**Ex No: 6 Diffie-Hellman key exchange algorithm**

**Date:**

**Aim:**

To implement the Diffie-Hellman Key Exchange algorithm for a given problem .

**Algorithm:**

1. [Alice and Bob](https://en.wikipedia.org/wiki/Alice_and_Bob) publicly agree to use a modulus *p* = 23 and base *g* = 5 (which is a primitive root modulo 23).
2. Alice chooses a secret integer ***a*** = 4, then sends Bob *A* = *g****a*** mod *p*
   * *A* = 5**4** mod 23 = 4
3. Bob chooses a secret integer ***b*** = 3, then sends Alice *B* = *g****b*** mod *p*
   * *B* = 5**3** mod 23 = 10
4. Alice computes ***s*** = *B****a*** mod *p*
   * ***s*** = 10**4** mod 23 = 18
5. Bob computes ***s*** = *A****b*** mod *p*
   * ***s*** = 4**3** mod 23 = 18
6. Alice and Bob now share a secret (the number 18).

**Program:**

class DiffieHellman {

public static void main(String args[]) {

int p = 27; /\* publicly known (prime number) \*/

int g = 5; /\* publicly known (primitive root) \*/

int x = 2; /\* only Alice knows this secret \*/

int y = 3; /\* only Bob knows this secret \*/

double aliceSends = (Math.pow(g, x)) % p;

double bobComputes = (Math.pow(aliceSends, y)) % p;

double bobSends = (Math.pow(g, y)) % p;

double aliceComputes = (Math.pow(bobSends, x)) % p;

double sharedSecret = (Math.pow(g, (x \* y))) % p;

System.out.println("simulation of Diffie-Hellman key exchange algorithm\n---------------------------------------------");

System.out.println("Alice Sends : " + aliceSends);

System.out.println("Bob Computes : " + bobComputes);

System.out.println("Bob Sends : " + bobSends);

System.out.println("Alice Computes : " + aliceComputes);

System.out.println("Shared Secret : " + sharedSecret);

/\* shared secrets should match and equality is transitive \*/

if ((aliceComputes == sharedSecret) && (aliceComputes == bobComputes))

System.out.println("Success: Shared Secrets Matches! " + sharedSecret);

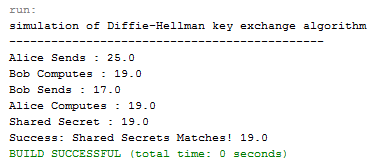
else

System.out.println("Error: Shared Secrets does not Match");

}

}

**Output:**

****

|  |  |
| --- | --- |
| Observation |  |
| Record |  |
| Total |  |

**Result:**

Thus the *Diffie-Hellman key exchange algorithm* has been implemented using Java Program and the output has been verified successfully.

**Ex No: 7 SHA-1 Algorithm**

**Date:**

**Aim:**

To calculate the message digests of a text using the SHA-1 algorithm.

**Algorithm:**

1. Append Padding Bits
2. Append Length - 64 bits are appended to the end
3. Prepare Processing Functions
4. Prepare Processing Constants
5. Initialize Buffers
6. Processing Message in 512-bit blocks (L blocks in total message)

**Program:**

import java.security.\*;

public class sha1 {

public static void main(String[] a) {

try {

MessageDigest md = MessageDigest.getInstance("SHA1");

System.out.println("Message digest object info:\n-----------------");

System.out.println("Algorithm=" + md.getAlgorithm());

System.out.println("Provider=" + md.getProvider());

System.out.println("ToString=" + md.toString());

String input = "";

md.update(input.getBytes());

byte[] output = md.digest();

System.out.println();

System.out.println("SHA1(\"" + input + "\")=" + bytesToHex(output));

input = "abcd3";

md.update(input.getBytes());

output = md.digest();

System.out.println();

System.out.println("SHA1(\"" + input + "\")=" + bytesToHex(output));

input = "abcdefghijklmnopqrstuvwxyz";

md.update(input.getBytes());

output = md.digest();

System.out.println();

System.out.println("SHA1(\"" + input + "\")=" + bytesToHex(output));

System.out.println();

} catch (Exception e) {

System.out.println("Exception:" + e);

}

}

private static String bytesToHex(byte[] b) {

char hexDigit[] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F' };

StringBuffer buf = new StringBuffer();

for (byte aB : b) {

buf.append(hexDigit[(aB >> 4) & 0x0f]);

buf.append(hexDigit[aB & 0x0f]);

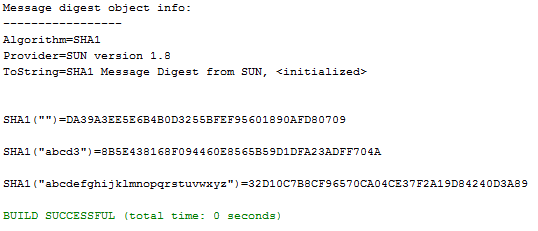
}

return buf.toString();

}

}

**Output:**

****

|  |  |
| --- | --- |
| Observation |  |
| Record |  |
| Total |  |

**Result:**

Thus the *Secure Hash Algorithm (SHA-1)* has been implemented and the output has been verified successfully.

**Ex No: 8 Digital Signature Standard**

**Date:**

**Aim:**

To implement the SIGNATURE SCHEME - Digital Signature Standard.

**Algorithm:**

1. Create a KeyPairGenerator object.
2. Initialize the KeyPairGenerator object.
3. Generate the KeyPairGenerator. ...
4. Get the private key from the pair.
5. Create a signature object.
6. Initialize the Signature object.
7. Add data to the Signature object
8. Calculate the Signature

**Program:**

import java.security.KeyPair;

import java.security.KeyPairGenerator;

import java.security.PrivateKey;

import java.security.Signature;

import java.util.Scanner;

public class DigitalSign {

public static void main(String args[]) throws Exception {

Scanner sc = new Scanner(System.in);

System.out.println("Enter some text");

String msg = sc.nextLine();

KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("DSA");

keyPairGen.initialize(2048);

KeyPair pair = keyPairGen.generateKeyPair();

PrivateKey privKey = pair.getPrivate();

Signature sign = Signature.getInstance("SHA256withDSA");

sign.initSign(privKey);

byte[] bytes = "msg".getBytes();

sign.update(bytes);

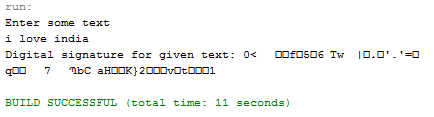
byte[] signature = sign.sign();

System.out.println("Digital signature for given text: "+new String(signature, "UTF8"));

}

}

**Output:**

****

|  |  |
| --- | --- |
| Observation |  |
| Record |  |
| Total |  |

**Result:**

Thus the Digital Signature Standard Signature Scheme has been implemented and the output has been verified successfully.

**Ex No: 9 Demonstration of Intrusion Detection System(IDS)**

**Date:**

**Aim:**

To demonstrate Intrusion Detection System (IDS) using Snort software tool.

**Steps on Configuring and Intrusion Detection:**

1. Download Snort from the Snort.org website. (http://www.snort.org/snort-downloads)

2. Download Rules(https://www.snort.org/snort-rules). You must register to get the rules. (You should download these often)

3. Double click on the .exe to install snort. This will install snort in the “C:\Snort” folder.It is important to have WinPcap (https://www.winpcap.org/install/) installed

4. Extract the Rules file. You will need WinRAR for the .gz file.

5. Copy all files from the “rules” folder of the extracted folder. Now paste the rules into *“C:\Snort\rules”* folder.

6. Copy “snort.conf” file from the “etc” folder of the extracted folder. You must paste it into “C:\Snort\etc” folder. Overwrite any existing file. Remember if you modify your snort.conf file and download a new file, you must modify it for Snort to work.

7. Open a command prompt (cmd.exe) and navigate to folder “C:\Snort\bin” folder. ( at the Prompt, type cd\snort\bin)

8. To start (execute) snort in sniffer mode use following command:

snort -dev -i 3

-i indicates the interface number. You must pick the correct interface number. In my case, it is 3.

-dev is used to run snort to capture packets on your network.

To check the interface list, use following command:

snort -W

Finding an interface

You can tell which interface to use by looking at the Index number and finding Microsoft. As you can see in the above example, the other interfaces are for VMWare. My interface is 3.



9. To run snort in IDS mode, you will need to configure the file “snort.conf” according to your network environment.

10. To specify the network address that you want to protect in snort.conf file, look for the following line.

var HOME\_NET 192.168.1.0/24 (You will normally see any here)

11. You may also want to set the addresses of DNS\_SERVERS, if you have some on your network.

Example:

example snort

12. Change the RULE\_PATH variable to the path of rules folder.

var RULE\_PATH c:\snort\rules

path to rules

13. Change the path of all library files with the name and path on your system. and you must change the path of snort\_dynamicpreprocessorvariable.

C:\Snort\lib\snort\_dynamiccpreprocessor

You need to do this to all library files in the “C:\Snort\lib” folder. The old path might be: “/usr/local/lib/…”. you will need to replace that path with your system path. Using C:\Snort\lib

14. Change the path of the “dynamicengine” variable value in the “snort.conf” file..

Example:

dynamicengine C:\Snort\lib\snort\_dynamicengine\sf\_engine.dll

15 Add the paths for “include classification.config” and “include reference.config” files.

include c:\snort\etc\classification.config

include c:\snort\etc\reference.config

16. Remove the comment (#) on the line to allow ICMP rules, if it is commented with a #.

include $RULE\_PATH/icmp.rules

17. You can also remove the comment of ICMP-info rules comment, if it is commented.

include $RULE\_PATH/icmp-info.rules

18. To add log files to store alerts generated by snort, search for the “output log” test in snort.conf and add the following line:

output alert\_fast: snort-alerts.ids

19. Comment (add a #) the whitelist $WHITE\_LIST\_PATH/white\_list.rules and the blacklist

Change the nested\_ip inner , \ to nested\_ip inner #, \

20. Comment out (#) following lines:

#preprocessor normalize\_ip4

#preprocessor normalize\_tcp: ips ecn stream

#preprocessor normalize\_icmp4

#preprocessor normalize\_ip6

#preprocessor normalize\_icmp6

21. Save the “snort.conf” file.

22. To start snort in IDS mode, run the following command:

snort -c c:\snort\etc\snort.conf -l c:\snort\log -i 3

(Note: 3 is used for my interface card)

If a log is created, select the appropriate program to open it. You can use WordPard or NotePad++ to read the file.

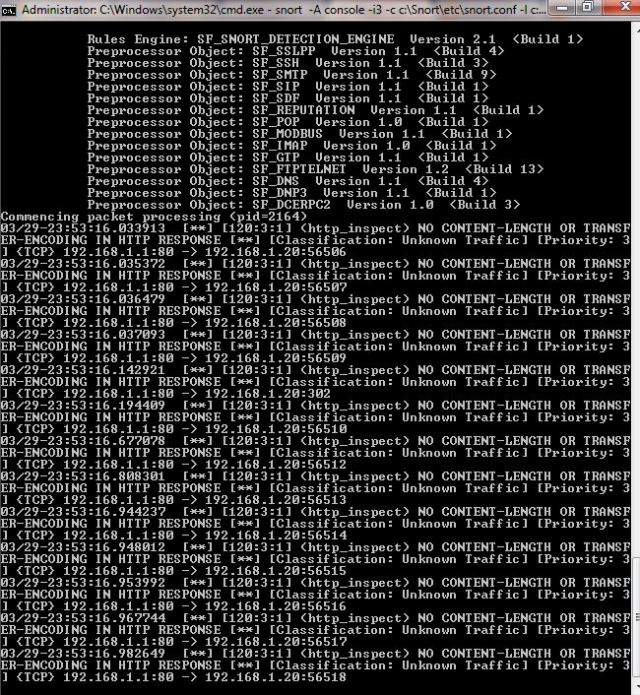
To generate Log files in ASCII mode, you can use following command while running snort in IDS mode:

snort -A console -i3 -c c:\Snort\etc\snort.conf -l c:\Snort\log -K ascii

23. Scan the computer that is running snort from another computer by using PING or NMap (ZenMap).

After scanning or during the scan you can check the snort-alerts.ids file in the log folder to insure it is logging properly. You will see IP address folders appear.

Snort monitoring traffic –



|  |  |
| --- | --- |
| Observation |  |
| Record |  |
| Total |  |

**Result:**

Thus the Intrusion Detection System (IDS) has been demonstrated by using the Open Source Snort Intrusion Detection Tool.

**Ex No: 10 Exploring N-Stalker, a Vulnerability Assessment Tool**

**Date:**

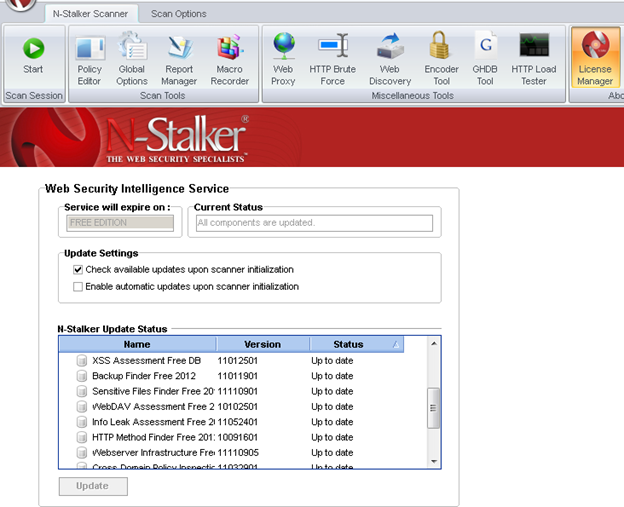
**Aim:**

To download the N-Stalker Vulnerability Assessment Tool and exploring the features.

**Exploring N-Stalker:**

* N-Stalker Web Application Security Scanner is a Web security assessment tool.
* It incorporates with a well-known N-Stealth HTTP Security Scanner and 35,000 Web attack signature database.
* This tool also comes in both free and paid version.
* Before scanning the target, go to “License Manager” tab, perform the update.
* Once update, you will note the status as up to date.
* You need to download and install N-Stalker from [www.nstalker.com](http://www.nstalker.com).

1. Start N-Stalker from a Windows computer. The program is installed under Start ➪ Programs ➪ N-Stalker ➪ N-Stalker Free Edition.
2. Enter a host address or a range of addresses to scan.
3. Click Start Scan.
4. After the scan completes, the N-Stalker Report Manager will prompt
5. you to select a format for the resulting report as choose Generate HTML.
6. Review the HTML report for vulnerabilities.



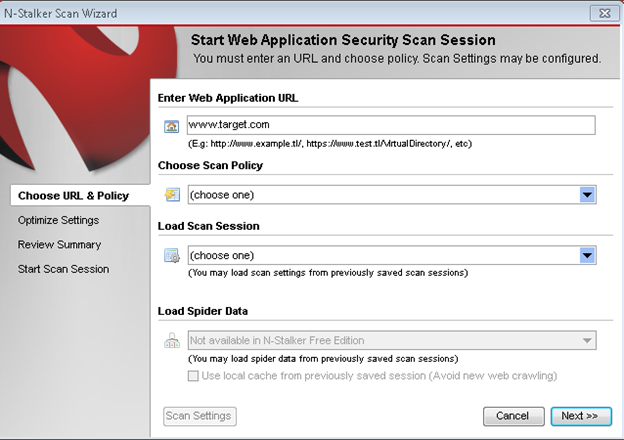
Now goto “Scan Session”, enter the target URL.

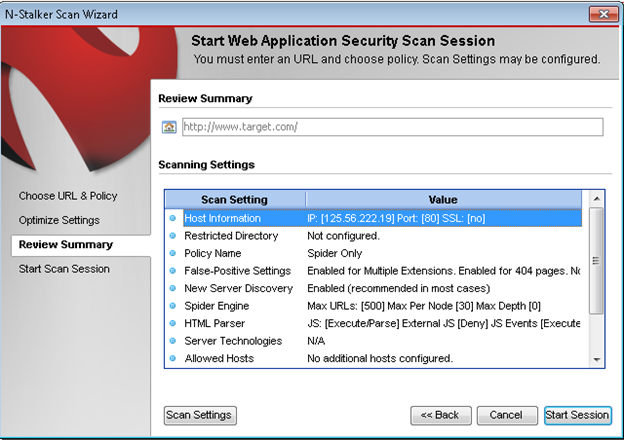
In scan policy, you can select from the four options,

* Manual test which will crawl the website and will be waiting for manual attacks.
* full xss assessment
* owasp policy
* Web server infrastructure analysis.

Once, the option has been selected, next step is “Optimize settings” which will crawl the whole website for further analysis.

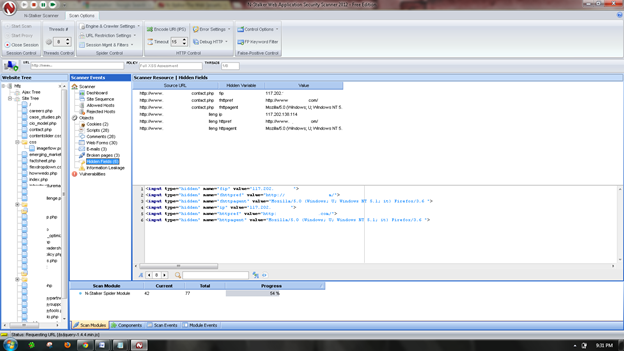
In review option, you can get all the information like host information, technologies used, policy name, etc.



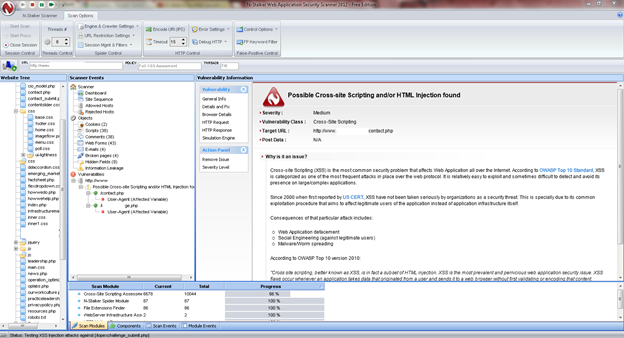


Once done, start the session and start the scan.

The scanner will crawl the whole website and will show the scripts, broken pages, hidden fields, information leakage, web forms related information which helps to analyze further.



Once the scan is completed, the NStalker scanner will show details like severity level, vulnerability class, why is it an issue, the fix for the issue and the URL which is vulnerable to the particular vulnerability?



|  |  |
| --- | --- |
| Observation |  |
| Record |  |
| Total |  |

**Result:**

Thus the N-Stalker Vulnerability Assessment tool has been downloaded installed and the feature has been explored by using a vulnerable website.

**Ex.No:11a Defeating Malware - Building Trojans**

**Date:**

**Aim:**

To build a Trojan and know the harmness of the trojan malwares in a computer system.

**Procedure:**

1. Create a simple trojan by using Windows Batch File (***.bat***)
2. Type these below code in notepad and save it as **Trojan.bat**
3. Double click on ***Trojan.bat***file.
4. When the trojan code executes, it will open MS-Paint, Notepad, Command Prompt, Explorer, etc., infinitely.
5. Restart the computer to stop the execution of this trojan.

**Trojan:**

* In computing, a Trojan horse,or trojan, is any malware which misleads users of its true intent.
* Trojans are generally spread by some form of social engineering, for example where a user is duped into executing an email attachment disguised to appear not suspicious, (e.g., a routine form to be filled in), or by clicking on some fake advertisement on social media or anywhere else.
* Although their payload can be anything, many modern forms act as a backdoor, contacting a controller which can then have unauthorized access to the affected computer.
* Trojans may allow an attacker to access users' personal information such as banking information, passwords, or personal identity.
* ***Example:****Ransomware* attacks are often carried out using a *trojan*.

**Code:**

***Trojan.bat***

@echo off

:x

start mspaint

start notepad

start cmd

start explorer

start control

start calc

goto x

**Output:**

(MS-Paint, Notepad, Command Prompt, Explorer will open infinitely)

**Result:**

Thus a trojan has been built and the harmness of the trojan viruses has been explored.

**Ex No: 11b Defeating Malware - Rootkit hunter**

**Date:**

**Aim:**

To install a rootkit hunter and find the malwares in a computer.

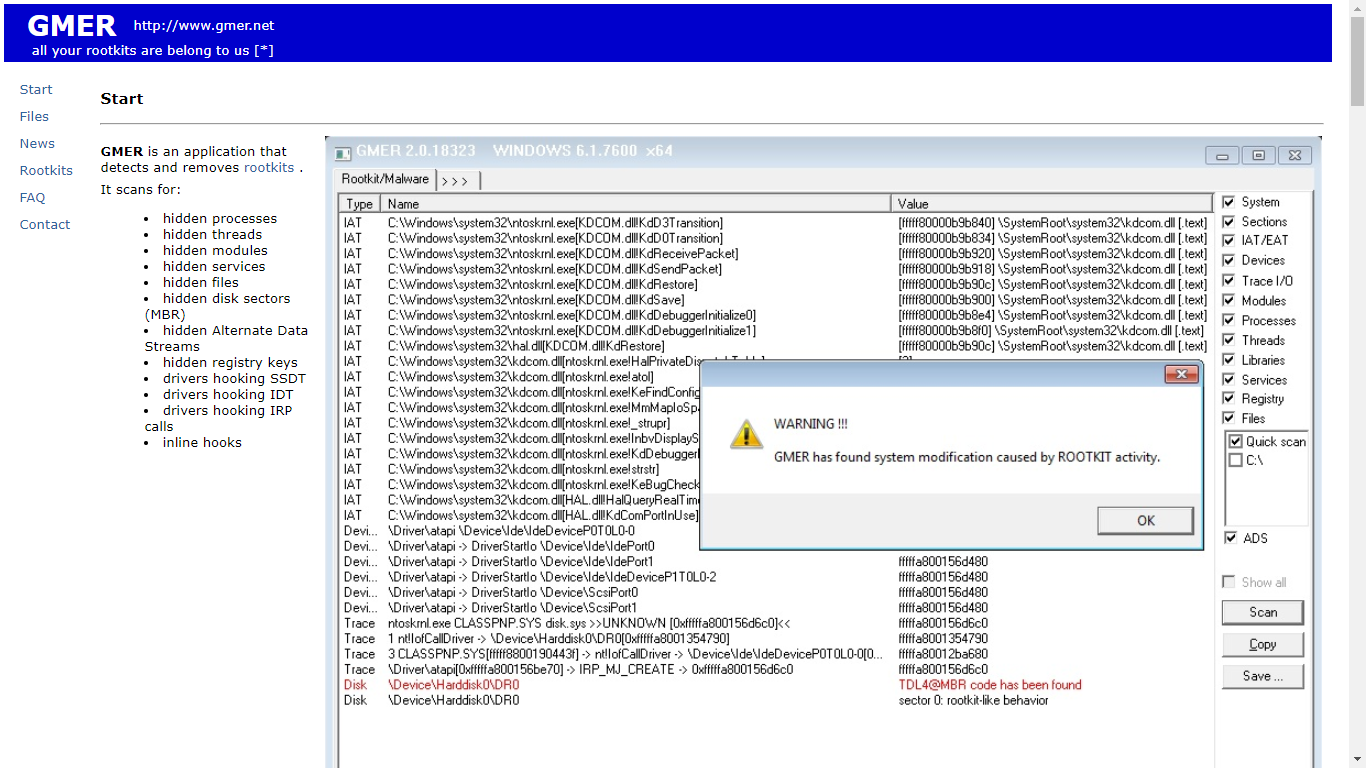
**Rootkit Hunter:**

* rkhunter (Rootkit Hunter) is a Unix-based tool that scans for rootkits, backdoors and possible local exploits.
* It does this by comparing SHA-1 hashes of important files with known good ones in online databases, searching for default directories (of rootkits), wrong permissions, hidden files, suspicious strings in kernel modules, and special tests for Linux and FreeBSD.
* rkhunter is notable due to its inclusion in popular operating systems (Fedora, Debian, etc.)
* The tool has been written in Bourne shell, to allow for portability. It can run on almost all UNIX-derived systems.

**GMER Rootkit Tool:**

* GMER is a software tool written by a Polish researcher Przemysław Gmerek, for detecting and removing rootkits.
* It runs on Microsoft Windows and has support for Windows NT, 2000, XP, Vista, 7, 8 and 10. With version 2.0.18327 full support for Windows x64 is added.

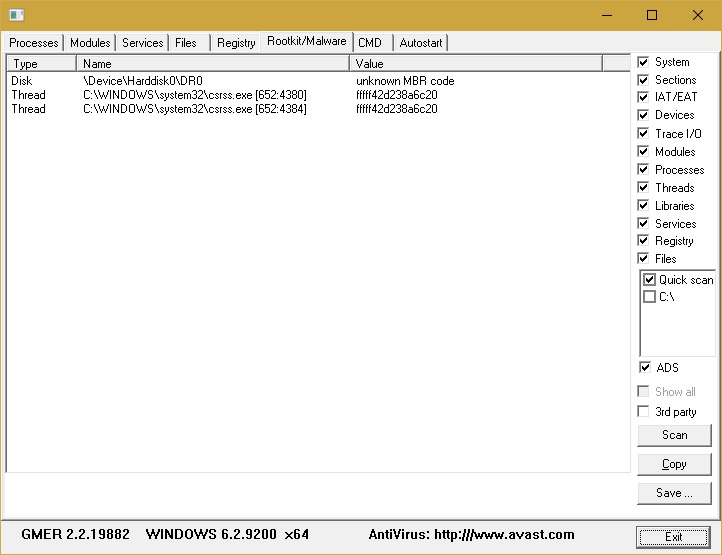
**Step 1**



Visit GMER's website (see Resources) and download the GMER executable.

Click the "Download EXE" button to download the program with a random file name, as some rootkits will close “gmer.exe” before you can open it.

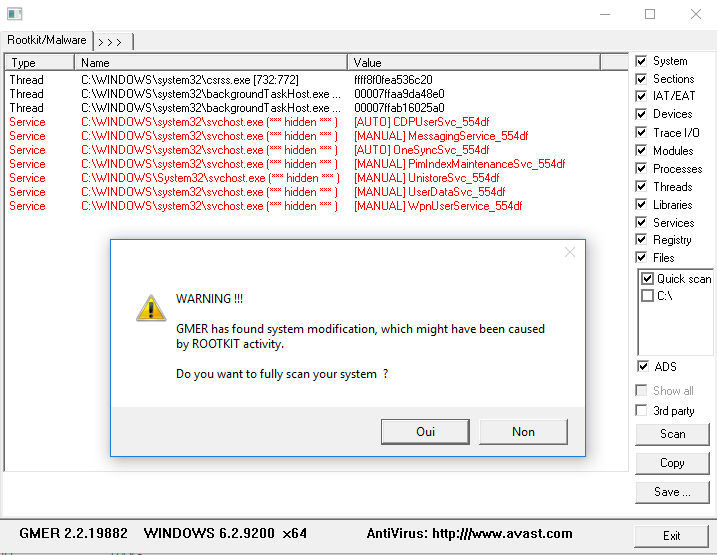
**Step 2**



Double-click the icon for the program.

Click the "Scan" button in the lower-right corner of the dialog box. Allow the program to scan your entire hard drive.

**Step 3**



When the program completes its scan, select any program or file listed in red. Right-click it and select "Delete."

If the red item is a service, it may be protected. Right-click the service and select "Disable." Reboot your computer and run the scan again, this time selecting "Delete" when that service is detected.

When your computer is free of Rootkits, close the program and restart your PC.

|  |  |
| --- | --- |
| Observation |  |
| Record |  |
| Total |  |

**Result:**

Thus the experiment a rootkit hunter software tool has been installed and the rootkits have been detected.