Substitution Cipher

Hiding some data is known as encryption. When plain text is encrypted it becomes unreadable and is known as ciphertext. In a Substitution cipher, any character of plain text from the given fixed set of characters is substituted by some other character from the same set depending on a key. For example with a shift of 1, A would be replaced by B, B would become C, and so on.

Note: Special case of Substitution cipher is known as Caesar cipher where the key is taken as 3.

Plaintext : abcd

Key 3

Ciphertext : defg

abcdefghijklmnopqrstuvwxyz

Plaintext :abcd

Key 3

Ciphertext:defg

Plain text xyz

Key 3

Cipher text abc

Mathematical representation

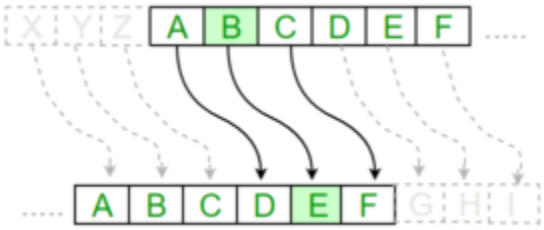
The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,…, Z = 25. Encryption of a letter by a shift n can be described mathematically as.

$$E_n(x) = (x + n) mod\ 26$$

Encryption Phase with shift n)

$$D_n(x) = (x - n) \bmod 26$$

Decryption Phase with shift n)



Plain Text:
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Output:
EFGHIJKLMNOPQRSTUVWXYZabcd

Transposition Cipher is a cryptographic algorithm where the order of alphabets in the plaintext is rearranged to form a cipher text. In this process, the actual plain text alphabets are not included

Example

A simple example for a transposition cipher is columnar transposition cipher where each character in the plain text is written horizontally with specified alphabet width. The cipher is written vertically, which creates an entirely different cipher text.

Consider the plain text hello world, and let us apply the simple columnar transposition technique as shown below

h	e	l	l
o	w	o	r
l	d		

hel

Low

Orl

d

Cipher text

Holewdlo lr

The plain text characters are placed horizontally and the cipher text is created with vertical format as : **holewdlo lr.** Now, the receiver has to use the same table to decrypt the cipher text to plain text.

product cipher, data encryption scheme in which the ciphertext produced by encrypting a plaintext document is subjected to further encryption. By combining two or more simple transposition ciphers or substitution ciphers, a more secure encryption may result.

(substitution cipher)

Implement rail fence cipher in java(transposition cipher)

Rail Fence Cipher

Given a plain-text message and a numeric key, cipher/de-cipher the given text using Rail Fence algorithm.

The rail fence cipher (also called a zigzag cipher) is a form of transposition cipher. It derives its name from the way in which it is encoded.

Encryption

Input : "GeeksforGeeks "
Key = 3
Output : GsGsekfrek eoe

AES

The more popular and widely adopted symmetric encryption algorithm likely to be

encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

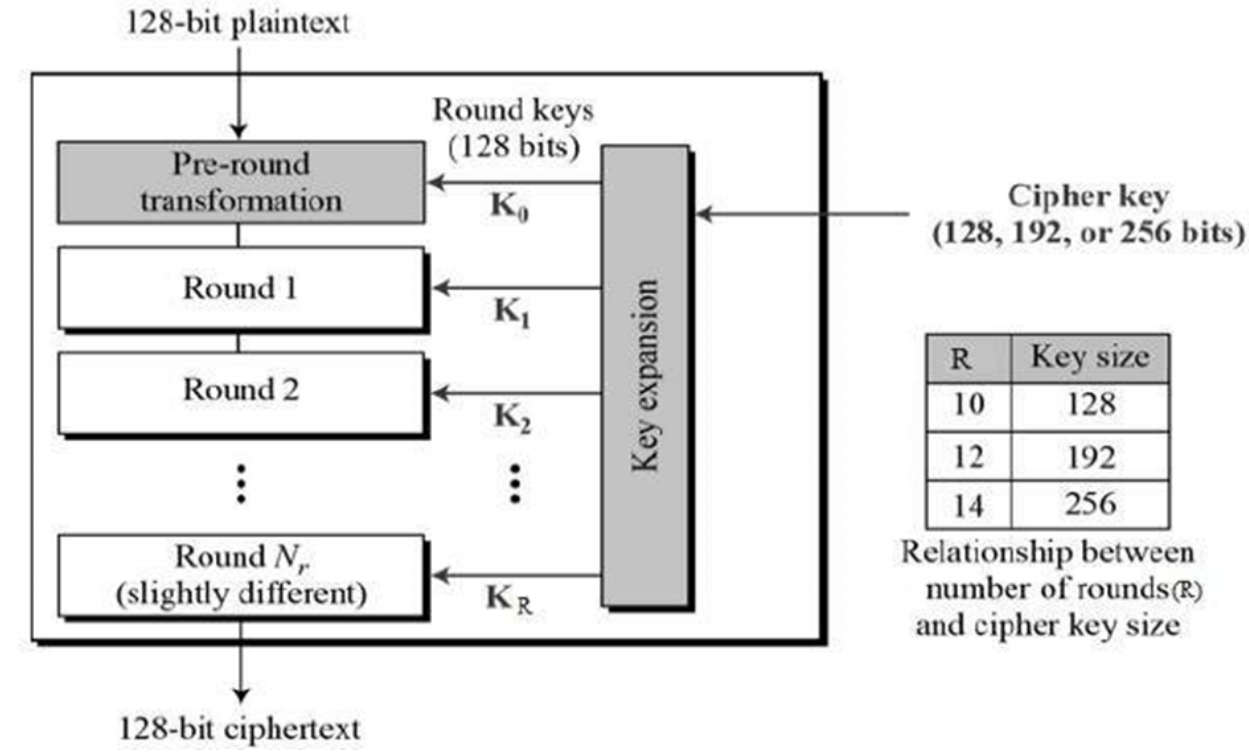
Operation of AES

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’.

It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

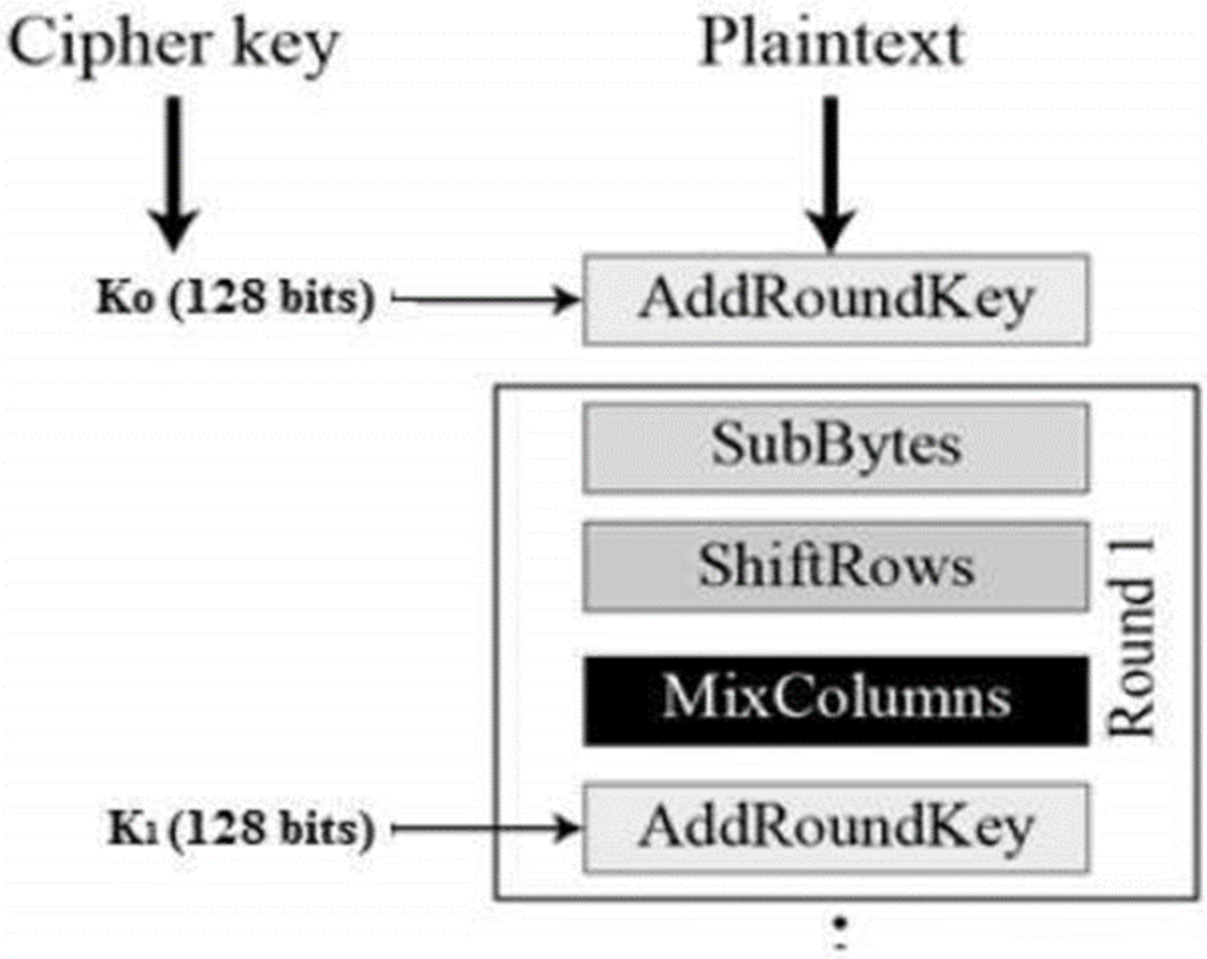


R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below −

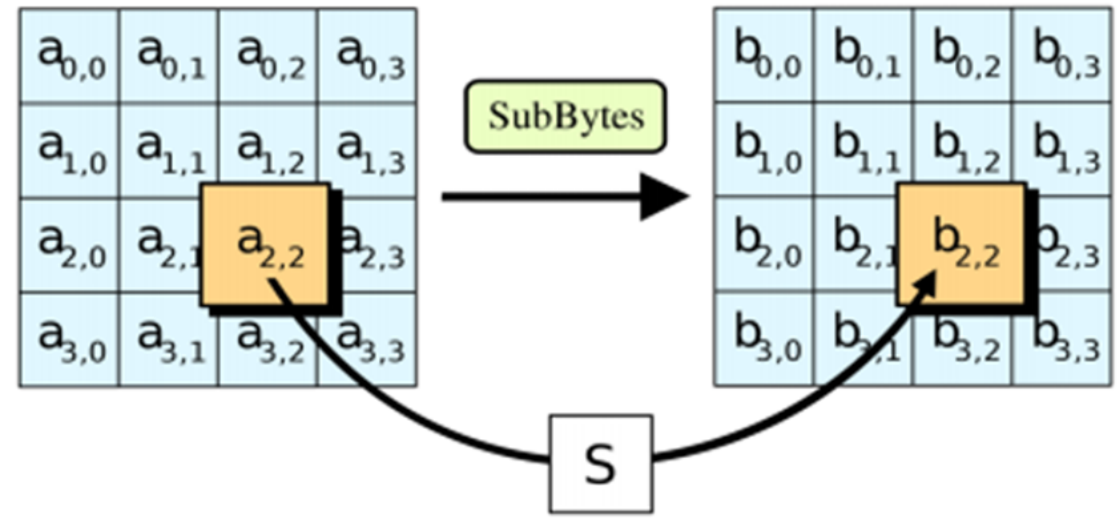


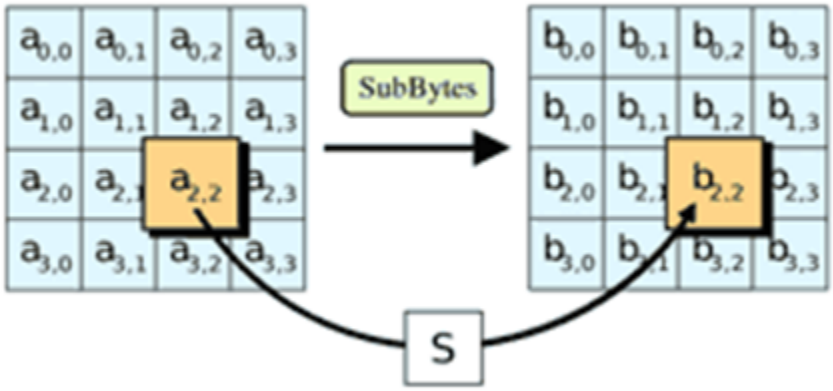
Steps in each round

Each round in the algorithm consists of four steps.

1.Substitution of the bytes

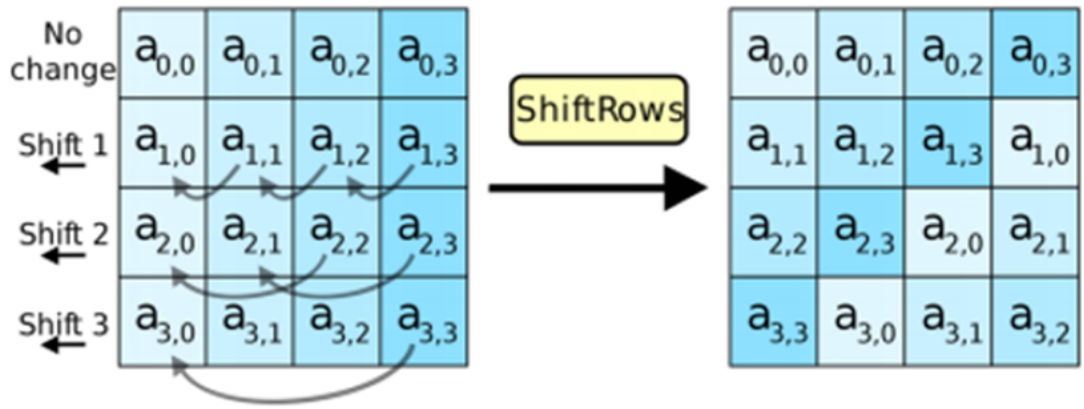
In the first step, the bytes of the block text are substituted based on rules dictated by predefined S-boxes (short for substitution boxes).

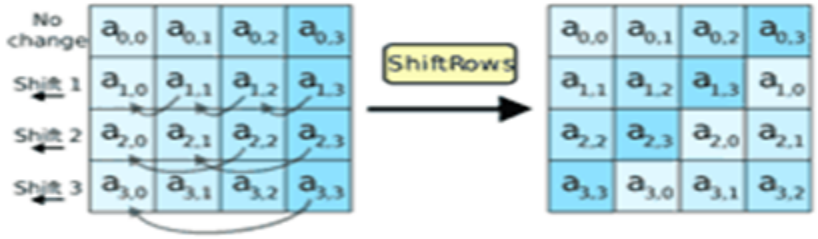




2. Shifting the rows

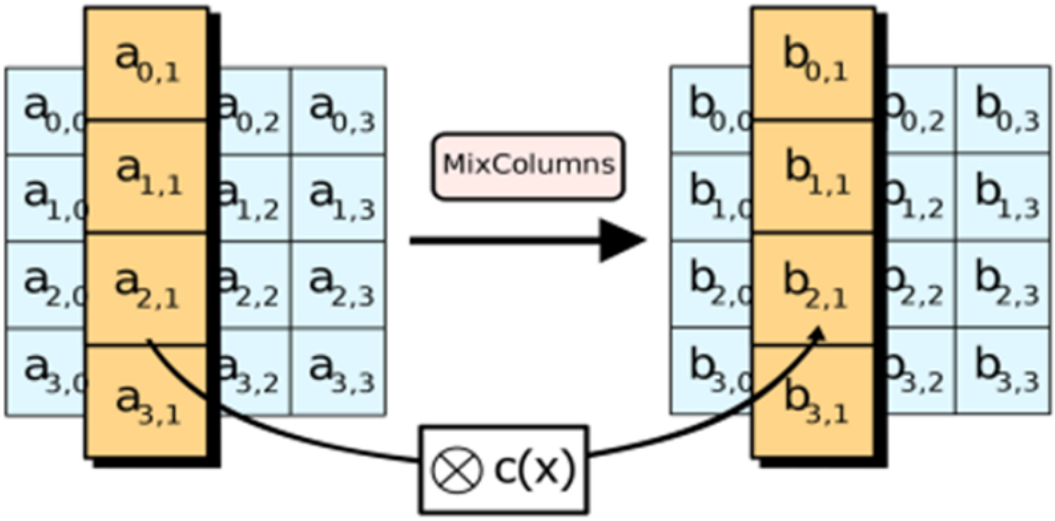
Next comes the permutation step. In this step, all rows except the first are shifted by one, as shown below.





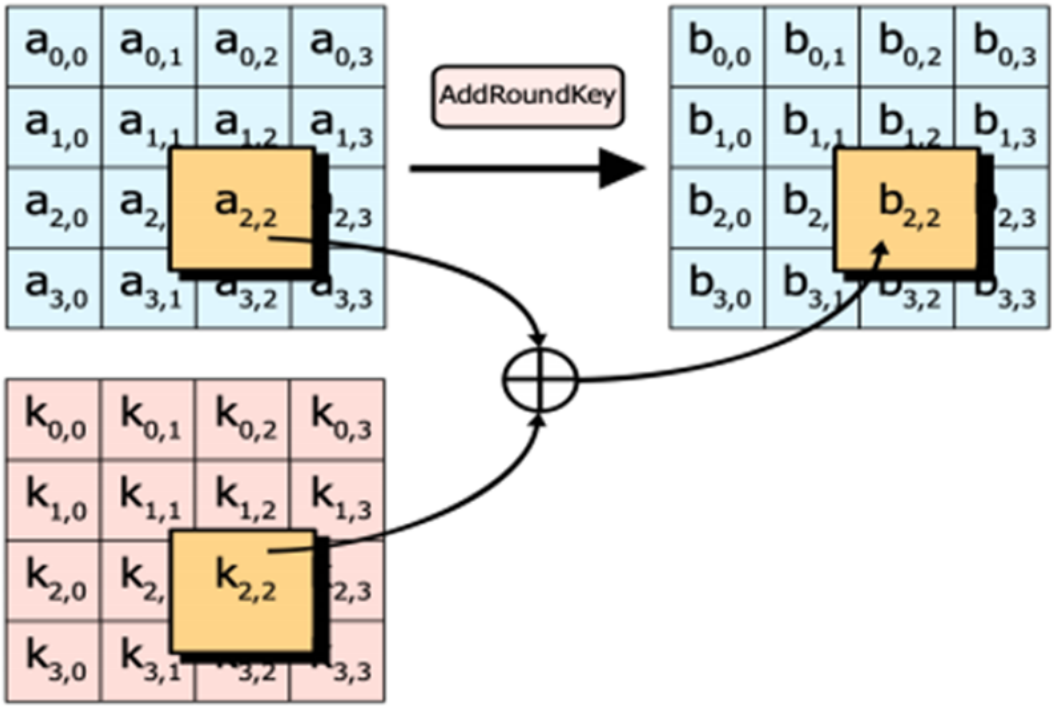
3. Mixing the columns

In the third step, the Hill cipher is used to jumble up the message more by mixing the block's columns



4. Adding the round key

In the final step, the message is XORed with the respective round key.



When done repeatedly, these steps ensure that the final ciphertext is secure.

Blowfish Algorithm:

Blowfish is a symmetric-key block cipher, designed in 1993 by Bruce Schneier and included in many cipher suites and encryption products. Blowfish provides a good encryption rate in software, and no

effective cryptanalysis of it has been found to date.

Blowfish has a 64-bit block size and a variable key length from 32 bits up to 448 bits. It is a 16-round Feistel cipher and uses large key-dependent S-boxes.

- Symmetric Key Algorithm
- Block cipher algo(64 bits)
- Encryption Technique
- Designed By Bruce Schneir(1993)
- It's an alternative to DES encryption technique.

Properties:

- It’s faster compared to DES
- Its compact(ie; it executes in limited/less memory)

- Simple:operations are performed using XOR & AND operation
- Secure: Variable Key length[the key size varies ,it has no fixed length like AES,DES etc]

SUMMERY:

Block Size/Plain Text in Block------64 bits
Key Size---- Variable(32 to 448)bits
No of Subkeys-----18(placed in a P-array[p0,p1,p2,p3…….p17] / [p1,p2,p3,p4…..p18] 32 bits each
No of Rounds----16
No of substitution boxes(s-boxes)-----4 (s-boxes) having 256 entries each will have 32 bits.

1)Generation of Sub-Keys:

*18 sub keys are used for encryption as well as decryption

These 18 sub keys are placed in a P-array[p0,p1,p2,p3…….p17] / [p1,p2,p3,p4…..p18])with each element being 32 bit entry

p[0]=”c5123896”

p[1]=”facd2345”

. [Hexadecimal representation of each sub key]

. [1 hexadecimal digit = 4 bits] [4*8= 32 bits]

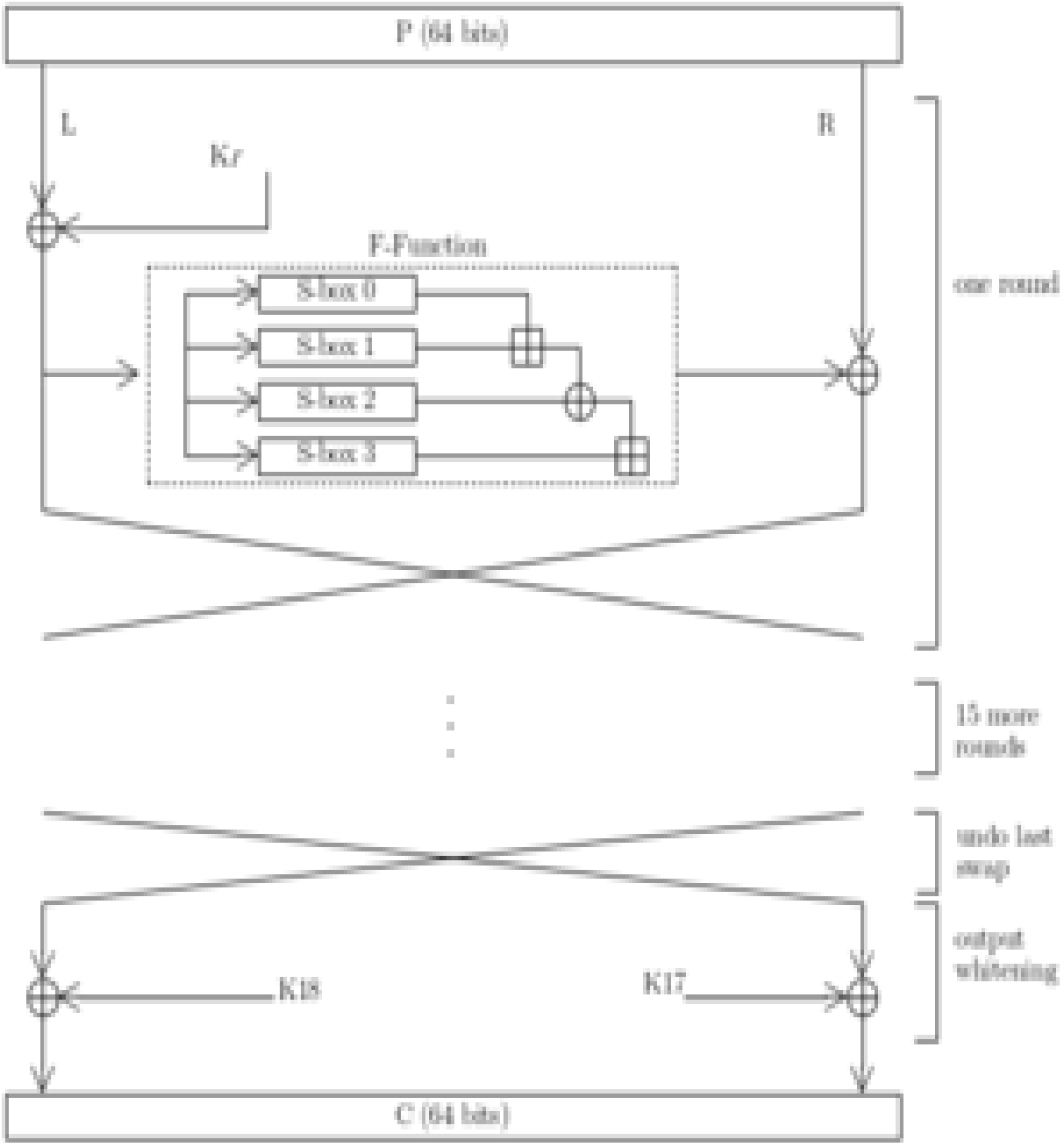
.

p[17]=”cg789654”

2)Key Generation

3)Initialize substitution boxes(s-boxes)

4) Encryption:



P=Plaintext; C=Ciphertext; K*x* = P-array-entry *x*

⊕ = xor ⊞ = addition mod 2^32

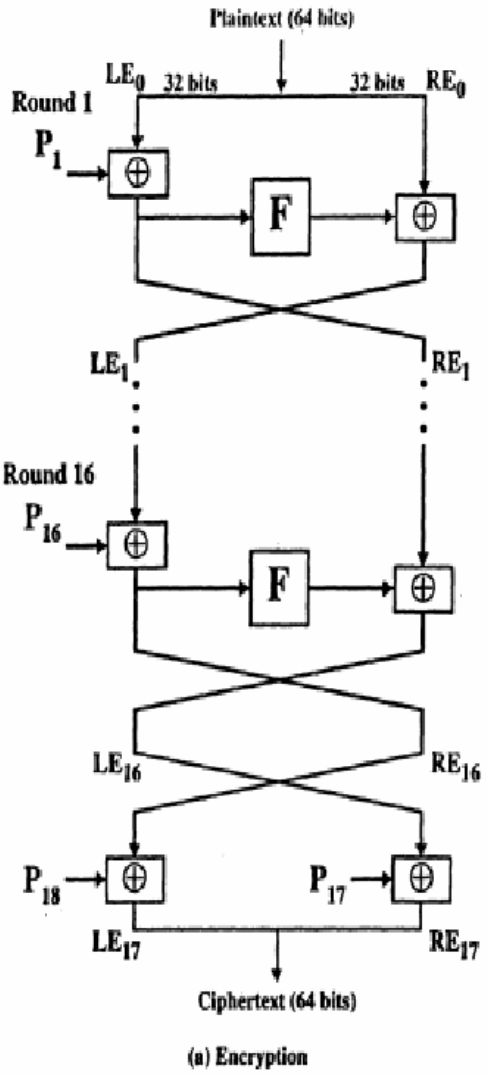
The above diagram shows Blowfish's encryption routine. Each line represents 32 bits. There are five subkey-arrays: one 18-entry P-array (denoted as K in the diagram, to avoid confusion with the Plaintext) and four 256-entry S-boxes (S0, S1, S2 and S3).

Every round *r* consists of 4 actions:

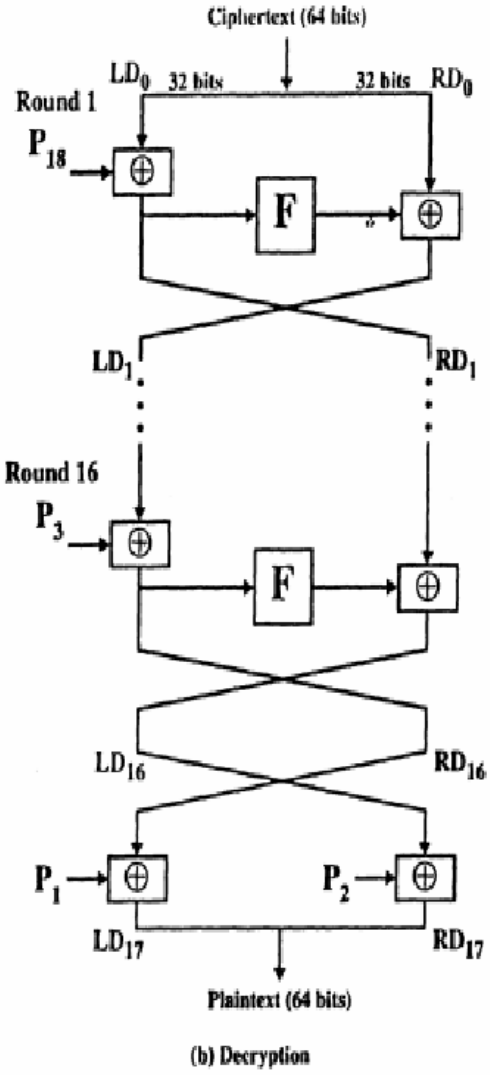
Action 1	XOR the left half (L) of the data with the *r* th P-array entry
Action 2	Use the XORed data as input for Blowfish's F-function

Action 3	XOR the F-function's output with the right half (R) of the data
Action 4	Swap L and R

The F-function splits the 32-bit input into four 8-bit quarters and uses the quarters as input to the S-boxes. The S-boxes accept 8-bit input and produce 32-bit output. The outputs are added modulo 2^{32} and XORed to produce the final 32-bit output

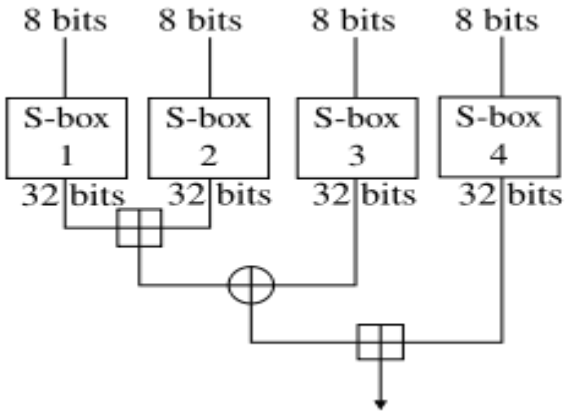


(a) Encryption



(b) Decryption

Function operation:



DIFFIE HELLMAN EXCHANGE KEY

- It is not an encryption Algorithm
- It is used to exchange Secret Key or Symmetric Key
- It uses ASYMMETRIC ENCRYPTION,ie; in order to send the secret key between sender and receiver we will use Public Key and Private Key.
- Here, Public key and Private key will be generated at both sides

STEPS:

1. Assume or select a Prime number ‘q’.
2. Select ‘α’ such that α should be a primitive root of q and (α<q)[alpha less than q]
3. After selecting α, (****v have to find Primitive Root)

Assume, at sender side

X_A (Private Key) (X_A < q)

---------private key of user A

Y_A(Public Key) [$Y_A = \alpha^{X_A} \bmod q$]

Assume , at receiver side

X_B(Private Key) (X_B<q)------ private key of user B

$Y_B = \alpha^{X_B} \bmod q$ Public key of B

Now v have q,α,Public and Private key of sender and receiver

4. Now v have to generate the KEY - K

5. The Key is going to be generated at user A and user B

USER A

USER B

$K = (Y_B)^{X}_{A} \mod q$

$K=(Y_A)^{X}_{B} \mod q$

******Finding Primitive Root:

Suppose, ‘a’ is primitive root of P

If, a mod p, a^2 mod p, a^3mod p……..a^{p-1}mod p gives the results 1,2,3,4,.....p-1.

[v have to get all the values from 1 to p-1 and the values must not repeat]

Example:

Let's choose a prime number
Consider a prime number q=11
Now v have to find out the primitive root for 11 ie; *α* [*α* mod q] where *α* is less than q]

Power →	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	5	10	9	7	3	6	1
3	**3**	**9**	**5**	**4**						
4										

5										
6										
7										
8										
9										
10										
Num↑										

Primitive root for 11 are 2,5,7 etc
From 2,5,7 we have to choose only one number as a primitive root,

So,lets select α = 2
Now v have q=11
α = 2
Now, Select X_A private key of user A which should be less than q
X_A=8

Now calculate Y_A which is Public Key of USER A

$$Y_A = \alpha^{X_A} \bmod q$$

$$Y_A = 2^8 \bmod 11$$
$$Y_A = 256 \bmod 11$$
$$Y_A = 3 \text{ [public key of A]}$$

Now calculate Y_B which is Public Key of USER B lets take X_B=4

$$Y_B = \alpha^{X_B} \bmod q$$

$$Y_B = 2^4 \bmod 11$$
$$Y_B = 16 \bmod 11$$
$$Y_B = 5 \text{ [public key of B]}$$

Now v have to generate the secret key at sender and receiver side

A

B

SENDER

RECEIVER

$K= (Y_B)^{X}_{A} \mod q$

$K=(Y_A)^{X}_{B} \mod q$

$k= 5^8 \mod 11$

$K= 3^4 \mod 11$

$K=390625 \mod 11$

$k=4$

$K=81 \mod 11$

$K=4$

Description of Blowfish

Blowfish is a block cipher that encrypts data in 8-byte blocks. The algorithm

consists of two parts: a key-expansion part and a data-encryption part. Key expansion converts a variable-length key of at most 56 bytes (448 bits) into several subkey arrays totaling 4168 bytes. (Note: the description in this article differs slightly from the one in the April 1994 issue of Dr. Dobb's Journal; there were typos in steps (5) and (6) of the subkey generation algorithm.)

Blowfish has 16 rounds. Each round consists of a key-dependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round.

Subkeys:

Blowfish uses a large number of subkeys. These keys must be precomputed before any data encryption or decryption. The P-array consists of 18 32-bit subkeys: P1, P2,..., P18. There are also four 32-bit S-boxes with 256 entries each: S1,0, S1,1,..., S1,255; S2,0, S2,1,..,, S2,255; S3,0, S3,1,..., S3,255; S4,0, S4,1,..,, S4,255.

Encryption and Decryption:

Blowfish has 16 rounds. The input is a 64-bit data element, x. Divide x into two 32-bit halves: xL, xR. Then, for i = 1 to 16:

xL = xL XOR Pi

xR = F(xL) XOR xR

Swap xL and xR

After the sixteenth round, swap xL and xR again to undo the last swap. Then, xR = xR XOR P17 and xL = xL XOR P18. Finally, recombine xL and xR to get the ciphertext.

Function F looks like this: Divide xL into four eight-bit quarters: a, b, c, and d. Then, F(xL) = ((S1,a + S2,b mod 232) XOR S3,c) + S4,d mod 232.

Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order.

Generating the Subkeys:

The subkeys are calculated using the Blowfish algorithm:

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the

hexadecimal digits of pi (less the initial 3): P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, etc.

2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)

3. Encrypt the all-zero string with the Blowfish algorithm, using the subkeys described in steps (1) and (2).

4. Replace P1 and P2 with the output of step (3).

5. Encrypt the output of step (3) using the Blowfish algorithm with the modified subkeys.

6. Replace P3 and P4 with the output of step (5).

7. Continue the process, replacing all entries of the P array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm.

In total, 521 iterations are required to generate all required subkeys. Applications can store the subkeys rather than execute this derivation process multiple times.

https://www.angelfire.com/moon/dmp/ link for blowfish

SHA-1(Secure Hash Algorithms)

In cryptography, SHA-1 (Secure Hash Algorithm 1) is a cryptographically brokenbut still widely used hash function which takes an input and produces a 160-bit (20-byte) hash value known as a message digest – typically rendered as a hexadecimal number, 40 digits long. It was designed by the United States National Security Agency,

and is a U.S. Federal Information Processing Standard

SHA stands for secure hashing algorithm. SHA is a modified version of MD5 and used for hashing data and certificates.

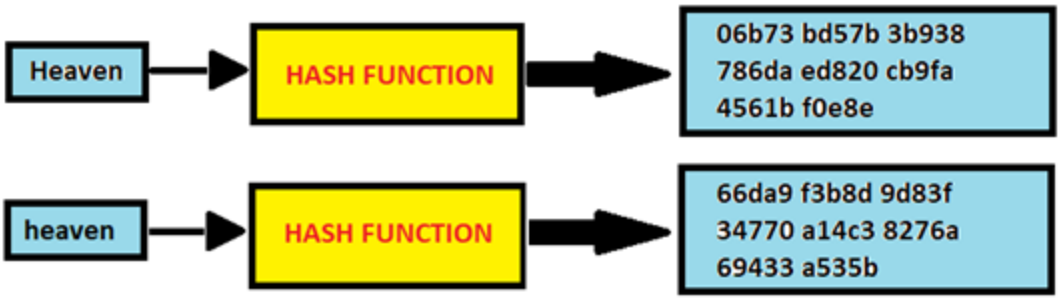
A hashing algorithm shortens the input data into a smaller form that cannot be understood by using bitwise operations, modular additions, and compression functions. You may be wondering, can hashing be cracked or decrypted?

Hashing is similar to encryption, the only difference between hashing and encryption is that hashing is one-way,

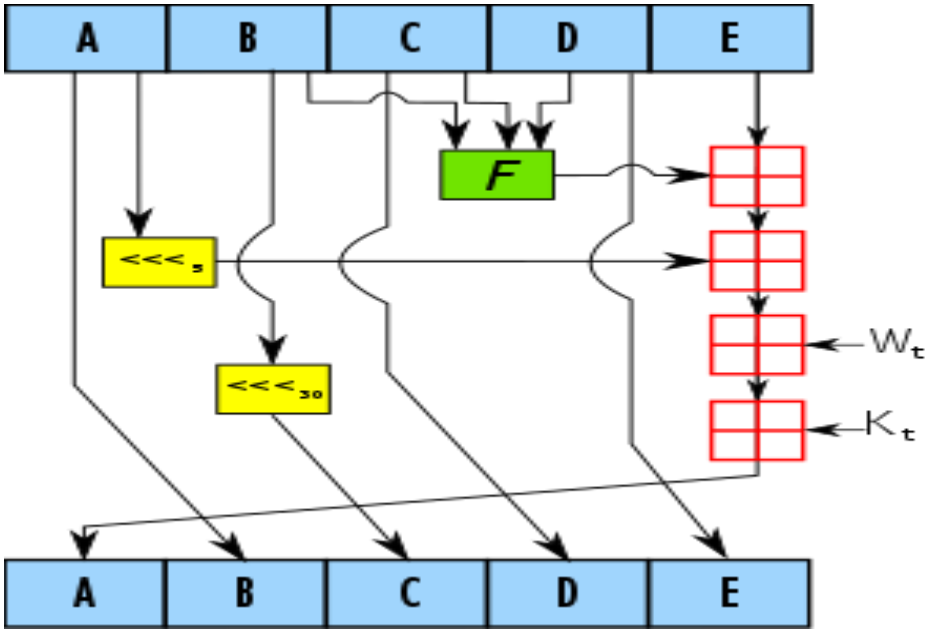
meaning once the data is hashed, the resulting hash digest cannot be cracked,

unless a brute force attack is used. See the image below for the working of SHA algorithm. SHA works in such a way even if a single character of the message changed, then it will generate a different hash.

For example, hashing of two similar, but different messages i.e., Heaven and heaven is different. However, there is only a difference of a capital and small letter.



The initial message is hashed with SHA-1, resulting in the hash digest "06b73bd57b3b938786daed820cb9fa4561bf0e8e". If the second, similar, message is hashed with SHA-1, the hash digest will look like "66da9f3b8d9d83f34770a14c38276a69433a535b". This is referred to as the avalanche effect. This effect is important in cryptography, as it means even the slightest change in the input message completely changes the output.



One iteration within the SHA-1 compression function:

A, B, C, D and E are 32-bit words of the state;

F is a nonlinear function that varies;

{\displaystyle \lll _{n}}

$\lll_n$ denotes a left bit rotation by *n* places;

***n* varies for each operation;**
W_t is the expanded message word of round t;
K_t is the round constant of round t;
$\boxplus$ denotes addition modulo 2^{32}.

- SHA-1 is not an encryption Algorithm .
- It's a HASHING Algorithm.
- The difference between **encryption** and **hashing**: Encrypted data can be decrypted to its original form because during encryption no information is lost, just represented differently. A hash can not be reversed to the original data because that information is lost during the hash computation .
- The goal of a hash is to have a very small "string" that merely represents the original data and can be compared to

another hash in order to determine whether or not the source data for these hashes was probably identical or not.

- It works by transforming the data using a hash function: an algorithm that consists of bitwise operations, modular additions, and compression functions. The hash function then produces a fixed-size string that looks nothing like the original. These algorithms are designed to be one-way functions, meaning that once they're transformed into their respective hash values, it's virtually impossible to transform them back into the original data.
- A common application of SHA is to encrypting passwords, as the server side only needs to keep track of a specific user's hash value, rather than the actual password. This is helpful in

case an attacker hacks the database, as they will only find the hashed functions and not the actual passwords, so if they were to input the hashed value as a password, the hash function will convert it into another string and subsequently deny access.

- Secure Hash Algorithm 1, or SHA-1, was developed in 1993 by the U.S. government's standards agency National Institute of Standards and Technology (NIST). It is widely used in security applications and protocols, including TLS, SSL, PGP, SSH, IPsec, and S/MIME.

Wireshark

https://www.javatpoint.com/wireshark

Wireshark is an open-source packet analyzer, which is used for **education, analysis, software development, communication protocol development, and network troubleshooting**.

It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a **sniffer, network protocol analyzer, and network analyzer**. It is also used by network security engineers to examine security problems.

Wireshark is a free to use application which is used to apprehend the data back and forth. It is often called as a free packet sniffer computer application. It puts the network card into an unselective mode, i.e., to accept all the packets which it receives.

Uses of wireshark

Wireshark can be used in the following ways:

1. It is used by network security engineers to examine security problems.
2. It allows the users to watch all the traffic being passed over the network.
3. It is used by network engineers to troubleshoot network issues.

4. It also helps to troubleshoot latency issues and malicious activities on your network.
5. It can also analyze dropped packets.
6. It helps us to know how all the devices like laptop, mobile phones, desktop, switch, routers, etc., communicate in a local network or the rest of the world.

What is a packet?

A packet is a unit of data which is transmitted over a network between the origin and the destination. Network packets are small, i.e., maximum **1.5 Kilobytes for Ethernet packets and 64 Kilobytes for IP packets**. The data packets in the Wireshark

can be viewed online and can be analyzed offline.

Functionality of Wireshark:

Wireshark is similar to tcpdump in networking. **Tcpdump** is a common packet analyzer which allows the user to display other packets and TCP/IP packets, being transmitted and received over a network attached to the computer. It has a graphic end and some sorting and filtering functions. Wireshark users can see all the traffic passing through the network.

Wireshark can also monitor the unicast traffic which is not sent to the network's MAC address interface. But, the switch does not pass all the traffic to the port. Hence, the promiscuous mode is not sufficient to see all the traffic. The various

network taps or **port mirroring** is used to extend capture at any point.

Port mirroring is a method to monitor network traffic. When it is enabled, the switch sends the copies of all the network packets present at one port to another port

What is color coding in Wireshark?

The packets in the Wireshark are highlighted with **blue**, **black**, and **green color**. These colors help users to identify the types of traffic. It is also called as **packet colorization**.

Features of Wireshark

- It is multi-platform software, i.e., it can run on Linux, Windows, OS X, FreeBSD, NetBSD, etc.

- It is a standard three-pane packet browser.
- It performs deep inspection of the hundreds of protocols.
- It often involves live analysis, i.e., from the different types of the network like the Ethernet, loopback, etc., we can read live data.
- It has sort and filter options which makes ease to the user to view the data.
- It is also useful in VoIP analysis.
- It can also capture raw USB traffic.
- Various settings, like timers and filters, can be used to filter the output.

- It can only capture packet on the PCAP (an application programming interface used to capture the network) supported networks.
- Wireshark supports a variety of well-documented capture file formats such as the PcapNg and Libpcap. These formats are used for storing the captured data.
- It is the no.1 piece of software for its purpose. It has countless applications ranging from the **tracing down, unauthorized traffic, firewall settings, etc**

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 of Administrator 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.

Rootkits

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.

GMER is an application that detects and removes rootkits .

It scans for:

hidden processes
hidden threads

hidden modules
hidden services
hidden files
hidden disk sectors (MBR)
hidden Alternate Data Streams
hidden registry keys

drivers hooking(steal,snatch,informal)
SSDT(System Service Descriptor Table, an internal data structure within Microsoft Windows)

Answer
Hooking. Modification of the SSDT **allows to redirect syscalls to routines outside the kernel**. These routines can be either used to hide the presence of software or to act as a backdoor to allow attackers

permanent code execution with kernel privileges.

drivers hooking IDT (Interrupt Descriptor Table (IDT))
Answer

Interrupt Descriptor Table (**IDT**) is a data structure used by the x86 architecture to implement an interrupt vector table. The IDT is used by the processor to determine the correct response to interrupts and exceptions.

drivers hooking IRP calls

I/O request packets (IRPs) are **kernel mode structures that are used by Windows Driver Model (WDM) and Windows NT device drivers to**

communicate with each other and with the operating system.

inline hooks

Answer

Inline Hooks are **outbound calls from Okta to your own custom code, triggered at specific points in Okta process flows**. They allow you to integrate custom functionality into those flows. You implement your custom code as a web service with an Internet-accessible endpoint.

PROCEDURE:

STEP-1: Download Rootkit Tool from GMER website 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.

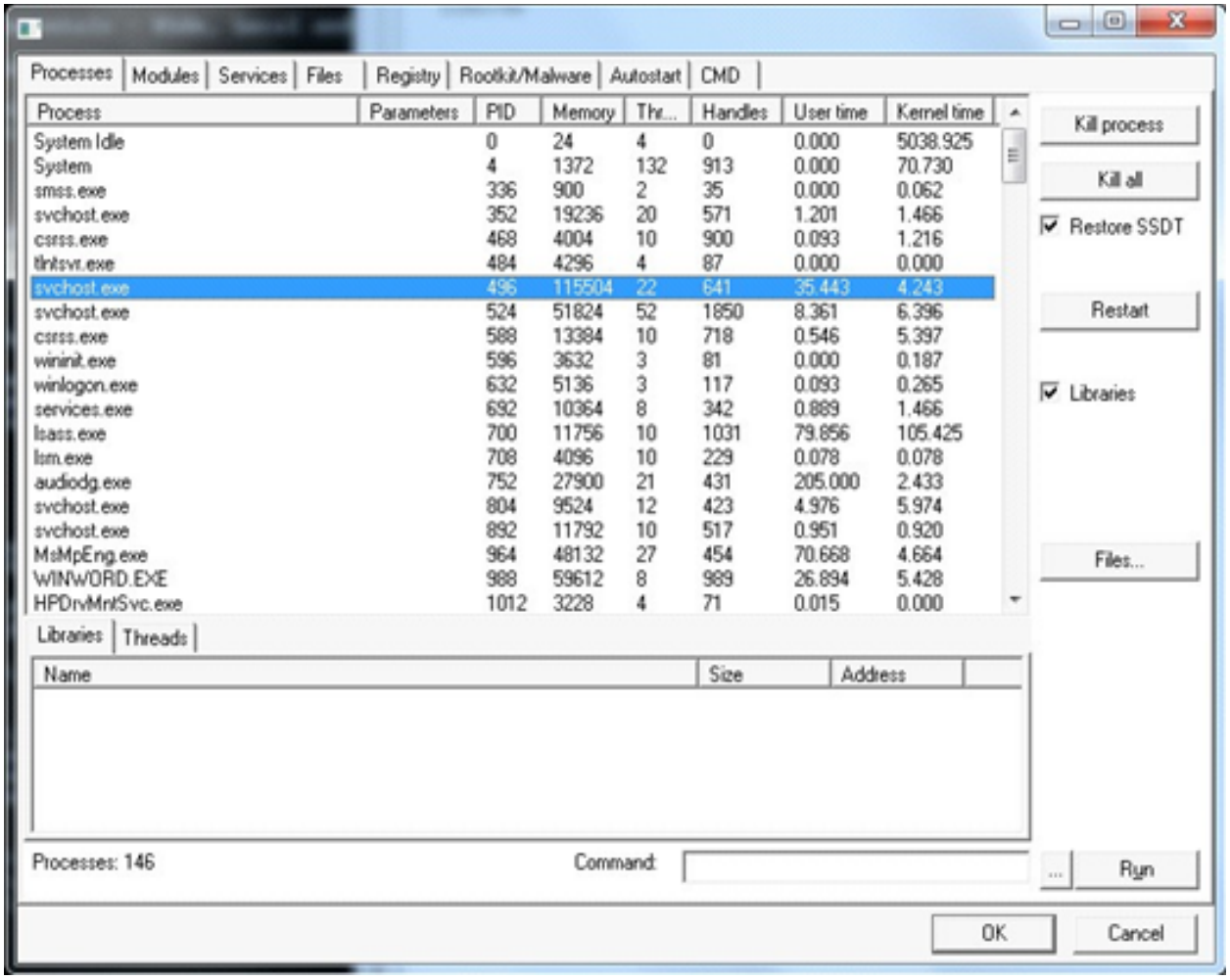
STEP-6: Files menu displays full files on Hard-Disk volumes.

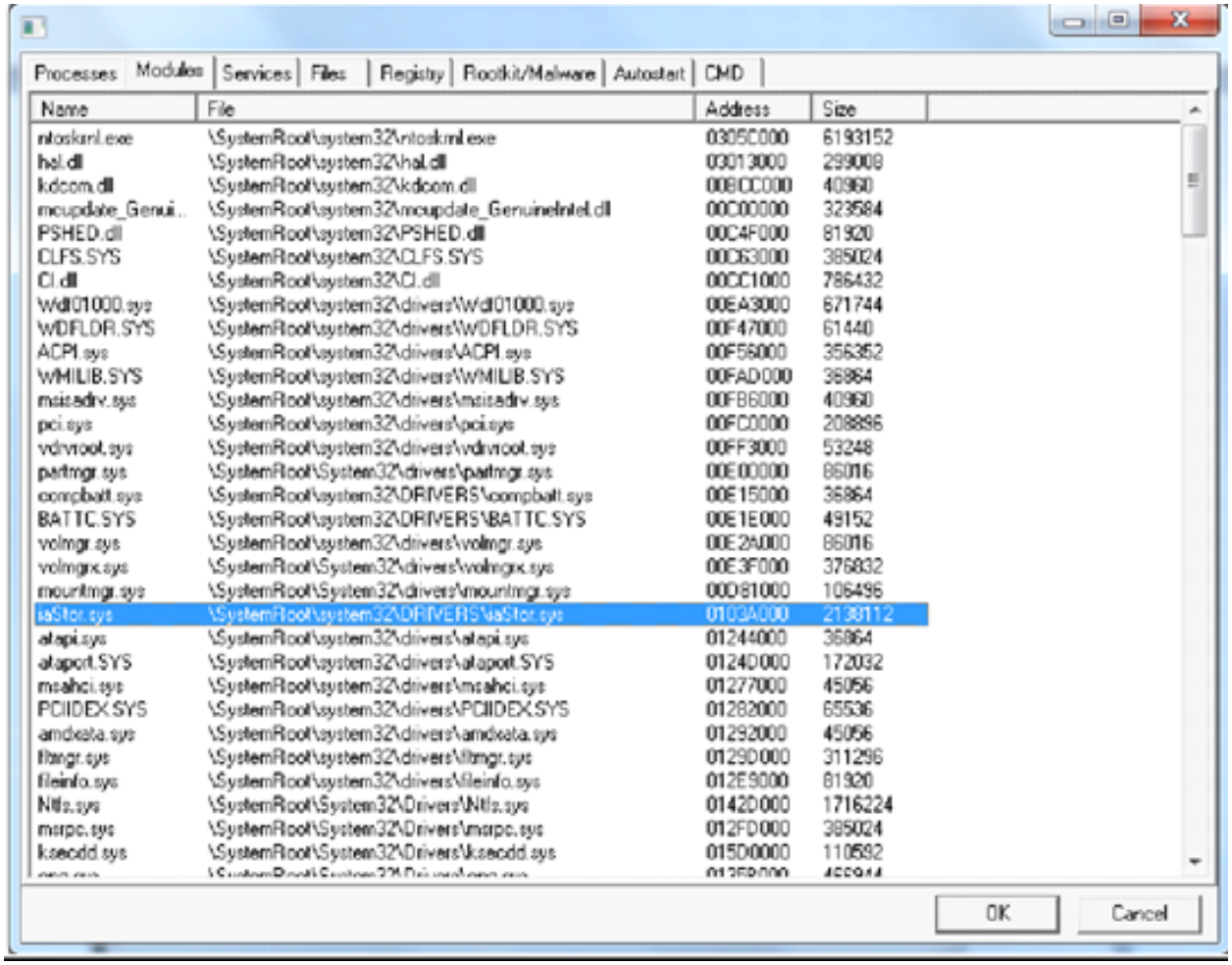
STEP-7: Registry displays Hkey_Current_user and Hkey_Local_Machine.

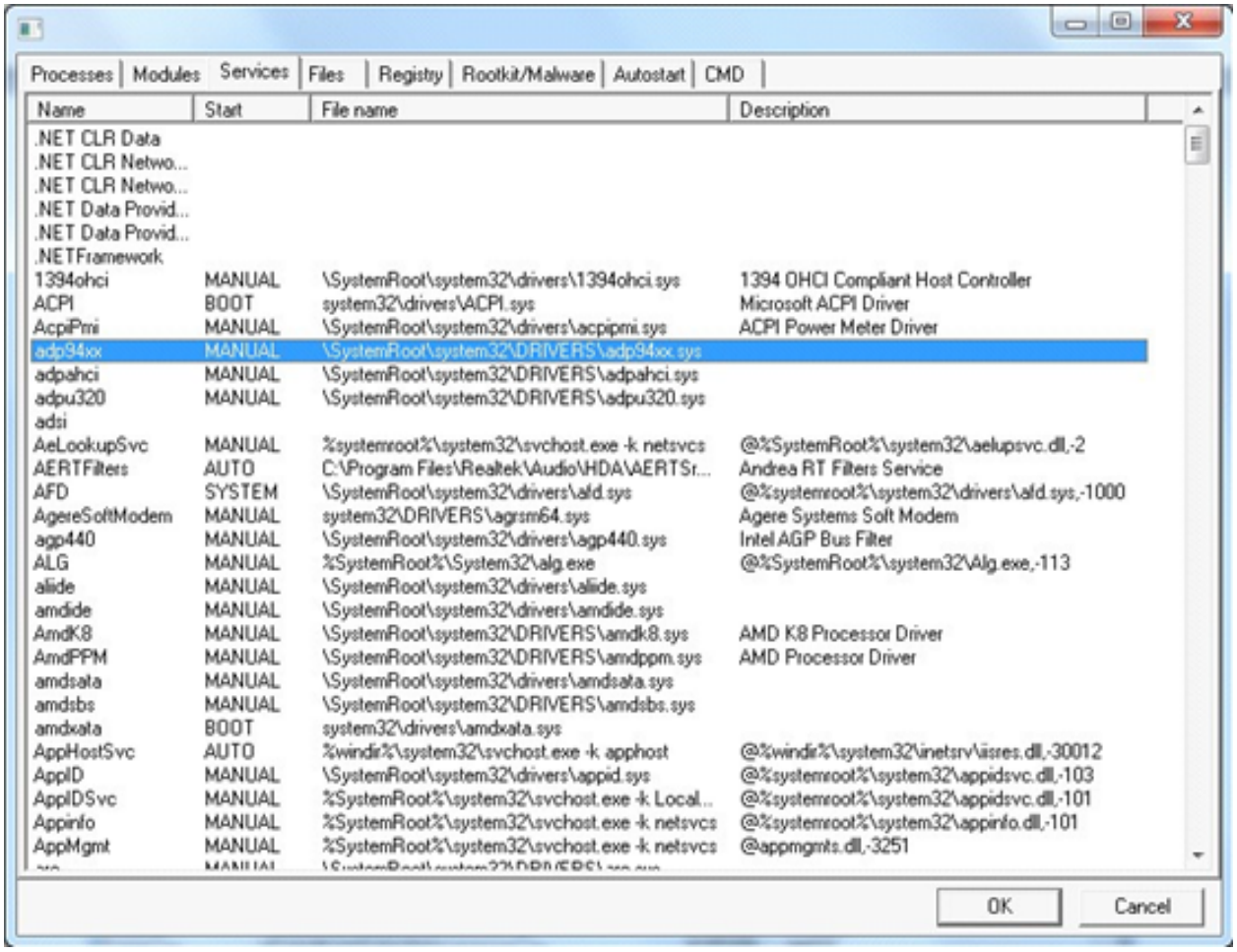
STEP-8: Rootkits / Malwares scans the local drives selected.

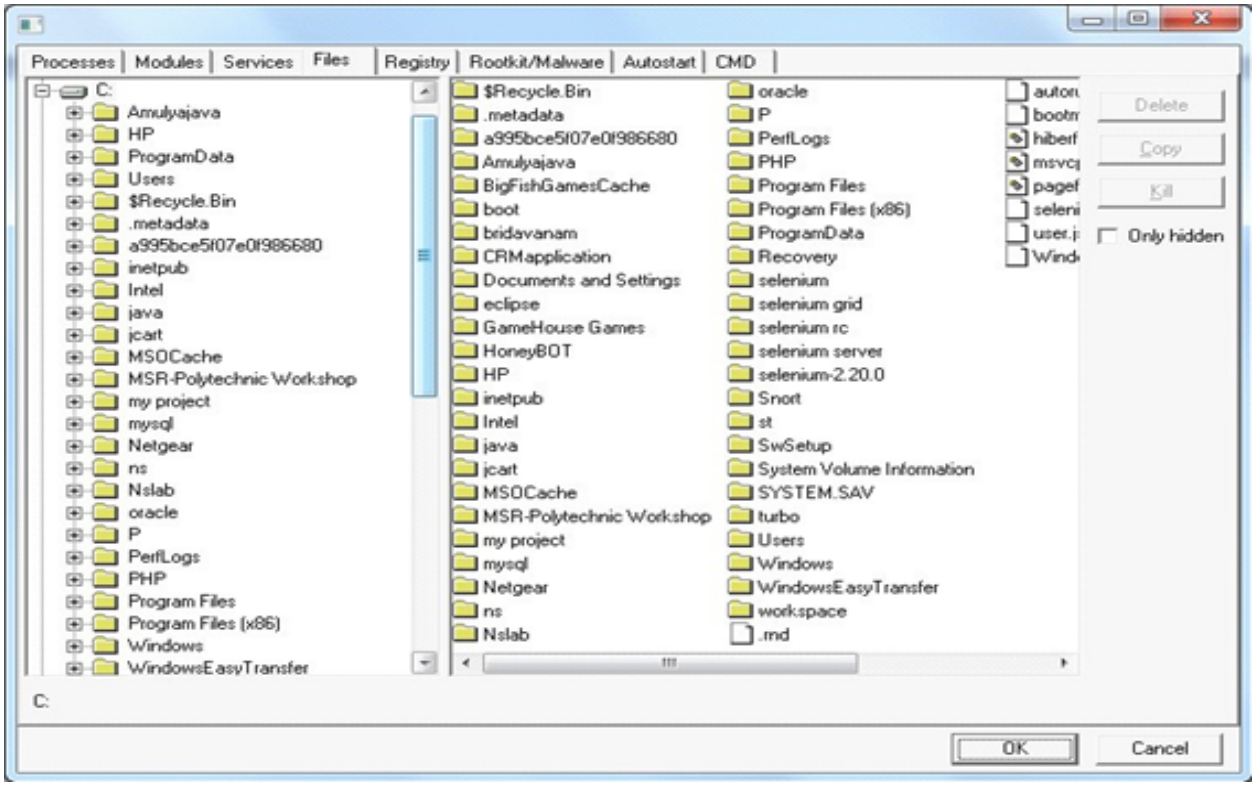
STEP-9: Autostart displays the registry base Autostart applications.

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


Processes
Modules
Services
Files
Registry
Rootkit/Malware
Autostart
CMD
Process
Parameters
PID
Memory
Thr...
Handles
User time
Kernel time
System Idle 0 24 4 0 0.000 5038.925
System 4 1372 132 913 0.000 70.730
smss.exe 336 900 2 35 0.000 0.062
svchost.exe 352 19236 20 571 1.201 1.466
csrss.exe 468 4004 10 900 0.093 1.216
tlntsvr.exe 484 4296 4 87 0.000 0.000
svchost.exe 496 115504 22 641 35.443 4.243
svchost.exe 524 51824 52 1650 8.361 6.396
csrss.exe 588 13384 10 718 0.546 5.397
wininit.exe 596 3632 3 81 0.000 0.187
winlogon.exe 632 5136 3 117 0.093 0.265
services.exe 692 10364 8 342 0.889 1.466
lsass.exe 700 11756 10 1031 79.856 105.425
lsm.exe 708 4096 10 229 0.078 0.078
audiodg.exe 752 27900 21 431 205.000 2.433
svchost.exe 804 9524 12 423 4.976 5.974
svchost.exe 892 11792 10 517 0.951 0.920
MsMpEng.exe 964 48132 27 454 70.668 4.664
WINWORD.EXE 988 59612 8 989 26.894 5.428
HPDrvMntSvc.exe 1012 3228 4 71 0.015 0.000
Kill process
Kill all
Restore SSDT
Restart
Libraries
Files...
Libraries
Threads
Name
Size
Address
Processes: 146
Command:
...
Run
OK
Cancel


Processes
Modules
Services
Files
Registry
Rootkit/Malware
Autostart
CMD
Name
File
Address
Size
ntoskrnl.exe
\SystemRoot\system32\ntoskrnl.exe
03050000
6193152
hal.dll
\SystemRoot\system32\hal.dll
03013000
299008
kdcom.dll
\SystemRoot\system32\kdcom.dll
00BCC000
40960
mcupdate_Genui...
\SystemRoot\system32\mcupdate_GenuineIntel.dll
00C00000
323584
PSHED.dll
\SystemRoot\system32\PSHED.dll
00C4F000
81920
CLFS.SYS
\SystemRoot\system32\CLFS.SYS
00C63000
385024
CI.dll
\SystemRoot\system32\CI.dll
00CC1000
786432
Wdf01000.sys
\SystemRoot\system32\drivers\Wdf01000.sys
00EA3000
671744
WDFLDR.SYS
\SystemRoot\system32\drivers\WDFLDR.SYS
00F47000
61440
ACPI.sys
\SystemRoot\system32\drivers\ACPI.sys
00F56000
356352
WMILIB.SYS
\SystemRoot\system32\drivers\WMILIB.SYS
00FAD000
36864
msisadrv.sys
\SystemRoot\system32\drivers\msisadrv.sys
00FB6000
40960
pci.sys
\SystemRoot\system32\drivers\pci.sys
00FC0000
208896
vdrvroot.sys
\SystemRoot\system32\drivers\vdrvroot.sys
00FF3000
53248
partmgr.sys
\SystemRoot\System32\drivers\partmgr.sys
00E00000
86016
compbatt.sys
\SystemRoot\system32\DRIVERS\compbatt.sys
00E15000
36864
BATTC.SYS
\SystemRoot\system32\DRIVERS\BATTC.SYS
00E1E000
49152
volmgr.sys
\SystemRoot\system32\drivers\volmgr.sys
00E2A000
86016
volmgrx.sys
\SystemRoot\System32\drivers\volmgrx.sys
00E3F000
376832
mountmgr.sys
\SystemRoot\System32\drivers\mountmgr.sys
00D81000
106496
iaStor.sys
\SystemRoot\system32\DRIVERS\iaStor.sys
0103A000
2138112
atapi.sys
\SystemRoot\system32\drivers\atapi.sys
01244000
36864
ataport.SYS
\SystemRoot\system32\drivers\ataport.SYS
0124D000
172032
msahci.sys
\SystemRoot\system32\drivers\msahci.sys
01277000
45056
PCIIDEX.SYS
\SystemRoot\system32\drivers\PCIIDEX.SYS
01282000
65536
amdxata.sys
\SystemRoot\system32\drivers\amdxata.sys
01292000
45056
fltmgr.sys
\SystemRoot\system32\drivers\fltmgr.sys
0129D000
311296
fileinfo.sys
\SystemRoot\system32\drivers\fileinfo.sys
012E9000
81920
Ntfs.sys
\SystemRoot\System32\Drivers\Ntfs.sys
0142D000
1716224
msrpc.sys
\SystemRoot\System32\Drivers\msrpc.sys
012FD000
385024
ksecdd.sys
\SystemRoot\System32\Drivers\ksecdd.sys
015D0000
110592
OK
Cancel


Processes
Modules
Services
Files
Registry
Rootkit/Malware
Autostart
CMD
Name
Start
File name
Description
.NET CLR Data
.NET CLR Netwo...
.NET CLR Netwo...
.NET Data Provid...
.NET Data Provid...
.NETFramework
1394ohci MANUAL \SystemRoot\system32\drivers\1394ohci.sys 1394 OHCI Compliant Host Controller
ACPI BOOT system32\drivers\ACPI.sys Microsoft ACPI Driver
AcpiPmi MANUAL \SystemRoot\system32\drivers\acpipmi.sys ACPI Power Meter Driver
adp94xx MANUAL \SystemRoot\system32\DRIVERS\adp94xx.sys
adpahci MANUAL \SystemRoot\system32\DRIVERS\adpahci.sys
adpu320 MANUAL \SystemRoot\system32\DRIVERS\adpu320.sys
adsi
AeLookupSvc MANUAL %systemroot%\system32\svchost.exe -k netsvcs @%SystemRoot%\system32\aelupsvc.dll,-2
AERTFilters AUTO C:\Program Files\Realtek\Audio\HDA\AERTSr... Andrea RT Filters Service
AFD SYSTEM \SystemRoot\system32\drivers\afd.sys @%systemroot%\system32\drivers\afd.sys,-1000
AgereSoftModem MANUAL system32\DRIVERS\agrsm64.sys Agere Systems Soft Modem
agp440 MANUAL \SystemRoot\system32\drivers\agp440.sys Intel AGP Bus Filter
ALG MANUAL %SystemRoot%\System32\alg.exe @%SystemRoot%\system32\Alg.exe,-113
aliide MANUAL \SystemRoot\system32\drivers\aliide.sys
amdide MANUAL \SystemRoot\system32\drivers\amdide.sys
AmdK8 MANUAL \SystemRoot\system32\DRIVERS\amdk8.sys AMD K8 Processor Driver
AmdPPM MANUAL \SystemRoot\system32\DRIVERS\amdppm.sys AMD Processor Driver
amdsata MANUAL \SystemRoot\system32\drivers\amdsata.sys
amdsbs MANUAL \SystemRoot\system32\DRIVERS\amdsbs.sys
amdxata BOOT system32\drivers\amdxata.sys
AppHostSvc AUTO %windir%\system32\svchost.exe -k apphost @%windir%\system32\inetsrv\iisres.dll,-30012
AppID MANUAL \SystemRoot\system32\drivers\appid.sys @%systemroot%\system32\appidsvc.dll,-103
AppIDSvc MANUAL %SystemRoot%\system32\svchost.exe -k Local... @%systemroot%\system32\appidsvc.dll,-101
Appinfo MANUAL %SystemRoot%\system32\svchost.exe -k netsvcs @%systemroot%\system32\appinfo.dll,-101
AppMgmt MANUAL %SystemRoot%\system32\svchost.exe -k netsvcs @appmgmts.dll,-3251
OK
Cancel



GMER is an application that detects and removes rootkits .

It scans for:

1.hidden processes
2.hidden threads
3.hidden modules
4.hidden services

5.hidden files
6.hidden disk sectors (MBR)
7.hidden Alternate Data Streams
8.hidden registry keys
9.drivers hooking SSDT
10. drivers hooking IDT
11. drivers hooking IRP calls
12. inline hooks

Frequently Asked Questions

Question: *Do I have a rootkit?*

Answer: You can scan the system for rootkits using GMER. Run **gmer.exe**, select **Rootkit** tab and click the "Scan" button. If you don't know how to interpret the output, please **Save** the log and send it to my email address.

Warning ! Please, do not select the

"Show all" checkbox during the scan.

Question: *How to create "3rd party" log ?*

Answer: Tick **"3rd party"** option and then click the "Scan" button. After the scan you can use "Remove signed" and "Remove duplicates" options to filter the scan results.

Question: *How to install the GMER software ?*

Answer: Just run **gmer.exe.** All required files will be copied to the system during the first lanuch.

Question: *How to uninstall/remove the GMER software from my machine ?*

Answer: Just delete the **exe** file.

Question: My computer is infected and GMER won't start:

Answer: Try to rename gmer.exe to iexplore.exe and then run it.

Question: How do I remove the Rustock rootkit ?

Answer: When GMER detects hidden service click "**Delete the service**" and answer **YES** to all questions.

Question: How do I show all NTFS Streams ?

Answer: On the "Rootkit Tab" select only: Files + ADS + Show all options and then click the Scan button.

Question: Can I launch GMER in Safe Mode ?

Answer: Yes, you can launch GMER in Safe Mode, however rootkits which don't work in Safe Mode won't be detected.

Question: I am confused as to use delete or disable the hidden "service".

Answer: Sometimes "delete the service" option wont work because the rootkit protects its service. So, in such case use: 1) "disable the service", 2) reboot your machine, and 3) "delete the service".

link

http://www.gmer.net/#files

KFSENSOR:

Web link for install

http://www.keyfocus.net/kfsensor/

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. If there are no alerts then green icon is displayed.

INTRODUCTION: HONEY POT:

A honeypot is a computer system that is set up to act as a decoy to lure cyber attackers, and to detect, deflect or study attempts to gain unauthorized access to information systems. Generally, it consists of a computer, applications, and data that simulate the behavior of a real system that appears to be part of a network but is actually isolated and closely monitored. All communications with a honeypot are considered hostile, as there's no reason for legitimate users to access a honeypot. Viewing and logging this activity can provide an insight into the level and types of threat a network infrastructure faces while distracting attackers away from assets of real value.

Honeypots can be classified based on their deployment (use/action) and based on their level

of involvement. Based on deployment, honeypots may be classified as:

1. Production honeypots
2. Research honeypots

Production honeypots are easy to use, capture only limited information, and are used primarily by companies or corporations. Production honeypots are placed inside the production network with other production servers by an organization to improve their overall state of security. Normally, production honeypots are low-interaction honeypots, which are easier to deploy. They give less information about the attacks or attackers than research honeypots.

Research honeypots are run to gather information about the motives and tactics of the Black hat community targeting different networks. These honeypots do not add direct value to a specific organization; instead, they

are used to research the threats that organizations face and to learn how to better protect against those threats.

KF SENSOR:

KFSensor is a Windows based honeypot Intrusion Detection System (IDS). It acts as a honeypot to attract and detect hackers and worms by simulating vulnerable system services and trojans. By acting as a decoy server it can divert attacks from critical systems and provide a higher level of information than can be achieved by using firewalls and NIDS alone. KFSensor is a system installed in a network in order to divert and study an attacker's behavior. This is a new technique that is very effective in detecting attacks.

The main feature of KFSensor is that every connection it receives is a suspect hence it results in very few false alerts. At the heart of KFSensor sits a powerful internet daemon service that is built to handle multiple ports and IP addresses. It is written to resist denial of service and buffer overflow attacks. Building on this flexibility KFSensor can respond to connections in a variety of ways, from simple port listening and basic services (such as echo), to complex simulations of standard system services. For the HTTP protocol KFSensor accurately simulates the way Microsoft's web server (IIS) responds to both valid and invalid requests. As well as being able to host a website it also handles complexities such as range requests and client side cache negotiations. This makes it extremely difficult for an attacker to fingerprint, or identify KFSensor as a honeypot.

KFSensor installs a new system tray (systray) icon in the shape of a siren on the desktop. You click the siren icon to launch the KFSensor monitor, and it is used liberally throughout the program to indicate KFSensor's current status. On the desktop, it will usually be gray, but it will flash red or yellow, based on event activity. The systray icon will flash until you view the KFSensor monitor, although its behavior can be customized. Low-priority events are just logged, and no alert is generated. Medium-priority events alert and make the systray icon flash yellow and orange. High-priority events alert and make the systray icon flash red and orange.

Emulating Services with KFSensor

Each port in the Port view represents a listener. Listeners are attached to actions. Actions can be close, close and read, or call up a simulated

server. The close action will immediately close the connection and log the event. Read and close will wait for the visitor to send a request, and then close the connection without sending a response. A listener can also be attached to a server action.

KFSensor calls emulated services *sim servers*, short for simulated servers. A single instance of KFSensor can have an unlimited number of sim servers defined, although only 256 can be active at once. KFSensor has two types of sim servers: *sim banner* and *sim standard*. Some services, like FTP and SMTP, exist as both sim banner servers and sim standard servers, and listen on TCP or UDP ports, depending on the requirements of the environment.

Sim Banner Servers

Sim banner servers are simple port listeners with the ability to serve up text or encoded data

as a banner in response to a visitor request. Each sim banner server can be edited, and new banner sim servers can be added.

The default sim banner servers include Echo (7), Daytime (13), Quote of the Day (17), Chargen (19), MyDoom worm (3127), Dameware (6129), and the SubSeven trojan (54283).

Sim Standard Servers

Sim standard servers entail a higher level of interaction than a mere one-tim

e banner response. KFSensor currently comes with the following emulated services:

- FTP (Guild, not Microsoft, on port 21)
- Telnet (port 23)
- SMTP (Microsoft Exchange Server 2003 on port 25)

- HTTP (IIS 6.0 and Apache on ports 80, 81, 82, and 83)
- POP3 (Exchange Server on port 110)
- NetBIOS (ports 137, 138, 139, and 445)
- SOCKS Proxy (port 1080)
- Microsoft SQL Server (ports 1433 and 1434)
- SubSeven trojan (ports 2794, 7215, and 27374)
- Hogle SMTP trojan (port 3355)
- Terminal Server (port 3389)
- HTTP Proxy (port 8080)
- VNC (port 5900)

honeypot should be set up just like the real server so that data can appear to be authentic by showing fake files, fake ports, fake directories, etc. As the honeypot creates the illusion of

being legitimate; the attacker tends to believe that they have gained accessed of the real deal.

KFSensor is a honeypot for a windows system. it also acts as an IDS. Its job is to attract and detect all the attackers in the network, hence the name ‘Honeypot’. It does so by imitating a vulnerable environment and disguising itself as a server and it way, it succeeds to not only catch the attacker but also helps to know their motive

KFSensor’s role is to be a decoy server for the attackers in order to protect the real thing.

It does its job perfectly by opening fake ports on the system where it’s installed and gathering the information when a connection is made. It does this in precisely the same way as a routine server program, such as a web server or an SMTP server. By doing this it sets up a target, or

a honeypot server, that will record the activities of an attacker.

Web link : http://www.keyfocus.net/kfsensor/

What is Nmap?

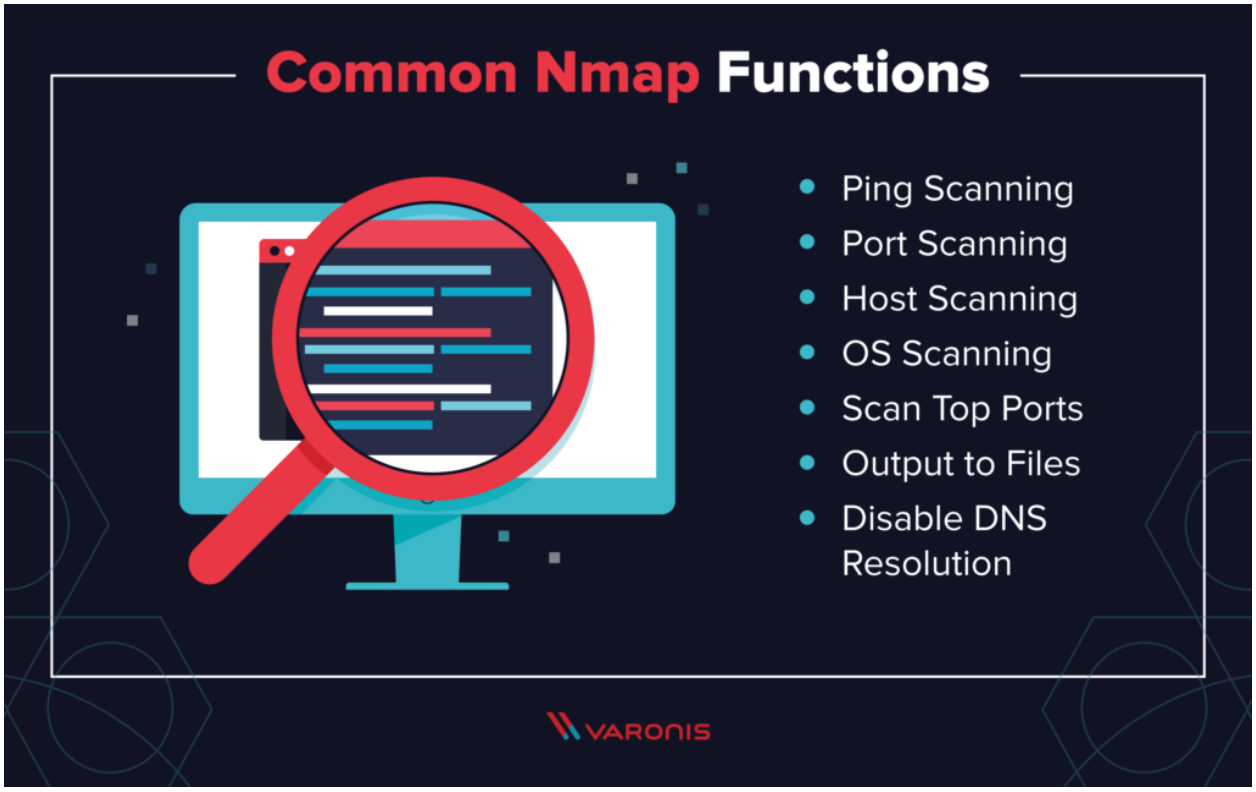
Nmap ,short for Network Mapper, is a network discovery and security auditing tool. It is known

for its simple and easy to remember flags that provide powerful scanning options. Nmap is widely used by network administrators to scan for:

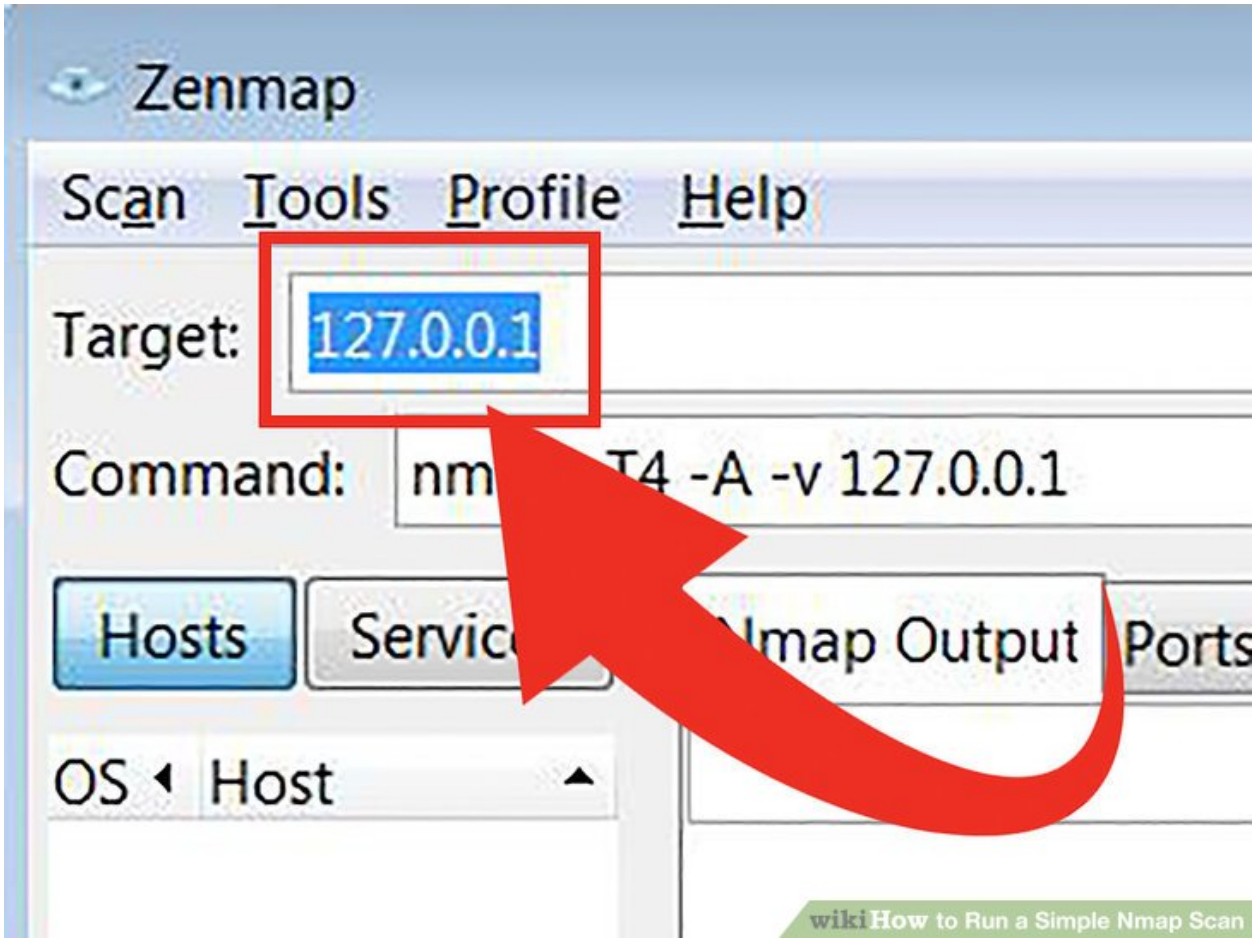
- Open ports and services
- Discover services along with their versions
- Guess the operating system running on a target machine
- Get accurate packet routes till the target machine
- Monitoring hosts

Nmap is a network mapper that has emerged as one of the most popular, free network discovery tools on the market. Nmap is now one of the core tools used by network administrators to map their networks. The program can be used to find live hosts on a

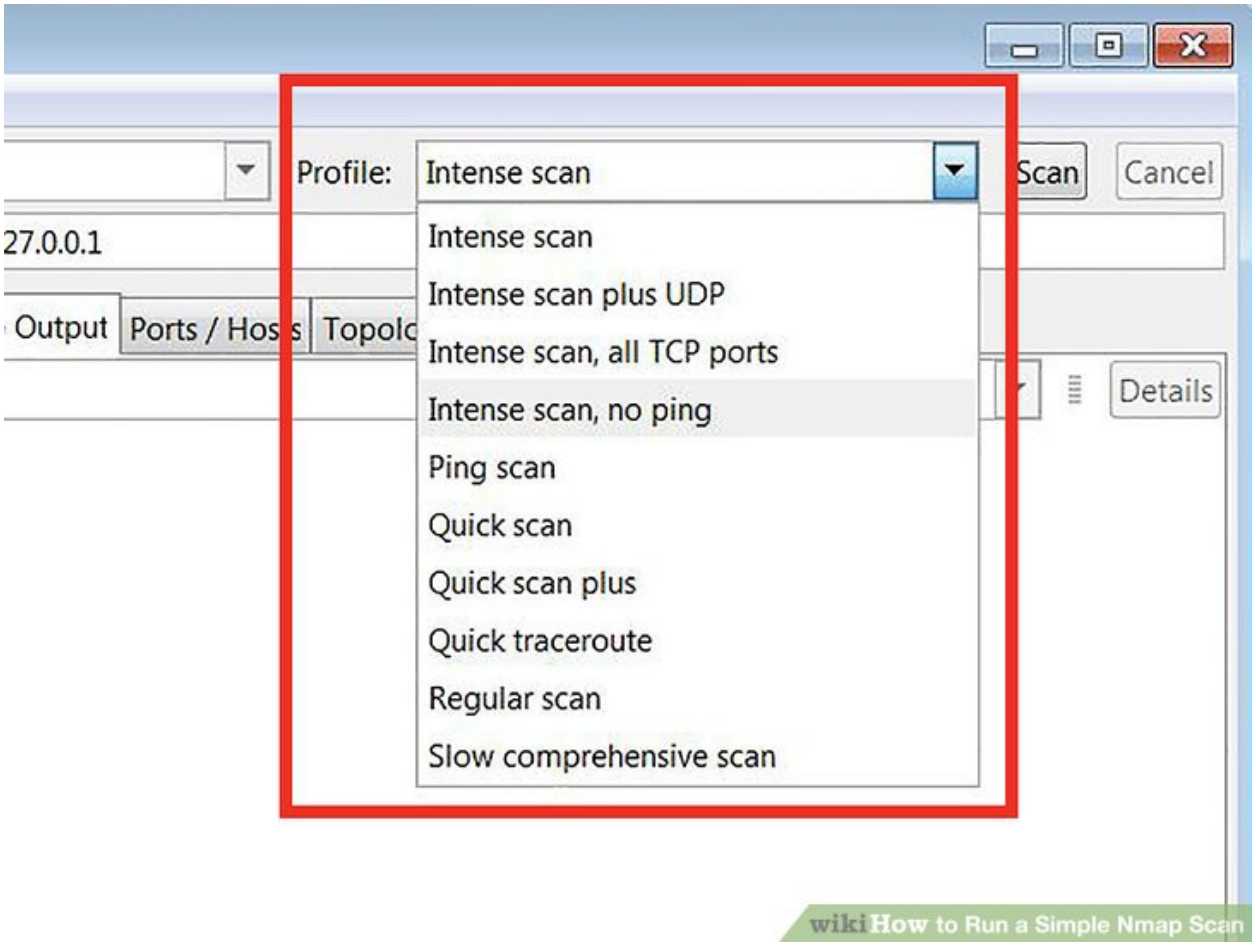
network, perform port scanning, ping sweeps, OS detection, and version detection.



Using Zenmap GUI in windows



Enter in the target for your scan. The Zenmap program makes scanning a fairly simple process. The first step to running a scan is choosing your target. You can enter a domain (example.com), an IP address (127.0.0.1), a network (192.168.1.0/24)



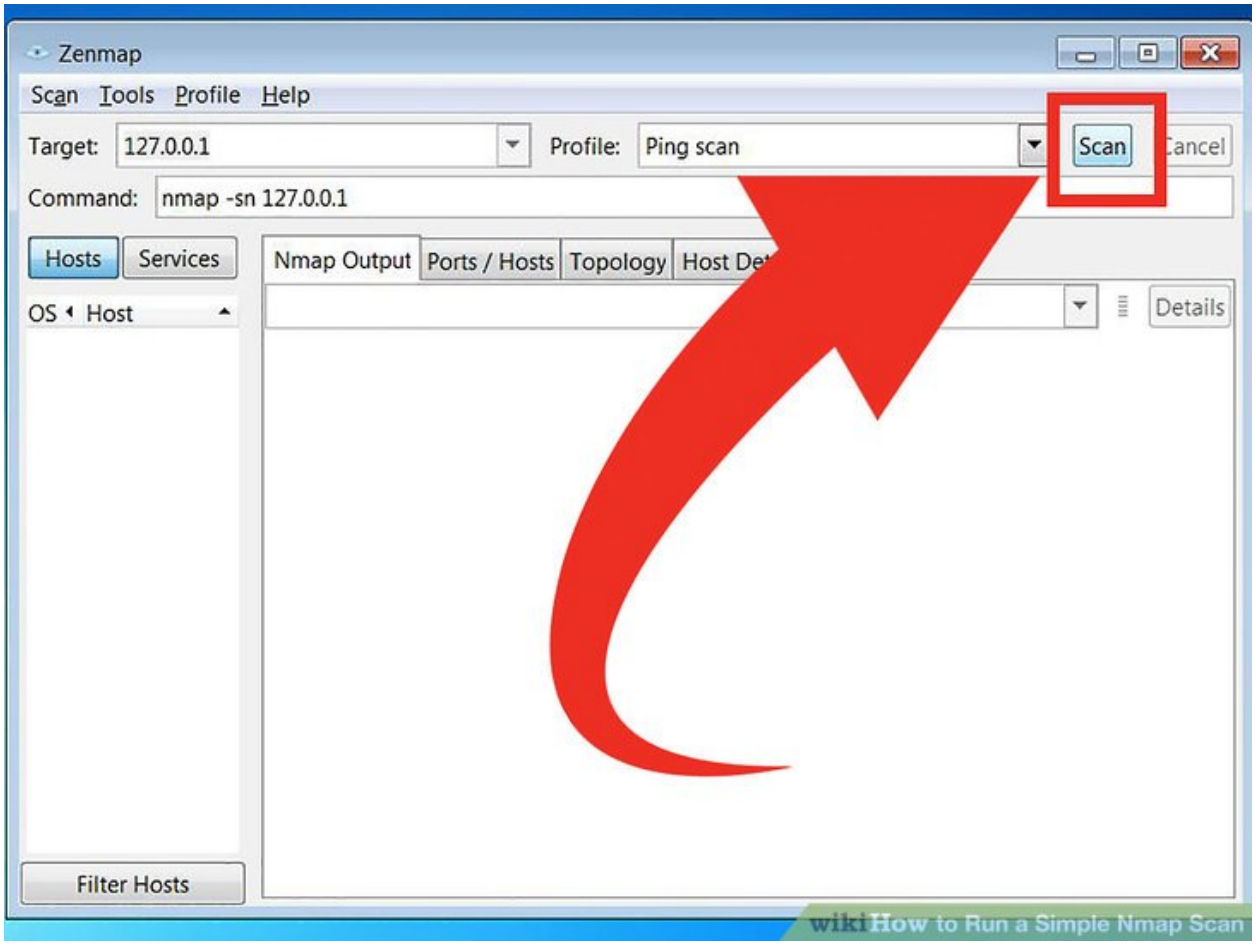
- **Intense scan** - A comprehensive scan. Contains Operating System (OS) detection, version detection, script scanning, traceroute, and has aggressive

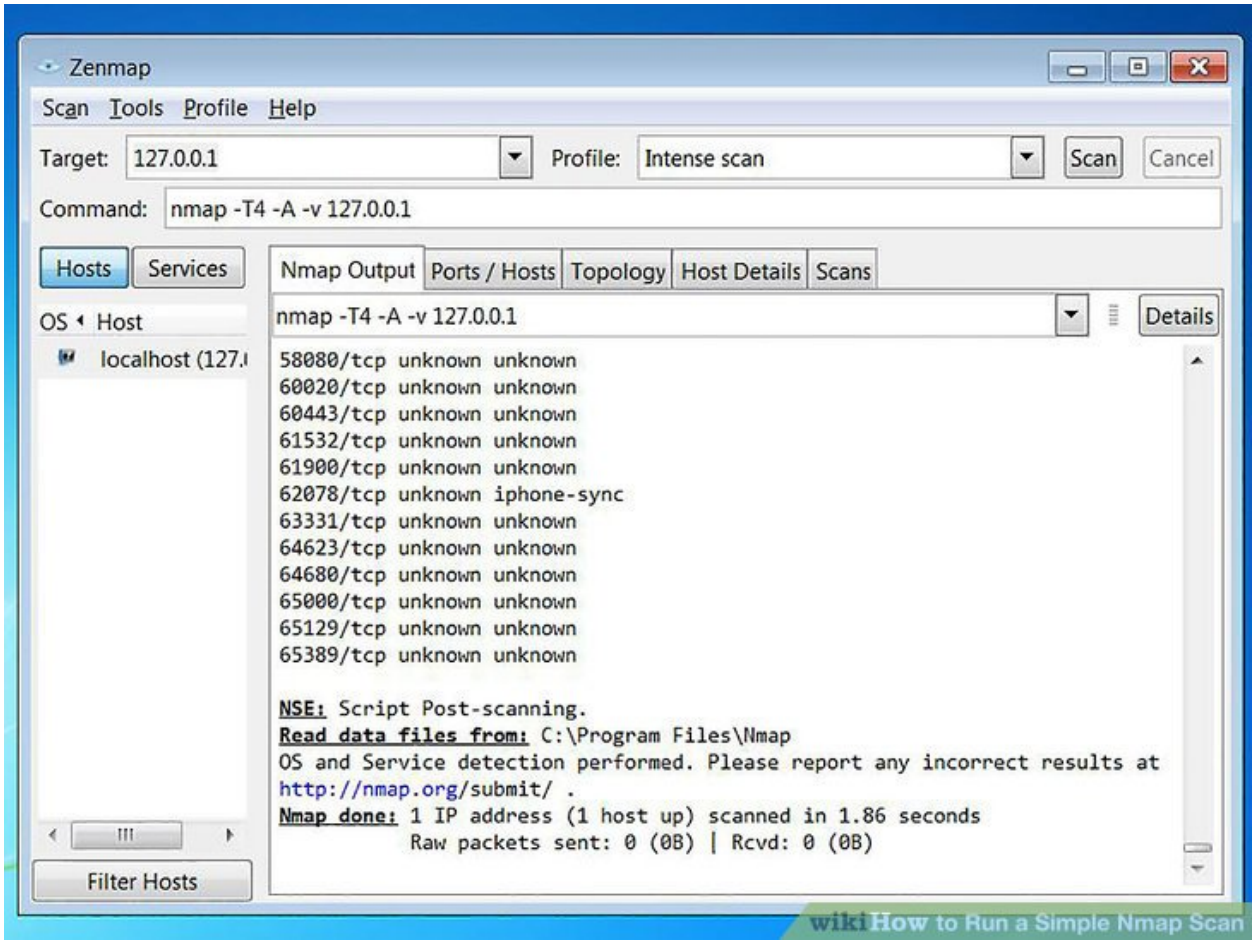
scan timing. This is considered an intrusive scan.

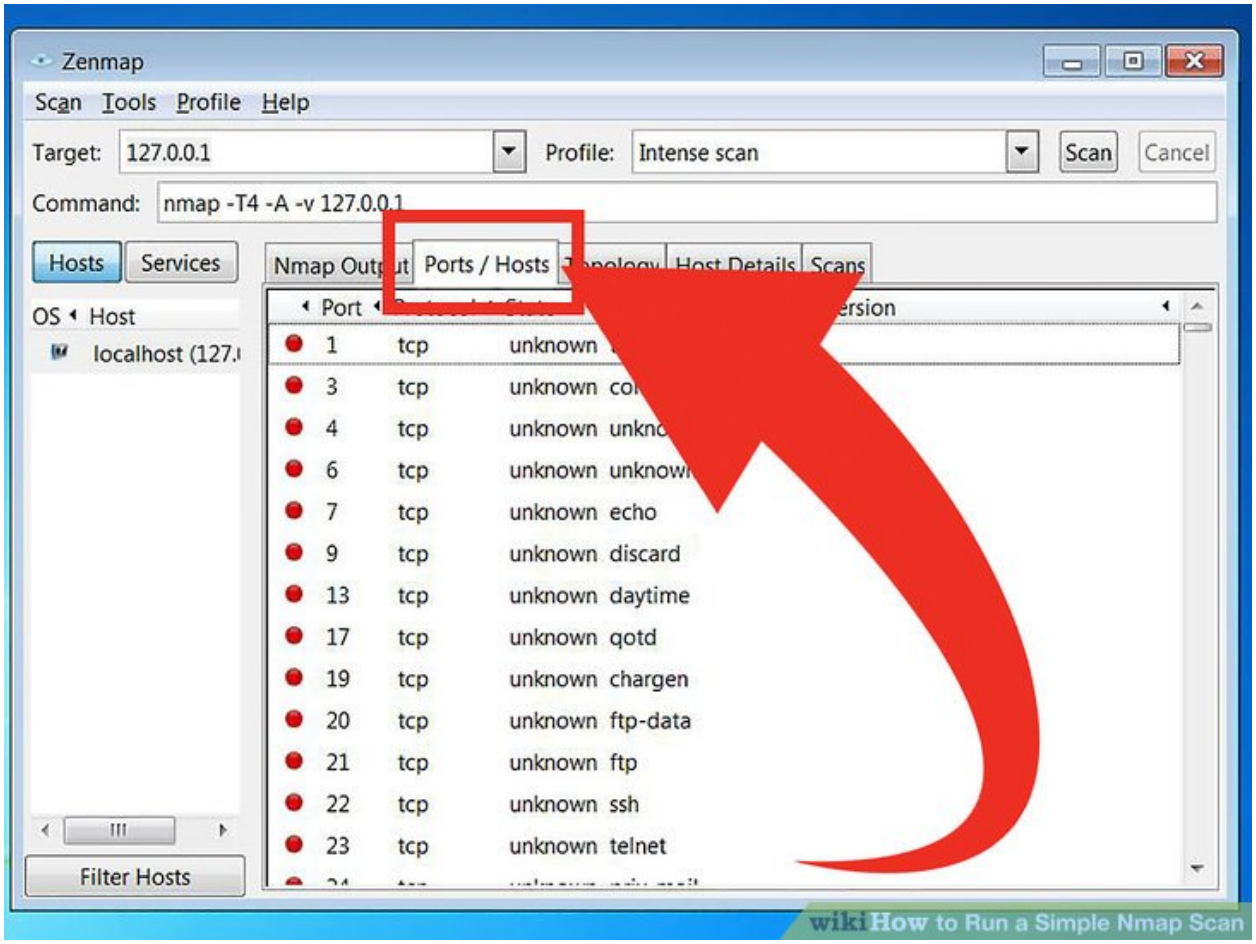
- **Ping scan** - This scan simply detects if the targets are online, it does not scan any ports.
- **Quick scan** - This is quicker than a regular scan due to aggressive timing and only scanning select ports.
- **Regular scan** - This is the standard Nmap scan without any modifiers. It will return ping and return open ports on the target.

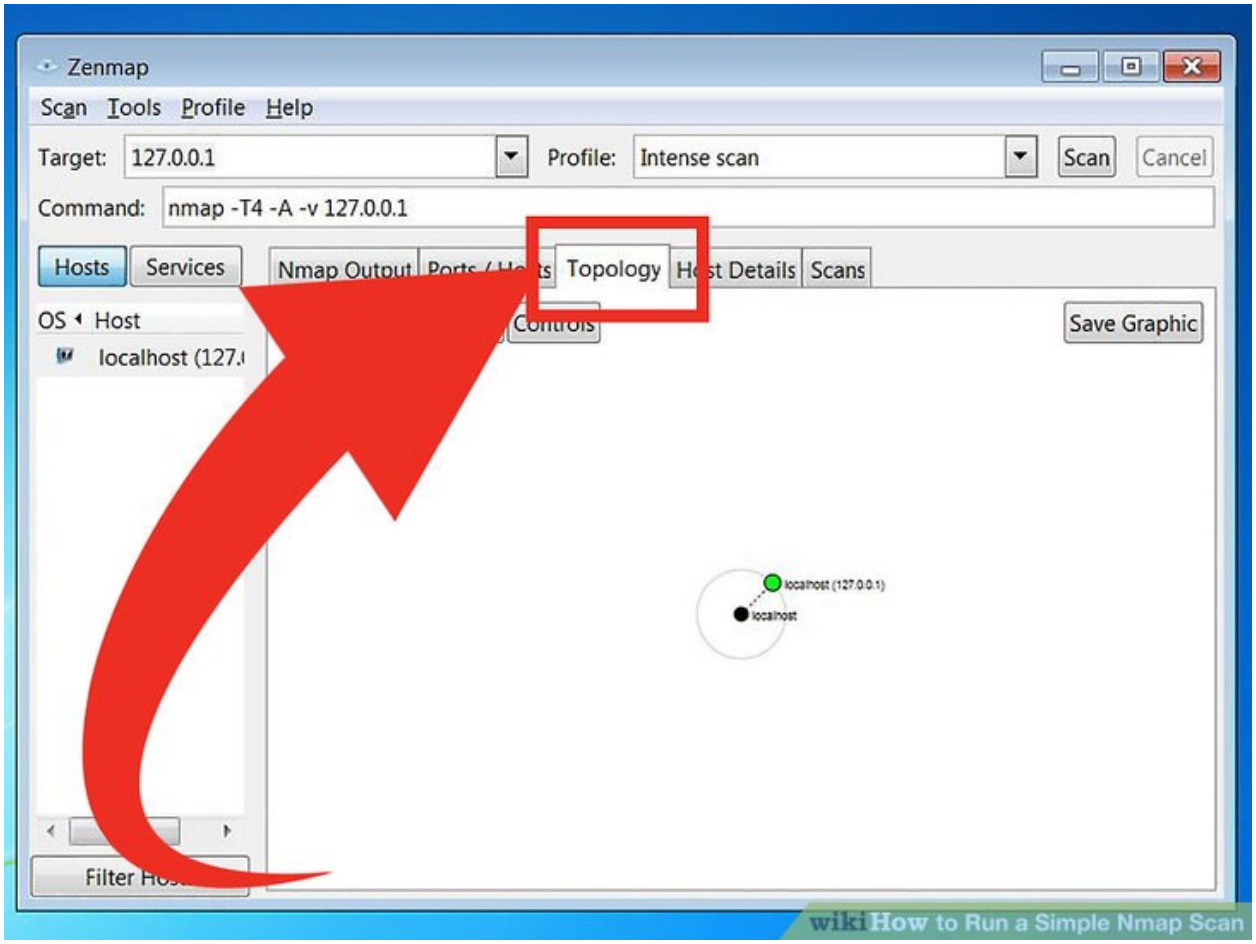
Click Scan to start scanning. The active results of the scan will be displayed in the Nmap Output tab. The time the scan takes will depend on the scan profile you chose,

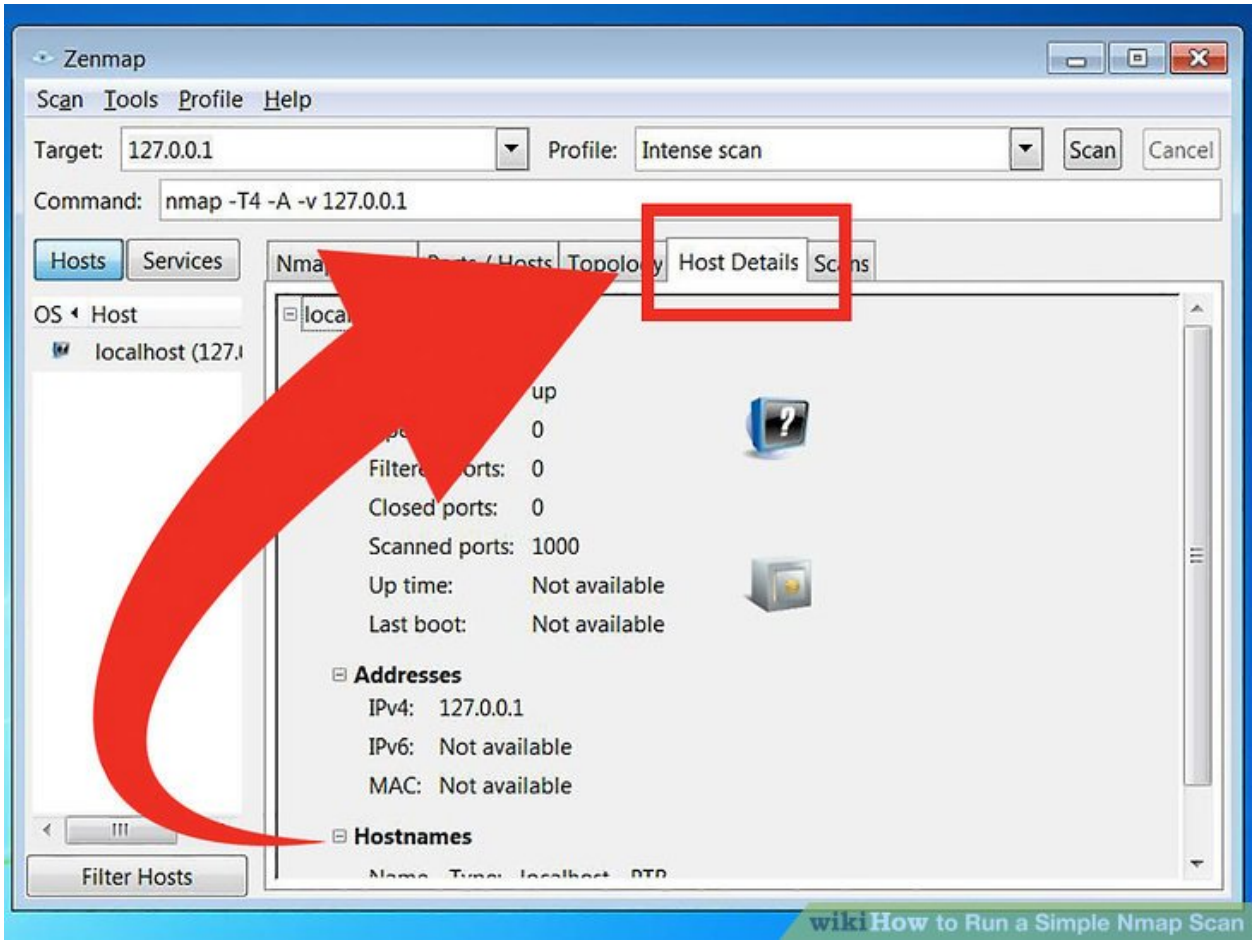
the physical distance to the target, and the target's network configuration.



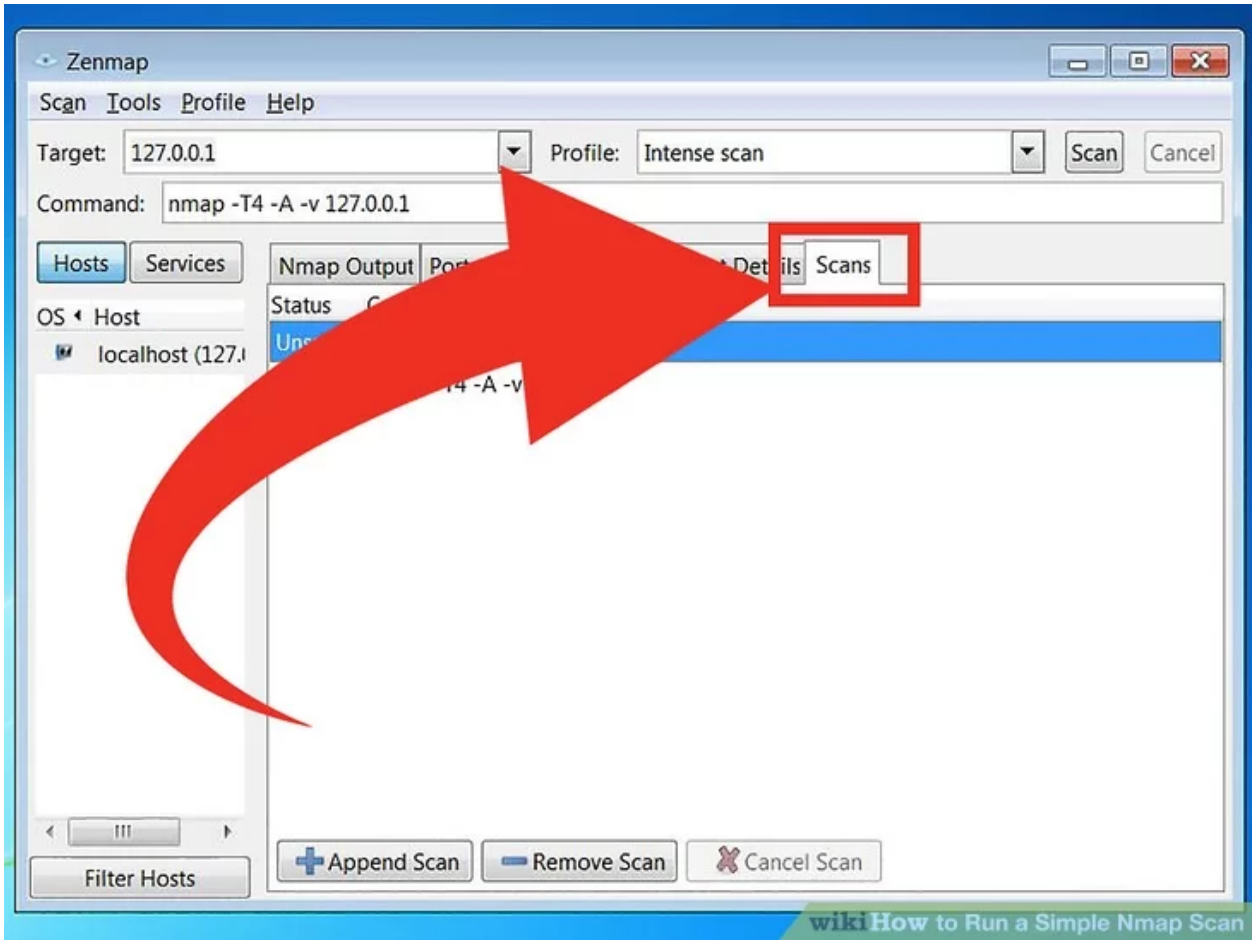

Zenmap
Scan Tools Profile Help
Target: 127.0.0.1
Profile: Intense scan
Scan
Cancel
Command: nmap -T4 -A -v 127.0.0.1
Hosts
Services
OS Host
localhost (127.
Filter Hosts
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
nmap -T4 -A -v 127.0.0.1
Details
58080/tcp unknown unknown
60020/tcp unknown unknown
60443/tcp unknown unknown
61532/tcp unknown unknown
61900/tcp unknown unknown
62078/tcp unknown iphone-sync
63331/tcp unknown unknown
64623/tcp unknown unknown
64680/tcp unknown unknown
65000/tcp unknown unknown
65129/tcp unknown unknown
65389/tcp unknown unknown
NSE: Script Post-scanning.
Read data files from: C:\Program Files\Nmap
OS and Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.86 seconds
Raw packets sent: 0 (0B) | Rcvd: 0 (0B)
wikiHow to Run a Simple Nmap Scan


Zenmap
Scan Tools Profile Help
Target: 127.0.0.1
Profile: Intense scan
Scan
Cancel
Command: nmap -T4 -A -v 127.0.0.1
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
OS Host
localhost (127.
Port
Version
1 tcp unknown
3 tcp unknown
4 tcp unknown
6 tcp unknown
7 tcp unknown echo
9 tcp unknown discard
13 tcp unknown daytime
17 tcp unknown qotd
19 tcp unknown chargen
20 tcp unknown ftp-data
21 tcp unknown ftp
22 tcp unknown ssh
23 tcp unknown telnet
Filter Hosts
wikiHow to Run a Simple Nmap Scan


Zenmap
Scan Tools Profile Help
Target: 127.0.0.1
Profile: Intense scan
Scan
Cancel
Command: nmap -T4 -A -v 127.0.0.1
Hosts
Services
Nmap Output
Topology
Scans
OS Host
localhost (127.
Save Graphic
localhost (127.0.0.1)
localhost
wikiHow to Run a Simple Nmap Scan



Scans - This tab stores the commands of your previously-run scans. This allows you to quickly re-scan with a specific set of parameters.



Nmap Commands

Scanning Techniques

Flag	Use	Example
-sS	TCP syn port scan	nmap -sS 192.168.1.1

-sT	TCP connect port scan	nmap -sT 192.168.1.1
–sU	UDP port scan	nmap –sU 192.168.1.1
–sA	TCP ack port scan	nmap –sA 192.168.1.1

Host Discovery

Flag	Use	Example
-Pn	only port scan	nmap -Pn 192.168.1.1
-sn	only host discover	nmap -sn 192.168.1.1
-PR	arp discovery on a local network	nmap -PR 192.168.1.1
-n	disable DNS resolution	nmap -n 192.168.1.1

Port Specification

Flag	Use	Example
-p	specify a port or port range	nmap -p 1-30 192.168.1.1
-p-	scan all ports	nmap -p- 192.168.1.1
-F	fast port scan	nmap -F 192.168.1.1

service Version and OS Detection

Flag	Use	Example
-sV	detect the version of services running	nmap -sV 192.168.1.1
-A	aggressive scan	nmap -A 192.168.1.1
-O	detect operating system of the target	nmap -O 192.168.1.1

Scanning Multiple Hosts

Nmap has the capability of scanning multiple hosts simultaneously. This feature comes in real handy when you are managing vast network infrastructure.

- Write all the IP addresses in a single row to scan all of the hosts at the same time
- nmap 192.164.1.1 192.164.0.2 192.164.0.2

Web links

Installation link for windows

https://nmap.org/download.html

Process that happen in windows

https://www.wikihow.com/Run-a-Simple-Nmap-Scan

https://www.unixmen.com/10-practical-examples-linux-nmap-command/

Message Digest Algorithm

MD5

• designed by Ronald Rivest (the R in RSA) •

latest in a series of MD2, MD4

• produces a 128-bit hash value

until recently was the most widely used hash algorithm – in recent times have both brute-force & cryptanalytic concerns • specified as Internet standard RFC1321

MD5 Overview

1. pad message so its length is 448 mod 512

• Padding of 1-512 bits is always used.

• Padding: 1000….0

2. append a 64-bit length value to message

• Generate a message with 512L bits in length

3. initialise 4-word (128-bit) MD buffer (A,B,C,D)

4. process message in 16-word (512-bit) blocks:

5. output hash value is the final buffer value

Digital Signature Algorithm

DSA stand for Digital Signature Algorithm. It is used for digital signature and its verification. It is based on mathematical concept of modular exponentiation and discrete logarithm. It was developed by National Institute of Standards and Technology (NIST) in 1991.

Digital signatures are the public-key primitives of message authentication in cryptography. In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind signatory to the message.

Digital signature is a technique that binds a person/entity to the digital data. This binding can be independently verified by receiver as well as any third party.

Digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer.

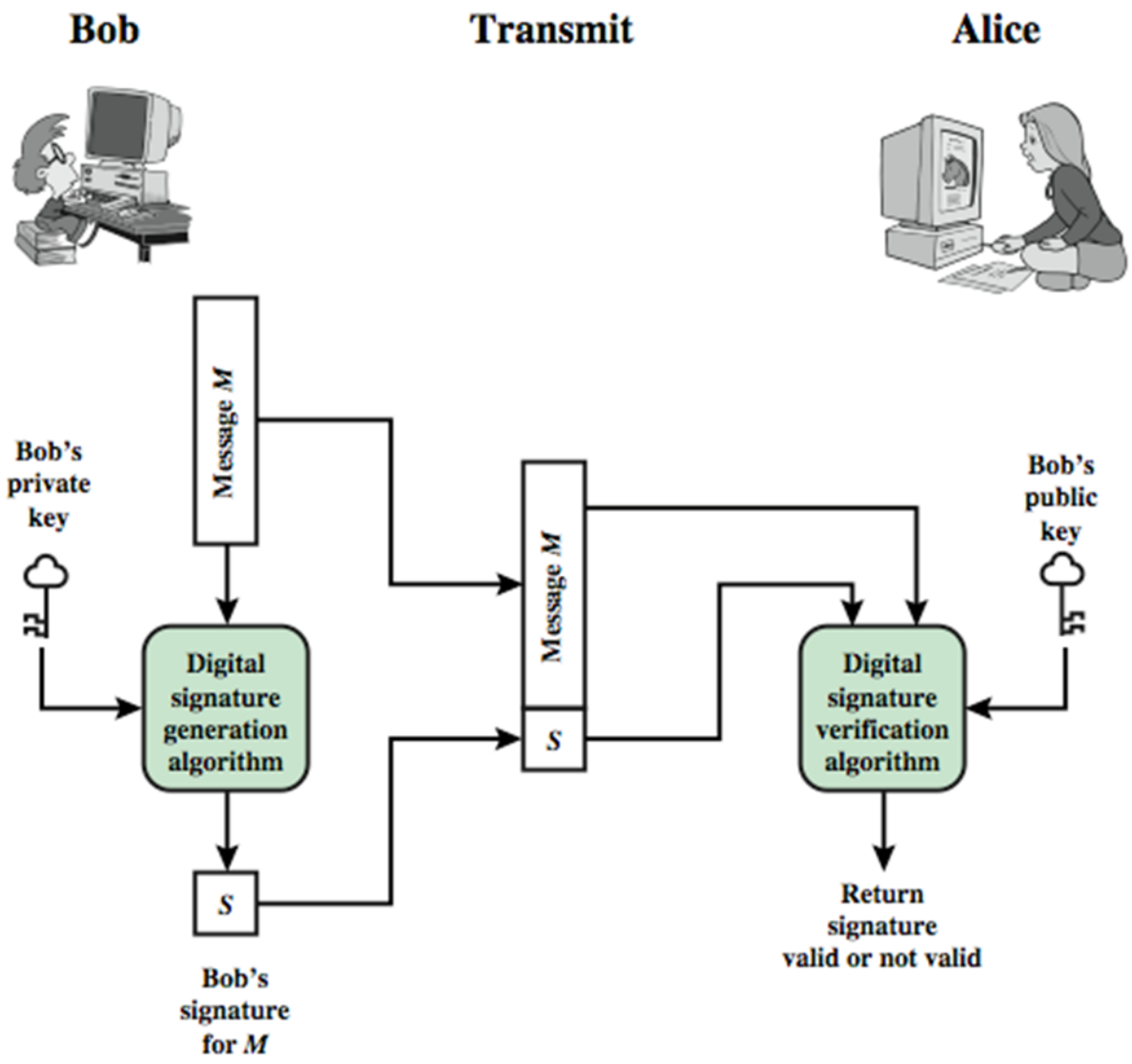
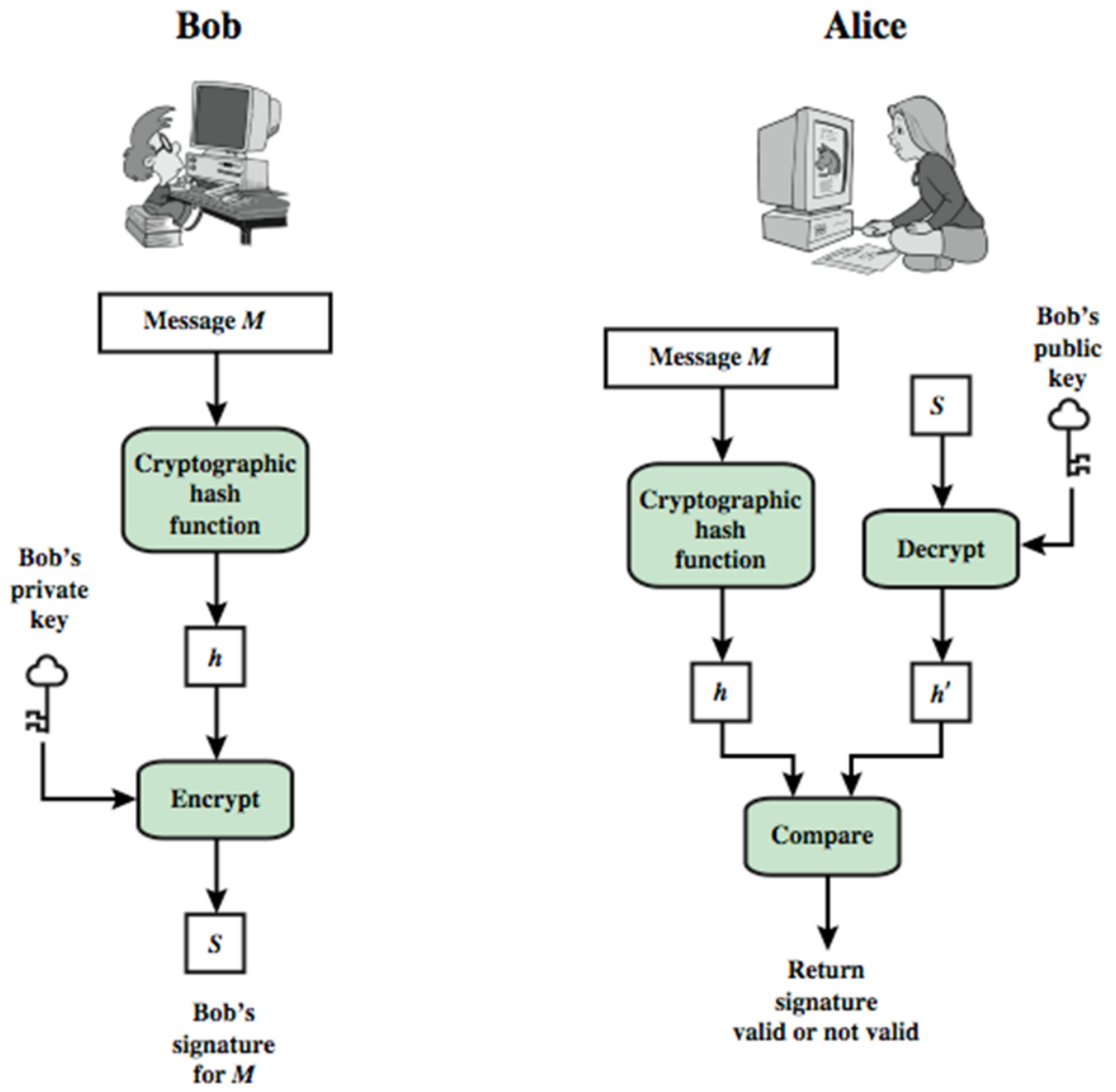
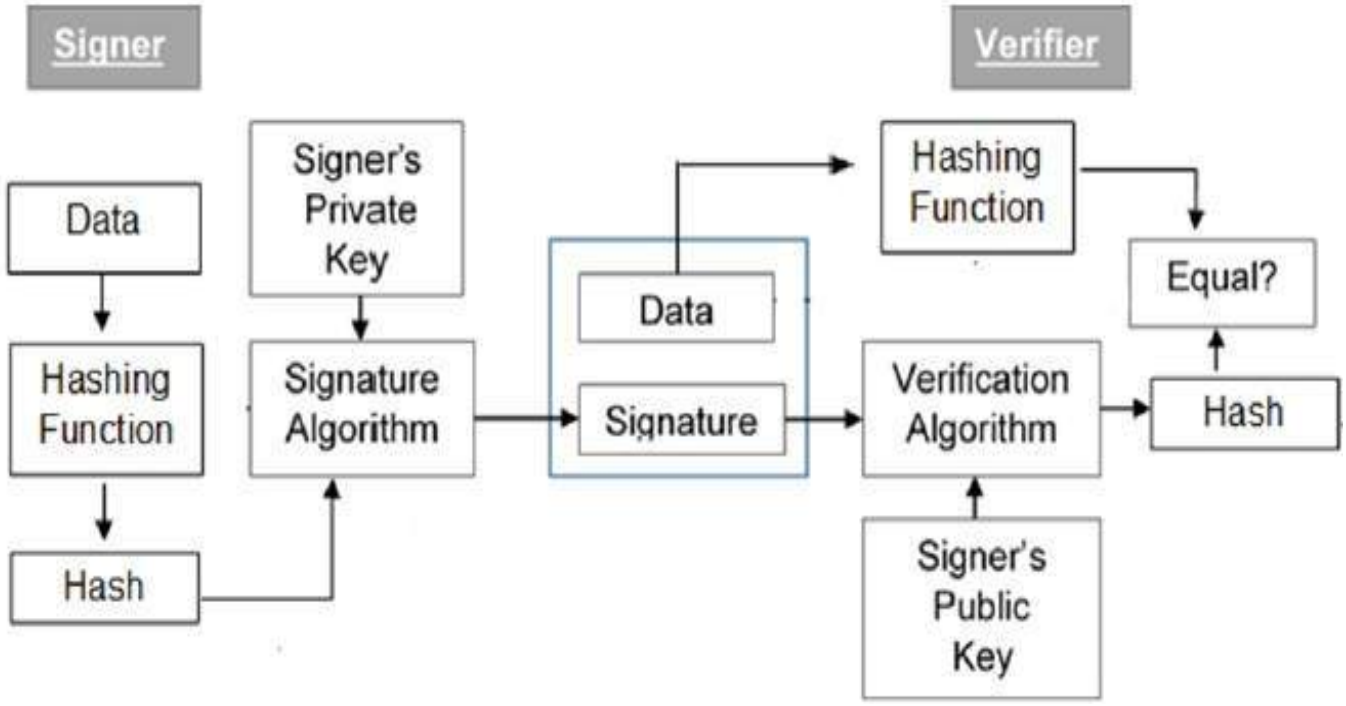


Figure :Digital Signature Model


Bob
Alice
Message M
Cryptographic hash function
Bob's private key
h
Encrypt
S
Bob's signature for M
Message M
Cryptographic hash function
S
Bob's public key
Decrypt
h
h'
Compare
Return signature valid or not valid



The following points explain the entire process in detail −

- Each person adopting this scheme has a public-private key pair.
- Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.

- Signer feeds data to the hash function and generates hash of data.
- Hash value and signature key are then fed to the signature algorithm which produces the digital signature on given hash. Signature is appended to the data and then both are sent to the verifier.
- Verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.
- Verifier also runs same hash function on received data to generate hash value.
- For verification, this hash value and output of verification algorithm are compared. Based on the comparison result, verifier decides whether the digital signature is valid.
- Since digital signature is created by ‘private’ key of signer and no one else

can have this key; the signer cannot repudiate signing the data in future.

Importance of Digital signature

Message authentication
Data Integrity
Non-repudiation

Snort

Snort is the foremost Open Source Intrusion Prevention System (IPS) in the world. Snort IPS uses a series of rules that help define malicious network activity and uses those rules to find packets that match against them and generates alerts for users.

Snort has three primary uses: As a packet sniffer like tcpdump, as a packet logger — which is useful for network traffic

debugging, or it can be used as a full-blown network intrusion prevention system.

Snort is a flexible, lightweight, and popular Intrusion Detection System that can be deployed according to the needs of the network, ranging from small to large networks, and provides all the features of a paid IDS.

Open terminal type ifconfig

output

enp2s0:
flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 172.20.15.234 netmask 255.255.240.0 broadcast 172.20.15.255

inet6 fe80::3922:4a0c:89ef:ca21 prefixlen 64 scopeid 0x20<link>

ether 50:9a:4c:01:fe:ec txqueuelen 1000 (Ethernet)

RX packets 1630908 bytes 234296084 (234.2 MB)

RX errors 0 dropped 35 overruns 0 frame 0

TX packets 25764 bytes 6450864 (6.4 MB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Open one more terminal and do installation of snort

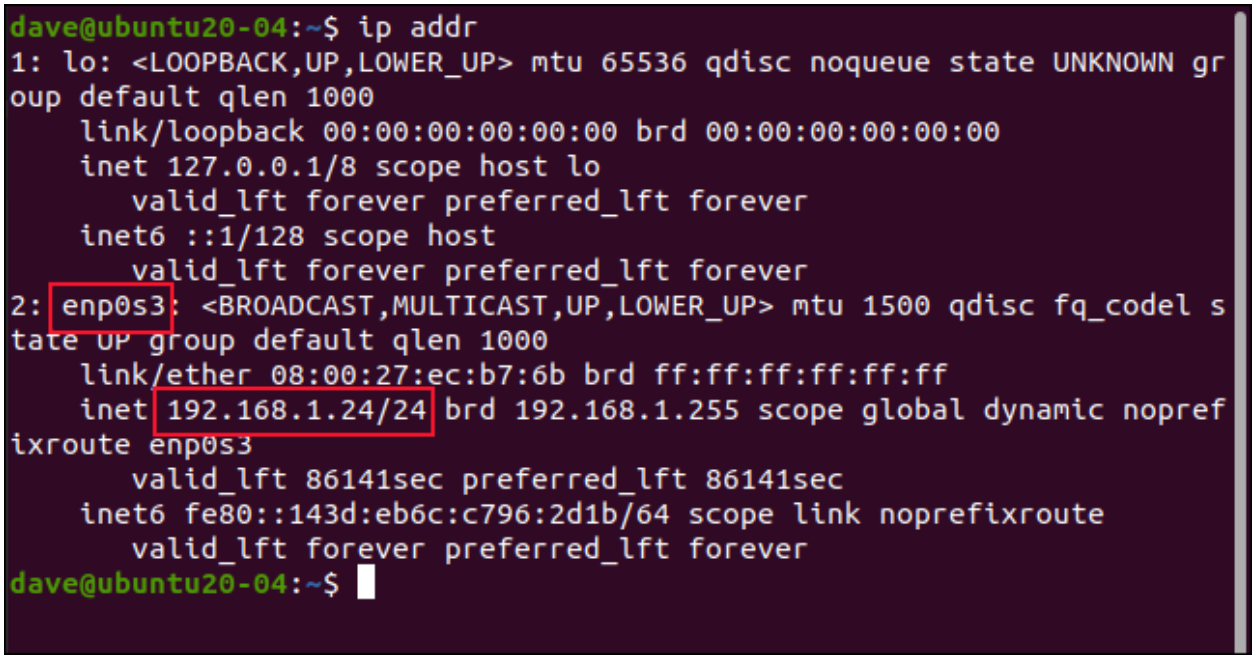
Open one more terminal after installation and perform ping operation followed by ip address

1.To install Snort on Ubuntu, use this command: sudo apt install snort

As the installation proceeds, you'll be asked a couple of questions. You can find the answers to these by using the `ip addr` command before starting the installation, or in a separate terminal window.

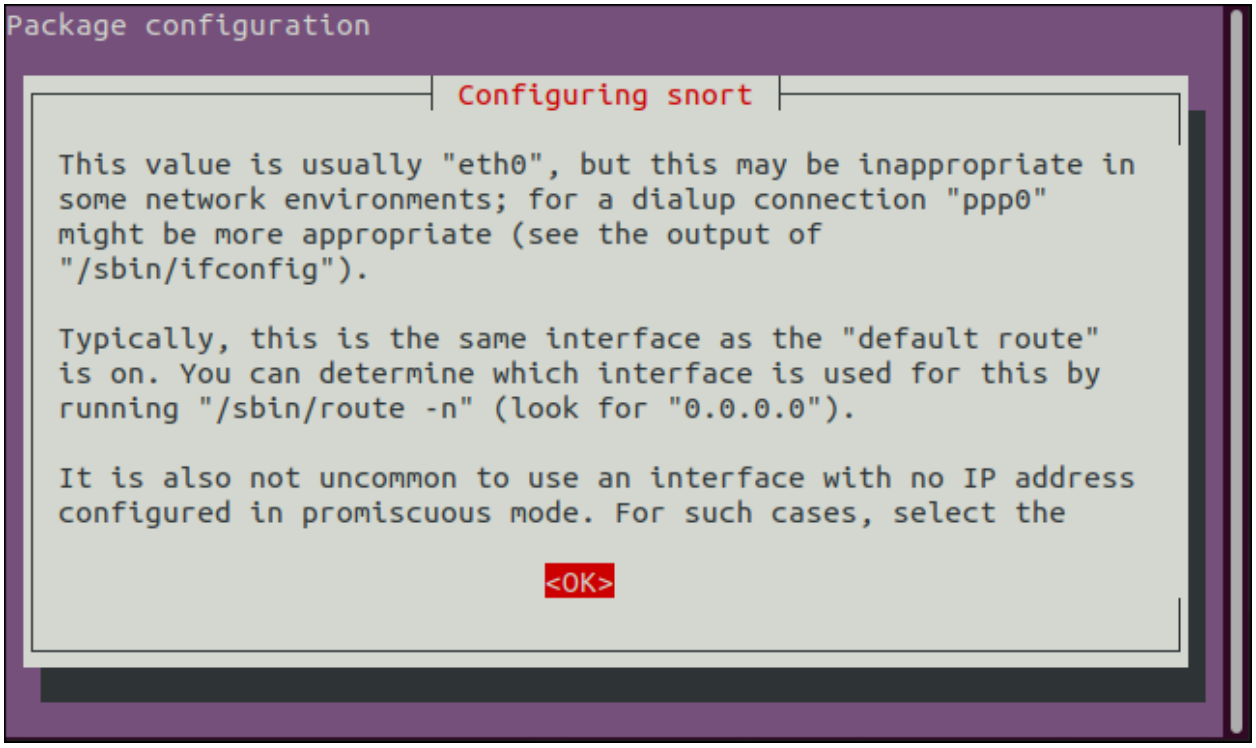
Type ip addr in separate terminal

You will get this output,it changes according to the your system

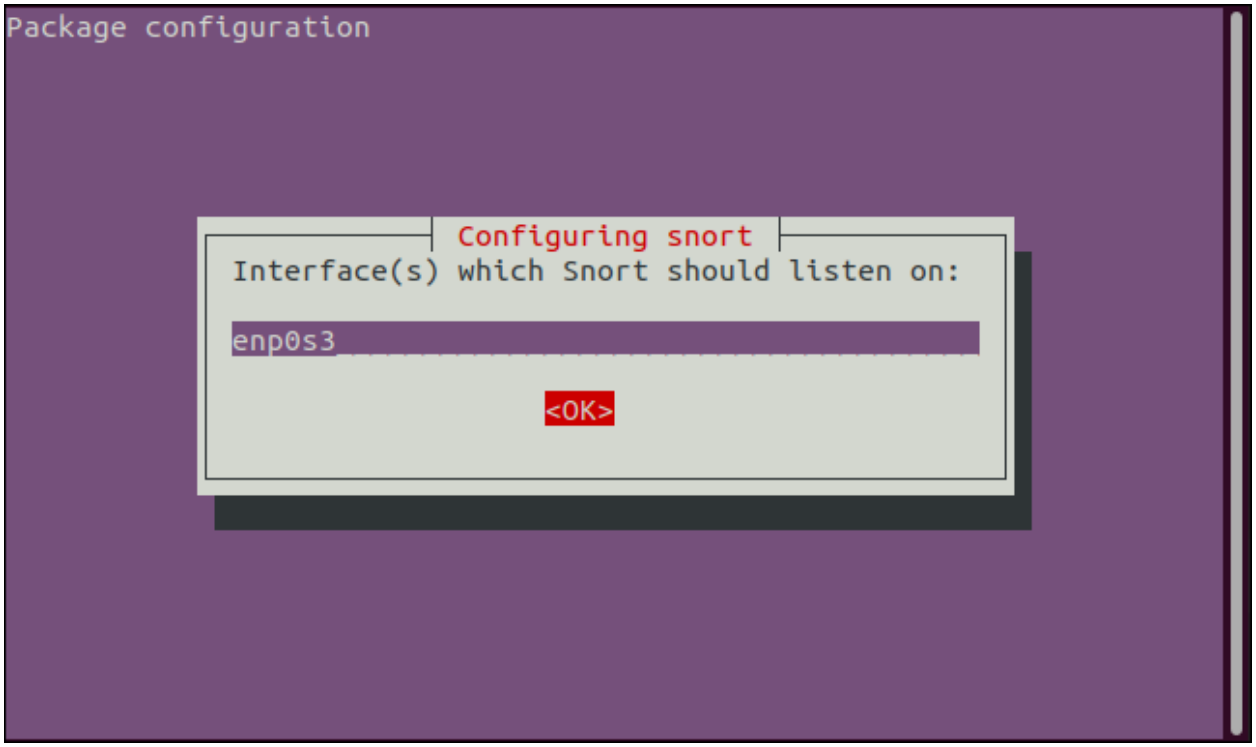


Take note of your network interface name. On this research computer, it is `enp0s3`

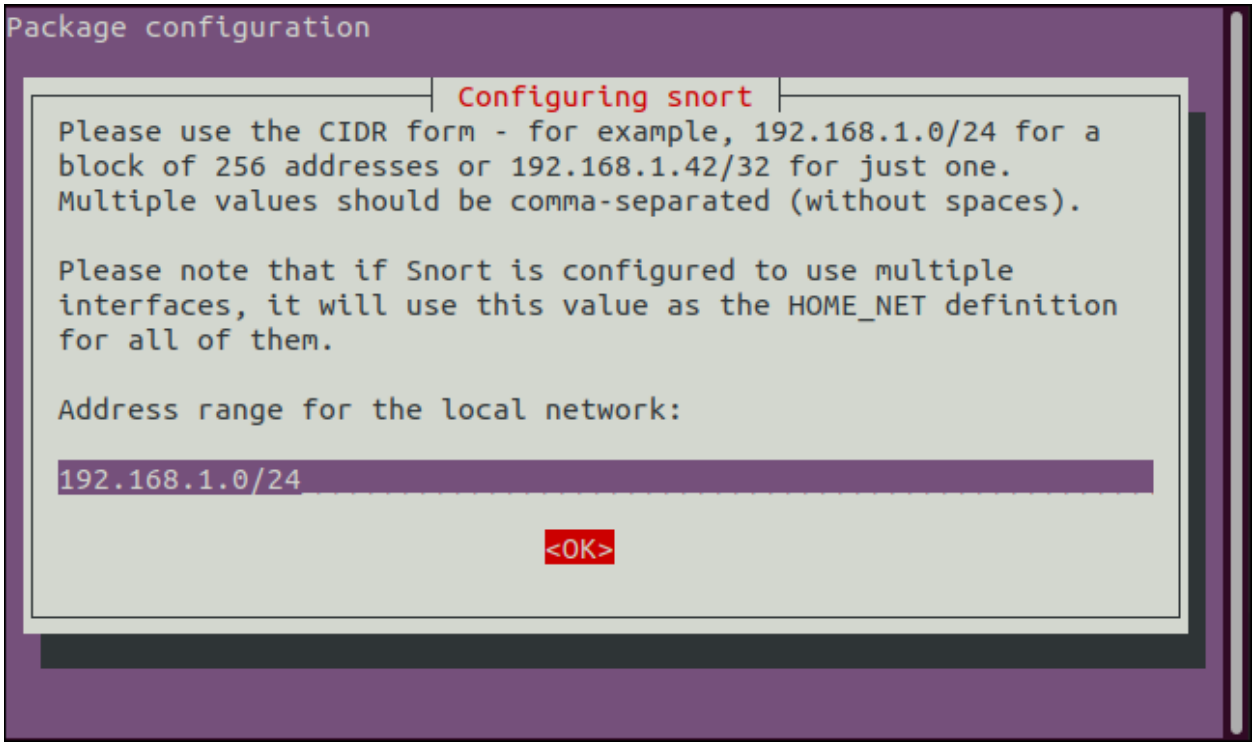
Press “Tab” to highlight the “OK” button, and press “Enter.”



Type the name of the network interface name and press “Tab” to highlight the “OK” button, and press “Enter.”



Type the network address range in CIDR format, press “Tab” to highlight the “OK” button, and press “Enter.”



To see whether snort installed or not
snort --help
You will get output

Then after that type the command **man snort**
Output
It will give descriptions and options then after enter q

Configuring Snort

There are a few steps to complete before we can run Snort. We need to edit the “snort.conf” file.

```
sudo gedit /etc/snort/snort.conf
```

Locate the line that reads “`ipvar HOME_NET any`” and edit it to replace the “any” with the CIDR notation address range of your network.

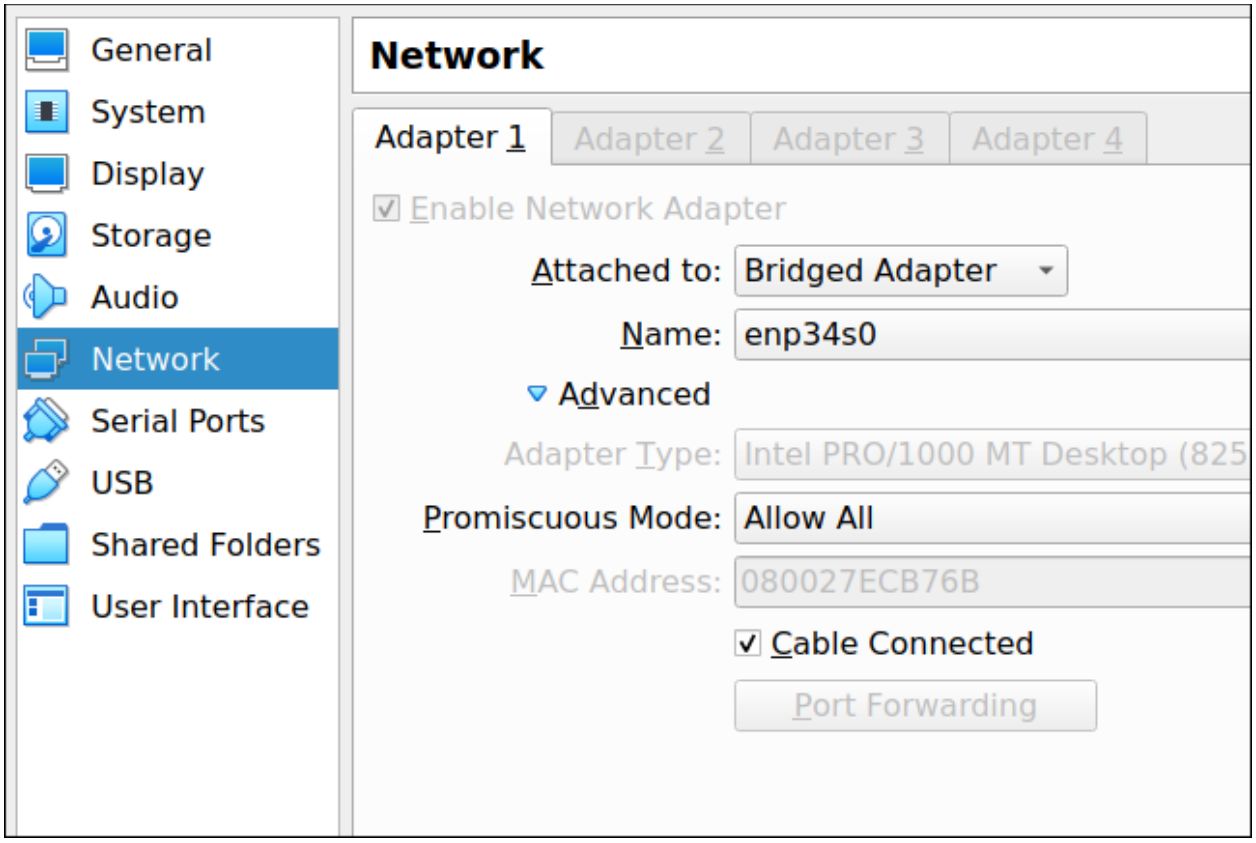
```
44 # Setup the network addresses you are protecting
45 #
46 # Note to Debian users: this value is overriden when starting
47 # up the Snort daemon through the init.d script by the
48 # value of DEBIAN_SNORT_HOME_NET s defined in the
49 # /etc/snort/snort.debian.conf configuration file
50 #
51 ipvar HOME_NET 192.168.1.0/24
52
53 # Set up the external network addresses. Leave as "any" in most situ
54 ipvar EXTERNAL_NET any
55 # If HOME_NET is defined as something other than "any", alternative,
56 # use this definition if you do not want to detect attacks from your
57 # IP addresses:
58 #ipvar EXTERNAL_NET !$HOME_NET
59
```

Save your changes and close the file.

To make the Snort computer's network interface listen to all network traffic, we need to set it to promiscuous mode.

```
sudo ip link set enp2s0 promisc
on
```

If you are running Snort in a virtual machine, also remember to adjust the settings in your hypervisor for the virtual network card used by your virtual machine. For example, in VirtualBox, you need to go to `Settings > Network > Advanced` and change the "Promiscuous Mode" drop-down to "Allow All."



sudo snort -T -i enp2s0 -c/etc/snort/snort.conf

Perform ping in command prompt by typing your inet address

Ping 192.168.0.167

Open other terminal in ubuntu and perform again ping

Ping 192.168.0.167

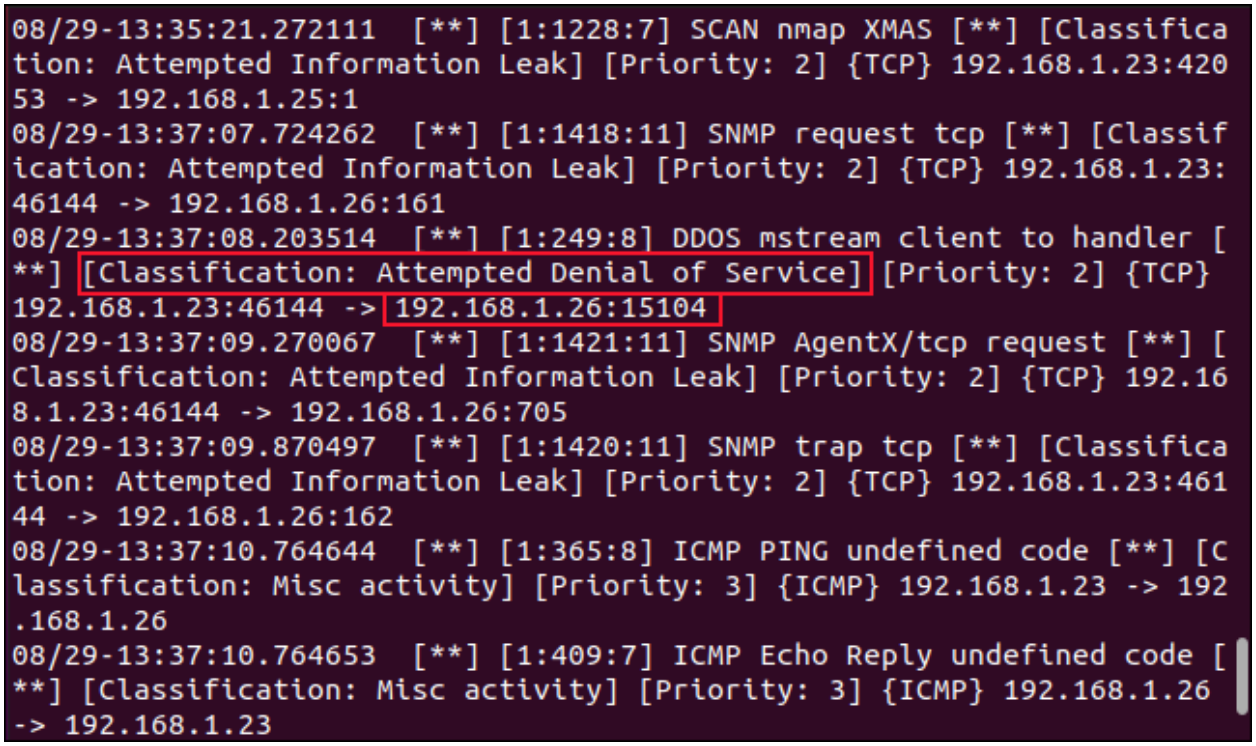
Find icmp packets

sudo snort -A console -i enp2s0 -c/etc/snort/snort.conf

```
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 2.4  <Buil
d 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
Commencing packet processing (pid=20037)
```

```
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
Commencing packet processing (pid=20037)
08/27-21:37:34.850356  [**] [1:1420:11] SNMP trap tcp [**] [Classifica
tion: Attempted Information Leak] [Priority: 2] {TCP} 192.168.1.23:643
99 -> 192.168.1.25:162
08/27-21:37:35.465875  [**] [1:1421:11] SNMP AgentX/tcp request [**] [
Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.16
8.1.23:64399 -> 192.168.1.25:705
08/27-21:37:35.841650  [**] [1:249:8] DDOS mstream client to handler [
**] [Classification: Attempted Denial of Service] [Priority: 2] {TCP}
192.168.1.23:64399 -> 192.168.1.25:15104
08/27-21:37:36.806899  [**] [1:1418:11] SNMP request tcp [**] [Classif
ication: Attempted Information Leak] [Priority: 2] {TCP} 192.168.1.23:
64399 -> 192.168.1.25:161
08/27-21:37:37.808042  [**] [1:365:8] ICMP PING undefined code [**] [C
lassification: Misc activity] [Priority: 3] {ICMP} 192.168.1.23 -> 192
.168.1.25
08/27-21:37:37.808067  [**] [1:409:7] ICMP Echo Reply undefined code [
**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.25
-> 192.168.1.23
```



After that Store icmp packets in a particular file

sudo snort -A console -i enp2s0 -c/etc/snort/snort.conf | tee -a cbitit.txt

TCPDUMP & DUMPCAP

It’s often more useful to capture packets using `tcpdump` rather than `wireshark`

Dumpcap is a network traffic dump tool. It captures packet data from a live network and writes the packets to a file. Dumpcap’s native capture file format is pcapng, which is also the format used by Wireshark.

By default, Dumpcap uses the pcap library to capture traffic from the first available network interface and writes the received raw packet data, along with the packets’ time stamps into a pcapng file. The capture filter syntax follows the rules of the pcap library.

tcpdump is a packet sniffing and packet analyzing tool for a System Administrator to troubleshoot connectivity issues in Linux. It is used to capture, filter, and analyze network traffic such as TCP/IP packets going through your system. It is many times used as a security tool as well. It saves the captured information in a pcap file, these pcap files can then be opened through Wireshark or through the command tool itself.

Installing tcpdump tool for Ubuntu

```
sudo apt install tcpdump
```

1. To capture the packets of current network interface

```
sudo tcpdump
```

This will capture the packets from the current interface of the network through which the system is connected to the internet.

2. To capture packets from a specific network interface

```
sudo tcpdump -i enp2s0
```

This command will now capture the packets from `enp2s0` network interface.

3. To capture specific number of packets

```
sudo tcpdump -c 4 -i enp2s0
```

This command will capture only 4 packets from the `enp2s0` interface.

4. To print captured packages in ASCII format

```
sudo tcpdump -A -i enp2s0
```

This command will now print the captured packets from `enp2s0` to ASCII value.

5. To display all available interfaces

```
sudo tcpdump -D
```

This command will display all the interfaces that are available in the system.

6.To capture packets with ip address

```
sudo tcpdump -n -i enp2s0
```

This command will now capture the packets with IP addresses.

7. To capture only TCP packets

```
sudo tcpdump -i enp2s0 tcp
```