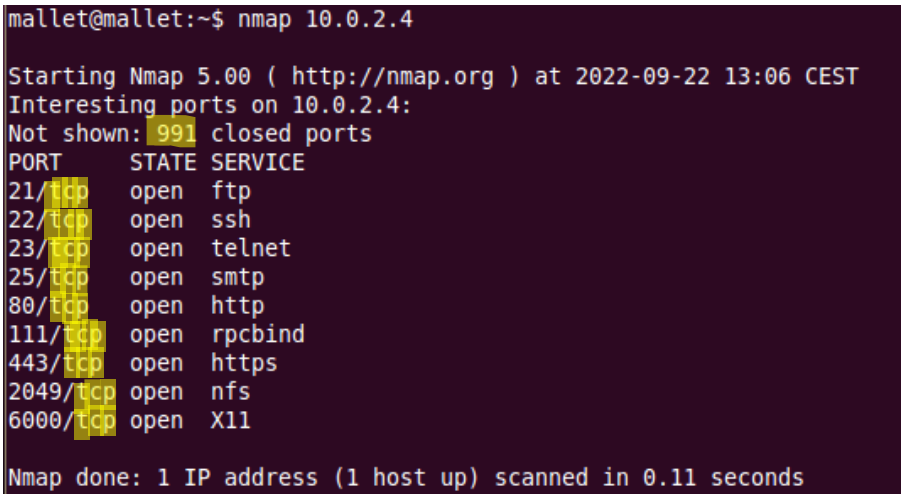
Práctica 2. Análisis de Servicios de Red

1. *Usando la página de manual de nmap, documente las funciones y opciones básicas de nmap.*

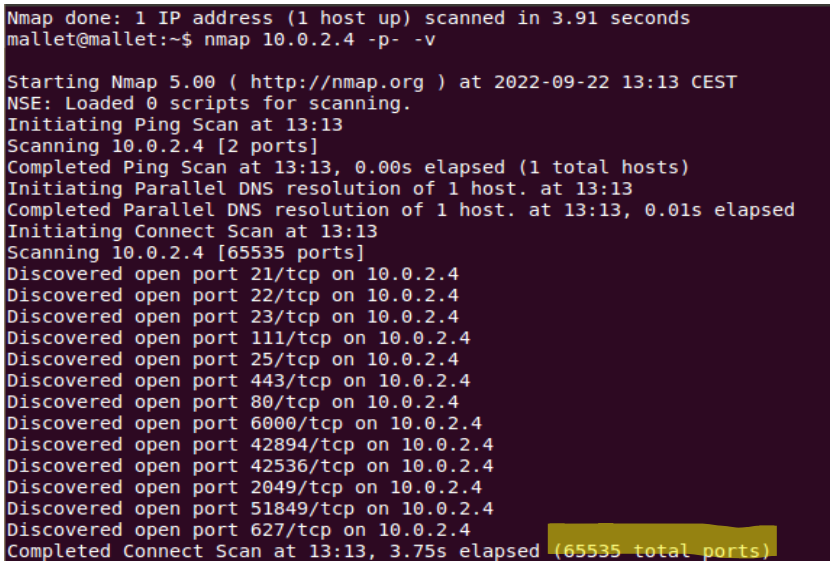
   Nmap es una herramienta utilizada para la exploración de red y auditoría de seguridad. Se diseñó para analizar rápidamente grandes redes, aunque también es utilizado como analizador de puertos para descubrir los servicios que se ejecutan en un servidor o máquina.

   Una vez que sabemos lo que es y para que sirve "nmap", vamos a ver que argumentos podemos utilizar para ejecutar algunas opciones básicas.

   Este comando analiza por defecto los 1000 servicios (puertos) más utilizado de tipo TCP.



   Aquí hemos hecho un nmap desde Mallet a la dirección IP de la maquina correspondiente a Alice y como se puede observar ha analizado solo 1000 puertos de tipo TCP. En la cabecera de los paquetes de red hay 16 bits, es decir, 2^{16} puertos (65535 puertos) por tanto analizar solo 1000 es un poco pobre. Para solucionar esto nos podemos valer del **argumento "-p-".**



El uso del **argumento "-v"** (verbose) nos permite ver los servicios que está explorando nmap.

Este argumento hace que se escaneen los 65535 servicios del servidor. Ahora que hemos analizado todos los puertos TCP tocaría analizar todos los puertos UDP. Esto se puede hacer añadiendo el **argumento "-sU"** al argumento anterior.

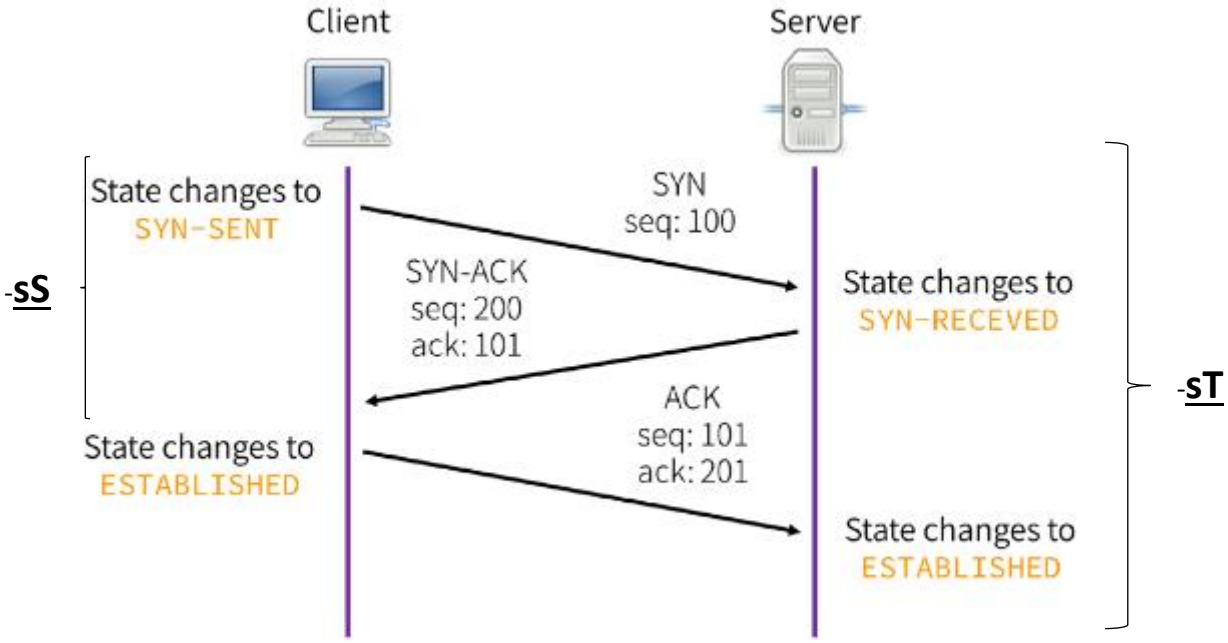
```
mallet@mallet:~$ sudo nmap 10.0.2.4 -p- -sU -v

Starting Nmap 5.00 ( http://nmap.org ) at 2022-09-22 13:17 CEST
NSE: Loaded 0 scripts for scanning.
Initiating ARP Ping Scan at 13:17
Scanning 10.0.2.4 [1 port]
Completed ARP Ping Scan at 13:17, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:17
Completed Parallel DNS resolution of 1 host. at 13:17, 0.01s elapsed
Initiating UDP Scan at 13:17
Scanning 10.0.2.4 [65535 ports]
Increasing send delay for 10.0.2.4 from 0 to 50 due to max_successful_tryno increase to 4
Increasing send delay for 10.0.2.4 from 50 to 100 due to max_successful_tryno increase to 5
```

Otro argumento que cabe mencionar es **"-n".** Con este lo que hacemos es evitar que haga una resolución inversa de DNS, es decir, si ya sabemos la IP que no consiga el nombre de la máquina. Con esto nos ahorraremos tiempo.

Para buscar un servicio en concreto nos valdremos del **argumento "-p"** seguido del número de puerto correspondiente.

Para la búsqueda de servicios TCP hay diferentes tipos de escaneo, el más sigiloso es el escaneo de tipo SYN, que se ejecuta mediante el **argumento "-sS"**. Para un escaneo TCP completo se utilizaría el **argumento "-sT".**



Todos los argumentos anteriores se pueden combinar para ejecutar una búsqueda los más apropiada posible. Por ejemplo, vamos a usar una búsqueda TCP completa en el puerto 80 sin resolución DNS.

```
mallet@mallet:~$ nmap 10.0.2.4 -n -p80 -sT

Starting Nmap 5.00 ( http://nmap.org ) at 2022-09-22 13:47 CEST
Interesting ports on 10.0.2.4:
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

Otro argumento interesante es **"-sV",** el cual nos dice la versión del servicio escaneado.

```
mallet@mallet:~$ nmap www.uva.es -p80 -sV

Starting Nmap 5.00 ( http://nmap.org ) at 2022-09-22 13:56 CEST
Stats: 0:00:21 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Interesting ports on innova.uva.es (157.88.25.8):
PORT   STATE SERVICE VERSION
80/tcp open  http?
```

En este caso no conseguimos saber la versión del servicio porque la versión de "nmap" es un poco antigua.

Otros argumentos de interés pueden ser **"-d",** que incrementa el nivel de depuración, haciendo que Nmap imprima detalles sobre su funcionamiento que pueden ser útiles para rastrear errores. Como es posible que se arrojen demasiados datos se puede modificar el nivel de profundidad del 1 al 9 para que leer los resultados sea más cómodo.

```
mallet@mallet:~$ nmap 10.0.2.4 -d1

Starting Nmap 5.00 ( http://nmap.org ) at 2022-09-23 14:48 CEST
PORTS: Using top 1000 ports found open (TCP:1000, UDP:0, SCTP:0)
--------------- Timing report ---------------
  hostgroups: min 1, max 100000
  rtt-timeouts: init 1000, min 100, max 10000
  max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
  parallelism: min 0, max 0
  max-retries: 10, host-timeout: 0
  min-rate: 0, max-rate: 0
---------------------------------------------
NSE: Loaded 0 scripts for scanning.
Initiating Ping Scan at 14:48
Scanning 10.0.2.4 [2 ports]
Completed Ping Scan at 14:48, 0.00s elapsed (1 total hosts)
Overall sending rates: 3252.03 packets / s.
mass_rdns: Using DNS server 212.166.132.116
mass_rdns: Using DNS server 212.166.132.104
Initiating Parallel DNS resolution of 1 host. at 14:48
mass_rdns: 13.01s 0/1 [#: 2, OK: 0, NX: 0, DR: 0, SF: 0, TR: 4]
Completed Parallel DNS resolution of 1 host. at 14:48, 13.01s elapsed
DNS resolution of 1 IPs took 13.01s. Mode: Async [#: 2, OK: 0, NX: 0, DR: 1, SF: 0, TR: 4, CN: 0]
Initiating Connect Scan at 14:48
Scanning 10.0.2.4 [1000 ports]
Discovered open port 111/tcp on 10.0.2.4
Discovered open port 443/tcp on 10.0.2.4
Discovered open port 25/tcp on 10.0.2.4
Discovered open port 80/tcp on 10.0.2.4
Discovered open port 23/tcp on 10.0.2.4
Discovered open port 22/tcp on 10.0.2.4
Discovered open port 21/tcp on 10.0.2.4
Discovered open port 6000/tcp on 10.0.2.4
Discovered open port 2049/tcp on 10.0.2.4
Completed Connect Scan at 14:48, 0.08s elapsed (1000 total ports)
Overall sending rates: 12494.53 packets / s.
Host 10.0.2.4 is up, received syn-ack (0.0014s latency).
Scanned at 2022-09-23 14:48:00 CEST for 13s
Interesting ports on 10.0.2.4:
Not shown: 991 closed ports
Reason: 991 conn-refused
```

--open muestra los hosts que tienen puertos abiertos, y sólo muestra los puertos abiertos para esos. Aquí, los "puertos abiertos" son cualquier puerto que tenga la posibilidad de estar abierto, lo que incluye abierto, abierto/filtrado y sin filtrar.

```
mallet@mallet:~$ nmap 10.0.2.4 --open -n

Starting Nmap 5.00 ( http://nmap.org ) at 2022-09-23 14:49 CEST
Interesting ports on 10.0.2.4:
Not shown: 991 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
80/tcp   open  http
111/tcp  open  rpcbind
443/tcp  open  https
2049/tcp open  nfs
6000/tcp open  X11

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

-6 es útil si queremos que analice el objetivo utilizando el protocolo IPv6.

--**reason** añade una columna a la tabla de puertos que describe por qué Nmap clasificó un puerto como lo hizo.

```
mallet@mallet:~$ nmap 10.0.2.4 --reason -n

Starting Nmap 5.00 ( http://nmap.org ) at 2022-09-23 14:57 CEST
Interesting ports on 10.0.2.4:
Not shown: 991 closed ports
Reason: 991 conn-refused
PORT     STATE SERVICE REASON
21/tcp   open  ftp     syn-ack
22/tcp   open  ssh     syn-ack
23/tcp   open  telnet  syn-ack
25/tcp   open  smtp    syn-ack
80/tcp   open  http    syn-ack
111/tcp  open  rpcbind syn-ack
443/tcp  open  https   syn-ack
2049/tcp open  nfs     syn-ack
6000/tcp open  X11     syn-ack

Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

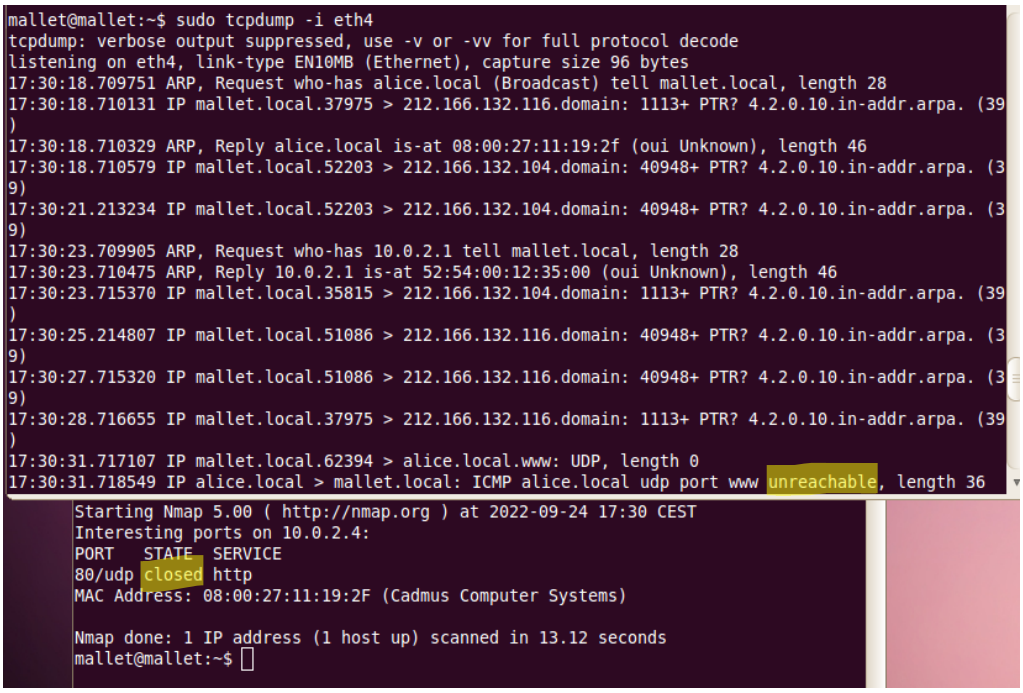
2. *Active un proceso de monitorización tcp en Mallet para poder seguir los diferentes métodos de scanning.*

   Activamos un proceso de monitorización tcp a través de la instrucción tcpdump. Le añadiremos el argumento -i para señalar cual es la interfaz de red que queremos analizar, en caso de Mallet es la eth4 (lo podemos ver con la instrucción ip a). También añadiremos el tipo de escaneo que vamos a realizar (tcp) y -v (verbose) para poder ver que está analizando. En nuestro caso vamos a analizar un puerto específico, el 80. Para ello añadimos el argumento port seguido del número correspondiente y para obtener más información acabaremos con el argumento -e.

```
mallet@mallet:~$ sudo tcpdump -i eth4 tcp -v port 80 -e
tcpdump: listening on eth4, link-type EN10MB (Ethernet), capture size 96 bytes
```

3. *Usando nmap, realice y documente un barrido de puertos UDP en Alice y compare con los datos que se obtienen a través de tcpdump.*

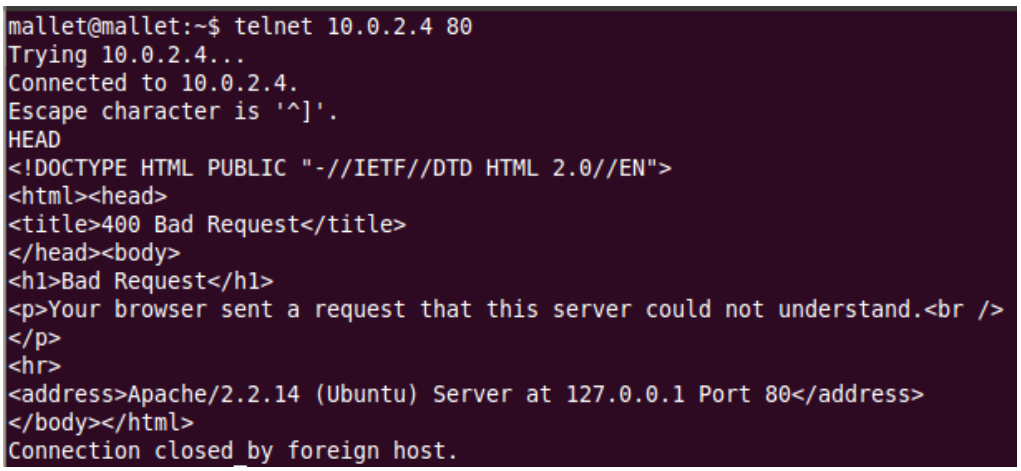
   Realizaremos el barrido con las instrucciones y argumentos explicados en el primer punto. Para analizar y comparar en otra terminal utilizaremos lo explicado en el punto anterior con respecto a tcpdump.



   Utilizando tcpdump podemos ver que el puerto 80 de Alice no es accesible, cosa que se puede confirmar al ver que el nmap nos ha dado como resultado que el puerto 80 está cerrado.

4. *Documente toda la información sobre el servicio web que se ejecuta en Alice usando una simple conexión telnet.*

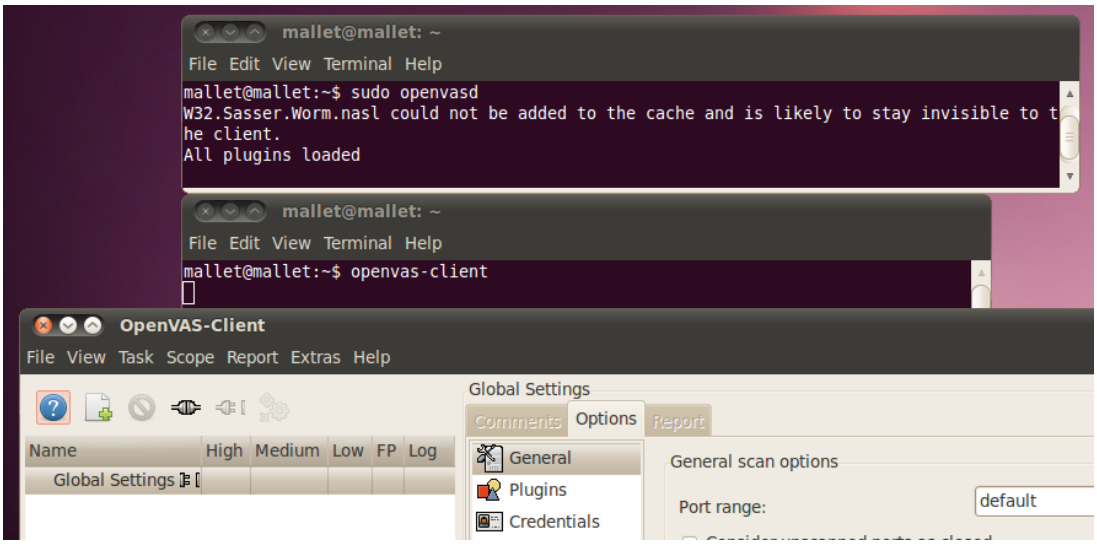
   Analizaremos el servicio web correspondiente al puerto 80 mediante la siguiente instrucción:



   Poniendo HEAD obtenemos información del HTML. Para obtenerlo en formato original podemos poner “i” en vez de HEAD. Se observa que se utiliza la tecnología Apache.

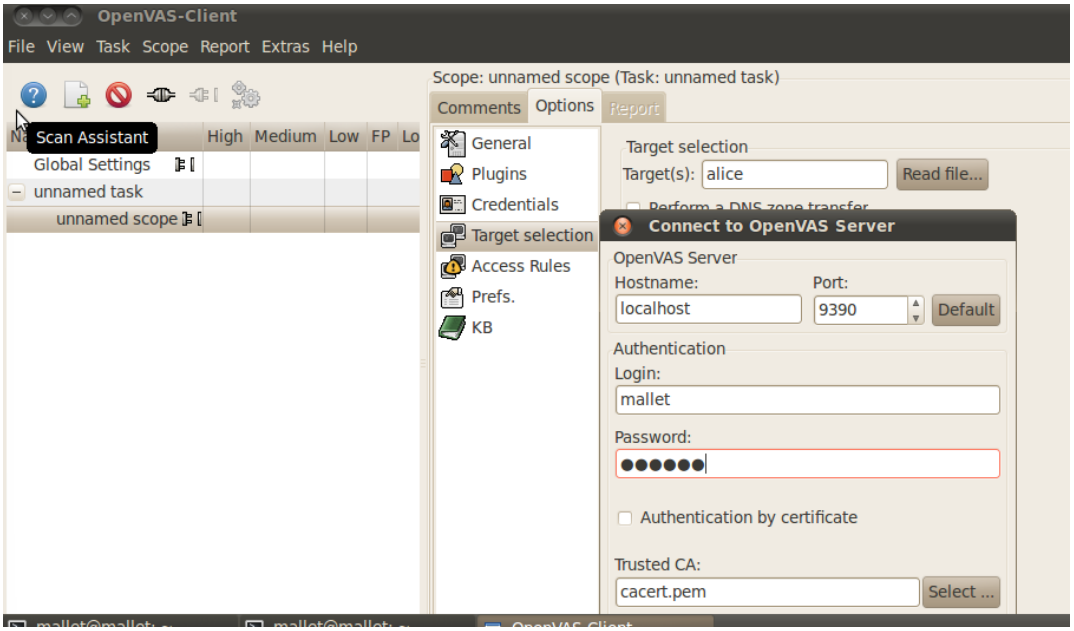
5. *Elabore un informe de vulnerabilidades de Alice usando el analizador de barrido openvas. Documente cómo debe iniciarse este analizador y los resultados que encuentre.*

Primero iniciamos el servidor de openvasd. Después iniciamos un cliente, se nos desplegará una interfaz.

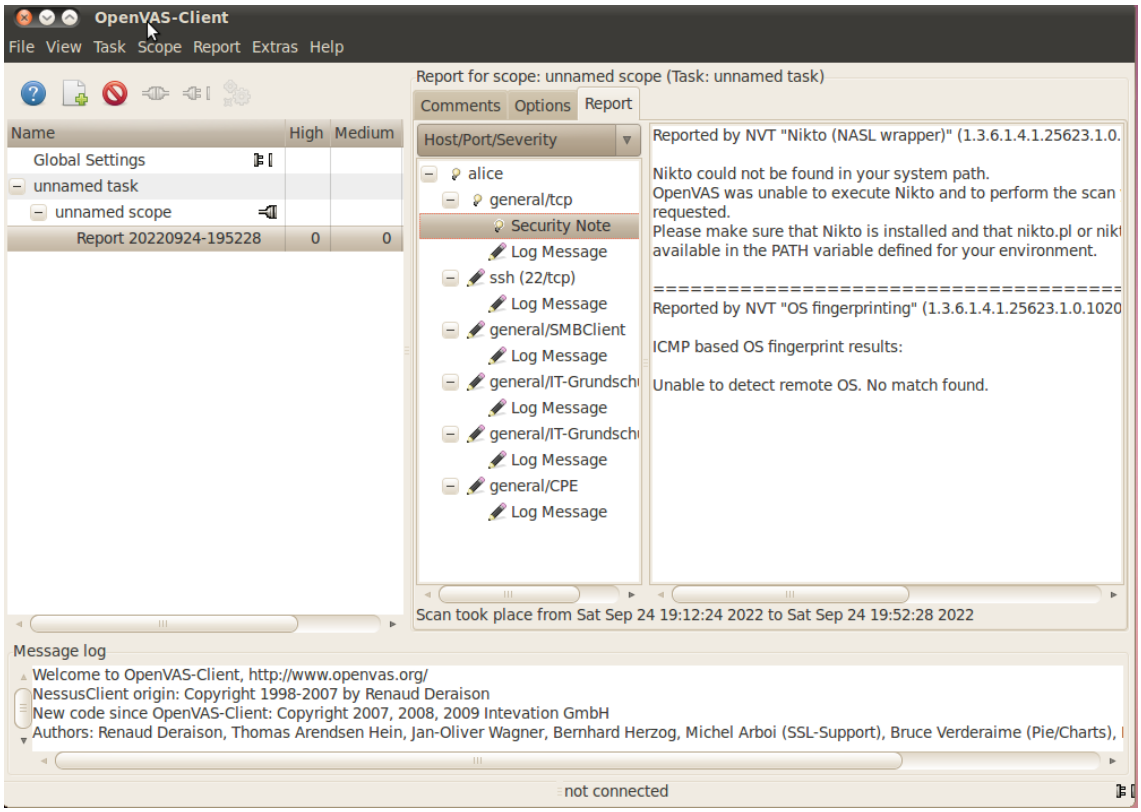


Una vez desplegada nos iremos al submenú de la parte superior y pulsaremos en Task, añadimos una nueva, depues nos vamos a Scope y también añadimos una nueva. Una vez creada, nos iremos a opciones y Target Selection, pulsaremos en los engranajes del menú superior izquierdo y añadiremos la contraseña de Mallet.

Para que comience el escaneo le daremos a Ok en la parte inferior derecha.



Tras unos 20 minutos obtendremos todas las posibles vulnerabilidades de la Máquina de Alice.



6. *Recopile la información anterior (desde el punto "b" al punto "e"), ahora para Bob en lugar de Alice.*

   Como estos pasos están descritos anteriormente, la debida explicación estará en el apartado correspondiente.

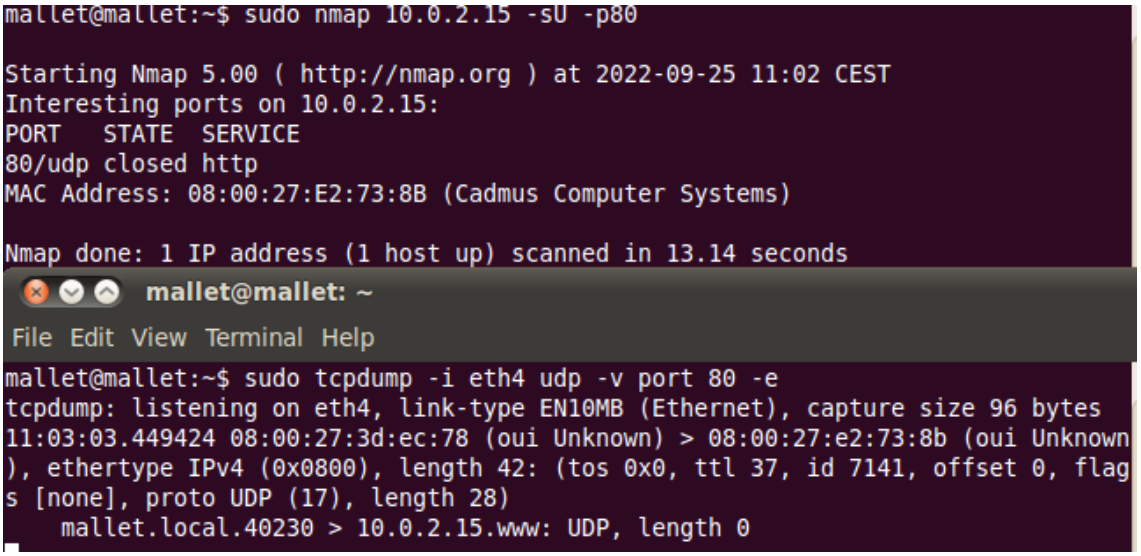
   Primero vemos cual es la dirección IP de Bob con la instrucción “ip a”. Vemos que la IP es una dirección de otra red, para poder cambiarla a la misma red que Mallet levantaremos la tarjeta de red, que en nuestro caso es la interfaz eth0. Usaremos la instrucción “ifup eth0” y para ver la nueva IP volveremos a utilizar “ip a”. Ahora la dirección IP está en la misma red (10.0.2.15/24). Haremos un ping desde Mallet hasta Bob para comprobar que ambas máquinas se pueden comunicar entre sí.

```
mallet@mallet:~$ ping 10.0.2.15 -c 4
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=1.88 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=1.25 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.991 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=1.30 ms

--- 10.0.2.15 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 0.991/1.357/1.884/0.327 ms
```

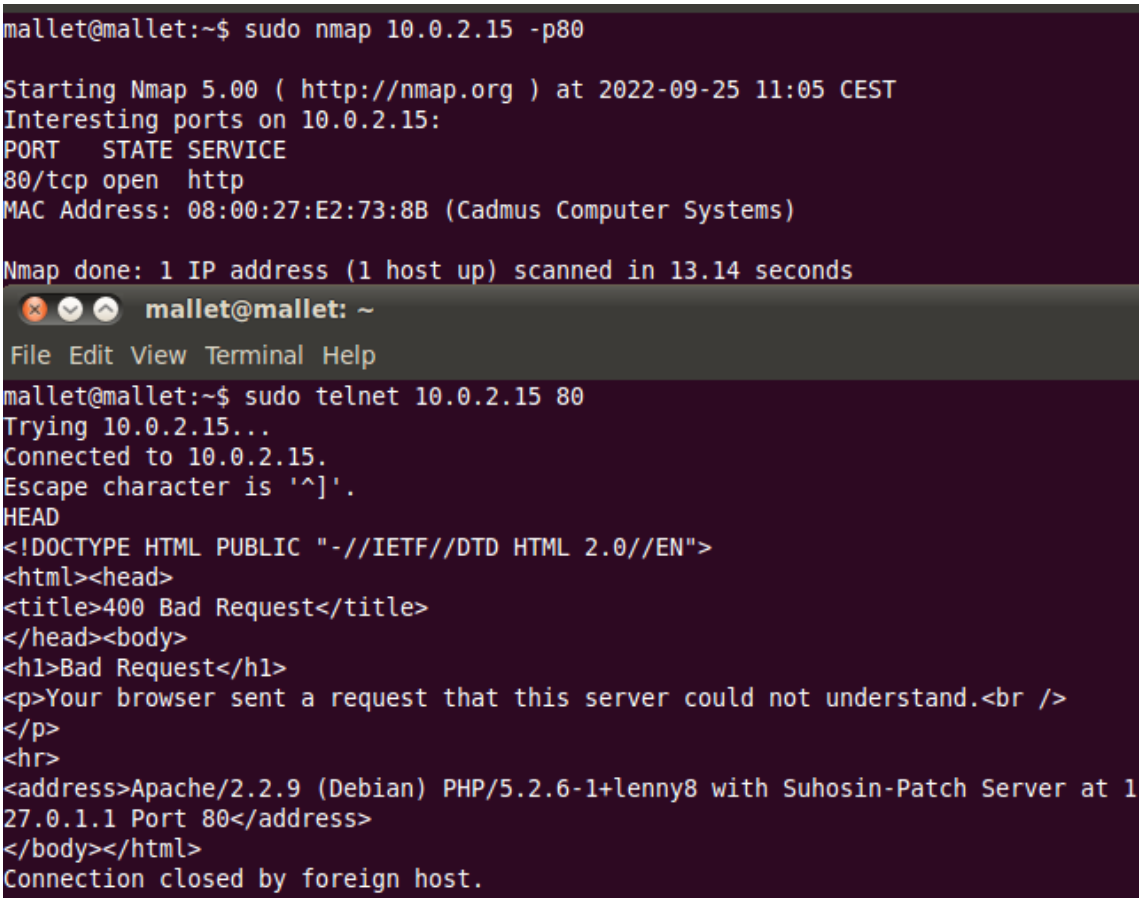
   Vemos que si que hay conexión entre ambas máquinas.

Ahora realizaremos un barrido UDP en el puerto 80, utilizando tcpdump para escanearlo y nmap para ver el estado del puerto.



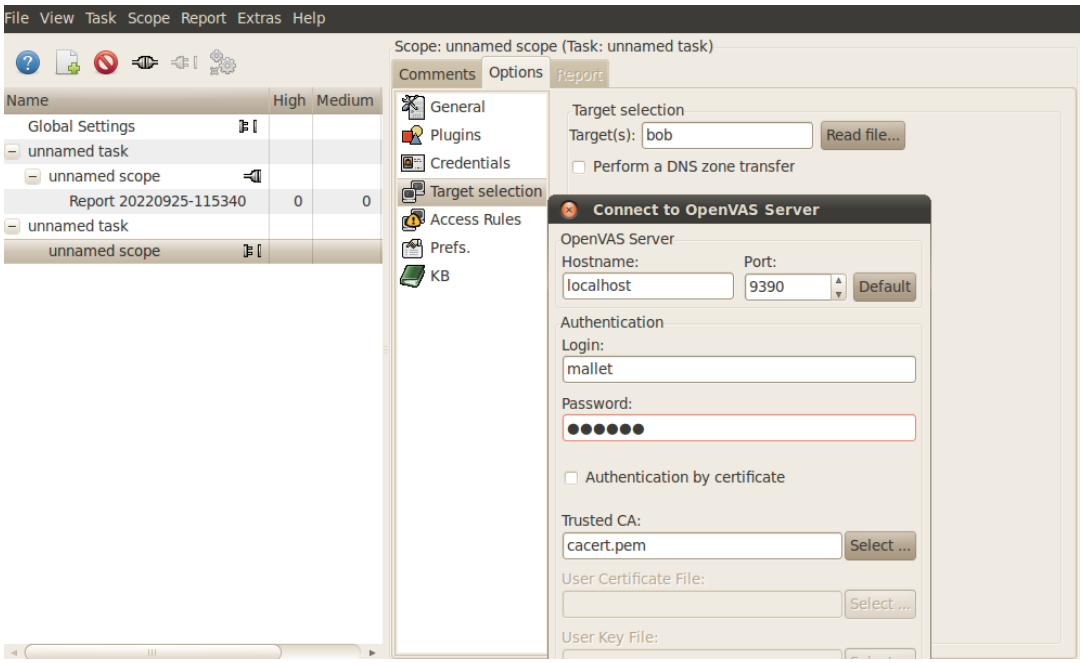
Podemos ver que el puerto 80 UDP está cerrado. En la terminal tcpdump se puede ver la comunicación entre máquinas.

Ahora comprobaremos el estado del puerto 80 TCP para luego realizar la instrucción telnet (HEAD).



Por último, veremos las vulnerabilidades de la máquina de bob usando openvas.

Sustituimos a Alice por Bob en las instrucciones descritas antes e iniciamos el escaneo.



Iniciamos el escaneo y tras unos minutos obtenemos el informe.