EXP@1 - To study IP spoofing and ARP spoofing over a local area network.

#ip spoofing

#start 3 pcs

vstart pc1 –eth0=A

vstart pc2 –eth0=A

vstart pc3 –eth0=A

Assign IP addresses to PC1,PC2 and PC3

ifconfig eth0 192.168.1.11

ifconfig eth0 192.168.1.12

ifconfig eth0 192.168.1.13

#ping from pc1 to pc3

Ping ipaddress

PC1 will spoof IP address of PC2

#in pc1

iptables -t nat -A POSTROUTING -p icmp -j SNAT --to-source 192.168.1.12

in pc2

Tcpdump -I any

#in pc1

Ping <pc3 ipaddess>

#arp spoofing

Sudo apt-get install arpwatch

Service arpwatch status

Ping <ipaddress>

#Now change the MAC address of your system using ifconfig command.

Sudo ifconfig enp1s0 hw ether <address = 00:1a:ff:0a:e7:1b>

#now ping the same address again

#to check log

Sudo tail -f /var/log/syslog

#to check the changed mac address

Systemctl daemon-reload

Systemctl start arpwatch@enp4s0

Sudo tail -f /var/log/syslog | grep arpwatch

EXP@3 - To Study use of Iptables to configure stateful Software firewall on Linux Host.

View Tables INPUT, OUTPUT and FORWARD.

Sudo iptables -L

Policy DROP for INPUT , OUTPUT and FORWARD chain.

Sudo iptables -P INPUT DROP

Sudo iptables -P OUTPUT DROP

Sudo iptables -L

#To allow connection for INPUT

Sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED, RELATED -j ACCEPT

Sudo iptables -L

#To allow connection for OUTPUT

Sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED, RELATED -j ACCEPT

Sudo iptables -L

#for ssh Allow All Incoming SSH

iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT

#The second command, which allows the outgoing traffic of established SSH connections, is only necessary if the OUTPUT policy is not set to ACCEPT.

Sudo ufw disable

Ssh apsit@192.168.104.3

Exit

#http and https

#first search on any browser error

iptables -A INPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate ESTABLISHED -j ACCEPT

#now search accept

#mysql

iptables -A INPUT -p tcp -s 15.15.15.0/24 --dport 3306 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -p tcp --sport 3306 -m conntrack --ctstate ESTABLISHED -j ACCEPT

#mysql start and run it

EXP@4 - To study analysis of network packets by sing open source sniffing tools like tcpdump and Wireshark in promiscuous and non-promiscuous mode

sudo apt-get install tcpdump

tcpdump -D : display all available interfaces

tcpdump -i any

tcpdump -i wlo1 (mam's command)

tcpdump -i enp4s0 port 80 (our command)

tcpdump -i enp4s0 -c 5

tcpdump -i enp4s0 tcp (captures tcp traffic)

sudo tcpdump -i enp4s0 -x -X -A -nvvv port 22 > ssh.text

sudo tcpdump -i wlo1 -x -X -A -nvvv port 22 > ssh.txt

ssh apsit@ipaddress

To capture telnet packet:

sudo tcpdump -i wlo1 -x -X -A -nvvv port 23 > telnet.txt

telnet <ipaddress>

sudo wireshark

#search

Ssh

Tcp.stream eq 17

EXP@5 - To use nmap for network discovery and security auditing.

Sudo apt-get install nmap

To scan a single system

Nmap -sP ipaddress

To scan the entire subnet

Nmap -sP ipaddress/24

To scan a multiple targets

Nmap -sL ipadress ipadresss

To scan the entire subnet but not a specific IP addresses

Nmap -sL ipadress/24 exclude ipaddress

To scan a specific port on the target machines

Nmap -p 80,21,23 ipaddress

To know the open ports on target system

Nmap -open ipadress

#syn packets

Sudo nmap -sS ipadress

#udp packets

Sudo nmap -sU ipadress

#version detection

Sudo nmap -sV ipadress

#OS detection

Sudo nmap -O ipadress