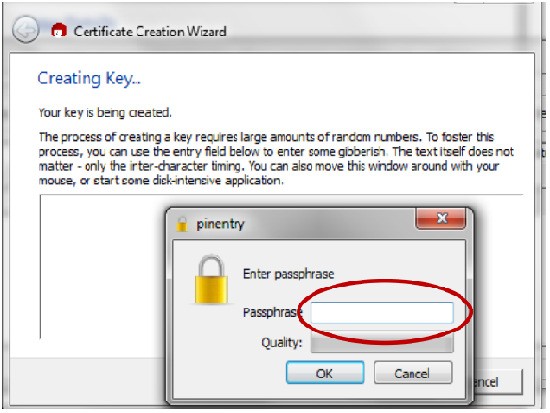
1. Enter your name and e-mail address. You may also enter an optional comment. Then, click the “Next” button



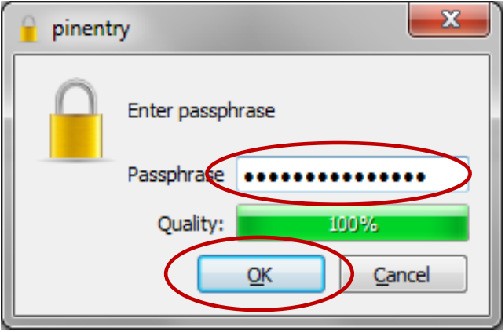
1. Review your entered values. If OK, click the “Create Key” button



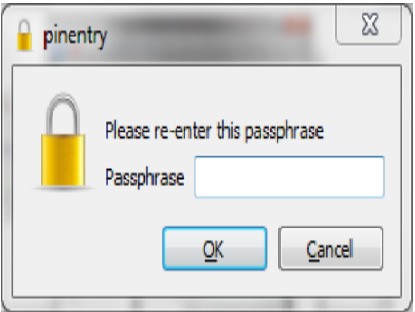
1. You will be asked to enter a passphrase



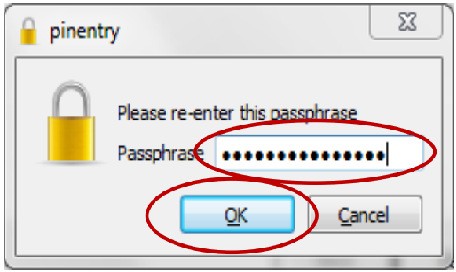
1. The passphrase should follow strong password standards. After you’ve entered your passphrase, click the “OK” button.



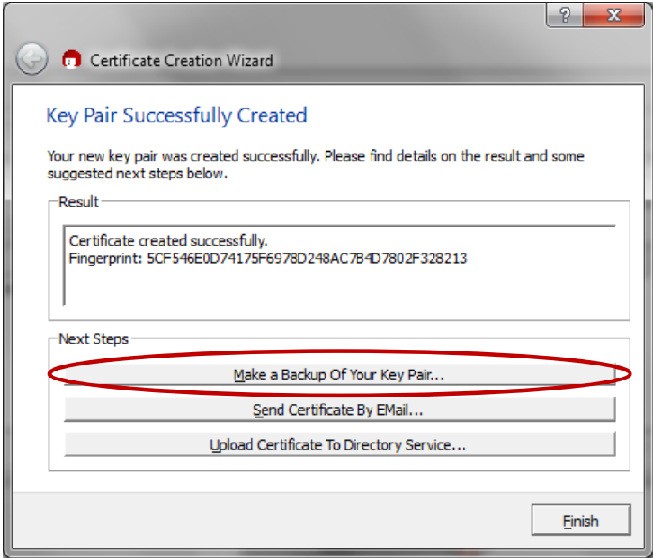
1. You will be asked to re-enter the passphrase



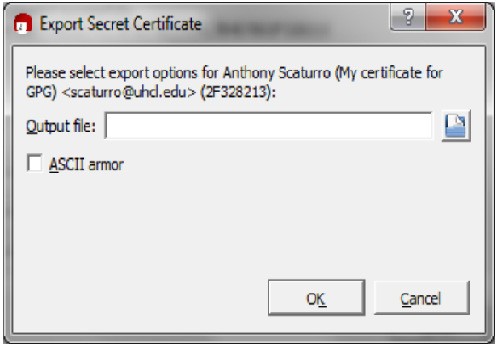
1. Re-enter the passphrase value. Then click the “OK” button. If the passphrases match, the certificate will be created.



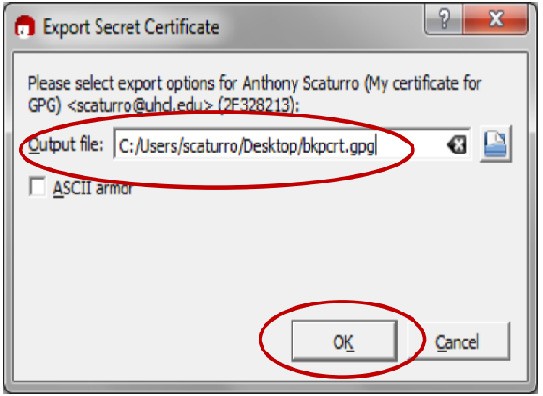
1. Once the certificate is created, the following screen will be displayed. You can save a backup of your public and private keys by clicking the “Make a backup Of Your Key Pair” button. This backup can be used to copy certificates onto other authorized computers.



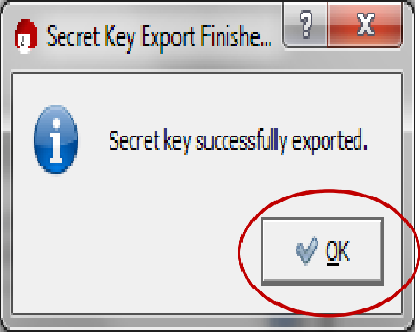
1. If you choose to backup your key pair, you will be presented with the following screen:



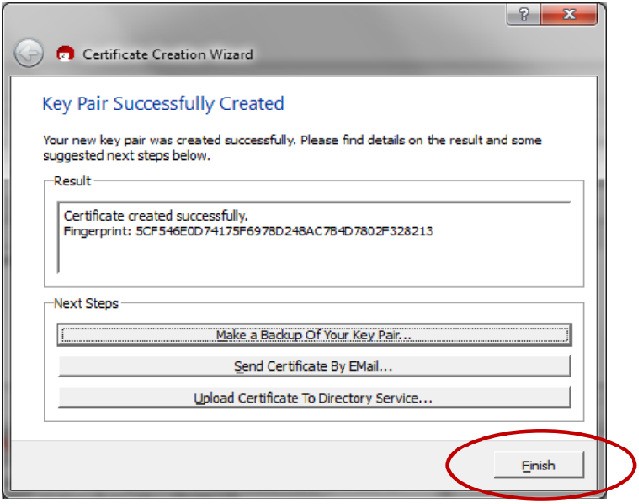
1. Specify the folder and name the file. Then click the “OK” button.



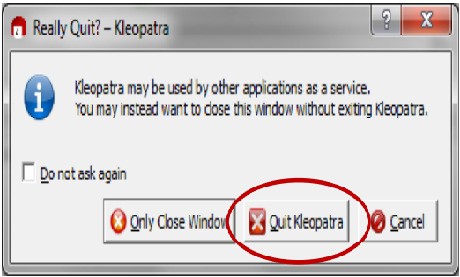
1. After the key is exported, the following will be displayed. Click the “OK” button.



1. You will be returned to the “Key Pair Successfully Created” screen. Click the “Finish” button.

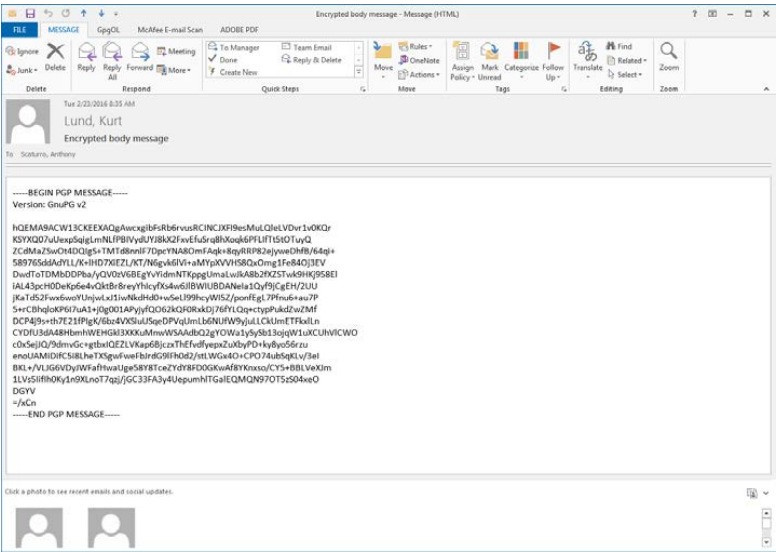


1. Before the program closes, you will need to confirm that you want to close the program by clicking on the “Quit Kleopatra” button

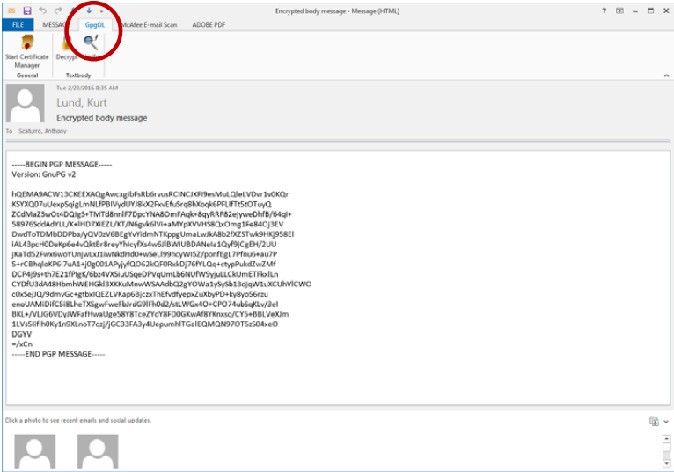


**DECRYPTING AN ENCRYPTED E-MAIL THAT HAS BEEN SENT TO YOU:**

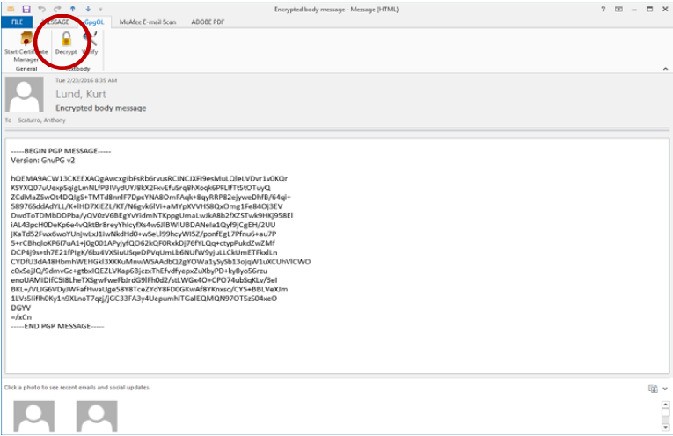
1. Open the e-mail message



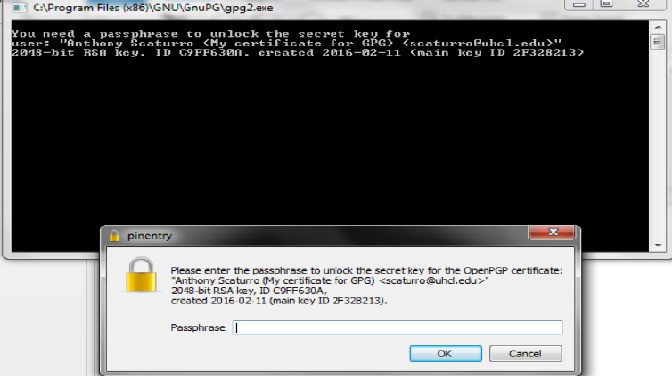
1. Select the GpgOL tab



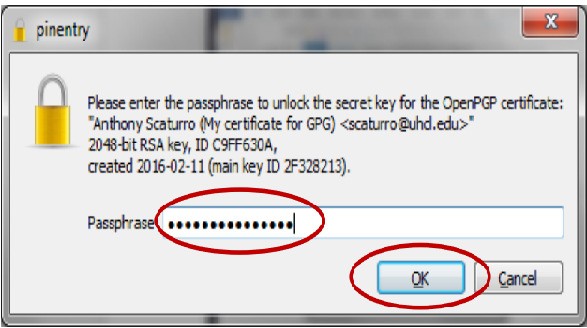
1. Click the “Decrypt” button



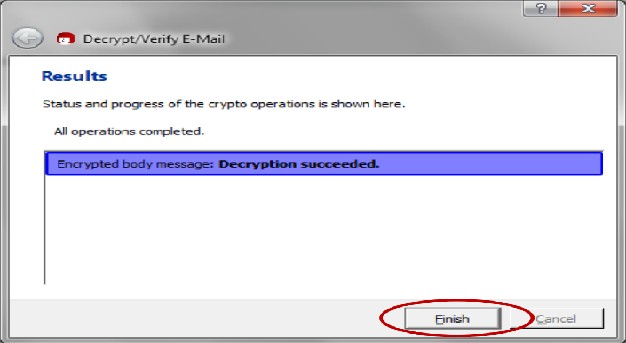
1. A command window will open along with a window that asks for the Passphrase to your private key that will be used to decrypt the incoming message.



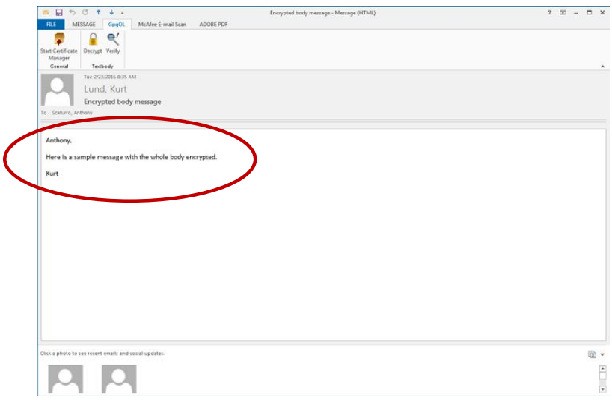
1. Enter your passphrase and click the “OK” button



1. The results window will tell you if the decryption succeeded. Click the “Finish” button top close the window



1. Your unencrypted e-mail message body will be displayed.



1. When you close the e-mail you will be asked if you want to save the e-mail message in its unencrypted form. For maximum security, click the “No” button. This will keep the message encrypted within the e-mail system and will require you to enter your passphrase each time you reopen the e-mail message

