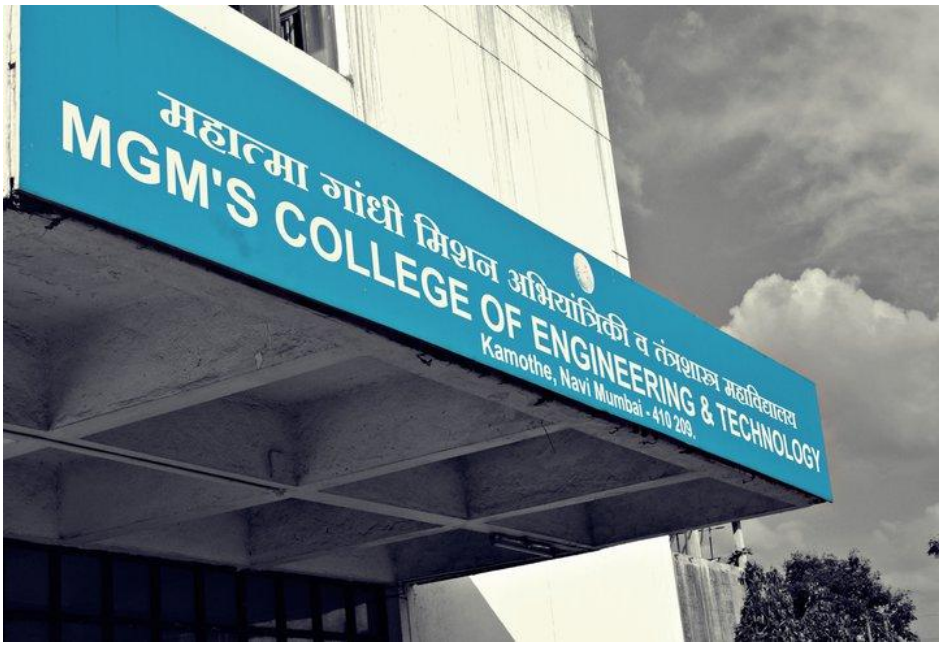
Mahatma Gandhi Mission's

College of Engineering and Technology, Kamothe, Navi Mumbai

Department of Information Technology



Name	:		**Department**	:	IT
Course /LaboratoryName	:	Security Lab	**Course Code**	:	ITL502
Class/Sem/Div	:	TE / V	**Academic Year**	:	2022-23
Email ID	:		**Mobile Number**	:	

Mahatma Gandhi Mission's

College of Engineering and Technology, Kamothe, Navi Mumbai

Vision	To become one of the outstanding Engineering Institute in India by providing a conductive and vibrant environment to achieve excellence in the field of Technology
Mission	To empower the aspiring professional students to be prudent enough to explore the world of technology and mould them to be proficient to reach the pinnacle of success in the competitive global economy.

Department of Information Technology

Vision	To emerge out as a prominent department offering a programme in its pursuit for academic excellence in order to develop professionally competent and socially responsible engineers capable of meeting industry demands and social obligations in a vibrant global environment.
Mission	To strive towards building an atmosphere that will be a catalyst for innovative ideas and learning, providing students with various opportunities and experiences that can help them to thrive and prosper through a blend of academics, practical exposure and research programs to pursue successful careers in a global environment.

Mahatma Gandhi Mission's

College of Engineering and Technology, Kamothe, Navi Mumbai

Department of Information Technology

Program Educational Objectives (PEOs)

PEO 1	To impart learners with sound knowledge of basic sciences and core engineering fundamentals.
PEO 2	To prepare learners use modern programming tools/technologies and develop competency to counter complicated engineering problems.
PEO 3	To prepare learners to be professionally competent and socially responsible to sustain and strive through the competitive, global/environment challenges.
PEO 4	To create a strong foundation in IT discipline and motivate learners undertake postgraduate studies, explore professional avenues or venture into entrepreneurship.
PEO 5	To inculcate personality traits and professional ethics.

Program Outcomes (POs)

PO 1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO 2	**Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO 3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO 4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO 5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO 6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO 8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO 9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO 10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO 11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO 12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1	Analyze real life problems and design user friendly solutions.
PSO2	To provide effective solutions for problems in sectors like healthcare, science, commerce, e-governance etc., by employing right set of tools and methodologies.
PSO3	Design and implement right IT infrastructural setups for any organization.

Mahatma Gandhi Mission's

College of Engineering and Technology, Kamothe, Navi Mumbai

Course: Information Technology

Course Outcomes (COs)

Academic Year	Class/Semester	Course Name	Course Code
2022-23	TE / V	SECURITY LAB	ITL502

CO 1	Illustrate symmetric cryptography by implementing classical ciphers.
CO 2	Demonstrate Key management, distribution and user authentication.
CO 3	Explore the different network reconnaissance tools to gather information about networks.
CO 4	Use tools like sniffers, port scanners and other related tools for analyzing packets in a network.
CO 5	Use open-source tools to scan the network for vulnerabilities and simulate attacks.
CO 6	Demonstrate the network security system using open source tools.

_____

(Name & Signature)

Mahatma Gandhi Mission's

College of Engineering and Technology, Kamothe, Navi Mumbai

CERTIFICATE

This is to certify that Mr./Ms. _____of

Department of _____ Roll No. _____ UID _____

has successfully performed _____ Experiments/Tutorials and _____

Assignments given in the subject _____ __during the

period _____ to _____

Date : _____

Place: Subject In charge

_____

HoD

Mahatma Gandhi Mission's

College of Engineering and Technology, Kamothe, Navi Mumbai

Lab Name: SECURITY LAB

Student Name : **Roll No** :

Class/Sem : TE/V **Academic Year** : 2022-23

Subject Name : SECURITY LAB **Course Code** : ITL502

CONTENTS

Sr. No.	Title of Experiment/Tutorial/Assignment	Page No.	Date of Performance	Marks Out of15	Sign	CO Covered
1	Implementation of the Mono-alphabetic Substitution Cipher using Frequency analysis method.					CO1
2	Design and Implement a product cipher using Substitution ciphers.					CO1
3	Implementation of Cryptanalysis Playfair cipher.					CO1
4	Encrypt long messages using various modes of operation using DES.					CO2
5	Implementation and analysis of RSA cryptosystem					CO2
6	Implementation of Cryptographic Hash Functions and Applications (HMAC):SHA					CO2
7	Study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup to gather information about networks and domain registrars					CO3
8	Study of packet sniffer tools wireshark: Show the packets can be traced based on different filters.					CO3
9	To study and implement various scanning techniques using Nmap.					CO4
10	Study of malicious software using different tools: Use the NESSUS/ISO Kali Linux tool to scan the network for vulnerabilities					CO5
11	Study of Network security by: Set up Snort and study the logs.					CO6
12	Explore the GPG tool to implement email security					CO6

Mahatma Gandhi Mission's

College of Engineering and Technology, Kamothe, Navi Mumbai

Average Score (Out of 25)

1	Evaluation 1	
2	Evaluation 2	
3	Evaluation 3	
Average Score Evaluation (Out of 15)		
Attendance Marks (Out of 5)		
Assignment Marks (Out of 5)		
Total term work Marks (Out of 25)		

Internal Assessment Marks

Parameter	CO1	CO2	CO3	CO4	CO5	CO6
Test 1						
Test 2						
Term Work						
Total						

Mahatma Gandhi Mission's

College of Engineering and Technology, Kamothe, Navi Mumbai

Lab Name: Security Lab

LABORATORY ASSESSMENT

Student Name :
Roll No :
Class/Sem : TE/V
UID No :
Subject Name : SECURITY LAB
Course Code : ITL502

Assessment Parameters for Practical's/Tutorials/Assignment

Experiments	Conceptual Understanding	Performance	Self Learning Initiative	Ethical Behavior	Punctuality	Total
Exp 1						
Exp 2						
Exp 3						
Exp 4						
Exp 5						
Exp 6						
Exp 7						
Exp 8						
Exp 9						
Exp 10						
Exp 11						
Exp 12						

Total marks can be allotted in the range from 1 to 3 for all points

Subject In-charge

Student

Mahatma Gandhi Mission's

College of Engineering and Technology, Kamothe, Navi Mumbai

Lab Name: Security Lab

LABORATORY ASSESSMENT

Student Name : **Roll No** :

Class/Sem : TE/V **UID No** :

Subject Name : SECURITY LAB **Course Code** : ITL502

Assessment Parameters for Practical's/Tutorials/Assignment

Experiments	Conceptual Understanding	Performance	Self Learning Initiative	Ethical Behavior	Punctuality	Total
Ass1						
Ass2						
Ass3 (QB)						

Total marks can be allotted in the range from 1 to 3 for all points

Subject In-charge

Student

Experiment No. 6

Cryptographic Hash Function: SHA

Aim: To implement Cryptographic Hash Function: SHA

Objective: Cryptographic Hash Functions and Applications (HMAC): to understand the need, design and applications of collision resistant hash functions.

Theory:

Hash functions are extremely useful and appear in almost all information security applications.

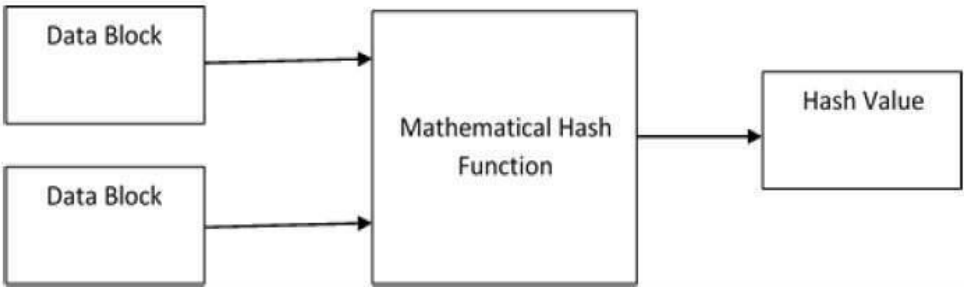
A hash function is a mathematical function that converts a numerical input value into another compressed numerical value. The input to the hash function is of arbitrary length but output is always of fixed length.

Values returned by a hash function are called **message digest** or simply **hash values**.

Design of Hashing Algorithms

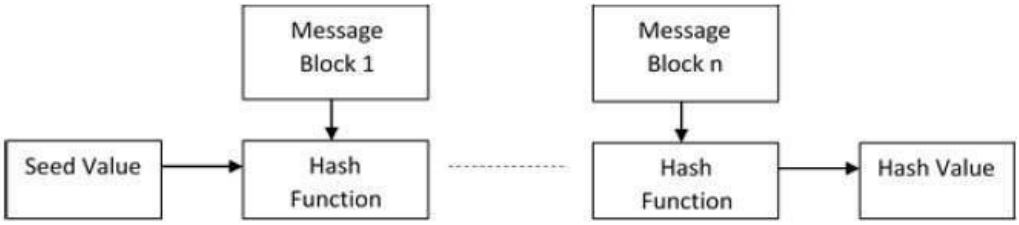
At the heart of a hashing is a mathematical function that operates on two fixed-size blocks of data to create a hash code. This hash function forms the part of the hashing algorithm.

The size of each data block varies depending on the algorithm. Typically the block sizes are from 128 bits to 512 bits. The following illustration demonstrates hash function −



Hashing algorithm involves rounds of above hash function like a block cipher. Each round takes an input of a fixed size, typically a combination of the most recent message block and the output of the last round.

This process is repeated for as many rounds as are required to hash the entire message. Schematic of hashing algorithm is depicted in the following illustration −



Since, the hash value of first message block becomes an input to the second hash operation, output of which alters the result of the third operation, and so on. This effect, known as an **avalanche** effect of hashing.

Popular Hash Functions

Message Digest (MD)

Secure Hash Function (SHA)

What is SHA-256?

The SHA-256 algorithm is one flavor of SHA-2 (Secure Hash Algorithm 2), which was created by the National Security Agency in 2001 as a successor to SHA-1. SHA-256 is a patented cryptographic hash function that outputs a value that is 256 bits long.

In cryptographic hashing, the hashed data is modified in a way that makes it completely unreadable. It would be virtually impossible to convert the 256-bit hash mentioned above back to its original 512-bit form. So why would you want to create a scrambled message that can't be recovered? The most common reason is to verify the content of data that must be kept secret. For example, hashing is used to verify the integrity of secure messages and files. The hash code of a secure file can be posted publicly so users who download the file can confirm they have an authentic version without the contents of the file being revealed. Hashes are similarly used to verify digital signatures.

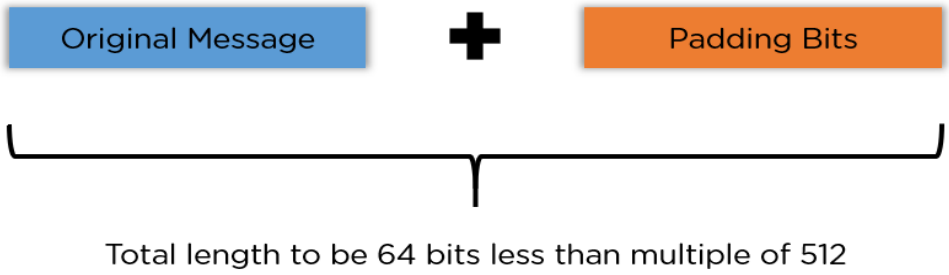
Password verification is a particularly important application for cryptographic hashing. Storing users' passwords in a plain-text document is a recipe for disaster; any hacker that manages to access the document would discover a treasure trove of unprotected passwords. That's why it's more secure to store the hash values of passwords instead. When a user enters a password, the hash value is calculated and then compared with the table. If it matches one of the saved hashes, it's a valid password and the user can be permitted access.

Three properties make SHA-256 this secure. First, it is almost impossible to reconstruct the initial data from the hash value. A brute-force attack would need to make 2^{256} attempts to generate the initial data. Second, having two messages with the same hash value (called a collision) is extremely unlikely. With 2^{256} possible hash values (more than the number of atoms in the known universe), the likelihood of two being the same is infinitesimally, unimaginably small. Finally, a minor change to the original data alters the hash value so much that it's not apparent the new hash value is derived from similar data; this is known as the avalanche effect.

Steps in SHA-256 Algorithm

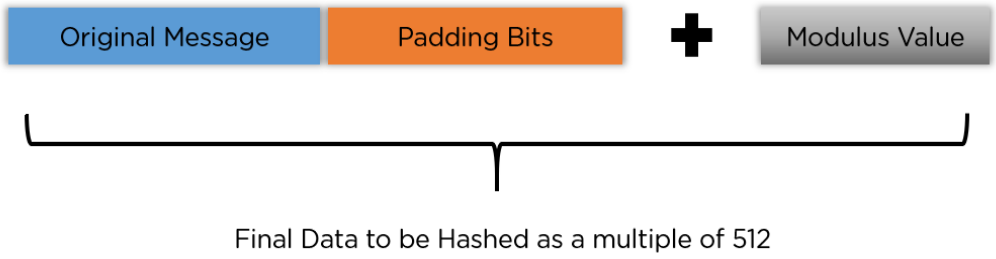
Padding Bits

It adds some extra bits to the message, such that the length is exactly 64 bits short of a multiple of 512. During the addition, the first bit should be one, and the rest of it should be filled with zeroes.



Padding Length

You can add 64 bits of data now to make the final plaintext a multiple of 512. You can calculate these 64 bits of characters by applying the modulus to your original cleartext without the padding.



Initialising the Buffers:

You need to initialize the default values for eight buffers to be used in the rounds as follows:

a = 0x6a09e667

b = 0xbb67ae85

c = 0x3c6ef372

d = 0xa54ff53a

e = 0x510e527f

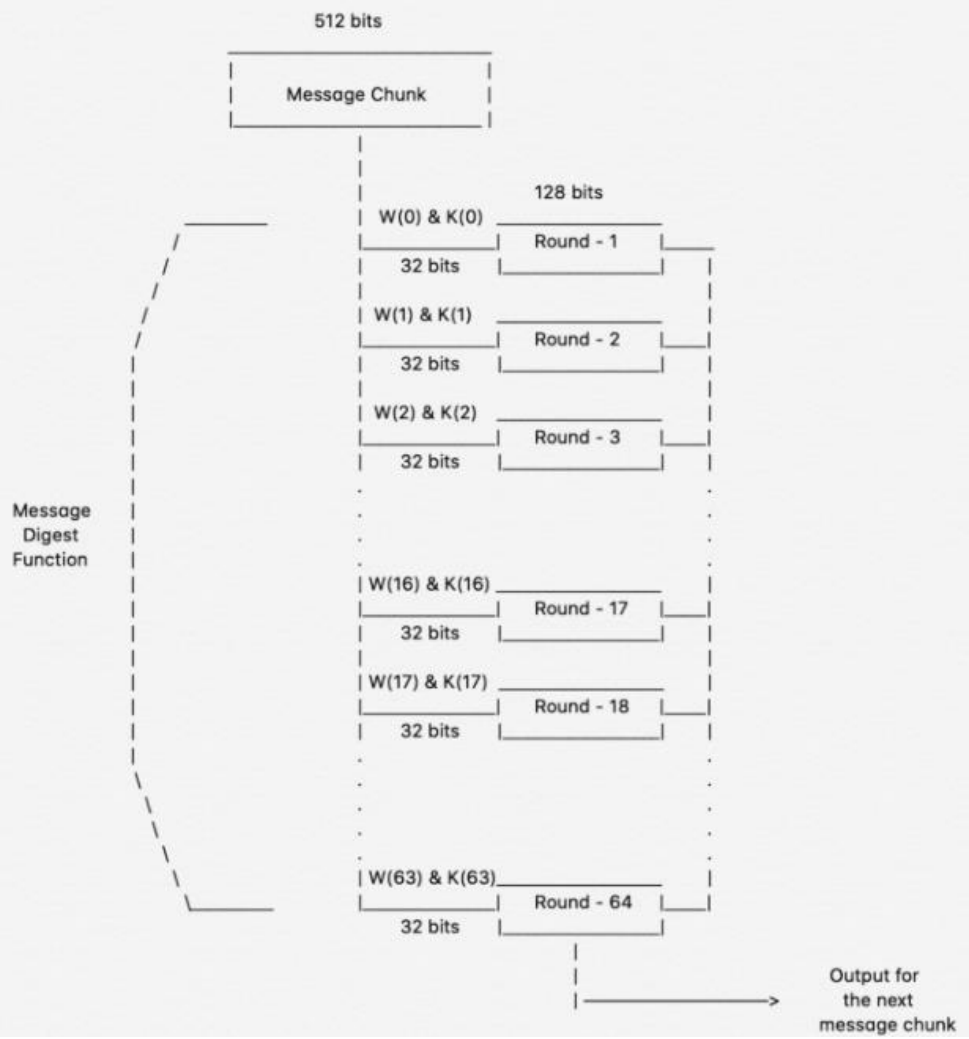
f = 0x9b05688c

g = 0x1f83d9ab

h = 0x5be0cd19

Compression Functions

The entire message gets broken down into multiple blocks of 512 bits each. It puts each block through 64 rounds of operation, with the output of each block serving as the input for the following block. The entire process is as follows:



Results:

SHA.java

```
// Java program to calculate SHA-1 hash value

import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class SHA {
	public static String encryptThisString(String input)
	{
		try {
			// getInstance() method is called with algorithm SHA-1
			MessageDigest md = MessageDigest.getInstance("SHA-1");

			// digest() method is called
			// to calculate message digest of the input string
			// returned as array of byte
			byte[] messageDigest = md.digest(input.getBytes());

			// Convert byte array into signum representation
			BigInteger no = new BigInteger(1, messageDigest);

			// Convert message digest into hex value
			String hashtext = no.toString(16);

			// Add preceding 0s to make it 32 bit
			while (hashtext.length() < 32) {
				hashtext = "0" + hashtext;
			}

			// return the HashText
			return hashtext;
		}

		// For specifying wrong message digest algorithms
		catch (NoSuchAlgorithmException e) {
			throw new RuntimeException(e);
		}
	}

	// Driver code
	public static void main(String args[]) throws
                                                        NoSuchAlgorithmException
	{
```

```
                System.out.println("HashCode Generated by SHA-1 for: ");

                String s1 = "ThisIsExperimentSixth";
                System.out.println("\n" + s1 + " : " + encryptThisString(s1));

                String s2 = "hello world";
                System.out.println("\n" + s2 + " : " + encryptThisString(s2));
        }
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

D:\AAKASH\MGM COLLEGE OF ENGINEERING\Sem-5\Security Practical\6\SHA>javac SHA.java

D:\AAKASH\MGM COLLEGE OF ENGINEERING\Sem-5\Security Practical\6\SHA>java SHA
HashCode Generated by SHA-1 for:

ThisIsExperimentSixth : 3fd882e4baa64c42826fba624f4908078be49aa8

hello world : 2aae6c35c94fcfb415dbe95f408b9ce91ee846ed
```

Conclusion:

Industrial Applications:

- Digital Signature Verification: Digital signatures follow asymmetric encryption methodology to verify the authenticity of a document/file. Hash algorithms like SHA 256 go a long way in ensuring the verification of the signature.
- Password Hashing: As discussed above, websites store user passwords in a hashed format for two benefits. It helps foster a sense of privacy, and it lessens the load on the central database since all the digests are of similar size.
- SSL Handshake: The SSL handshake is a crucial segment of the web browsing sessions, and it's done using SHA functions. It consists of your web browsers and the web servers agreeing on encryption keys and hashing authentication to prepare a secure connection.
- Integrity Checks: As discussed above, verifying file integrity has been using variants like SHA 256 algorithm and the MD5 algorithm. It helps maintain the full value functionality of files and makes sure they were not altered in transit.

Qustionnaire:

1. What is SHA?

2. What is the number of round computation steps in the SHA-256 algorithm?
a) 80 b) 76 c) 64 d) 70

3. When a hash function is used to provide message authentication, the hash function value is referred to as
a) Message Field b) Message Digest c) Message Score d) Message Leap

4. Message authentication code is also known as
a) key code b) hash code c) keyed hash function d) message key hash function

5. Which one of the following is not an application hash functions?
a) One-way password file b) Key wrapping c) Virus Detection d) Intrusion detection

Ex. No: 7	Study the use of network reconnaissance tools

AIM:

Study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup to gather information about networks and domain registrars.

Theory:

Study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup to gather information about networks and domain registrars.

1. **traceroute:**

The traceroute, tracert, or tracepath command is similar to ping, but provides information about the path a packet takes. traceroute sends packets to a destination, asking each Internet router along the way to reply when it passes on the packet. This will show you the path packets take when you send them between your location and a destination.

In computing, traceroute is a computer network diagnostic tool for displaying the route (path) and measuring transit delays of packets across an Internet Protocol (IP) network. The history of the route is recorded as the round-trip times of the packets received from each successive host (remote node) in the route (path); the sum of the mean times in each hop indicates the total time spent to establish the connection. Traceroute proceeds unless all (three) sent packets are lost more than twice, then the connection is lost and the route cannot be evaluated.

Installation: sudo apt-get install traceroute Commands: traceroute google.com

2. **whois:**

The whois command looks up the registration record associated with a domain name. This can show you more information about who registered and owns a domain name, including their contact information.

Installation: sudo apt-get install whois Commands: whois google.com

The WHOIS protocol had its origin in the ARPANET NICNAME protocol and was based on the NAME/FINGER Protocol, described in RFC 742 (1977). The NICNAME/WHOIS protocol was first described in RFC 812 in 1982 by Ken Harrenstien and Vic White of the Network Information Centre at SRI International.

WHOIS was originally implemented on the Network Control Program (NCP) but found its major use when the TCP/IP suite was standardized across the ARPANET and later the Internet.

```
root@kali:~# whois google.com

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

Aborting search 50 records found .....
   Server Name: GOOGLE.COM.AFRICANBATS.ORG
   Registrar: TUCOWS DOMAINS INC.
   Whois Server: whois.tucows.com
   Referral URL: http://www.tucowsdomains.com

   Server Name: GOOGLE.COM.ANGRYPIRATES.COM
   IP Address: 8.8.8.8
   Registrar: NAME.COM, INC.
   Whois Server: whois.name.com
   Referral URL: http://www.name.com

   Server Name: GOOGLE.COM.AR
   Registrar: ENOM, INC.
   Whois Server: whois.enom.com
   Referral URL: http://www.enom.com

   Server Name: GOOGLE.COM.AU
   Registrar: PLANETDOMAIN PTY LTD.
   Whois Server: whois.planetdomain.com
   Referral URL: http://www.planetdomain.com

   Server Name: GOOGLE.COM.BAISAD.COM
   IP Address: 92.53.96.24
   IP Address: 91.218.229.20
   Registrar: REGISTRAR OF DOMAIN NAMES REG.RU LLC
   Whois Server: whois.reg.com
   Referral URL: http://www.reg
```

3. nslookup

The nslookup command will look up the IP addresses associated with a domain name. For example, you can run nslookup howtogeek.com to see the IP address of How-To Geek's server

Installation: sudo apt-get install nslookup Commands: nslookup google.com

```
root@kali:~# nslookup google.com
Server:         192.168.223.2
Address:        192.168.223.2#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.196.14
```

4. dig

Dig is a networking tool that can query DNS servers for information. It can be very helpful for diagnosing problems with domain pointing and is a good way to verify that your configuration is working.

Installation: sudo apt-get install dig Commands: dig google.com

```
root@kali:~# dig google.com

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33983
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             5       IN      A       216.58.220.14

;; Query time: 81 msec
;; SERVER: 192.168.223.2#53(192.168.223.2)
;; WHEN: Thu Oct  8 00:00:38 2015
;; MSG SIZE  rcvd: 44
```

The dig command output has the following sections:

i. Header: This displays the dig command version number, the global options used by the dig command, and few additional header information.

ii. QUESTION SECTION: This displays the question it asked the DNS. i.e This is your input. Since we said 'dig redhat.com', and the default type dig command uses is A record, it indicates in this section that we asked for the A record of the redhat.com website

iii. ANSWER SECTION: This displays the answer it receives from the DNS. i.e This is your output. This displays the A record of redhat.com

iv. AUTHORITY SECTION: This displays the DNS name server that has theauthority to respond to this query. Basically, this displays available name servers of redhat.com

v. ADDITIONAL SECTION: This displays the ip address of the name servers listed in the AUTHORITY SECTION.

vi. Stats section at the bottom displays few dig command statistics including how much time it took to execute this query.

To view all the record types (A, MX, NS, etc.), use ANY as the record typeas shown below.

```
root@kali:~# dig google.com any

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> google.com any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39659
;; flags: qr rd ra; QUERY: 1, ANSWER: 14, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                    IN      ANY

;; ANSWER SECTION:
google.com.             5       IN      A       216.58.220.14
google.com.             5       IN      AAAA    2404:6800:4009:804::200e
google.com.             5       IN      SOA     ns2.google.com. dns-admin.google.com. 104874572 900 900 1800 60
google.com.             5       IN      MX      50 alt4.aspmx.l.google.com.
google.com.             5       IN      NS      ns2.google.com.
google.com.             5       IN      TXT     "v=spf1 include:_spf.google.com ~all"
google.com.             5       IN      NS      ns4.google.com.
google.com.             5       IN      NS      ns3.google.com.
google.com.             5       IN      MX      30 alt2.aspmx.l.google.com.
google.com.             5       IN      TYPE257 \# 19 0005697373756573796D616E7465632E636F6D
google.com.             5       IN      MX      40 alt3.aspmx.l.google.com.
google.com.             5       IN      MX      10 aspmx.l.google.com.
google.com.             5       IN      MX      20 alt1.aspmx.l.google.com.
google.com.             5       IN      NS      ns1.google.com.

;; Query time: 99 msec
;; SERVER: 192.168.223.2#53(192.168.223.2)
;; WHEN: Thu Oct  8 00:02:34 2015
;; MSG SIZE  rcvd: 377
```

Result:

1. Tracert

C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aakas>tracert google.com

Tracing route to google.com [142.250.182.238]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.0.1
  2     1 ms     1 ms     1 ms  1.186.179.1.dvois.com [1.186.179.1]
  3     1 ms     1 ms     1 ms  114.79.129.97.dvois.com [114.79.129.97]
  4     3 ms     2 ms     2 ms  72.14.208.165
  5     4 ms     4 ms     4 ms  142.251.76.27
  6     2 ms     2 ms     2 ms  142.250.214.105
  7     2 ms     2 ms     2 ms  bom07s29-in-f14.1e100.net [142.250.182.238]

Trace complete.
```

2. Whois

C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aakas\Desktop\IP\WhoIs>whois google.com

Whois v1.21 - Domain information lookup
Copyright (C) 2005-2019 Mark Russinovich
Sysinternals - www.sysinternals.com

Connecting to COM.whois-servers.net...

WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2019-09-09T15:39:04Z
   Creation Date: 1997-09-15T04:00:00Z
   Registry Expiry Date: 2028-09-14T04:00:00Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2086851750
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1.GOOGLE.COM
   Name Server: NS2.GOOGLE.COM
   Name Server: NS3.GOOGLE.COM
   Name Server: NS4.GOOGLE.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2022-09-22T17:22:20Z <<<

For more information on Whois status codes, please visit https://icann.org/epp
```

3. Nslookup

C:\Windows\System32\cmd.exe

```
C:\Users\Aakas\Desktop\IP\WhoIs>nslookup google.com
Server:  UnKnown
Address:  192.168.0.1

Non-authoritative answer:
Name:    google.com
Addresses:  2404:6800:4009:81c::200e
          172.217.174.78
```

4. Dig

C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aakas\Desktop\IP\Bind>dig google.com

; <<>> DiG 9.16.33 <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54217
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             273     IN      A       142.250.199.142

;; AUTHORITY SECTION:
google.com.             150827  IN      NS      ns4.google.com.
google.com.             150827  IN      NS      ns1.google.com.
google.com.             150827  IN      NS      ns3.google.com.
google.com.             150827  IN      NS      ns2.google.com.

;; ADDITIONAL SECTION:
ns3.google.com.         150979  IN      A       216.239.36.10
ns3.google.com.         211754  IN      AAAA    2001:4860:4802:36::a
ns2.google.com.         174681  IN      A       216.239.34.10
ns2.google.com.         150827  IN      AAAA    2001:4860:4802:34::a
ns1.google.com.         151618  IN      A       216.239.32.10
ns1.google.com.         153719  IN      AAAA    2001:4860:4802:32::a
ns4.google.com.         174681  IN      A       216.239.38.10
ns4.google.com.         150827  IN      AAAA    2001:4860:4802:38::a

;; Query time: 0 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Thu Sep 22 23:01:20 India Standard Time 2022
;; MSG SIZE  rcvd: 303
```

Ex. No: 8	Study of packet sniffer tools wireshark:

AIM:

Study of packet sniffer tools wireshark: - a. Observer performance in promiscuous as well as non-promiscuous mode. b. Show the packets can be traced based on different filters.

Theory:

Wireshark :-

Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 dueto trademark issues.

Wireshark is cross-platform, using the GTK+ widget toolkit in current releases, and Qt in the development version, to implement its user interface, and using pcap to capture packets; it runs on Linux, OS X, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows. There is also a terminal-based (non-GUI) version called TShark. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of the GNU General Public License.

Functionality of wireshark:-

Wireshark lets the user put network interface controllers that support promiscuous mode into that mode, so they can see all traffic visible on that interface, not just traffic addressed to one of the interface's configured addresses and broadcast/multicast traffic. However, when capturing with a packet analyzer in promiscuous mode on a port on a network switch, not all traffic through the switch is necessarily sent to the port where the capture is done, so capturing in promiscuous mode is not necessarily sufficient to see all network traffic.

Wireshark uses pcap to capture packets, so it can only capture packets on the types of networks that pcap supports.

i. Data can be captured "from the wire" from a live network connection or read from a file ofalready-captured packets.

ii. Live data can be read from a number of types of network, including Ethernet, IEEE 802.11, PPP, and loopback.

iii. Captured network data can be browsed via a GUI, or via the terminal (command line)version of the utility, TShark.

iv. Captured files can be programmatically edited or converted via command-line switchesto the "editcap" program.

v. Data display can be refined using a display filter.

vi. Plug-ins can be created for dissecting new protocols.

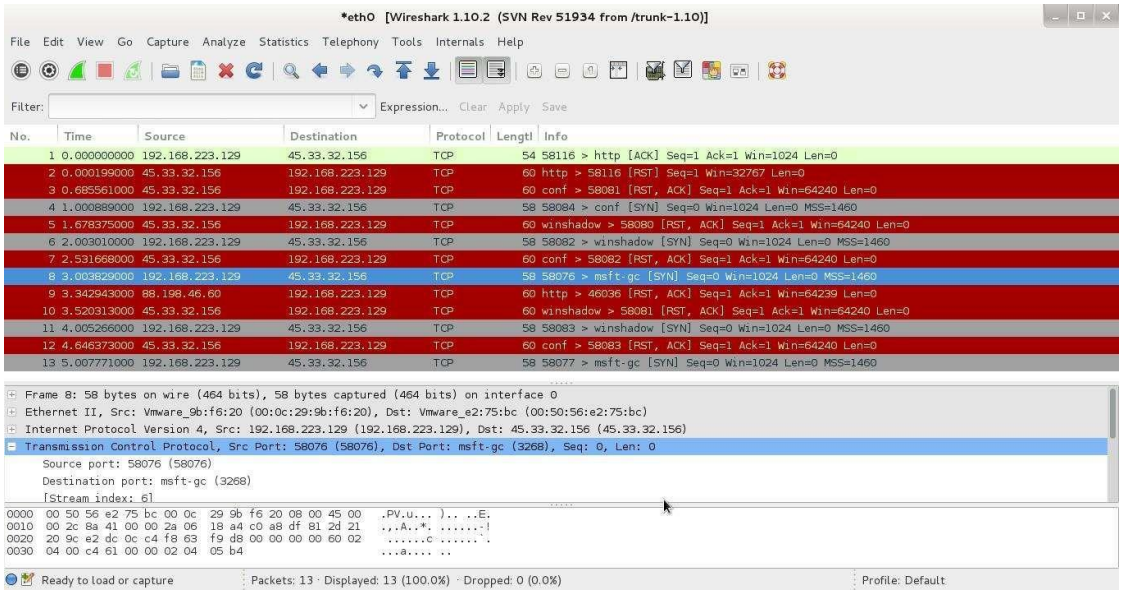
vii. VoIP calls in the captured traffic can be detected. If encoded in a compatible encoding,the media flow can even be played.

viii. Raw USB traffic can be captured.

ix. Wireless connections can also be filtered as long as they transverse the monitored Ethernet.

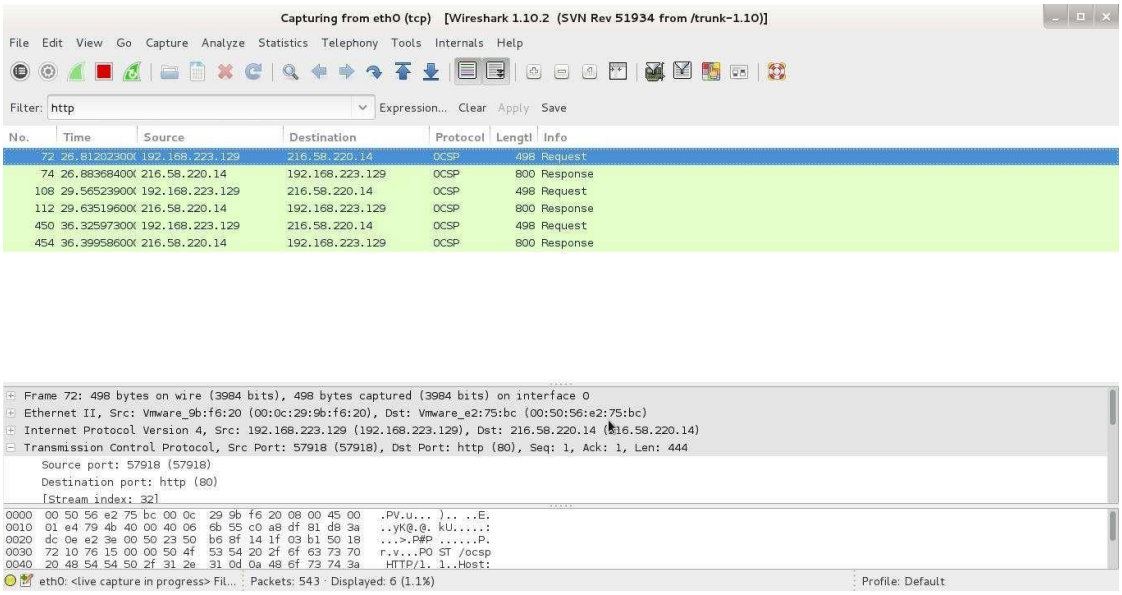
x. Various settings, timers, and filters can be set that ensure only triggered traffic appear.

Packets captured in promiscuous mode.(Packets are captured in promiscuous mode by default.)



Settings for applying filter and to set the mode:

Packets captured in non-promiscuous mode and with http filter :



Tcpdump :

tcpdump is a common packet analyzer that runs under the command line. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached. Distributed under the BSD license, tcpdump is free software.

Tcpdump works on most Unix-like operating systems: Linux, Solaris, BSD, OS X, HP- UX, Android and AIX among others. In those systems, tcpdump uses the libpcap library to capture packets. The port of tcpdump for Windows is called WinDump; it uses WinPcap, the Windows port of libpcap.

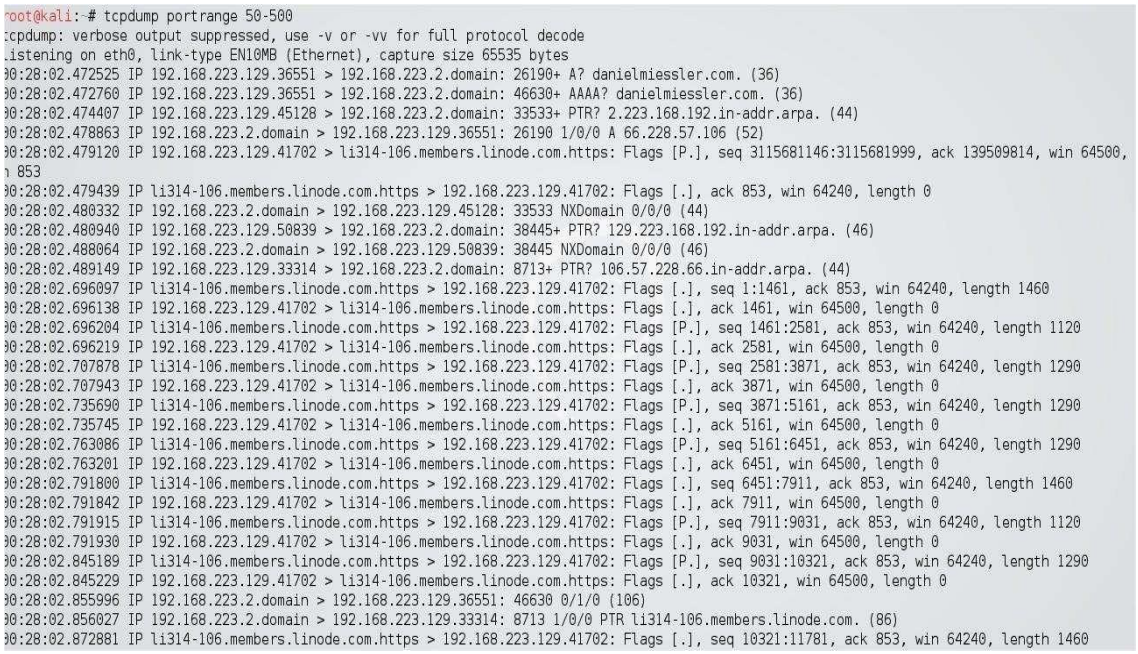
Command : tcpdump

```
root@kali:~# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
00:21:16.146887 IP scanme.nmap.org.9999 > 192.168.223.129.58084: Flags [R.], seq 177890604, ack 4167825873, win 64240, length 0
00:21:16.149056 IP 192.168.223.129.46954 > 192.168.223.2.domain: 16056+ PTR? 129.223.168.192.in-addr.arpa. (46)
00:21:16.155847 IP 192.168.223.2.domain > 192.168.223.129.46954: 16056 NXDomain 0/0/0 (46)
00:21:16.178165 IP 192.168.223.129.37669 > 192.168.223.2.domain: 26238+ PTR? 156.32.33.45.in-addr.arpa. (43)
00:21:16.240842 IP 192.168.223.2.domain > 192.168.223.129.37669: 26238 1/0/0 PTR scanme.nmap.org. (72)
00:21:16.241866 IP 192.168.223.129.58928 > 192.168.223.2.domain: 21169+ PTR? 2.223.168.192.in-addr.arpa. (44)
00:21:16.248125 IP 192.168.223.2.domain > 192.168.223.129.58928: 21169 NXDomain 0/0/0 (44)
00:21:16.677053 IP 192.168.223.129.58080 > scanme.nmap.org.8701: Flags [S], seq 4167563740, win 1024, options [mss 1460], length 0
00:21:17.139072 IP scanme.nmap.org.8701 > 192.168.223.129.58076: Flags [R.], seq 1957823006, ack 4167301593, win 64240, length 0
00:21:17.677741 IP 192.168.223.129.58081 > scanme.nmap.org.8701: Flags [S], seq 4167498205, win 1024, options [mss 1460], length 0
00:21:18.682095 IP 192.168.223.129.58082 > scanme.nmap.org.8701: Flags [S], seq 4167694802, win 1024, options [mss 1460], length 0
00:21:19.174265 IP scanme.nmap.org.8701 > 192.168.223.129.58077: Flags [R.], seq 415924006, ack 4167236058, win 64240, length 0
00:21:19.684682 IP 192.168.223.129.58083 > scanme.nmap.org.8701: Flags [S], seq 4167629267, win 1024, options [mss 1460], length 0
00:21:20.162911 IP scanme.nmap.org.8701 > 192.168.223.129.58078: Flags [R.], seq 1466125676, ack 4167432671, win 64240, length 0
00:21:20.686862 IP 192.168.223.129.58084 > scanme.nmap.org.8701: Flags [S], seq 4167825872, win 1024, options [mss 1460], length 0
00:21:21.235184 IP scanme.nmap.org.8701 > 192.168.223.129.58079: Flags [R.], seq 1292259147, ack 4167367136, win 64240, length 0
00:21:21.689266 ARP, Request who-has 192.168.223.2 tell 192.168.223.129, length 28
00:21:21.690413 ARP, Reply 192.168.223.2 is-at 00:50:56:e2:75:bc (oui Unknown), length 46
00:21:21.690693 IP 192.168.223.129.58076 > scanme.nmap.org.49157: Flags [S], seq 4167301592, win 1024, options [mss 1460], length 0
^C
19 packets captured
20 packets received by filter
0 packets dropped by kernel
```

Tcpdump done on a port range :

Command : tcpdump port range 50-

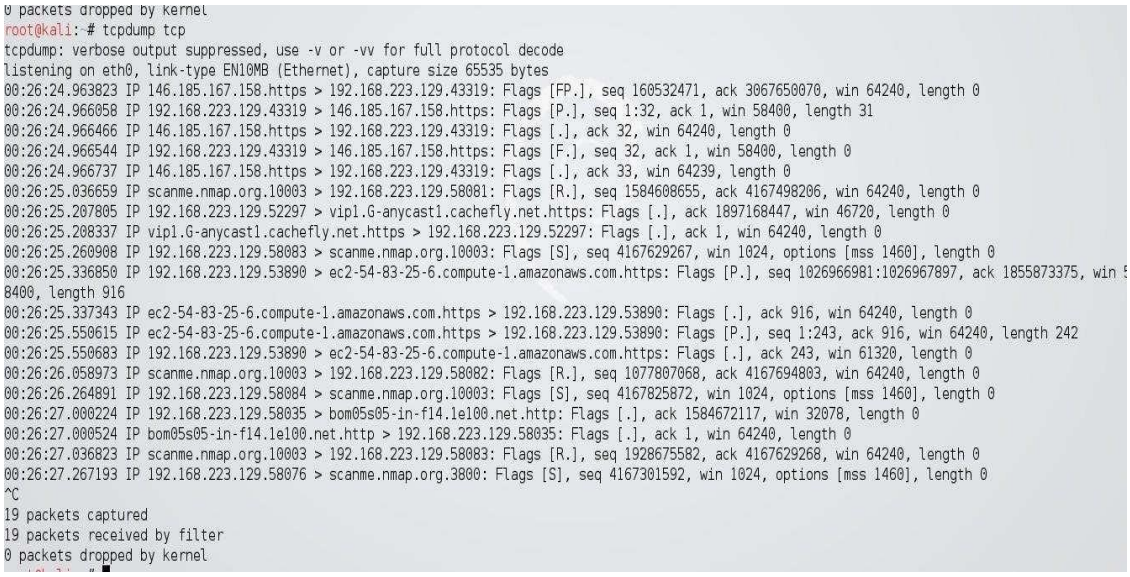
500



```
root@kali:~# tcpdump portrange 50-500
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
00:28:02.472525 IP 192.168.223.129.36551 > 192.168.223.2.domain: 26190+ A? danielmiessler.com. (36)
00:28:02.472760 IP 192.168.223.129.36551 > 192.168.223.2.domain: 46630+ AAAA? danielmiessler.com. (36)
00:28:02.474407 IP 192.168.223.129.45128 > 192.168.223.2.domain: 33533+ PTR? 2.223.168.192.in-addr.arpa. (44)
00:28:02.478863 IP 192.168.223.2.domain > 192.168.223.129.36551: 26190 1/0/0 A 66.228.57.106 (52)
00:28:02.479120 IP 192.168.223.129.41702 > li314-106.members.linode.com.https: Flags [P.], seq 3115681146:3115681999, ack 139509814, win 64500,
n 853
00:28:02.479439 IP li314-106.members.linode.com.https > 192.168.223.129.41702: Flags [.], ack 853, win 64240, length 0
00:28:02.480332 IP 192.168.223.2.domain > 192.168.223.129.45128: 33533 NXDomain 0/0/0 (44)
00:28:02.480940 IP 192.168.223.129.50839 > 192.168.223.2.domain: 38445+ PTR? 129.223.168.192.in-addr.arpa. (46)
00:28:02.488064 IP 192.168.223.2.domain > 192.168.223.129.50839: 38445 NXDomain 0/0/0 (46)
00:28:02.489149 IP 192.168.223.129.33314 > 192.168.223.2.domain: 8713+ PTR? 106.57.228.66.in-addr.arpa. (44)
00:28:02.696097 IP li314-106.members.linode.com.https > 192.168.223.129.41702: Flags [.], seq 1:1461, ack 853, win 64240, length 1460
00:28:02.696138 IP 192.168.223.129.41702 > li314-106.members.linode.com.https: Flags [.], ack 1461, win 64500, length 0
00:28:02.696204 IP li314-106.members.linode.com.https > 192.168.223.129.41702: Flags [P.], seq 1461:2581, ack 853, win 64240, length 1120
00:28:02.696219 IP 192.168.223.129.41702 > li314-106.members.linode.com.https: Flags [.], ack 2581, win 64500, length 0
00:28:02.707878 IP li314-106.members.linode.com.https > 192.168.223.129.41702: Flags [P.], seq 2581:3871, ack 853, win 64240, length 1290
00:28:02.707943 IP 192.168.223.129.41702 > li314-106.members.linode.com.https: Flags [.], ack 3871, win 64500, length 0
00:28:02.735690 IP li314-106.members.linode.com.https > 192.168.223.129.41702: Flags [P.], seq 3871:5161, ack 853, win 64240, length 1290
00:28:02.735745 IP 192.168.223.129.41702 > li314-106.members.linode.com.https: Flags [.], ack 5161, win 64500, length 0
00:28:02.763086 IP li314-106.members.linode.com.https > 192.168.223.129.41702: Flags [P.], seq 5161:6451, ack 853, win 64240, length 1290
00:28:02.763201 IP 192.168.223.129.41702 > li314-106.members.linode.com.https: Flags [.], ack 6451, win 64500, length 0
00:28:02.791800 IP li314-106.members.linode.com.https > 192.168.223.129.41702: Flags [.], seq 6451:7911, ack 853, win 64240, length 1460
00:28:02.791842 IP 192.168.223.129.41702 > li314-106.members.linode.com.https: Flags [.], ack 7911, win 64500, length 0
00:28:02.791915 IP li314-106.members.linode.com.https > 192.168.223.129.41702: Flags [P.], seq 7911:9031, ack 853, win 64240, length 1120
00:28:02.791930 IP 192.168.223.129.41702 > li314-106.members.linode.com.https: Flags [.], ack 9031, win 64500, length 0
00:28:02.845189 IP li314-106.members.linode.com.https > 192.168.223.129.41702: Flags [P.], seq 9031:10321, ack 853, win 64240, length 1290
00:28:02.845229 IP 192.168.223.129.41702 > li314-106.members.linode.com.https: Flags [.], ack 10321, win 64500, length 0
00:28:02.855996 IP 192.168.223.2.domain > 192.168.223.129.36551: 46630 0/1/0 (106)
00:28:02.856027 IP 192.168.223.2.domain > 192.168.223.129.33314: 8713 1/0/0 PTR li314-106.members.linode.com. (86)
00:28:02.872881 IP li314-106.members.linode.com.https > 192.168.223.129.41702: Flags [.], seq 10321:11781, ack 853, win 64240, length 1460
```

Tcpdump using filters :

Command :



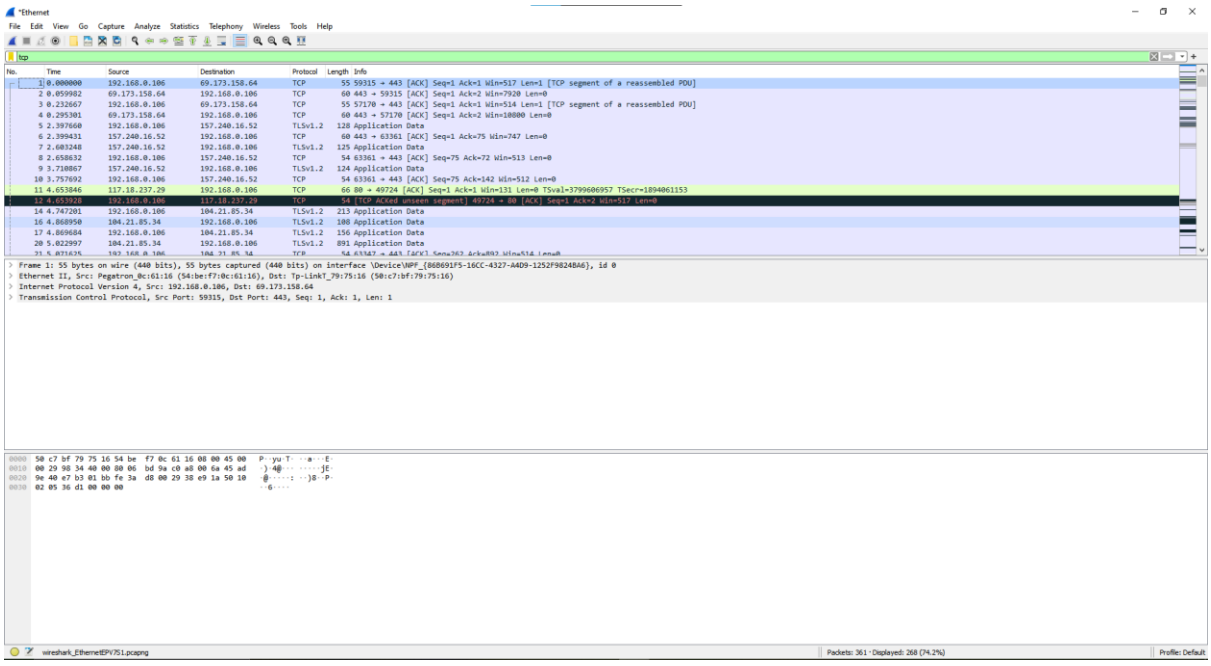
```
0 packets dropped by kernel
root@kali:~# tcpdump tcp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
00:26:24.963823 IP 146.185.167.158.https > 192.168.223.129.43319: Flags [FP.], seq 160532471, ack 3067650070, win 64240, length 0
00:26:24.966058 IP 192.168.223.129.43319 > 146.185.167.158.https: Flags [P.], seq 1:32, ack 1, win 58400, length 31
00:26:24.966466 IP 146.185.167.158.https > 192.168.223.129.43319: Flags [.], ack 32, win 64240, length 0
00:26:24.966544 IP 192.168.223.129.43319 > 146.185.167.158.https: Flags [F.], seq 32, ack 1, win 58400, length 0
00:26:24.966737 IP 146.185.167.158.https > 192.168.223.129.43319: Flags [.], ack 33, win 64239, length 0
00:26:25.036659 IP scanme.nmap.org.10003 > 192.168.223.129.58081: Flags [R.], seq 1584608655, ack 4167498206, win 64240, length 0
00:26:25.207805 IP 192.168.223.129.52297 > vip1.G-anycast1.cachefly.net.https: Flags [.], ack 1897168447, win 46720, length 0
00:26:25.208337 IP vip1.G-anycast1.cachefly.net.https > 192.168.223.129.52297: Flags [.], ack 1, win 64240, length 0
00:26:25.260908 IP 192.168.223.129.58083 > scanme.nmap.org.10003: Flags [S], seq 4167629267, win 1024, options [mss 1460], length 0
00:26:25.336850 IP 192.168.223.129.53890 > ec2-54-83-25-6.compute-1.amazonaws.com.https: Flags [P.], seq 1026966981:1026967897, ack 1855873375, win 5
8400, length 916
00:26:25.337343 IP ec2-54-83-25-6.compute-1.amazonaws.com.https > 192.168.223.129.53890: Flags [.], ack 916, win 64240, length 0
00:26:25.550615 IP ec2-54-83-25-6.compute-1.amazonaws.com.https > 192.168.223.129.53890: Flags [P.], seq 1:243, ack 916, win 64240, length 242
00:26:25.550683 IP 192.168.223.129.53890 > ec2-54-83-25-6.compute-1.amazonaws.com.https: Flags [.], ack 243, win 61320, length 0
00:26:26.058973 IP scanme.nmap.org.10003 > 192.168.223.129.58082: Flags [R.], seq 1077807068, ack 4167694803, win 64240, length 0
00:26:26.264891 IP 192.168.223.129.58084 > scanme.nmap.org.10003: Flags [S], seq 4167825872, win 1024, options [mss 1460], length 0
00:26:27.000224 IP 192.168.223.129.58035 > bom05s05-in-f14.1e100.net.http: Flags [.], ack 1584672117, win 32078, length 0
00:26:27.000524 IP bom05s05-in-f14.1e100.net.http > 192.168.223.129.58035: Flags [.], ack 1, win 64240, length 0
00:26:27.036823 IP scanme.nmap.org.10003 > 192.168.223.129.58083: Flags [R.], seq 1928675582, ack 4167629268, win 64240, length 0
00:26:27.267193 IP 192.168.223.129.58076 > scanme.nmap.org.3800: Flags [S], seq 4167301592, win 1024, options [mss 1460], length 0
^C
19 packets captured
19 packets received by filter
0 packets dropped by kernel
```

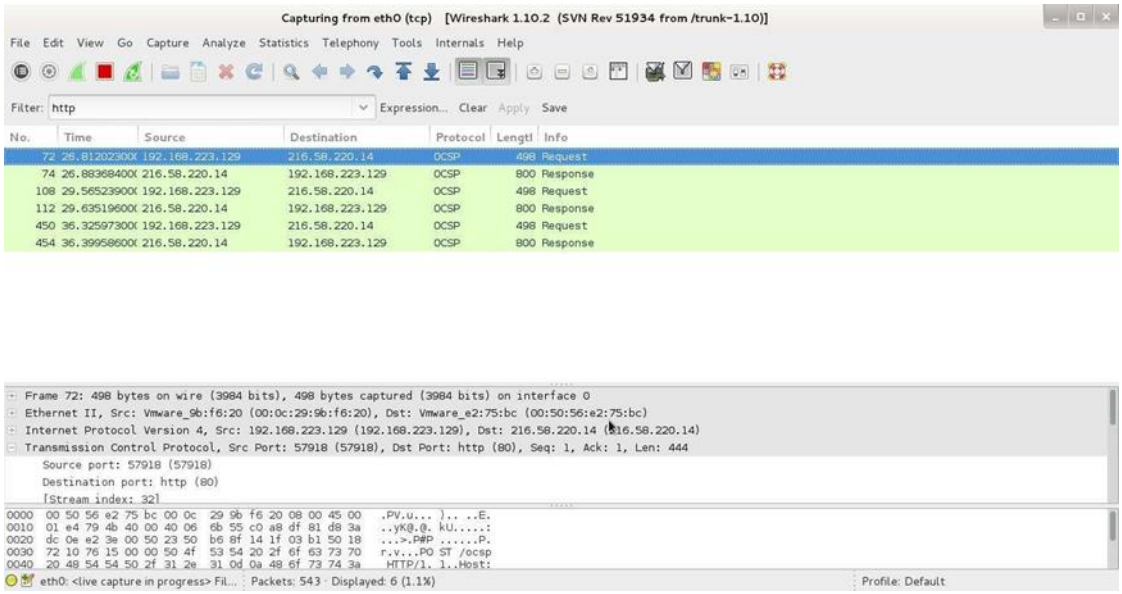
Tcpdump in non-promiscuous

mode :Command : tcpdump -p

```
root@kali:~# tcpdump -p
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
Listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
00:22:41.289850 ARP, Request who-has 192.168.223.2 tell 192.168.223.1, length 46
00:22:41.291542 IP 192.168.223.129.42939 > 192.168.223.2.domain: 53788+ PTR? 2.223.168.192.in-addr.arpa. (44)
00:22:41.297942 IP 192.168.223.2.domain > 192.168.223.129.42939: 53788 NXDomain 0/0/0 (44)
00:22:41.298803 IP 192.168.223.129.34567 > 192.168.223.2.domain: 62072+ PTR? 1.223.168.192.in-addr.arpa. (44)
00:22:41.305268 IP 192.168.223.2.domain > 192.168.223.129.34567: 62072 NXDomain 0/0/0 (44)
00:22:41.306193 IP 192.168.223.129.52313 > 192.168.223.2.domain: 43622+ PTR? 129.223.168.192.in-addr.arpa. (46)
00:22:41.313096 IP 192.168.223.2.domain > 192.168.223.129.52313: 43622 NXDomain 0/0/0 (46)
00:22:41.432584 IP 192.168.223.129.58002 > bom05s05-in-f14.1e100.net.http: Flags [.], ack 75386135, win 33570, length 0
00:22:41.434710 IP bom05s05-in-f14.1e100.net.http > 192.168.223.129.58002: Flags [.], ack 1, win 64240, length 0
00:22:41.440545 IP 192.168.223.129.60906 > 192.168.223.2.domain: 54905+ PTR? 14.220.58.216.in-addr.arpa. (44)
00:22:41.511785 IP 192.168.223.2.domain > 192.168.223.129.60906: 54905 4/0/0 PTR bom05s05-in-f14.1e100.net., PTR bom05s05-in-f14.1e100.net., PTR bom0
5s05-in-f14.1e100.net., PTR bom05s05-in-f14.1e100.net. (125)
00:22:41.586946 IP scanme.nmap.org.1021 > 192.168.223.129.58084: Flags [R.], seq 806036961, ack 4167825873, win 64240, length 0
00:22:41.587415 IP 192.168.223.129.42381 > 192.168.223.2.domain: 15054+ PTR? 156.32.33.45.in-addr.arpa. (43)
00:22:41.593669 IP 192.168.223.2.domain > 192.168.223.129.42381: 15054 1/0/0 PTR scanme.nmap.org. (72)
00:22:41.816093 IP 192.168.223.129.58077 > scanme.nmap.org.3052: Flags [S], seq 4167236057, win 1024, options [mss 1460], length 0
00:22:41.974795 ARP, Request who-has 192.168.223.2 tell 192.168.223.1, length 46
00:22:42.585700 IP scanme.nmap.org.3052 > 192.168.223.129.58076: Flags [R.], seq 1631808286, ack 4167301593, win 64240, length 0
00:22:42.816763 IP 192.168.223.129.58078 > scanme.nmap.org.3052: Flags [S], seq 4167432670, win 1024, options [mss 1460], length 0
00:22:42.974888 ARP, Request who-has 192.168.223.2 tell 192.168.223.1, length 46
00:22:43.599499 IP scanme.nmap.org.3052 > 192.168.223.129.58077: Flags [R.], seq 561654480, ack 4167236058, win 64240, length 0
00:22:43.817977 IP 192.168.223.129.58079 > scanme.nmap.org.3052: Flags [S], seq 4167367135, win 1024, options [mss 1460], length 0
^C
21 packets captured
21 packets received by filter
0 packets dropped by kernel
```

Result:




Capturing from eth0 (tcp) [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: http
Expression... Clear Apply Save
No. Time Source Destination Protocol Lengtl Info
72 28.81202300(192.168.223.129 216.58.220.14 OCSP 498 Request
74 26.88368400(216.58.220.14 192.168.223.129 OCSP 800 Response
108 29.56523900(192.168.223.129 216.58.220.14 OCSP 498 Request
112 29.63519600(216.58.220.14 192.168.223.129 OCSP 800 Response
450 36.32597300(192.168.223.129 216.58.220.14 OCSP 498 Request
454 36.39958600(216.58.220.14 192.168.223.129 OCSP 800 Response
Frame 72: 498 bytes on wire (3984 bits), 498 bytes captured (3984 bits) on interface 0
Ethernet II, Src: Vmware_9b:f6:20 (00:0c:29:9b:f6:20), Dst: Vmware_e2:75:bc (00:50:56:e2:75:bc)
Internet Protocol Version 4, Src: 192.168.223.129 (192.168.223.129), Dst: 216.58.220.14 (216.58.220.14)
Transmission Control Protocol, Src Port: 57918 (57918), Dst Port: http (80), Seq: 1, Ack: 1, Len: 444
Source port: 57918 (57918)
Destination port: http (80)
[Stream index: 32]
0000 00 50 56 e2 75 bc 00 0c 29 9b f6 20 08 00 45 00 .PV.u...)..
0010 01 e4 79 4b 40 00 40 06 6b 55 c0 a8 df 81 d8 3a ..yK@.@. kU.....:
0020 dc 0e e2 3e 00 50 23 50 b6 8f 14 1f 03 b1 50 18 ...>.P#P ......P.
0030 72 10 76 15 00 00 50 4f 53 54 20 2f 6f 63 73 70 r.v...PO ST /ocsp
0040 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a HTTP/1. 1..Host:
eth0: <live capture in progress> Fil... Packets: 543 Displayed: 6 (1.1%)
Profile: Default

Conclusion:

_____

_____

Ex. No: 9	To study and implement various scanning techniques using Nmap.

AIM:

Download, install nmap and use it with different options to scan open ports, performOS fingerprinting, ping scan, tcp port scan, udp port scan, etc.

Theory:

Theory:

Nmap (*Network Mapper*) is a security scanner originally written by Gordon Lyon (also known by his pseudonym *Fyodor Vaskovich*) used to discover hosts and services on a computer network, thus creating a "map" of the network. To accomplish its goal, Nmap sends specially crafted packets to the target host and then analyzes the responses. The software provides a number of features for probing computer networks, including host discovery and service and operating system detection. These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features. Nmap is also capable of adapting to network conditions including latency and congestion during a scan. Nmap is under development and refinement by its user community.

Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics.

It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff),and a packet generation and response analysis tool (Nping).

Features:

Nmap features include:

- Host discovery – Identifying hosts on a network. For example, listing the hosts thatrespond to TCP and/or ICMP requests or have a particular port open.
- Port scanning – Enumerating the open ports on target hosts.
- Version detection – Interrogating network services on remote devices to determineapplication name and version number.
- OS detection – Determining the operating system and hardware characteristics of network devices.
- Scriptable interaction with the target – using Nmap Scripting Engine (NSE) and Lua programming language.

Nmap can provide further information on targets, including reverse DNS names, device types, and MAC addresses.

Uses of Nmap:

- Auditing the security of a device or firewall by identifying the network connectionswhich can be made to, or through it.
- Identifying open ports on a target host in preparation for auditing.
- Network inventory, network mapping, and maintenance and asset management.
- Auditing the security of a network by identifying new servers.
- Generating traffic to hosts on a network.
- Find and exploit vulnerabilities in a network.

Port scanning:

A port scan or port scanning can be defined as a process that sends client requests to a range of server port addresses on a host, with the goal of finding an active port. While not a nefarious process in and of itself, it is one used by hackers to probe target machine services with the aim of exploiting a known vulnerability of that service, however the majority of usesof a port scan are not attacks and are simple probes to determine services available on a remote machine.

Fig.: PORT SCANNING.

Os fingerprinting:

TCP/IP stack fingerprinting is the passive collection of configuration attributes from a remote device during standard layer 4 network communications. The combination of parameters may then be used to infer the remote machine's operating system (aka, **OS fingerprinting**), or incorporated into a device fingerprint.

Certain parameters within the TCP protocol definition are left up to the implementation. Different operating systems, and different versions of the same operating system, set different defaults for these values. By collecting and examining these values, one may differentiate among various operating systems, and implementations of TCP/IP. The TCP/IP fields that may vary include the following:

- Initial packet size (16 bits)
- Initial TTL (8 bits)
- Window size (16 bits)
- Max segment size (16 bits)

- Window scaling value (8 bits)
- "don't fragment" flag (1 bit)
- "sackOK" flag (1 bit)
- "nop" flag (1 bit)

These values may be combined to form a 67-bit signature, or fingerprint, for the target machine. Just inspecting the Initial TTL and window size fields is often enough in order to successfully identify an operating system, which eases the task of performing manual OS fingerprinting.

Fig.: OS Fingerprinting.

Actual text of OS Fingerprinting:

root@kali:~# nmap -v -O scanme.nmap.org
Starting Nmap 6.47 (http://nmap.org) at 2015-10-08 08:03 IST Initiating Ping Scan at 08:03
Scanning scanme.nmap.org (45.33.32.156) [4 ports]
Completed Ping Scan at 08:03, 0.01s elapsed (1 total hosts) Initiating Parallel DNS resolution of 1 host. at 08:03
Completed Parallel DNS resolution of 1 host. at 08:03, 0.28s elapsed Initiating SYN Stealth Scan at 08:03
Scanning scanme.nmap.org (45.33.32.156) [1000 ports]
Discovered open port 22/tcp on 45.33.32.156
Discovered open port 80/tcp on 45.33.32.156
Discovered open port 31337/tcp on 45.33.32.156

tcp scan:

TCP connect scan is the default TCP scan type when SYN scan is not an option. This is the case when a user does not have raw packet privileges. Instead of writing raw packets as most other scan types do, Nmap asks the underlying operating system to establish a connectionwith the target machine and port by issuing the connect system call. This is the same high- level system call that web browsers, P2P clients, and most other network-enabled applications use to establish a connection. It is part of a programming interface known as the Berkeley Sockets API. Rather than read raw packet responses off the wire, Nmap uses this API to obtain status information on each connection attempt.

```
root@kali:~# nmap -sT scanme.nmap.org

Starting Nmap 6.47 ( http://nmap.org ) at 2015-10-08 01:32 IST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.00028s latency).
All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are filtered

Nmap done: 1 IP address (1 host up) scanned in 21.77 seconds
```

Fig.: TCP scan.

Udp scan:

While most popular services on the Internet run over the TCP protocol, UDP services are widely deployed. DNS, SNMP, and DHCP (registered ports 53, 161/162, and 67/68) are three of the most common. Because UDP scanning is generally slower and more difficult than TCP, some security auditors ignore these ports. This is a mistake, as exploitable UDP services are quite common and attackers certainly don't ignore the whole protocol. Fortunately, Nmap can help inventory UDP ports.

UDP scan is activated with the -sU option. It can be combined with a TCP scan type such as SYN scan (-sS) to check both protocols during the same run.

Nmap detects rate limiting and slows down accordingly to avoid flooding the network with useless packets that the target machine will drop. Unfortunately, a Linux-style limit of one packet per second makes a 65,536-port scan take more than 18 hours. Ideas for speeding your UDP scans up include scanning more hosts in parallel, doing a quick scan of just the popular ports first, scanning from behind the firewall, and using --host-timeout to skip slow hosts.

```
root@kali:~# nmap -sU scanme.nmap.org

Starting Nmap 6.47 ( http://nmap.org ) at 2015-10-08 01:33 IST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.021s latency).
Not shown: 999 open|filtered ports
PORT    STATE SERVICE
123/udp open  ntp

Nmap done: 1 IP address (1 host up) scanned in 14.57 seconds
```

Result:

Port scanning:

C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aakas>nmap -sP scanme.nmap.org google.com yahoo.in amazon.in
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-06 12:11 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.24s latency).
Nmap scan report for google.com (142.250.67.206)
Host is up (0.0010s latency).
rDNS record for 142.250.67.206: bom12s08-in-f14.1e100.net
Nmap scan report for yahoo.in (106.10.248.150)
Host is up (0.057s latency).
Other addresses for yahoo.in (not scanned): 212.82.100.150 124.108.115.100 74.6.136.150 98.136.103.23
rDNS record for 106.10.248.150: w2.src.vip.sg3.yahoo.com
Nmap scan report for amazon.in (52.95.120.67)
Host is up (0.13s latency).
Other addresses for amazon.in (not scanned): 54.239.33.92 52.95.116.115
Nmap done: 4 IP addresses (4 hosts up) scanned in 1.66 seconds
```

Os fingerprinting:

tcp scan:

C:\Windows\system32\cmd.exe

```
C:\Users\Aakas>nmap -sT scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-06 12:18 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.0080s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
110/tcp open  pop3
111/tcp open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 67.51 seconds
```

Udp scan:

C:\Windows\system32\cmd.exe

```
C:\Users\Aakas>nmap -sU scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-06 12:21 India Standard Time
Stats: 0:06:14 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 15.40% done; ETC: 13:01 (0:34:15 remaining)
Stats: 0:14:03 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 33.25% done; ETC: 13:03 (0:28:12 remaining)
Stats: 0:26:31 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 66.95% done; ETC: 13:00 (0:13:05 remaining)
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.24s latency).
Not shown: 999 open|filtered udp ports (no-response)
PORT    STATE SERVICE
123/udp open  ntp

Nmap done: 1 IP address (1 host up) scanned in 2326.75 seconds
```

Conclusion:

_____

_____

Experiment No 11

Snort

Aim: Set up snort and study the logs.

Theory:

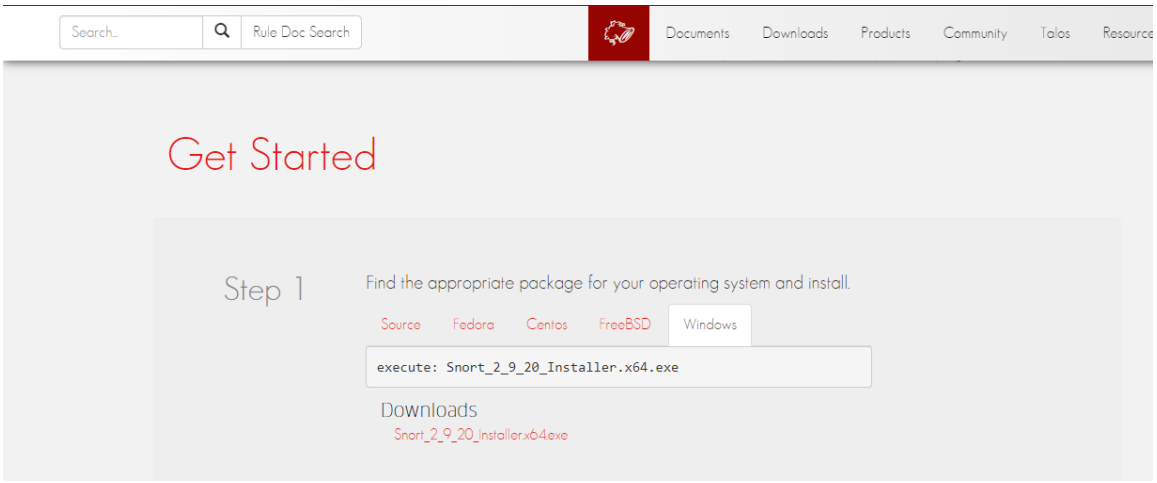
What is Snort?

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 can be deployed inline to stop these packets, as well. 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 can be downloaded and configured for personal and business use alike.

How to download Snort

Visit snort.org

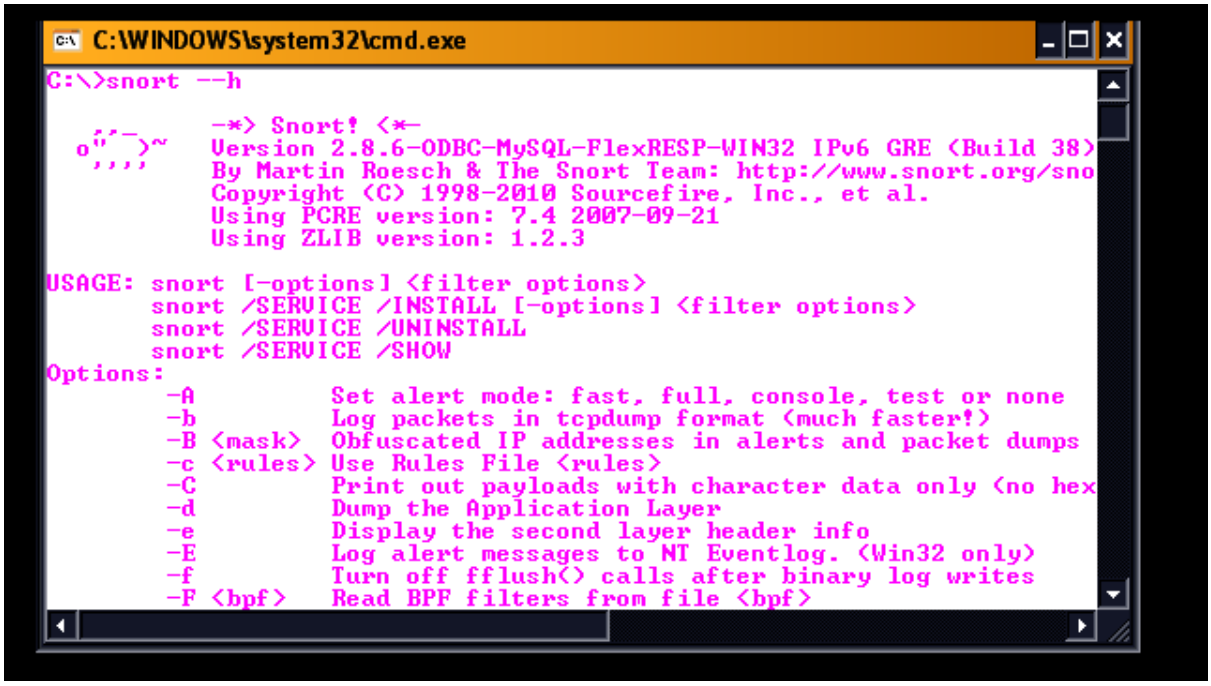


Snort commands:

After installing snort, open command prompt. Set the path in command prompt

Run the following commands.

1. Dir
2. Snort.exe
3. Snort –h



. Depending on your needs, Snort runs in three different modes: Sniffer, Packet Logger, and Network Intrusion modes.

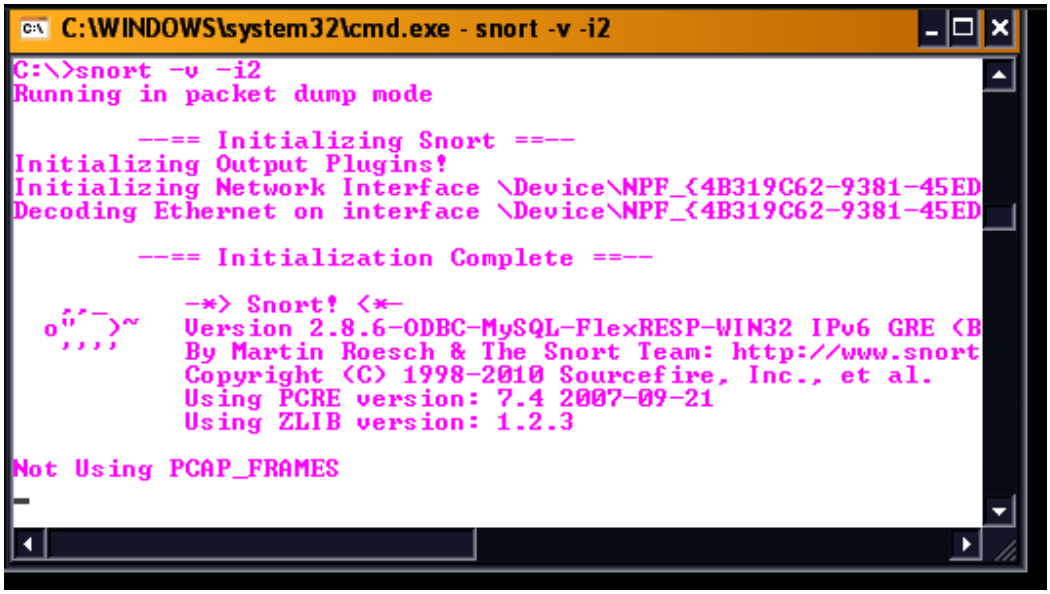
Snort in Sniffer mode

If you're running Snort from the command line with two network adapters, specify which adapter to monitor:

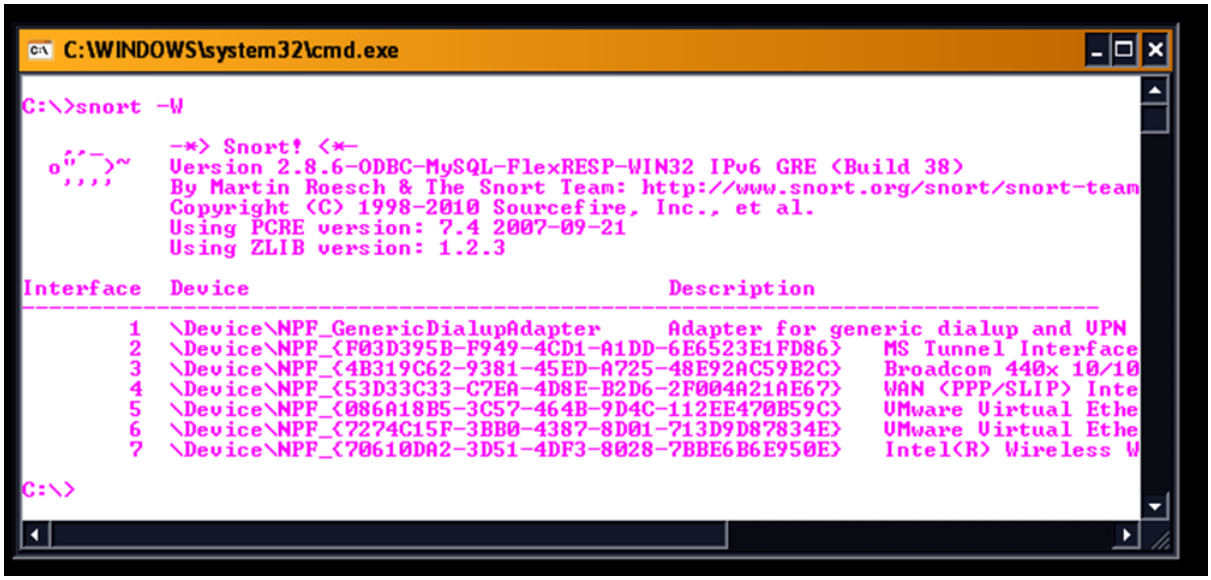
C:\>snort -v -i#

is the number of the applicable adapters (as shown on the output of the **snort -W** command).
You must use this -i switch whenever you run the snort program on the command line. Sniffer mode is the simplest iteration of Snort. To run it, follow these steps: from the command line (within the **%SnortPath%\bin** directory and in our case, we can run it from any Windows path) type:

C:\>snort -v -i2



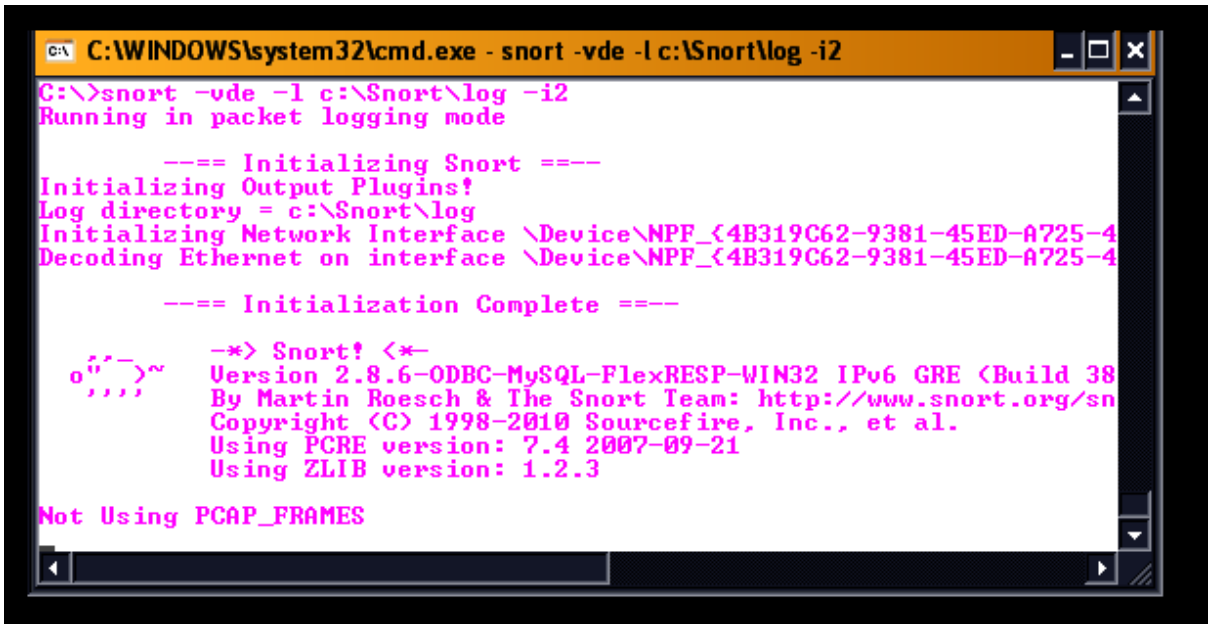
C:\>snort –W



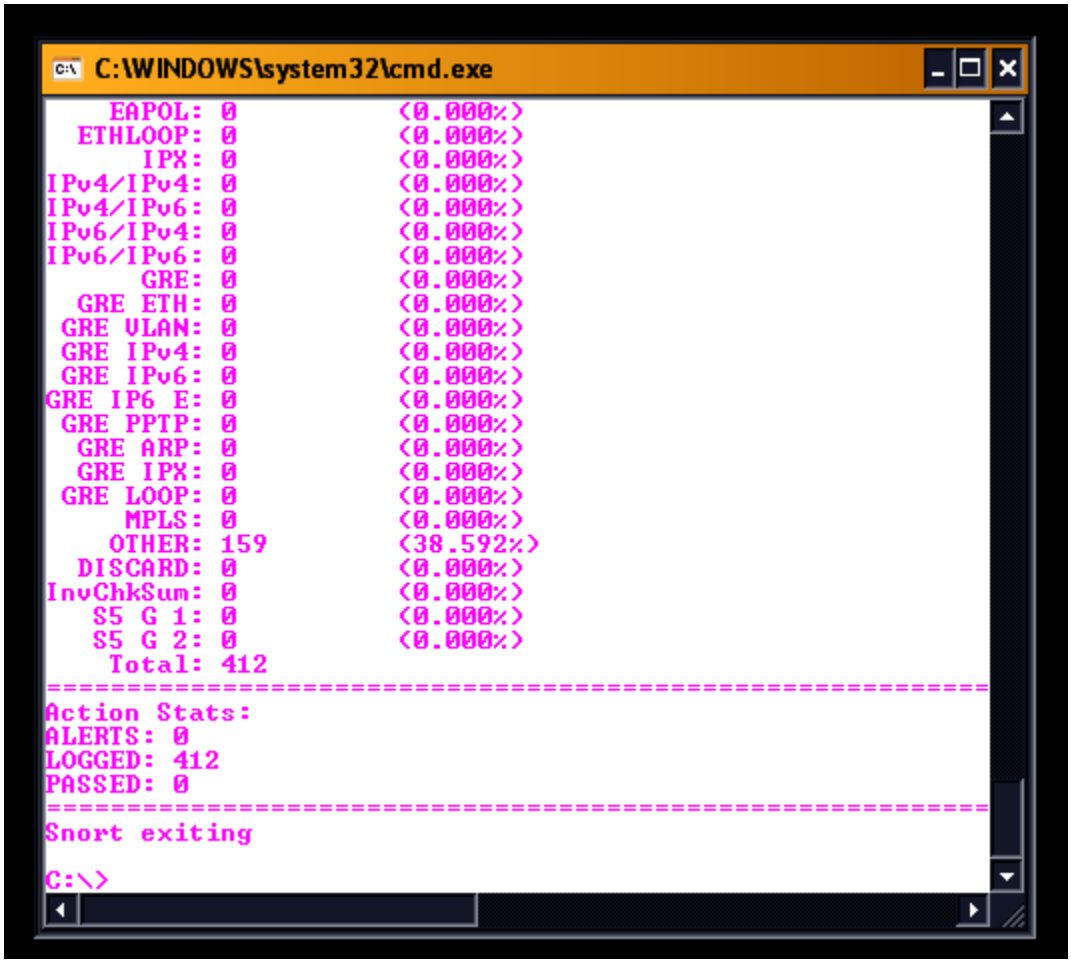
Snort as Packet Logger

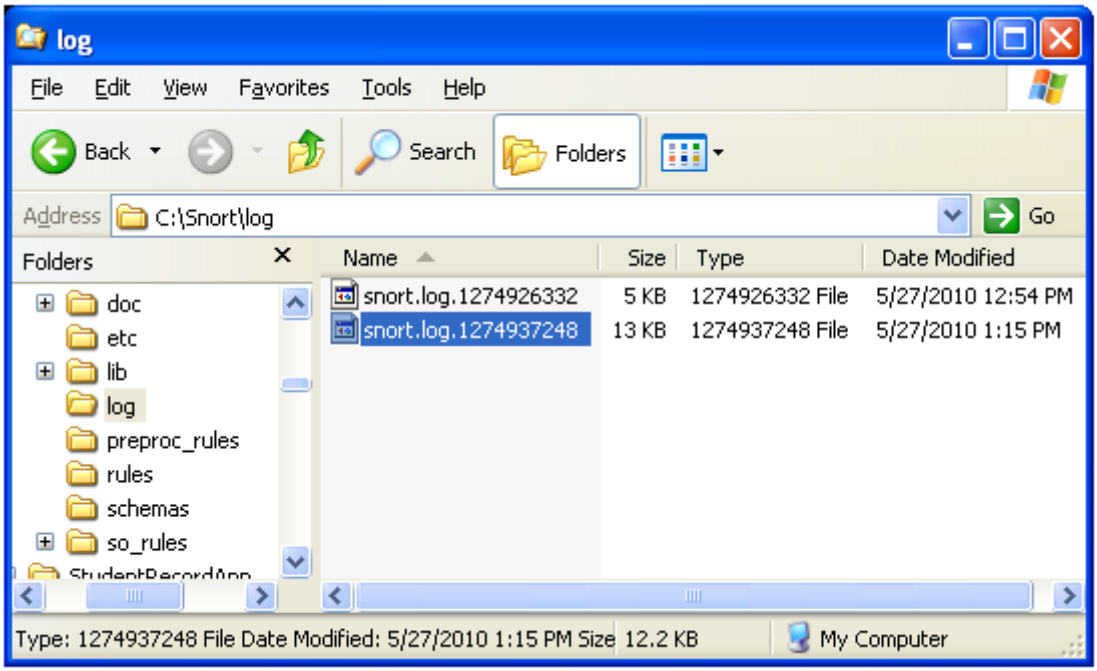
You can test Snort's logging abilities with the -l (log) switch, by typing (take note on the order of the options):

C:\>snort -vde -l c:\Snort\log -i2



Please press Ctrl+C to stop.





Result:

snort –h

C:\Windows\System32\cmd.exe

```
C:\Snort\bin>snort --h
snort: option `--h' is ambiguous

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20-WIN64 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11

USAGE: snort [-options] <filter options>
       snort /SERVICE /INSTALL [-options] <filter options>
       snort /SERVICE /UNINSTALL
       snort /SERVICE /SHOW
Options:
        -A         Set alert mode: fast, full, console, test or none  (alert file alerts only)
        -b         Log packets in tcpdump format (much faster!)
        -B <mask>  Obfuscated IP addresses in alerts and packet dumps using CIDR mask
        -c <rules> Use Rules File <rules>
        -C         Print out payloads with character data only (no hex)
        -d         Dump the Application Layer
        -e         Display the second layer header info
        -E         Log alert messages to NT Eventlog. (Win32 only)
        -f         Turn off fflush() calls after binary log writes
        -F <bpf>   Read BPF filters from file <bpf>
        -G <0xid>  Log Identifier (to uniquely id events for multiple snorts)
        -h <hn>    Set home network = <hn>
                   (for use with -l or -B, does NOT change $HOME_NET in IDS mode)
        -H         Make hash tables deterministic.
        -i <if>    Listen on interface <if>
        -I         Add Interface name to alert output
        -k <mode>  Checksum mode (all,noip,notcp,noudp,noicmp,none)
        -K <mode>  Logging mode (pcap[default],ascii,none)
        -l <ld>    Log to directory <ld>
        -L <file>  Log to this tcpdump file
        -n <cnt>   Exit after receiving <cnt> packets
        -N         Turn off logging (alerts still work)
        -O         Obfuscate the logged IP addresses
        -p         Disable promiscuous mode sniffing
        -P <snap>  Set explicit snaplen of packet (default: 1514)
        -q         Quiet. Don't show banner and status report
        -r <tf>    Read and process tcpdump file <tf>
        -R <id>    Include 'id' in snort_intf<id>.pid file name
        -s         Log alert messages to syslog
        -S <n=v>   Set rules file variable n equal to value v
        -T         Test and report on the current Snort configuration
        -U         Use UTC for timestamps
        -v         Be verbose
        -V         Show version number
        -W         Lists available interfaces. (Win32 only)
        -X         Dump the raw packet data starting at the link layer
        -x         Exit if Snort configuration problems occur
        -y         Include year in timestamp in the alert and log files
        -z <file>  Set the preproc_memstats file path and name
        -Z <file>  Set the performonitor preprocessor file path and name
        -?         Show this information
<Filter Options> are standard BPF options, as seen in TCPDump
Longname options and their corresponding single char version
   --logid <0xid>                Same as -G
   --perfmon-file <file>         Same as -Z
   --pid-path <dir>              Specify the directory for the Snort PID file
```

Snort in Sniffer mode

snort -v -i2

C:\Windows\System32\cmd.exe - snort -v -i2

```
C:\Snort\bin>snort -v -i2
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{CEC9B938-60FF-4DF4-9066-68B5D17E6E17}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20-WIN64 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=12252)
```

snort -W

C:\Windows\System32\cmd.exe

```
C:\Snort\bin>snort -W

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20-WIN64 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11

Index   Physical Address       IP Address      Device Name     Description
-----   ----------------        ----------      -----------     -----------
    1   00:00:00:00:00:00       disabled        \Device\NPF_{ACF529BC-31BA-4F20-AA59-F5F06D62B046}      WAN Miniport (Network Monitor)
    2   00:00:00:00:00:00       disabled        \Device\NPF_{CEC9B938-60FF-4DF4-9066-68B5D17E6E17}      WAN Miniport (IPv6)
    3   00:00:00:00:00:00       disabled        \Device\NPF_{AF5DCC73-3F25-4455-84A5-B1D6DA4DCC68}      WAN Miniport (IP)
    4   54:BE:F7:0C:61:16       192.168.0.106   \Device\NPF_{86B691F5-16CC-4327-A4D9-1252F9824BA6}      Intel(R) 82579V Gigabit Network Connection
    5   00:00:00:00:00:00       0000:0000:0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback    Adapter for loopback traffic capture
```

Snort as Packet Logger

snort -vde -l c:\Snort\log -i2

C:\Windows\System32\cmd.exe - snort -vde -l c:\Snort\log -i2

```
C:\Snort\bin>snort -vde -l c:\Snort\log -i2
Running in packet logging mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Log directory = c:\Snort\log
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{CEC9B938-60FF-4DF4-9066-68B5D17E6E17}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.20-WIN64 GRE (Build 82)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=10324)
```

Conclusion:

_____

_____

Questionnaire:

1. What is snort?

_____

_____

2. What are the uses of snort?

_____

_____

3. In which modes, does the snort run?

_____

_____

Experiment No. 12
GPG tool to implement email security

Aim: Explore the GPG tool to implement email security

Theory:

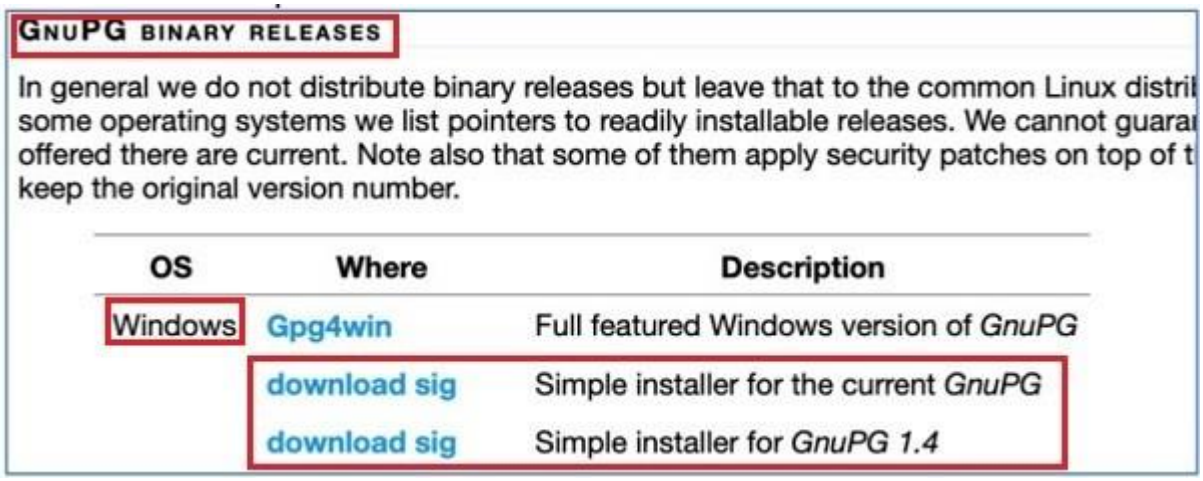
GnuPG is a cryptography tool that helps you manage public and private keys as well as perform encrypt, decrypt, sign, and verify operations. It is an open-source version of PGP.

The end result is a PGP-encrypted ZIP file (**.zip.pgp**) that is ready to be uploaded to the IBM sFTP server.

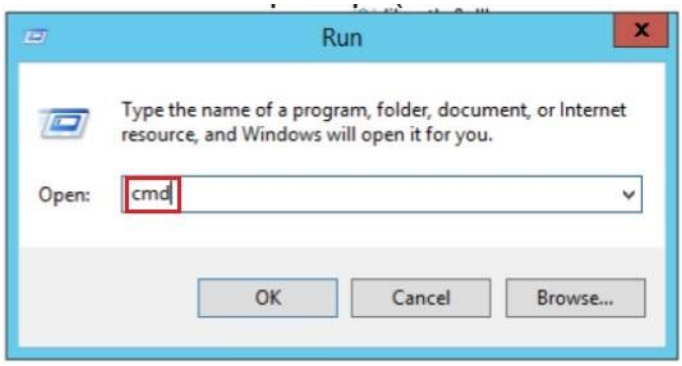
To do this task, you need the following from the **Welcome email**: **public PGP key** (in an **.asc** file).

STEP 1 - DOWNLOAD AND INSTALL GNUPG

1. Go to the GnuPG website to download the software: https://gnupg.org/download/index.html.
2. Scroll to **GnuPG Binary Releases**.

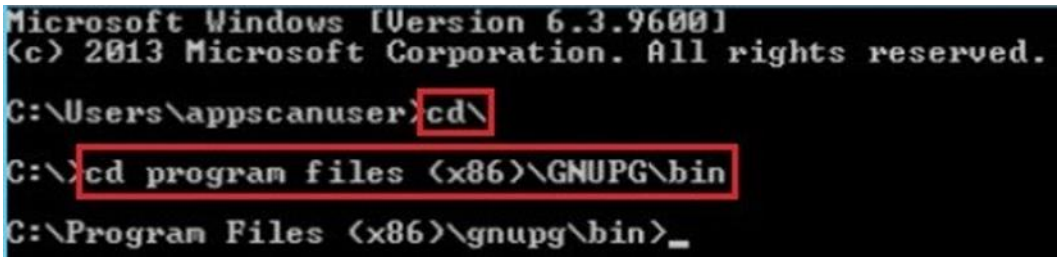


3. For the **Windows** OS, select the **Download Sig** link either for **Simple Installer for the CurrentGnuPG** or **Simple Installer for GnuPG 1.4**.
4. Select **Run** and follow the steps to install the software.
5. Open a **command prompt** (Windows > Run > cmd > OK or Enter key).



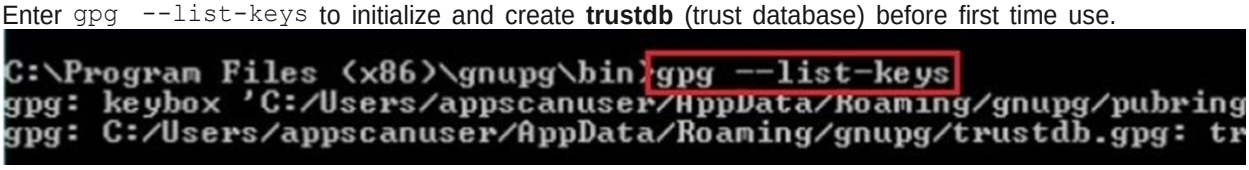
6. Enter command `cd\` and press the **Enter** key to move to the root directory (for example,

enter: C:\).



7. Change the directory where GNUPG is installed by entering a command like `cd Program Files(x86)\gnupg\bin\`.

Enter `gpg --list-keys` to initialize and create **trustdb** (trust database) before first time use.



STEP 2 - FINISH INSTALL FOR OPERATING SYSTEM

The following shows what you enter in a **Command Prompt** window for **each operating system**. Thisassumes you already went to the GnuPG website and downloaded/installed the software.

In all the operating systems, to check if your software installed correctly, enter `gpg --help` in thecommand line.

STEP 3 - IMPORT PUBLIC PGP KEY AND ENCRYPT ZIP FILE

The following procedure shows you how to do this.

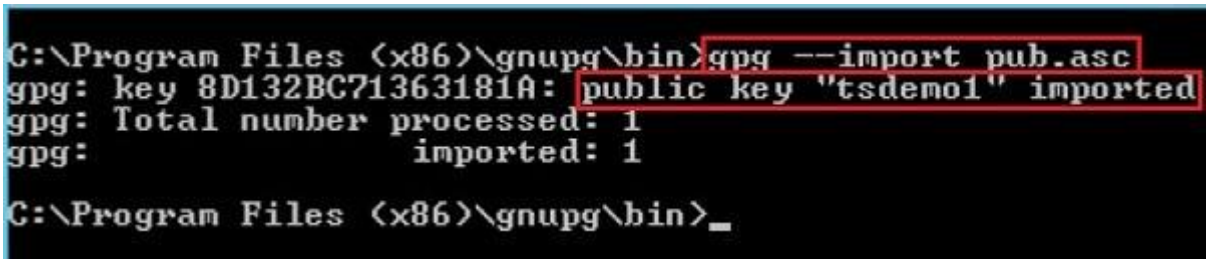
Procedure

Import the Public PGP Key

1. Download the **public PGP key** (provided in Welcome email, in an **.asc** file) to your machine. An .asc file isused by PGP encryption.
2. Open a **command prompt** and enter the `path to the .asc file` so that you can import the key. **Note**:This is a one time task.

   Format: `gpg --import <complete_path_to_.asc_file>`

   Example: `gpg --import pub.asc`

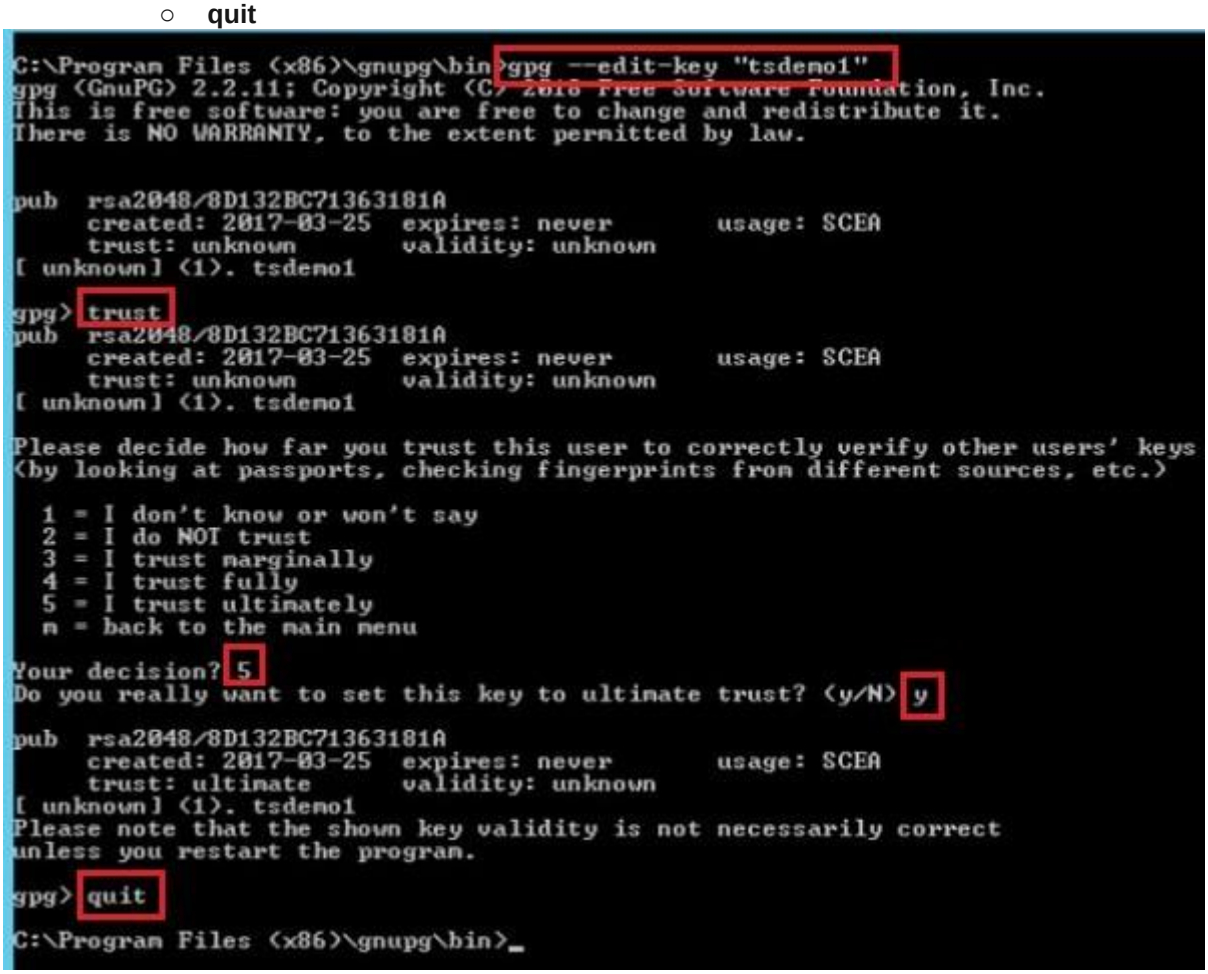


Note: If the public key is successfully imported, the name of the **key** (a user ID (**uid** as provided in the Welcome email) displays. In this example, **tsdemo1** is the **name of the key**. You need the key name for encryption.

3. If you enter `gpg --list-keys` in the command prompt, all available public keys on this particular machinedisplay, including the **public key you imported**.

Encrypt the File

4. Enter `gpg --edit-key "tsdemo1"` to **open the public key for editing**. This step ensures you are **readyfor encrypting files using this key**. Then enter the following, one at a time in the prompt:
    - **trust**
    - **5**
    - **y**
    - **quit**



5. Navigate to the **path** where the **ZIP file you intend to encrypt** is located.
6. Enter the following in the command prompt to identify the **key** (**tsdemo1** in this example), and the **ZIP filename**. `-u` indicates you are using a key, and `-e` indicates a ZIP file name follows.

    Format: `gpg -u "<key_uid>" -e <zip_file_to_be_encrypted>`

    Example: `gpg -u "tsdemo1" -e testfile.zip`

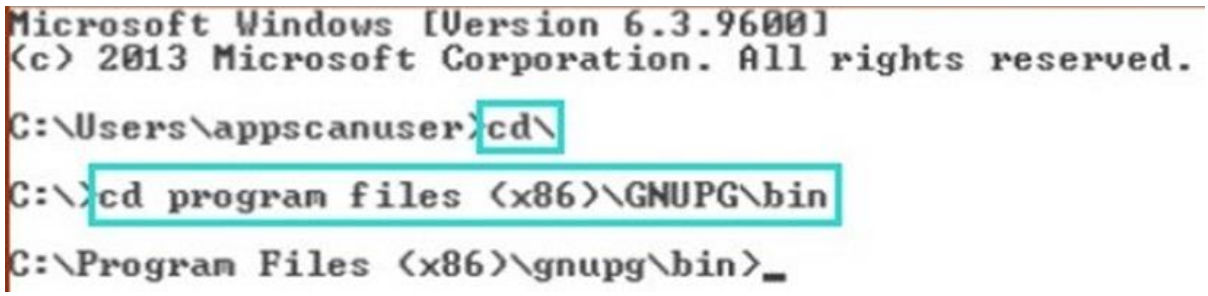
7. Enter the following in the command prompt to identify the **key** (**tsdemo1** in this example), and the **path to theZIP file**.

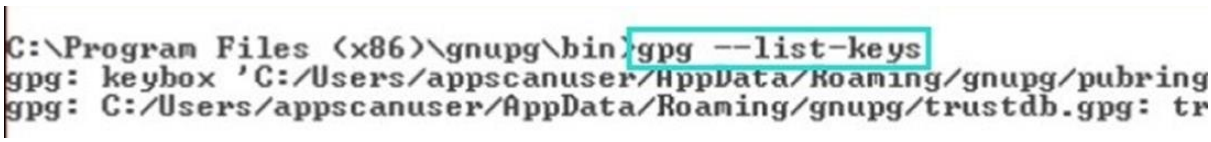
   Format: `gpg --encrypt --recipient "<key_uid>" "<complete_path_to_zip_file>"`

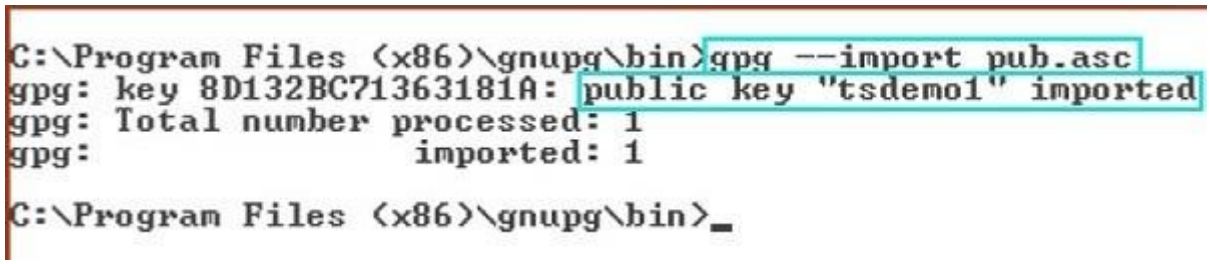
   Example: `gpg --encrypt --recipient "tsdemo1" "C:\Testing\testfile.zip"`

8. **Rename** the file name suffix from **.gpg** to **.pgp**. You now have a PGP-encrypted ZIP file, ready for upload tothe IBM sFTP server.

Results:







```
C:\Program Files (x86)\gnupg\bin>gpg --edit-key "tsdemo1"
gpg (GnuPG) 2.2.11; Copyright (C) 2018 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.


pub  rsa2048/8D132BC71363181A
     created: 2017-03-25  expires: never       usage: SCEA
     trust: unknown       validity: unknown
[ unknown] (1). tsdemo1

gpg> trust
pub  rsa2048/8D132BC71363181A
     created: 2017-03-25  expires: never       usage: SCEA
     trust: unknown       validity: unknown
[ unknown] (1). tsdemo1

Please decide how far you trust this user to correctly verify other users' keys
(by looking at passports, checking fingerprints from different sources, etc.)

  1 = I don't know or won't say
  2 = I do NOT trust
  3 = I trust marginally
  4 = I trust fully
  5 = I trust ultimately
  m = back to the main menu

Your decision? 5
Do you really want to set this key to ultimate trust? (y/N) y

pub  rsa2048/8D132BC71363181A
     created: 2017-03-25  expires: never       usage: SCEA
     trust: ultimate      validity: unknown
[ unknown] (1). tsdemo1
Please note that the shown key validity is not necessarily correct
unless you restart the program.

gpg> quit

C:\Program Files (x86)\gnupg\bin>_
```

```
➜  20180727  gpg -u "tsdemo1" -e testfile.zip
```

```
C:\Program Files (x86)\gnupg\bin>gpg --list-keys
C:/Users/appscanuser/AppData/Roaming/gnupg/pubring.kbx
------------------------------------------------
pub   rsa2048 2017-03-25 [SCEA]
      610DACFBE8E06921D77512988D132BC71363181A
uid           [ultimate] tsdemo1


C:\Program Files (x86)\gnupg\bin>gpg --encrypt --recipient tsdemo1 C:\Testing\S
mpleFile_For_zippignAndEncrypting.zip

C:\Program Files (x86)\gnupg\bin>_
```

Conclusion:

_____

_____

Industrial Application:

- Allows for the secure transmission of information between parties and can be used **to verify thatthe origin of a message is genuine**.
- Secure email and file encryption

Questionnaire:

1. **What is GPG?**

_____

_____

_____

2. **What is the use of GPG tool?**

_____

_____

_____

3. **How to list public keys stored locally using GPG tool?**

_____

_____

_____

4. **How to create a new private key in GPG tool?**

_____

_____

_____

5. **How to delete a private key from local storage?**

_____

_____

_____

Ex. No : 11	Study of malicious software using different tools:

AIM: use of nessus tool to scan the network for vulnerabilities

Theory:

NESSUS Vulnerability Scanner – Basics

If you are looking for a vulnerability scanner, you might have came across several expensive commercial products and tools, with wide range of features and benefits. If a full featured free vulnerability scanner is on your mind, then it's time to know about Nessus. The article covers installation, configuring and select policies, starting a scan, analyzing the reports using NESSUS Vulnerability Scanner. Nessus was founded by Renuad Deraison in the year 1998 to provide to the Internet community a free remote security scanner. It is one of the full fledged vulnerability scanners which allow you to detect potential vulnerabilities in the systems. Nessus is the world's most popular vulnerability scanning tool and supported by most of the research teams around the world. The tool is free of cost and non-commercial for non-enterprises. Nessus uses web interface to set up, scan and view repots. It has one of the largest vulnerability knowledge bases and because of this KB the tool is very popular. Nessus supports wide range of operating systems that include Windows XP/7, Linux, Mac OS X, Sun Solaris, etc.

Key Features:

- Identifies Vulnerabilities that allow a remote attacker to access sensitive information from the system.
- Checks whether the systems in the network has the latest software patches.
- Tries with Default passwords, common passwords, on systems account
- Configuration audits.
- Vulnerability analysis.
- Mobile Device audits.
- Customized reporting.

Installation & Configuration:

i. You can download the Nessus home feed (free) or professional feed from Nessus website.
ii. Once you download the Nessus home tool, you need to register for generating an activation key. The activation key will be sent to your email id.
iii. Install the tool (Installation of nessus tool will be quite confusing and the installation guide comes handy).

iv. Open the Nessus in the browser, normally it runs on the port 8834 – http://localhost:8834/WelcomeToNessus-Install/welcome and follow the screen.

v. Create an account with Nessus.

vi. Enter the activation code you have obtained by registering with the Nessus website. Also you can configure the proxy if needed by giving proxy hostname, proxy username and password.

vii. Then scanner gets registered and creates the user account.

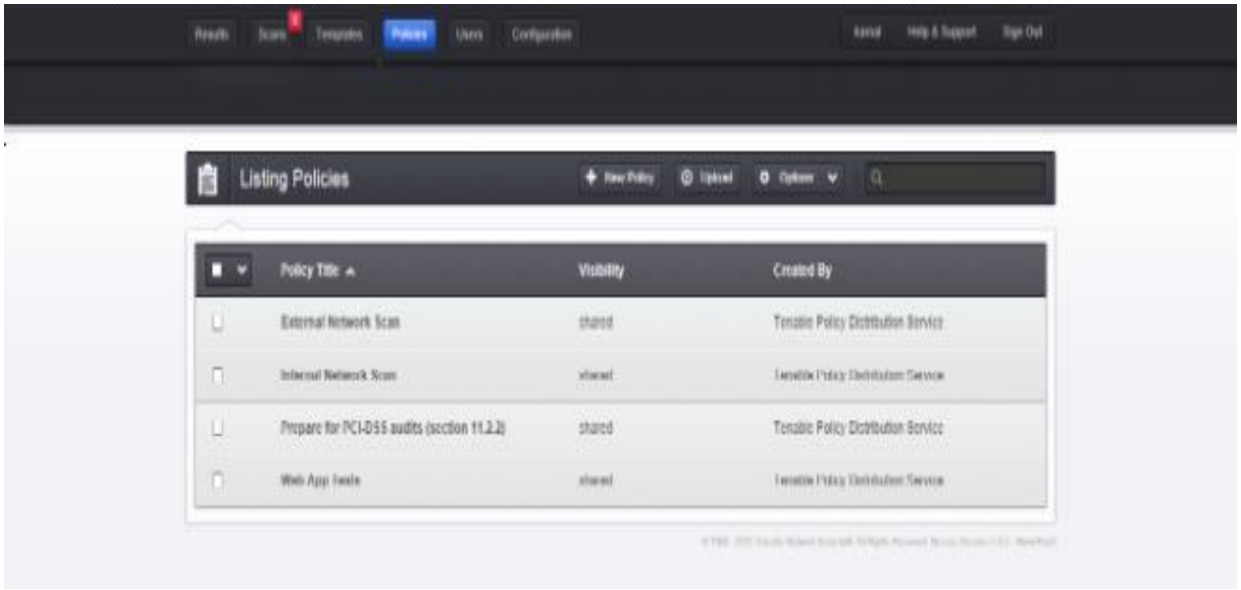
viii. Then downloads the necessary plugins (It takes some time for downloading the plugins).

ix. Once the plug-ins are downloaded then it will automatically redirects you to a login screen. Provide the Username and password that you have created earlier to login.

One of the foundations for discovering the vulnerabilities in the network are:

- Knowing which systems exist
- Knowing which ports are open and which listening services are available in those ports
- Determining which Operating System is running in the remote machine

POLICIES: Policies are nothing but the vulnerability tests that you can perform on the target machine. By default Nessus has 4 policies.



Above figure shows the default polices that comes with Nessus tool.

External Network Scan:

The policy is pre-configured in such a way that Nessus scans externally facing hosts, which provides services to the host. It scans all 65,535 ports of the target machine. It is also configured with Plugins required for web application vulnerabilities tests like XSS.

Internal Network Scan:

This policy is configured to scan large internal networks with many hosts, services, embedded systems like printers, etc… This policy scans only standard ports instead of scanning all 65,535 ports.

Web App Tests:

Nessus uses this policy to detect different types of vulnerabilities exist in the web applications. It has the capability to spider the entire web site and discovers the content and links in the application. Once the spider process has been completed then Nessus starts to discover the vulnerabilities that exist in the application.

Prepare for PCI DSS audits:

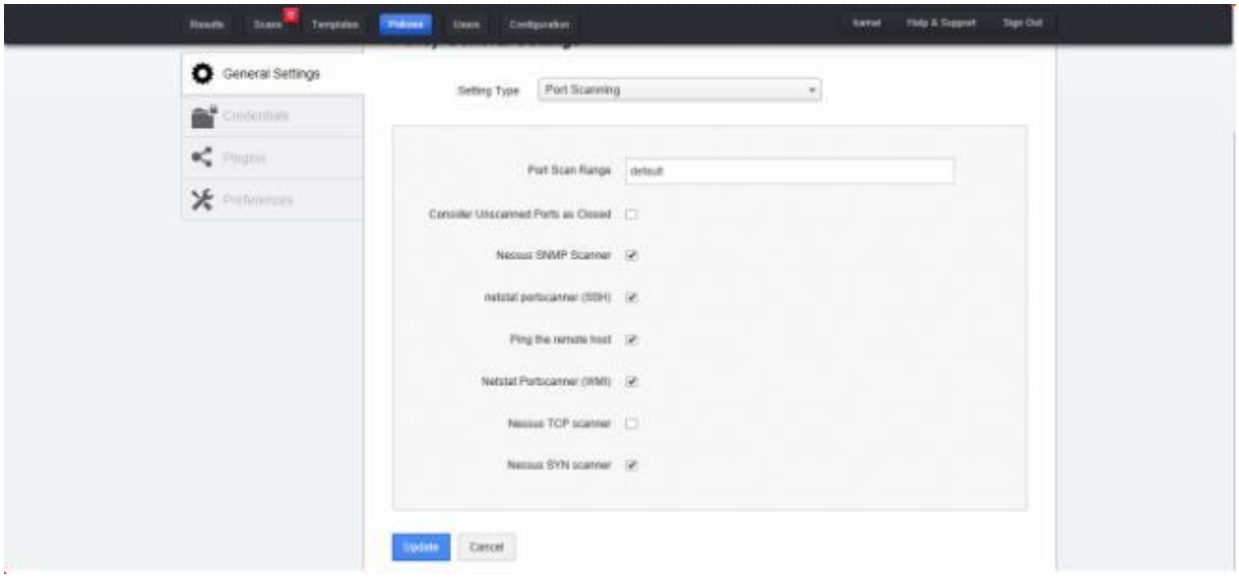
This policy consists of PCI DSS (Payment Card Industry Data Security Standards) enabled. Nessus compares the results with the standards and produces a report for the scan. The scan doesn't guarantee for a secure infrastructure. Industries or Organizations preparing for PCI-DSS can use this policy to prepare their network and systems.

Apart from these pre-configured policies you can also upload a policy by clicking on "Upload" or configure your own policy as per your scan requirement by clicking on "New Policy".

Configuring the Policy:

- Click on the policies tab on the top of the screen
- Click on the New Policy button to create a new policy

Under the General settings tab select the "setting type" based on scan requirement, like Port Scanning, Performance scanning etc… Based on the type Nessus prompts different options that has to be filled. For example 'Port Scanning' has the following options

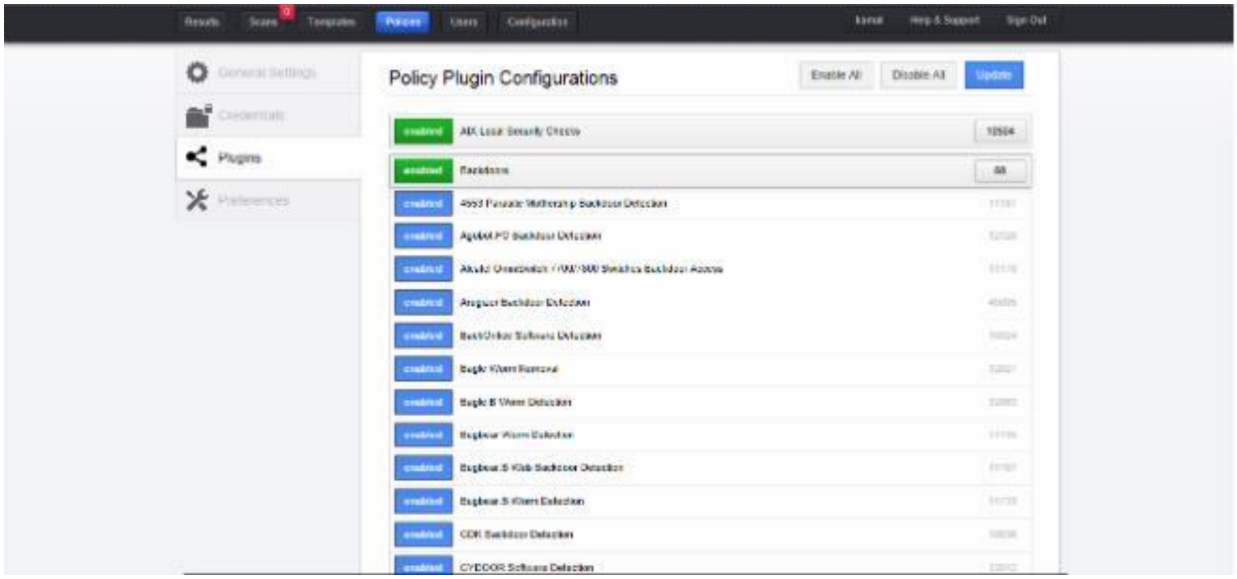


Above figure shows configuring options of Port Scanning.

Enter the port scan range. By default Nessus scans all the TCP ports in /etc/services file. You can limit the ports by specifying it manually (like 20-30). You have different scanners like Nessus SNMP scanner, SSH scanner, ping remote host, TCP Scanner, SYN scanner, etc…. Enable by selecting the check box as per the scan requirement.

- Enter the credentials for scan to use. You can use single set of credentials or multiple set of credentials if you have. You can also work it out without entering the credentials.

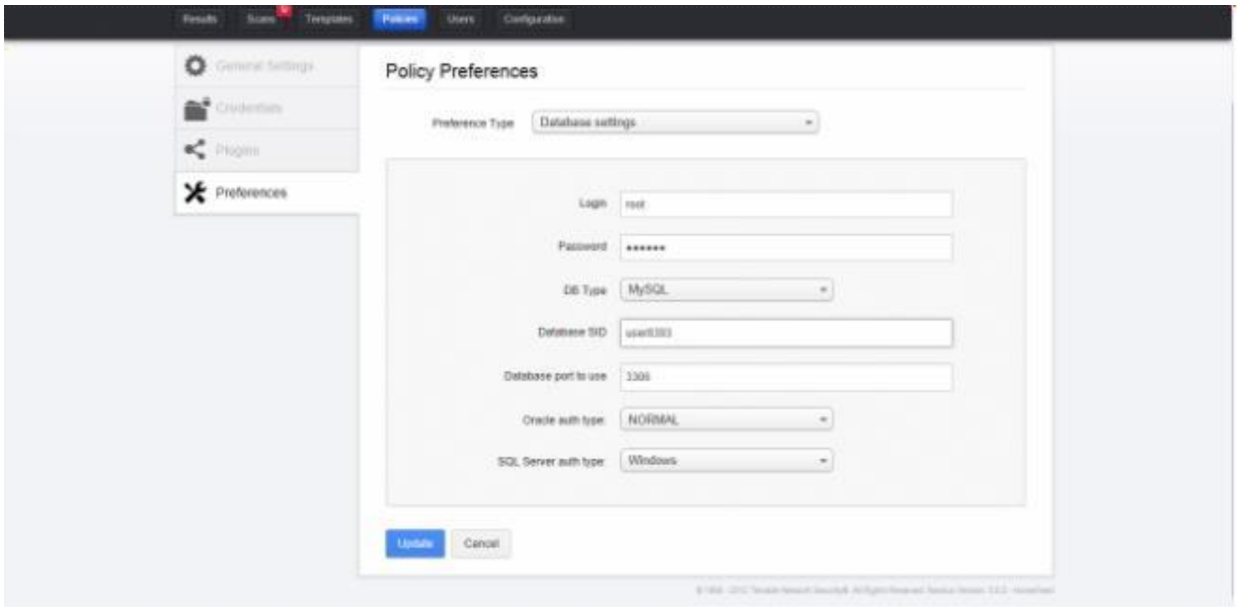
- The plugins tab has number of plugins. By default Nessus will have all the plugins enabled. You can enable or disable all the plugins at a time or enable few from the plug-in family as per the scan you'd like to perform. You can also disable some unwanted plugins from the plug-in family by clicking on particular plug-in.



The above figure shows the sub-plugins for the plugin Backdoors.

In the above Figure the green one shows the parent plugin and the blue once shows the sub-plugins or the plugins under the plugin (backdoor). You can enable or disable by simply clicking on the enabled button.

- In the Preferences, you are provided with a drop down box to select different types of plugins. Select the plugin based on the scan requirement and specify the settings as per the plugins requirement. Click finish once completed. For example: configure the database.



The above figure shows the configuration of Database settings plugin.

SCANS:

Once you are done with configuring the policies as per your scan requirement, you need to configure the scan details properly. You can do it under Scan tab.

Under the Scan tab, you can create a new scan by clicking *New Scan* on the top right. Then a pop up appears where you need to enter the details like Scan Name, Scan Type, Scan Policy & Target.

- Scan Name: The name that you are willing to give to the scan.
- Scan Type: You have options to RUN the scan instantly by selecting *RUN NOW*. Or you can make a template which you can launch later when you are willing to run. All the templates are moved under the TEMPLATE tab beside the SCAN tab.
- Scan Policy: Select the policy that you have configured previous in the policies section.
- Select Target: Enter the target machine which you are planning to test. Depending upon the targets Nessus takes time to scan the targets.

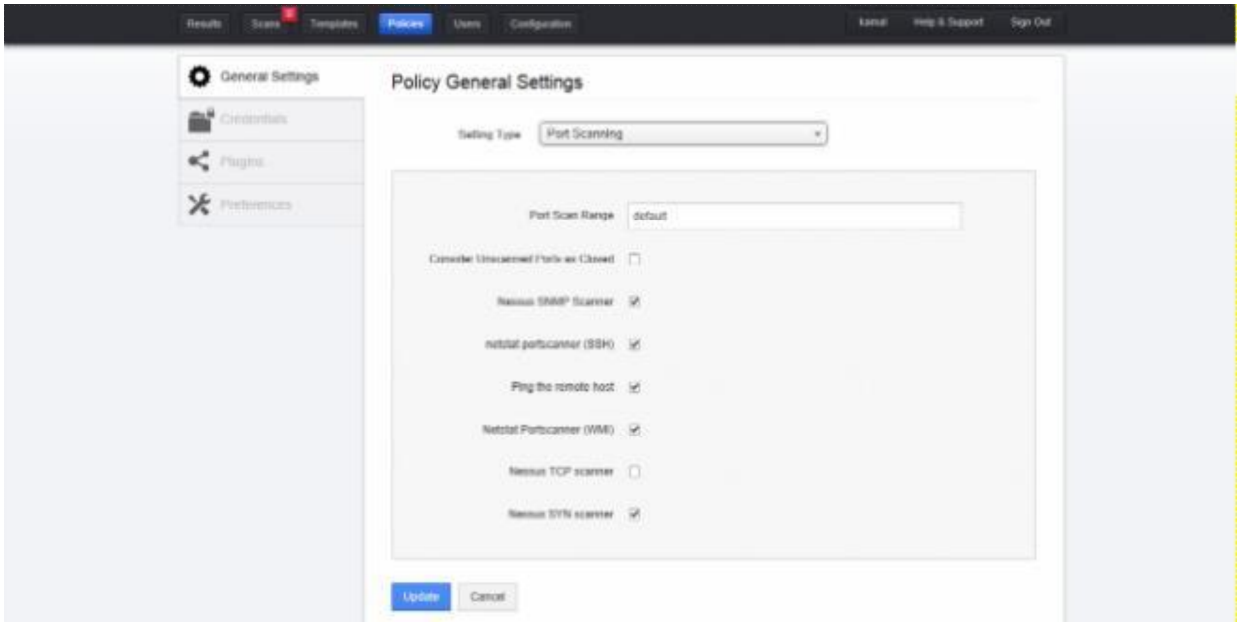
Results:

Once the scanning process has been completed successfully, results can be analyzed from RESULTS menu.

- Once the scan has been completed, you can see the name of the scan under the results section. Click on the name to see the report.
- Hosts: Specifies all the target systems that you have scanned.
- Vulnerabilities: Displays all the vulnerabilities on the target machine that has been tested.
- Export Results: You can export the results into difference formats like html, pdf, etc… You can also select an individual section or complete result to export based on your requirement.

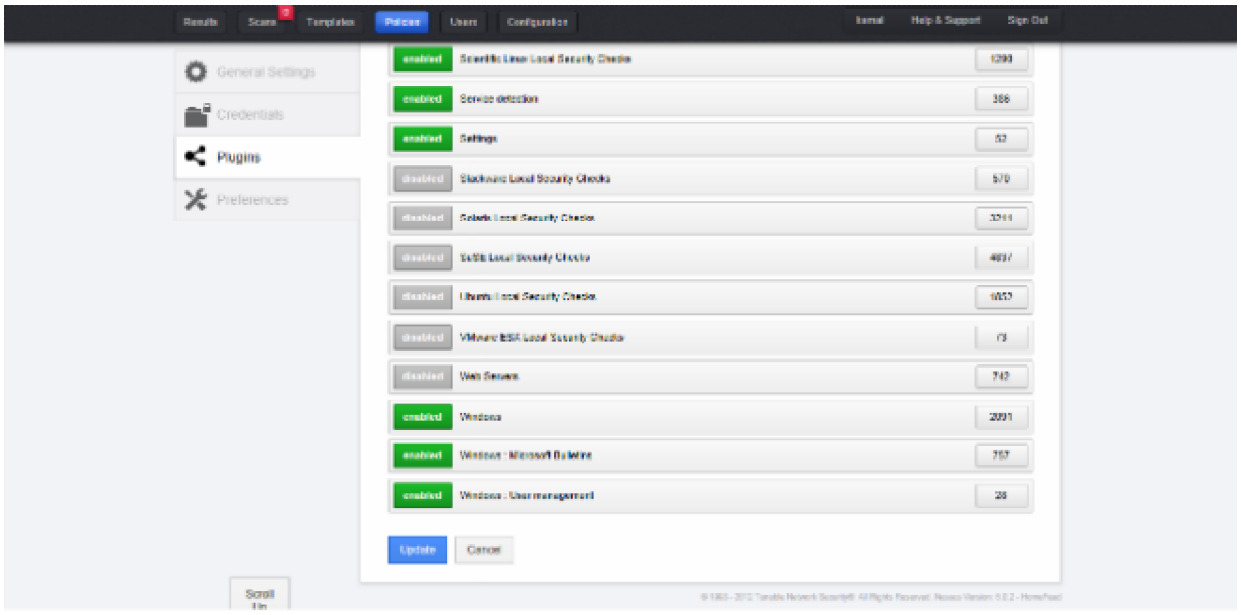
Let us try out an example now-

I have configured a policy named *Basic Scan*. We have many options while configuring or building the policy like port scanners, performance of the tool, Advanced etc.

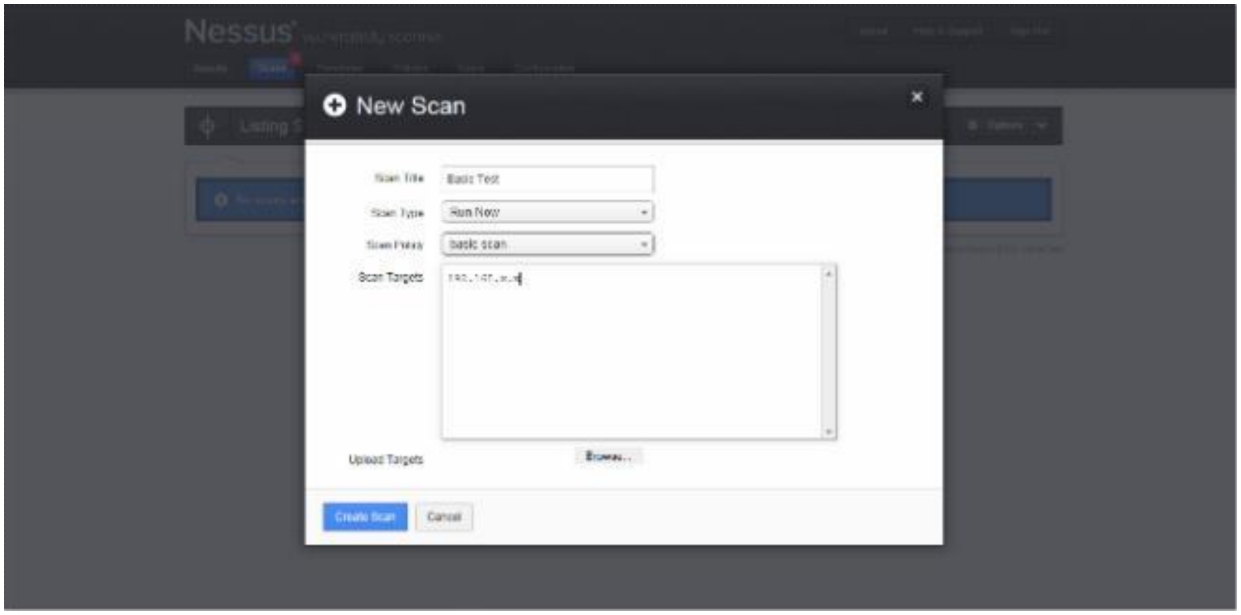


The above figure shows configuration settings of Port Scanning for the policy *Basic Scan*.

You don't need credentials now, so skip the credentials tab and move to Plugins tab. You need to configure the specific plug-in as per the scan requirement that you are willing to perform on remote machine.



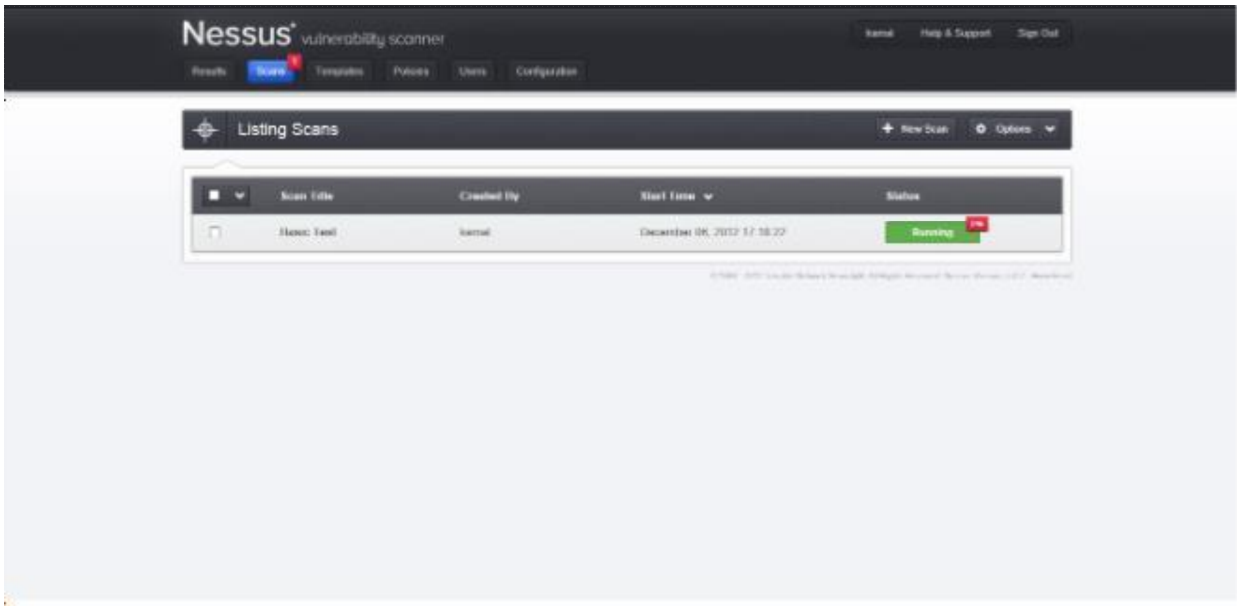
The above figure shows the plugins that I have enabled for the policy *Basic Scan*. I have enabled few plugins for windows machine scan.



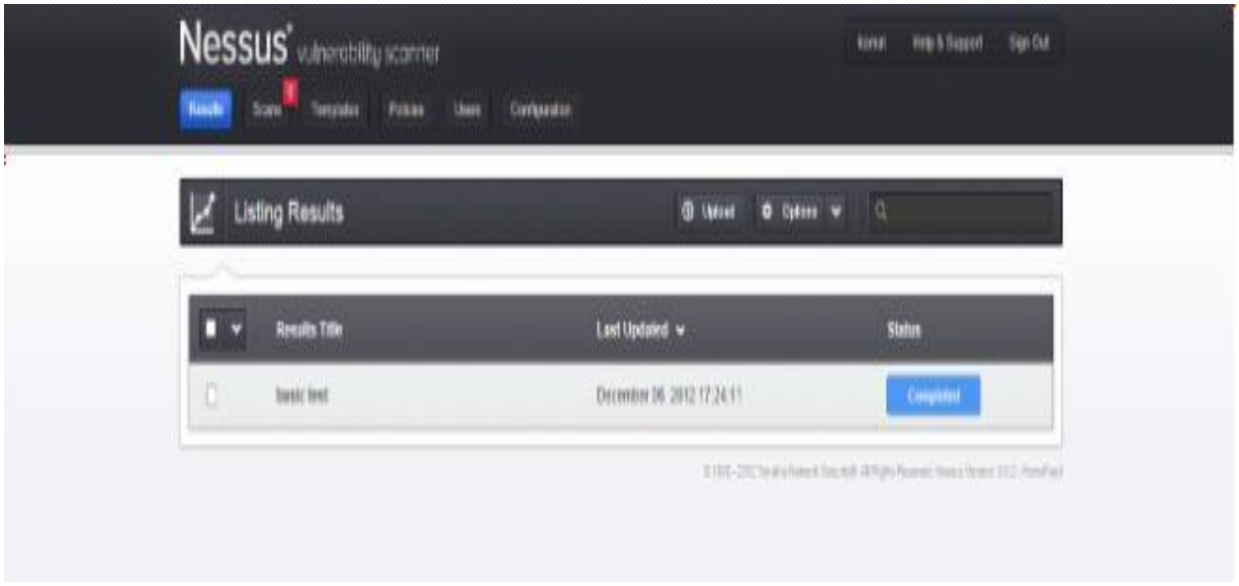
The above figure shows the configuration of the Scan.

I have configured the scan to run instantly with the policy that I have created earlier. And the scan target specify the IP address I am willing to scan.

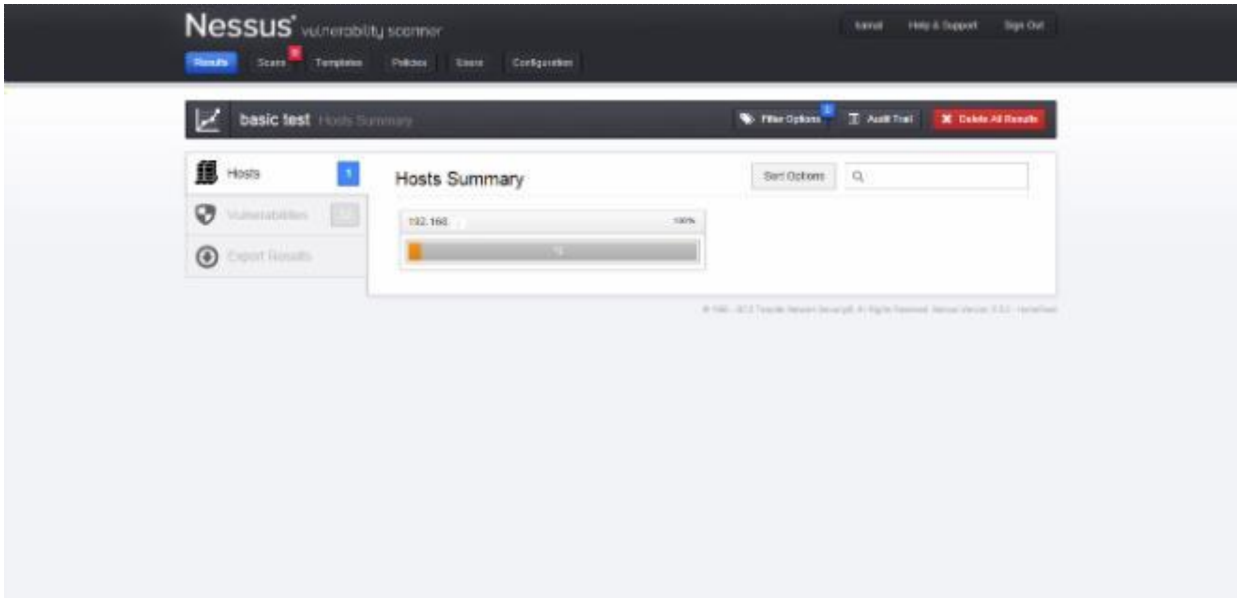
Once all the details has been entered click on *Create Scan* which shows the Scan is running as shown in the below Figure.



Once the scanning has been completed then you can see the results in Results tab. Below Figure shows the same.



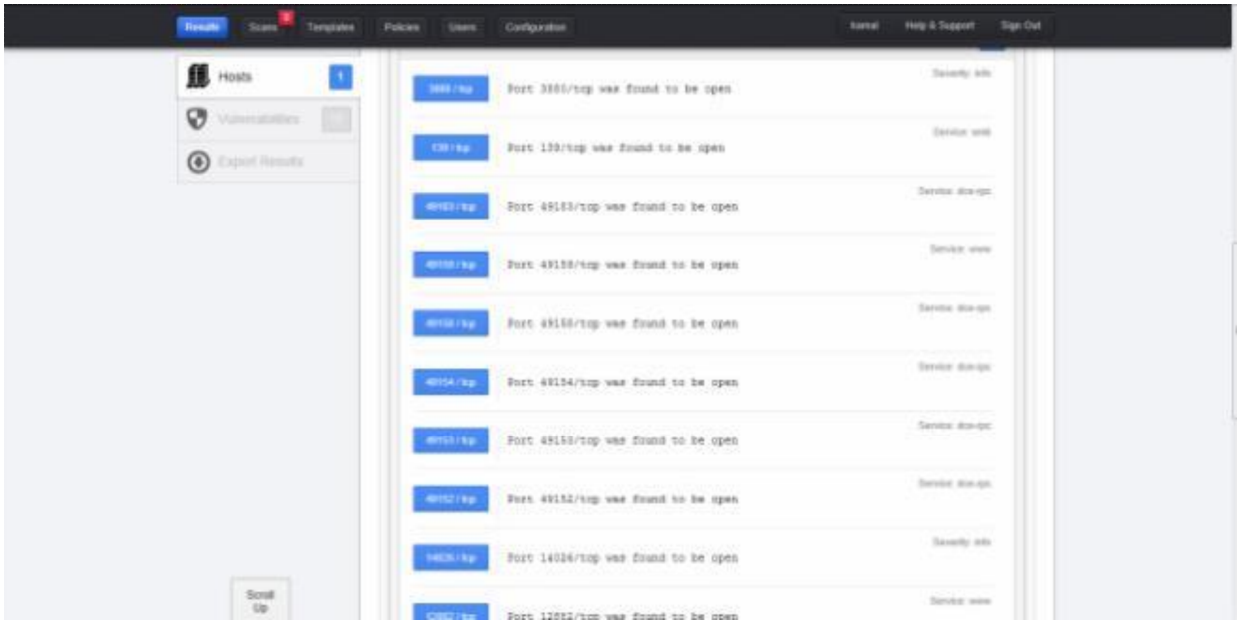
Double clicking on the title displays the scan results.



The above figure shows the Hosts details. It includes all the targets that you have scanned during the test. Double clicking on the host address displays the vulnerabilities Nessus have identified during the test. You can also click on Vulnerabilities tab to check out the vulnerabilities.

The above figure shows the Vulnerabilities that Nessus found during its scan. Based on the Risk Nessus marks it as high, medium, info etc… Clicking on the Vulnerability gives you brief description of it.

For example let us go with Netstat portscanner, displays you the following information



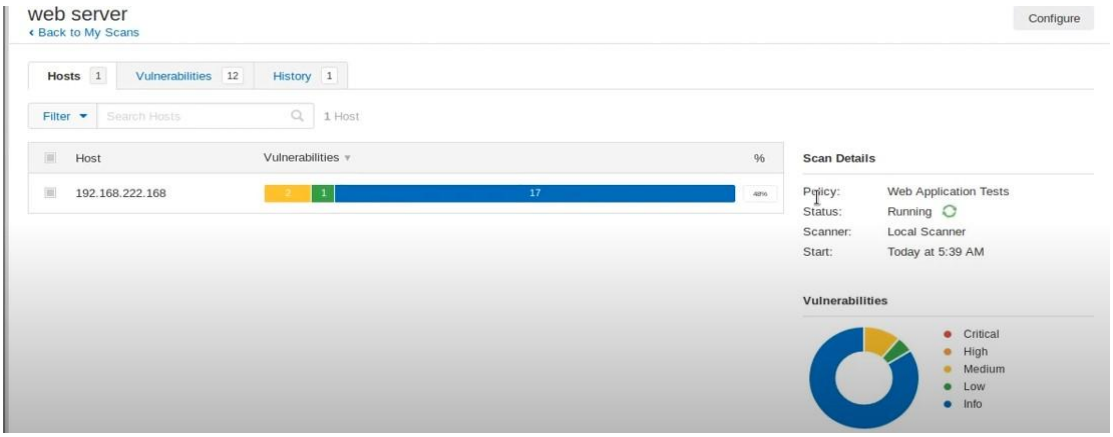
The above figure shows the ports opened in the target machine.
In the same manner you can analyze complete details by clicking on the vulnerabilities.
Conclusion:

Nessus is a tool which automates the process of scanning the network and web applications for the vulnerabilities also suggests solutions for the vulnerabilities that are identified during the scan.

OUTPUT :

NETWORK SCAN:

WEB APP TESTS:



PREPARE FOR PCI DSS AUDITS: