LABORATORY MANUAL

For

System Security Lab

(CSL602)

Semester – VI

(Academic year: 2022-2023)

Bachelor of Engineering

Computer Engineering

Vasantdada Patil Pratishthan's College of Engineering & Visual Arts, Sion, Mumbai-22

UNIVERSITY OF MUMBAI

LAB MANUAL

Course Name : Cryptography & System Security Lab

Course code : CSC 602

Lab Code : CSL 602

Class : TE Computer Engineering

Semester : VI

Div : A & B

Academic year : 2022-2023(REV-'C')

Prof. Atul Shintre

Prof. Asharani Shinde
(Subject In-charge)

Dr. Rais Mulla
(HOD of Computer Department)

Vision of Computer Department

- To inculcate skills for overall development of students to be a leader in the world of computer engineering and contribute in favour of society.

Mission of Computer Department

- To provide students with a fundamental knowledge of theory, practical and problem solving skills with an exposure to emerging technologies.
- Provide a platform for overall growth and adapting challenges in rapidly changing technology.
- To produce globally competent computer professionals with moral values and leadership abilities for sustainable development of the society.

Program Outcomes (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identity, formulate complex engineering problems reaching substantiated conclusions using principles of Computer Engineering.
3. **Design/development of solutions:** Design / develop solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the society.
4. **Conduct investigations of complex problems:** Use knowledge for the design of experiments, analysis, interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select and apply appropriate techniques and modern engineering tools, including predictions and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply the knowledge to assess social issues and the responsibilities relevant to engineering practices.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in social and environmental contexts, and demonstrate the knowledge for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively such as being able to comprehend and write effective reports and design documentation, make effective presentations.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management skills and apply the skills to manage projects effectively.
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 technologic

Academic Year 2022-2023

CLASS: TE (DIV A & B) SEM:VI

SUBJECT: Cryptography & System Security Lab (CSL602)

EXPERIMENTS LIST

Sr. No.	Title	LO
1	Design and implement the Ceaser Cipher, Product cipher using Substitution technique and Keyed Transposition ciphers technique.	LO1
2	To implement and analyze the RSA cryptosystem.	LO1
3	To implementation of Diffie Hellman Key exchange algorithm	LO1
4	To implement the MD-5 hash algorithm.	LO1
5	Study the use of network reconnaissance tools like WHOIS, dig, traceroute, ns lookup to gather information about networks and domain registry.	LO2
6	Study of packet sniffer tools: wireshark,: 1. Download and install wireshark and capture icmp, tcp, and http packets in promiscuous mode. 2. Explore how the packets can be traced based on different filters.	LO3
7	Download and install the nmap tool. Use it with different options to scan open ports, perform OS fingerprinting, do a ping scan, tcp port scan, udp port scan, xmas scan etc.	LO3
8	Simulate the SQL injection attack, Cross-Cite Scripting attack.	LO5
9	To set up, configuration and use the SNORT for Intrusion Detection	LO4
10	To design personal Firewall using IP-tables	LO4

Prof. Atul Shintre
Prof. Asharani Shinde
Subject Incharge

Dr. Rais Mulla
HOD of Computer Department

Experiment No. 1

Aim: Design and implement the Ceaser Cipher, Product cipher using Substitution technique and Keyed Transposition ciphers technique.

Theory:

The Ceasar Cipher is a type of substitution cipher in which each letter in the plaintext is shifted by a certain number of places. The key is the number of characters to shift the cipher alphabet.

First we translate all of our characters to numbers, 'a'=0, 'b'=1, 'c'=2, ... , 'z'=25.

We can now represent the caesar cipher encryption function, e(x), where x is the character we are encrypting, as:

$$e(x) = (x + k) \pmod{26}$$

Where k is the key (the shift) applied to each letter

After applying this function the result is a number which must then be translated back into a letter.

The decryption function is :

$$e(x) = (x - k) \pmod{26}$$

Product cipher is also a type of substitution cipher in which each letter in the plaintext is substuted with other letter . The key is the number which is used to decide the letter for substitution.

First we translate all of our characters to numbers, 'a'=0, 'b'=1, 'c'=2, ... , 'z'=25.

We can now represent the caesar cipher encryption function, e(x), where x is the character we are encrypting, as:

$$e(x) = (x * k) \pmod{26}$$

Where k is the key (the shift) applied to each letter

After applying this function the result is a number which must then be translated back into a letter.

The decryption function is :

$$e(x) = (x * k^{-1}) \pmod{26}$$

In **keyled transpositon technique** where a random key is used to describe the transposition sequence and carry out the transposition. This is also called Columnar Transpostion Cipher.

Algorithm:

1. Arrange the plaintext in a column under the given key.

2. Rearrange the plaintext column-wise in key's alphabetic order.

Suppose Key is 'ENCRYPT' and the plaintext is "Friday is the attack day", then to form the cipher :

1. Write the plaintext below given secret key cloumnwise like a table shown below.

2. Mark the alplhabets in the key in order alphabetically. For example for the given key the alphabet 'C' comes first in the order of 26 alphabets so it assigned to number 1 . Then comes 'E' and hence marked 2. Likewise mark all the alphabes in the key according to their occurence in alphabets.

E	N	C	R	Y	P	T
2	3	1	5	7	4	6
f	r	i	d	a	y	i
s	t	h	e	a	t	t
a	c	k	d	a	y	x

3. Take the letters from columns in order and form the cipher .

So ciphertext will be : *ihkfsartcytydeditxaaa.*

Conclusion: Thus we have studied & implemented Ceasar cipher , Product cipher and keyed transposition technique.

Experiment No. 2

Aim: To implement the RSA algorithm.

Take input as p,q and encryption key e
Find values for decryption key d
Input the message
Display the encrypted text and again decrypted text

Steps:

Randomly choose two LARGE distinct primes p and q.

Let n=pq

Calculate $\Phi(n)=(p-1)(q-1)$

Choose an integer e such that $(e,\Phi(n))=1$

Find an integer d which satisfies $ed \equiv 1 \bmod \Phi(n)$

Your public key will be the ordered pair (e,n), while your private key is the ordered pair (d,n).

Conclusion : The implementation of RSA algorithm shows the working of asymmetric key cryptographic technique.

Experiment No. 3

Aim: To Implement the Diffie Hellman Algorithm.

Take input as p,q, x and y
Calculate values of R1 and R2 and Print them
Calculate values of K using R2, x and p
Calculate values of K using R1, y and p

Diffie Hellman is a Symmetric Key Agreement Algorithm .

Alice and Bob will agree upon a symmetric key $K=q^{xy} \bmod p$

Steps:

Alice and Bob select two numbers, p(large) and q respectively where p is prime and q is a generator of group<Zp, *> of order p-1

Alice selects a large random number, x such that $0 \leq x \leq p-1$

Alice Calculates $R1=qx \bmod p$

Bob selects large random number, y such that $0 \leq y \leq p-1$

Bob Calculates $R2=qy \bmod p$

Alice sends R1 to Bob. (Print the value of R1)

Bob sends R2 to Alice. (Print the value of R2)

Alice Calculates $K=R2x \bmod p$

Bob Calculates $K=R1y \bmod p$

Conclusion: Thus the Diffie Hellman key exchange algorithm helps to the exchange of same secret key without shareing it over the network.

Experiment No. 4

Aim: To implementation the MD5 algorithm.

Theory:-

MD5 (Message Digest algorithm 5) is a widely used cryptographic hash function with a 128 bit hash value. An MD5 hash is typically expressed as a 32 digit hexadecimal number. MD5 processes a variable length message into a fixed length output of 128 bits. The input message is broken up into chunks of 512 bit blocks (sixteen 32bit little endian integers) ; The message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits less than a multiple of 512. The remaining bits are filled up with a 64bit integer representing the length of the original message, in bits.

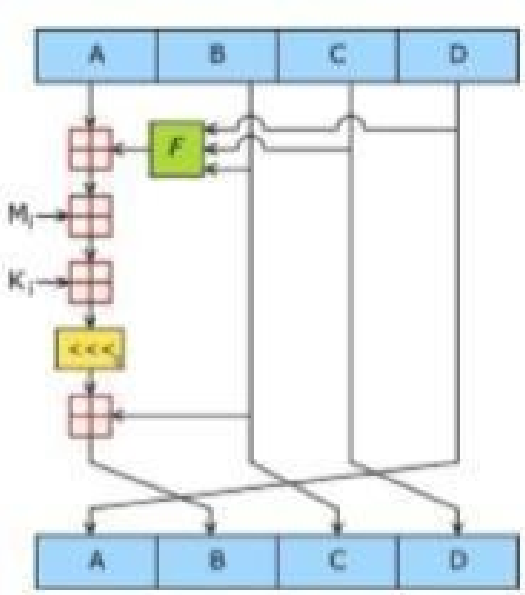

Figure 1: One MD5 operation. MD5 consists of 64 of these operations, grouped in four rounds of 16 operations. F is a nonlinear function; one function is used in each round. Mi denotes a 32bit block of the message input, and Ki denotes a 32bit constant, different for each operation.

The main MD5 algorithm operates on a 128bit state, divided into four 32bit words, denoted A, B, C and D. These are initialized to certain fixed constants. The main algorithm then operates on each 512bit message block in turn, each block modifying the state. The processing of a message block consists of four similar stages, termed rounds; each round is composed of 16 similar operations based on a nonlinear function F, modular addition, and left rotation.

Figure 1 illustrates one operation within a round. There are four possible functions F; a different one is used in each round:

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

$\oplus, \wedge, \vee, \neg$ denote the XOR, AND, OR and NOToperations respectively.

Algorithm:

1. Append Padding Bits The message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. That is, the message is extended so that it is just 64 bits shy of being a multiple of 512 bits long. Padding is always performed, even if the length of the message is already congruent to 448, modulo 512. Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448, modulo 512. In all, at least one bit and at most 512 bits are appended.

2. Append Length A 64 bit representation of b (the length of the message before the padding bits were added) is appended to the result of the previous step. In the unlikely event that b is greater than 2^64, then only the low order 64 bits of b are used. (These bits are appended as two 32bit words and appended loworder word first in accordance with the previous conventions.) At this point the resulting message (after padding with bits and with b) has a length that is an exact multiple of 512 bits. Equivalently, this message has a length that is an exact multiple of 16 (32 bit) words. Let M[0 ... N1] denote the words of the resulting message, where N is a multiple of 16.
3. Initialize MD Buffer A forward buffer (A,B,C,D) is used to compute the message digest. Here each of A, B, C, D is a 32bit register. These registers are initialized to the following values in hexadecimal, lowered bytes first):
4. Process Message in 16Word Blocks We first define four auxiliary functions that each take as input three 32bit words and produce as output one 32bit word.
5. Output The message digest produced as output is A, B, C, D. That is, we begin with the low order byte of A, and end with the high order byte of D

Conclusion : The implementation of MD5 algorithm helps to understand how the hash values gets calculated in cryptographic technique.

Experiment No. 5

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

Theory:-Steps:

Open ubuntu terminal.
Get root access by typing "sudo su root". Put the pc password.
Install the tool using the following command

#apt-get install whois
#apt-get install dig
#apt-get install traceroute
#apt-get install nslookup

whois

Example: Querying tsec.edu

student@lab:~#whoistsec.edu

dig

Simple dig Command Usage

student@lab:~# dig www.google.com
The dig command output has the following sections:

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

QUESTION SECTION: This displays the question it asked the DNS. i.e. input. Since we said 'dig google.com', it indicates in this section that we asked for the record of the google.com website.

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

AUTHORITY SECTION: This displays the DNS name server that has the authority to respond to this query. Basically this displays available name servers of google.com.

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

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

Display Only the ANSWER SECTION of the Dig command Output

All you need to look at is the "ANSWER SECTION" of the dig command. So, we can turn off all other sections as shown below.

student@lab:~ #dig google.com +noquestion
student@lab:~ #dig google.com +nocomments – Turn off the comment lines
student@lab:~ # dig google.com +noauthority – Turn off the authority section
student@lab:~ #dig google.com +noadditional – Turn off the additional section
student@lab:~ #dig google.com +nostats – Turn off the stats section
student@lab:~ #dig google.com +noanswer – Turn off the answer section

Query MX Records Using dig MX

To query MX records, pass MX as an argument to the dig command as shown below.

student@lab:~ #dig google.com MX +noall +answer

Query NS Records Using dig NS

To query the NS record use the type NS as shown below.

student@lab:~ #dig google.com NS +noall +answer

View ALL DNS Records Types Using dig -t ANY

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

student@lab:~ #dig -t ANY google.com +noall +answer

View Short Output Using dig +short

To view just the ip-address of a web site (i.e the A record), use the short form option as shown below.

student@lab:~ #dig google.com +short

DNS Reverse Look-up Using dig –x

To perform a DNS reverse look up using the ip address using dig -x as shown below
student@lab:~ #dig -x 209.132.183.81

traceroute

Command:**student@lab:~**#traceroute google.com

nslookup

Simple nslookup command :student@lab:~#nslookup google.com
Query the MX Record using -query=mx
student@lab:~#nslookup -query = mx google.com

MX (Mail Exchange) record maps a domain name to a list of mail exchange servers for that domain

Query the NS Record using -type=ns

student@lab: ~ #nslookup -type = ns google.com
NS (Name Server) record maps a domain name to a list of DNS servers authoritative for that domain.

Query the SOA Record using -type=soa

student@lab: ~ #nslookup -type = soagoogle.com

SOA record (start of authority) provides the authoritative information about the domain, the e-mail address of the domain admin, the domain serial number, etc

View available DNS records using -query=any

student@lab: ~ #nslookup -type = any google.

Conclusion: The netwrok tool like whois, dig, traceroute and nslookup helps to gather the valuable information from netwrok, which can be used for better network maintainance.

Experiment No. 6

Aim: Study of packet sniffer tools: wireshark,: 1. Download and install wireshark and capture icmp, tcp, and http packets in promiscuous mode. 2. Explore how the packets can be traced based on different filters.

Theory:
Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.

Wireshark is used for:

Network administrators use it to *troubleshoot network problems*

Network security engineers use it to *examine security problems*

Developers use it to *debug protocol implementations*

People use it to *learn network protocol* internals

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, color-coding and other features that let you dig deep into network traffic and inspect individual packets. Features of Wireshark :

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.

Commands:-

a) Open ubuntu terminal

b) Install wireshark: #apt-get install wireshark

c) To know the name of your Ethernet interface: (Mostly it is "etht0") : #ifconfig

d) Start wireshark: #sudo wireshark

e) Once wireshark window opens, select the interface and click on start

Capturing Packets

f) After downloading and installing wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface.

g) For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options.

h) As soon as you click the interface's name, you'll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system.

i) Click the stop capture button near the top left corner of the window when you want to stop capturing traffic

Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.

Wireshark can record the capturing information in the file with extension .pcap (packet capture).

This file can be again reopened for analysis in offline mode.There is no need to remember filtering commands. Filters can be applied by putting predefined strings in Wireshark.

Commands:-

1. Capturing packets of a particular host :- ip.addr = = 192.168.42.3

Sets a filter for any packet with 192.168.42.3, as either the source or destination.

2. To capture a conversation between specified hosts

ip.addr == 10.0.5.119 &&ip.addr == 91.189.94.25

Sets a conversation filter between the two defined IP addresses.

Filtering Packets The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type —dns and you'll see only DNS packets. When you start typing, Wireshark will help you auto complete your filter.

1. To filter packets for a specific protocol http or dns
2. Sets a filter to display all http and dns requests.
3. To filter packets for specific port tcp.port==4000
4. Sets a filter for any TCP packet with 4000 as a source or destination port.
5. Filter specific packets tcp.flags.reset== 0 Displays all TCP resets.
6. Filter for http request packets Displays all HTTP GET requests. Http. request
7. To filter traffic except given protocol packets: !(arp or icmp or dns)
8. Masks out arp, icmp, dns, or whatever other protocols may be background noise, allowing you to focus on the traffic of interest.
9. Capturing packets after applying multiple filters not (tcp.port == 80) and not (tcp port == 25)

Get all packets which are not HTTP or UDP.

To stop capturing click on the "red square"

To capture packets of FTP server. (Login ID and Password)What is FTP?

FTP stands for **F**ile **T**ransfer **P**rotocol. As the name suggest this network protocol allows you to transfer files or directories from one host to another over the network whether it is your LAN or Internet.

The package required to install FTP is known as VSFTPD (Very Secure File Transfer Protocol Daemon)

Steps:-

1. Get root access: $ sudosu root
2. Find your ip address: # ifconfig Installation of FTP server in Ubuntu

Name of Packages required: VSFTPD, XINETD

1. # sudo apt-get install vsftpd
2. # sudo apt-get install xinetd

The above command will install and start the xinetdsuperserver on your system. The chances are that you already have xinetdinstalled on your system. In that case you can omit the above installation command.

In the next step we need to edit the FTP server's configuration file which is present in

/etc/vsftpd.conf

3. # cd /etc
4. # ls
5. # geditvsftpd.conf Change the following line:

Anonymous_enable=NO

To Anonymous_enable=YES

This will instruct the FTP server to allow connecting with an anonymous client.

6. Save and close the gedit file

Now, that we are ready we can start the FTP server in the normal mode with:

7. # servicexinetd restart
8. # servicevsftpd restart OR
9. # init.d/vsftpd restart

Start WIRESHARK. In the FILTER field put FTP. This will filter all FTP packets Connectimg to a client present in other machine

$ ftp ip address of the FTP server

Name: anonymous

Please specify the password. Password:

Login successful. ftp>

ftp> quit Goodbye.

While the client is establishing a connection with the FTP server, the wireshark running in the background of the FTP server is able to capture all FTP packets. So, the Name and Password entered by the client is visible in plain text in Wireshark. Apart from that the source and destination address is also visible. If many clients are trying to connect with the server then source address, name and password are visible for all of them.

Conclusion: Thus packet snifer software is explored for it's different benefits.

Experiment No. 7

Aim: Download and install nmap. Use it with different options to scan open ports, perform OS fingerprinting, do a ping scan, tcp port scan, udp port scan, xmas scan etc

Nmap features include:

Host Discovery – Identifying hosts on a network. For example, listing the hosts which respond to pings or have a particular port open.

Port Scanning – Enumerating the open ports on one or more target hosts.

Version Detection – Interrogating listening network services listening on remote devices to determine the application name and version number.

OS Detection – Remotely determining the operating system and some hardware characteristics of network devices.

Basic commands working in Nmap:

3. For target specifications: nmap<target's URL or IP with spaces between them>
4. For OS detection: nmap -O <target-host's URL or IP>
5. For version detection: nmap -sV<target-host's URL or IP>

SYN scan is the default and most popular scan option for good reasons. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by restrictive firewalls. It is also relatively unobtrusive and stealthy since it never completes TCP connections

Steps:-

1. Get root access: $ sudosu root
2. #ifconfig
3. # apt-get install nmap

Commands:-

1. # nmap -V

It gives the version of Nmap

2. # nmap 192.168.23.20

It gives information about a single host. It gives the output in column form where first column is the PORT, second column is the STATE and third column is the SERVICE

3. #nmap –v 192.168.23.20

It gives the detailed information about remote host.

4. #nmap –O 192.168.23.20

It finds the remote host operating system and version (OS detection)

5. # nmap –sP 192.168.23.0/24

It scans a network and discover which servers and devices are up and running(ping

scan)

6. # nmap -sA 192.168.23.20

To discover if a host/network is protected by a firewall. The output has the word

FILTERED which shows presence of firewall. **UNFILTERED**means no firewall**.**

7. # nmap -p T:23 192.168.23.20

It scans TCP port 23

8. #nmap -p 80,443 192.168.23.20

It scansmultiple ports at one time

9. # nmap -sV 192.168.23.20

It detect remote services (server / daemon) version numbers. Version numbers are displayed only if the Port is open

10. nmap -sS 192.168.23.20

It performs SYN scan or Stealth scan. Open wireshark.

Set the Filter to TCP.

See the grey and red color packets

Double click any grey color TCP packet where destination address is the neighbour's address

See the Flag field of TCP: SYN bit should be set to 1

11. # nmap -sN 192.168.23.20

It performs TCP Null Scan. It does not set any bits (TCP flag header is 0) Open wireshark.

Set the Filter to TCP.

Double click any grey color TCP packet where destination address is the neighbour's address

See the Flag field of TCP: No flag bits should be set

12. # nmap –sF192.168.23.20

It performs FIN scan. It sets just the TCP FIN bit. Open wireshark.

Set the Filter to TCP.

Double click any grey color TCP packet where destination address is the neighbour's address

See the Flag field of TCP: FIN flag should be set to 1

13. # nmap -sX 192.168.23.20

It performs TCP Xmas. It sets the FIN, PSH, and URG flags.

Open wireshark.

Set the Filter to TCP.

Double click any grey color TCP packet where destination address is the neighbour's address

See the Flag field of TCP: FIN, PSH, and URG flagsshould be set to 1

14. # nmap –sO192.168.23.20

It performs IP protocol scan and allows us to determine which IP protocols) are supported by target machines.

15. #nmap –sU192.168.23.20

It performs UDP port scan.

Conclusion: Thus the nmap tool is explored with the different option available for the better result of network scanning.

Experiment No. 8

Aim: Simulate the SQL injection attack, Cross-Cite Scripting attack.

Sqlmap for sql injection attack:

Sqlmap is written in python, the first thing you need is the python interpreter.

Download the python interpreter from python.org. There are two series of python, 2.7.x and 3.3.x. Sqlmap should run fine with either.So download and install.

Next download the sqlmap zip file from sqlmap.org. Extract the zip files in any directory.Launch the dos prompt and navigate to the directory of sqlmap.

Now run the sqlmap.py script with the python interpreter.Start with a simple command:

sqlmap .py -u <URL to inject>.

sqlmap.py -u http://testphp.vulnweb.com/listproducts.php?cat=1

Use –time-sec to speed up the process in case of slow server responses:

sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --time-sec 15.

It will show the Mysql version along with useful information about the database. Database

Obtain the names of available databases by adding –dbs to

the previous command:sqlmap.py -u http://testphp.vulnweb.com/listproducts.php?cat=1 –dbs

Tables

Specify the desired database using –D and tell SQLmap to list the tables using – - tablescommand.

sqlmap.py -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acurt – - tables

Columns

Specify the database using –D, table using – T and columns using –columns:

sqlmap.py -u http://testphp.vulnweb.com/listproducts.php?cat=1 - D acurt –T artists –-columns

Data

As usual, use –D for database, -T for table, -C for column and – dump for data. The finalcommand to fetch data will appear as shown below:

sqlmap.py -u "http://testphp.vulnweb.com/listproducts.php?cat=1" -D acuart -T artists –Caname --dump

Conclusion: The Sqlmap tool is useful for identifying the need of input validation and Sql injection vulnerability.

Experiment 9

Aim: To set up, configuration and use of SNORT for Intrusion Detection

Theory:

Snort is an open source network intrusion prevention and detection system (IDS/IPS) developed by Sourcefire. Combining the benefits of signature, protocol, and anomaly-based inspection, Snort is the most widely deployed IDS/IPS technology worldwide. With millions of downloads and nearly 400,000 registered users, Snort has become the de facto standard for IPS.

Snort can be configured to run in three modes:

Sniffer mode : It simply reads the packets off of the network and displays them for you in a continuous stream on the console (screen)

Packet Logger mode : logs the packets to disk

Network Intrusion Detection System (NIDS) mode: it performs detection and analysis on network traffic. This is the most complex and configurable mode

Steps:

j) Get root access

$ sudosu root

k) Do updation # apt-get update

l) Installation

apt-get install snort During installation:

Put the name of network interface (by default it is eth0, change it to the interface name of your machine)

Put the IP address of the machine followed by /24 (by default it is the network address. Replace it with your IP addr/24)

m) Configuration # cd /etc

ls

cd /snort # ls

geditsnort.conf Go to line no. 51

ipvar HOME_NET any

Replace "any" with your ip address i.e. ipvar HOME_NET 192.168._._

Save and close the file

n) Monitoring

snort –q –A console –i enp2s0 enp2s0 is the name of the interface

o) Perform the following nmap command on neighbour's machine and observe the output in your machine

$ nmapipaddr of your machine (This command is to be performed on neignbour's machine)

Output to be observed in SNORT terminal: IP address of the neighbour who is performing Intrusion i.e. Port Scanning

Conclusion: The SNORT can help to understand & implementation of intrusion detection process.

Experiment 10

Aim: Design personal Firewall using Iptables Theory:

All packets inspected by iptables pass through a sequence of built-in tables (queues) for processing. Each of these queues is dedicated to a particular type of packet activity and is controlled by an associated packet transformation/filtering chain.

Filter Table

Filter is default table for iptables.

Iptables's filter table has the following built-in chains.

- INPUT chain – Incoming to firewall. For packets coming to the local server.
- OUTPUT chain – Outgoing from firewall. For packets generated locally and going out of the local server.
- FORWARD chain – Packet for another NIC on the local server. For packets routed through the local server.

NAT Table

This table is consulted when a packet that creates a newconnection is encountered. Iptable's NAT table has the following built-in chains.

- PREROUTING chain – Alters packets before routing. i.e Packet translation happens immediately after the packet comes to the system (and before routing). This helps to translate the destination ip address of the packets to something that matches the routing on the local server. This is used for DNAT (destination NAT).
- POSTROUTING chain – Alters packets after routing. i.e Packet translation happens when the packets are leaving the system. This helps to translate the source ip address of the packets to something that might match the routing on the destination server. This is used for SNAT (source NAT).
- OUTPUT chain – NAT for locally generated packets on the firewall.

Mangle Table

Iptables's Mangle table is for specialized packet alteration. This alters QOS bits in the TCP header. Mangle table has the following built-in chains.

- PREROUTING chain

- OUTPUT chain
- FORWARD chain
- INPUT chain
- POSTROUTING chain

Raw Table

Iptable's Raw table is for configuration exemptions. Raw table has the following built-in chains.

- PREROUTING chain
- OUTPUT chain

Security Table

This table is used for Mandatory Access Control (MAC) networking rules, such as those enabled by the SECMARK and CONNSECMARK targets. Mandatory Access Control is implemented by Linux Security Modules such as SELinux. The security table is called after the filter table, allowing any Discretionary Access Control (DAC) rules in the filter table to take effect before MAC rules. This table provides the following built-in chains: INPUT (for packets coming into the box itself), OUTPUT (for altering locally-generated packets before routing), and FORWARD (for altering packets being routed through the box).

Chains :-Tables consist of *chains*, Rules are combined into different chains. The kernel uses chains to manage packets it receives and sends out. A chain is simply a checklist of rules which are lists of rules which are followed in order. The rules operate with an if-then -else structure.

Input – This chain is used to control the behaviour for incoming connections. For example, if a user attempts to SSH into your PC/server, iptables will attempt to match the IP address and port to a rule in the input chain.

Forward – This chain is used for incoming connections that aren't actually being delivered locally. Think of a router – data is always being sent to it but rarely actually destined for the router itself; the data is just forwarded to its target.

Output – This chain is used for outgoing connections. For example, if you try to ping howtogeek.com, iptables will check its output chain to see what the rules are regarding ping and howtogeek.com before making a decision to allow or deny the connection attempt.

Targets:

ACCEPT: Allow packet to pass through the firewall. DROP: Deny access by the packet.

REJECT: Deny access and notify the server. QUEUE: Send packets to user space.

RETURN: jump to the end of the chain and let the default target process it

iptables command Switch	Description
-L	Listing of rules present in the chain
-n	Numeric output of addresses and ports
-v	Displays the rules in verbose mode
-t <-table->	If you don't specify a table, then the filter table is assumed. As discussed before, the possible built-in tables include: filter, nat, mangle
-j <target>	Jump to the specified target chain when the packet matches the current rule.
-A	Append rule to end of a chain
-F	Flush. Deletes all the rules in the selected table
-p <protocol-type>	Match protocol. Types include, icmp, tcp, udp, and all
-s <ip-address>	Match source IP address
-d <ip-address>	Match destination IP address
-i <interface-name>	Match "input" interface on which the packet enters.
-o <interface-name>	Match "output" interface on which the packet exits

Steps:-

Get root access: $ sudosu root

apt-get install iptables

Commands:-

To see the list of iptables rules

iptables -L

. Initially it is empty

To block outgoing traffic to a particular destination for a specific protocol from a machine

Syntax: iptables -I OUTPUT -s <your ip> -d <neighbourip> -p <protocol> -j <action> Open one terminal and Ping the neighbour. Let the ping run.

#ping 192.168.208.6

Open another terminal and run the iptables command

iptables -I OUTPUT -s 192.168.208.18 -d 192.168.208.6 -p icmp -j DROP

To allow outgoing traffic to a particular destination for a specific protocol from a machine

iptables -I OUTPUT -s 192.168.208.18 -d 192.168.208.6 -p icmp -j ACCEPT

To block outgoing traffic to a particular destination for a specific protocol from a machine for sometime

iptables -I OUTPUT -s 192.168.208.18 -d 192.168.208.6 -p icmp -j REJECT

Allow the traffic again by using ACCEPT instead of REJECT

To block incoming traffic from particular destination for a specific protocol to machine

Syntax: iptables -I INPUT -s <neighbourip> -d <firewall ip> -p <protocol> -j <action> Open one terminal and Ping the neighbour. Let the ping run.

#ping 192.168.208.6

Open another terminal and run the iptables command

#iptables -I INPUT -s 192.168.208.6 -d 192.168.208.18 -p icmp -j DROP

To allow incoming traffic from particular destination for a specific protocol to machine

Syntax: iptables -I INPUT -s <neighbourip> -d <firewall ip> -p <protocol> -j <action>

Open another terminal and run the iptables command

#iptables -I INPUT -s 192.168.208.6 -d 192.168.208.18 -p icmp -j ACCEPT

Check the ping status on the other terminal

To clear the rules in iptables

iptables -F

To block specific URL from machine

 # iptables -t filter -I INPUT -m string --string facebook.com -j REJECT --algo kmp It will block facebook.com by performing string matching. The algorithm used for string matching is KMP.

If we change target *from REJECT to ACCEPT*, the site can be visited again.

Observations:

In case of OUTPUT chain, for DROP and REJECT chain, at source machine we get two different messages.

For DROP – 'Operation Not Permitted'. Here No acknowledgement is provided. For REJECT – 'Destination Port Unreachable'. Here acknowledgement is given.

In case of INPUT chain for DROP and REJECT chain at source machine we get two different responses as follows:

For DROP – No message. Here No acknowledgement is provided.

For REJECT – 'Destination Port Unreachable'. Here acknowledgement is given.

Conclusion : The implementation of Iptables helps to understand the working principal of Firawall in Network security.