

Práctica de laboratorio: Detección de redes con Wireshark

Objetivos

En esta práctica de laboratorio, utilizará la utilidad de Linux **tcpdump** para capturar y guardar el tráfico de red. Luego utilizará Wireshark para investigar la captura de tráfico.

- Prepare el host para capturar el tráfico de red.
- Capture y guarde el tráfico de red.
- Ver y analizar la captura de paquetes.

Aspectos básicos/Situación

Wireshark es una utilidad de captura de paquetes de red que pueden utilizar los administradores de red para solucionar problemas de red. También se puede utilizar para espiar las comunicaciones de red para recopilar pasivamente información sobre usuarios y servicios. Wireshark se considera una herramienta pasiva porque no crea tráfico en la red.

Recursos necesarios

- Kali VM personalizada para el curso de Ethical Hacker
- Acceso a Internet

Instrucciones

Parte 1: Prepare el host para capturar el tráfico de red.

Paso 1: Iniciar la máquina virtual e iniciar sesión

- a. Inicie la máquina virtual de la estación de trabajo Kali. Utilicen las siguientes credenciales de usuario:
Nombre de usuario: **kali**
Contraseña: **kali**
- b. Inicie una sesión de terminal.

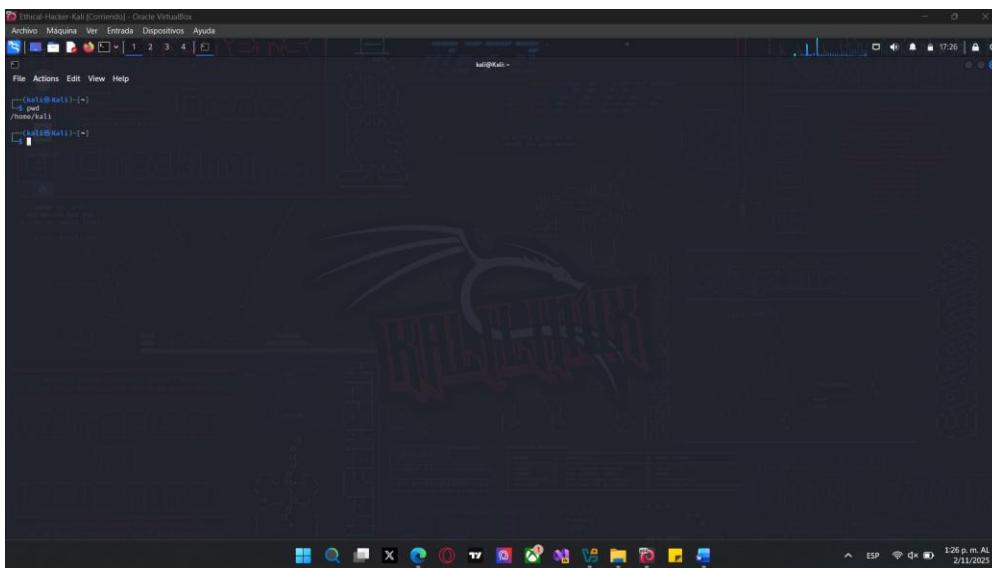
Paso 2: Verifique el entorno.

- a. Verifique el directorio de usuarios que se usará para almacenar el tráfico capturado. Utilice el comando **pwd** para ver el directorio actual. Esto mostrará la ruta completa al directorio de trabajo actual.

```
└─(kali㉿Kali)-[~]
└─$ pwd
```

Registre la ubicación del directorio.

Práctica de laboratorio: Detección de redes con Wireshark



/home/kali

- b. Determine la dirección IP de la interfaz Ethernet de Kali con el comando **ifconfig**. La interfaz ethernet generalmente se denomina **eth0**.

```
└─(kali㉿Kali)-[~]
└─$ ifconfig
```

Registre la dirección IP y la dirección MAC de la interfaz de red eth0. Esta será la dirección de origen de los paquetes que se originan en la máquina Kali.

A screenshot of a terminal window showing the output of the 'ifconfig' command. The output lists several network interfaces, with 'eth0' being the primary one. Key details from the output include:

eth0: flags=413c0^{UP,BROADCAST,RUNNING,MULTICAST} mtu 1500
 inet 192.168.0.1 brd 255.255.255.0 broadcast 192.168.0.255
 netmask 255.255.255.0 broadcast 192.168.0.255
 inet6 fe80::42:ff:fe01:9aa brd fe80::ff:ff:fe01:9aa
 prefixlen 64
 scoprid 0x20<link>
 ether 02:42:01:00:00:9aa txqueuelen 0 (Ethernet)
 RX packets 2244 bytes 94926 (92.7 Kib)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 231 bytes 20000 (2.8 Kib)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

br-35ee94a19a5c: flags=4111^{UP,BROADCAST,RUNNING,MULTICAST} mtu 1500
 inet 192.168.0.1 brd 255.255.255.0 broadcast 192.168.0.255
 netmask 255.255.255.0 broadcast 192.168.0.255
 inet6 fe80::42:ff:fe01:9aa brd fe80::ff:ff:fe01:9aa
 prefixlen 64
 scoprid 0x20<link>
 ether 02:42:01:00:00:9aa txqueuelen 0 (Ethernet)
 RX packets 83 bytes 5482 (5.4 Kib)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 83 bytes 17880 (17.8 Kib)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth-internal: flags=413c0^{UP,BROADCAST,RUNNING,MULTICAST} mtu 1500
 inet 192.168.0.1 brd 255.255.255.0 broadcast 192.168.0.255
 netmask 255.255.255.0 broadcast 192.168.0.255
 inet6 fe80::42:ff:fe01:9aa brd fe80::ff:ff:fe01:9aa
 prefixlen 64
 scoprid 0x20<link>
 ether 02:42:01:00:00:9aa txqueuelen 0 (Ethernet)
 RX packets 2244 bytes 94926 (92.7 Kib)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 232 bytes 17880 (17.8 Kib)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mon0: flags=413c0^{UP,BROADCAST,RUNNING,MULTICAST} mtu 1500
 inet 192.168.0.1 brd 255.255.255.0 broadcast 192.168.0.255
 netmask 255.255.255.0 broadcast 192.168.0.255
 inet6 fe80::42:ff:fe01:9aa brd fe80::ff:ff:fe01:9aa
 prefixlen 64
 scoprid 0x20<link>
 ether 02:42:01:00:00:9aa txqueuelen 1000 (Ethernet)
 RX packets 17604 bytes 1760448 (1.6 MiB)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 9599 bytes 1552478 (1.4 MiB)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

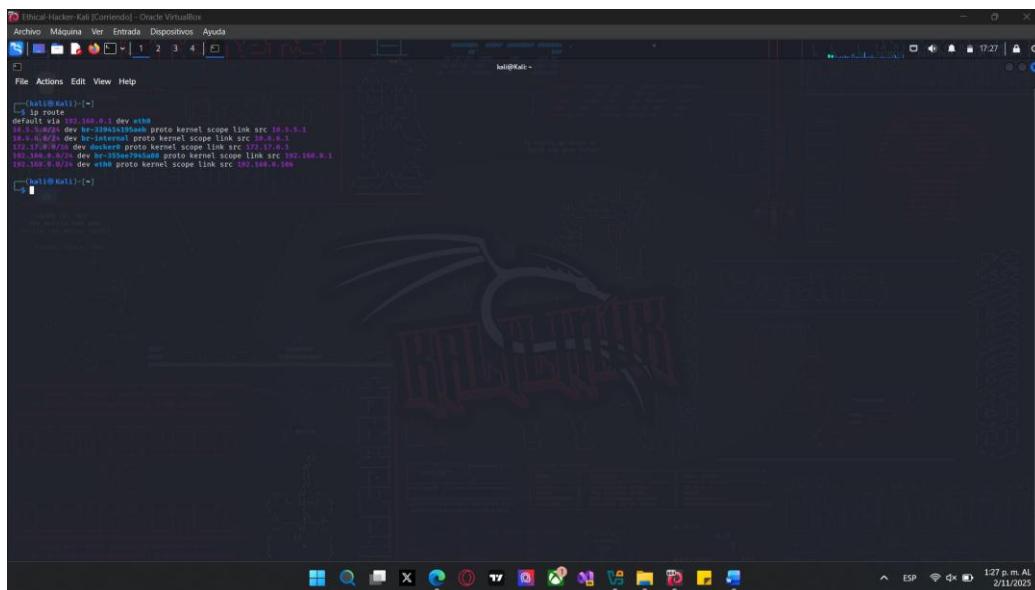
lo: flags=734^{UP,LOOPBACK,RUNNING} mtu 65536
 inet 127.0.0.1 brd 127.0.0.1
 netmask 255.255.255.255
 inet6 ::1
 prefixlen 128
 scoprid 0x8<host>
 loop txqueuelen 1000 (local Loopback)

Las respuestas pueden variar. Un ejemplo de eth0 es 10.0.2.15. El MAC podría ser 08:00:27:13:93:44

- c. Determine la puerta de enlace predeterminada asignada al host Kali mediante el comando **ip route**.

```
└─(kali㉿Kali)-[~]
└─$ ip route
```

Registre la dirección IP de la puerta de enlace predeterminada. La puerta de enlace predeterminada responde a las solicitudes ARP de direcciones IP de destino ubicadas fuera de la red de origen.



```
(kali㉿Kali)-[~]
$ ip route
default via 10.0.2.1 dev eth0
    proto kernel scope link src 10.0.2.3
    metric 100
10.0.2.0/16 dev br-int brd 10.0.2.255 scope link
    link-layer brd 00:0c:29:ff:ff:ff
    proto kernel scope link src 10.0.2.1
    metric 100
10.0.2.0/24 dev wlp4s0 brd 10.0.2.255 scope link
    link-layer brd 00:0c:29:ff:ff:ff
    proto kernel scope link src 10.0.2.4
    metric 100
--(kali㉿Kali)-[~]
```

Las respuestas pueden variar. Un ejemplo de dirección de puerta de enlace predeterminada es 10.0.2.2.

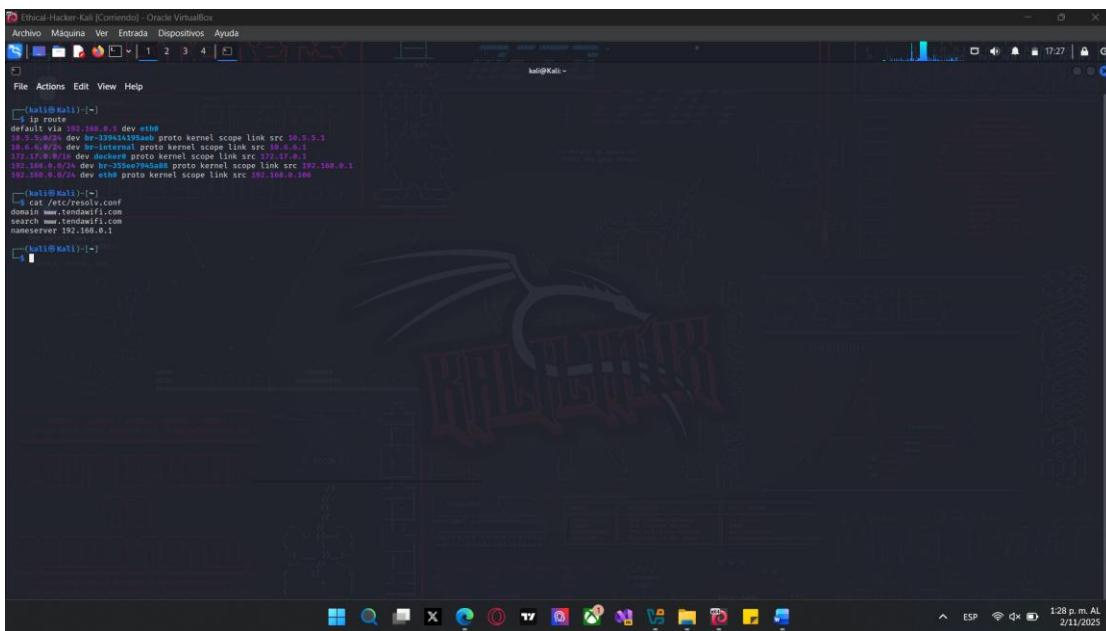
Nota: La dirección IP de la puerta de enlace predeterminada debe estar en la misma subred IP que la dirección de la interfaz Ethernet (eth0).

- d. Determine la dirección del servidor DNS predeterminado configurado mostrando el contenido del archivo **/etc/resolv.conf**. Puede ver el archivo con el comando **cat** .

```
└── (kali㉿Kali)-[~]
    └─$ cat /etc/resolv.conf
```

Registre la dirección IP del servidor DNS predeterminado configurado. La dirección IP del servidor DNS será la dirección de destino de los paquetes de consulta estándar y la dirección de origen de los paquetes de respuesta DNS.

Práctica de laboratorio: Detección de redes con Wireshark



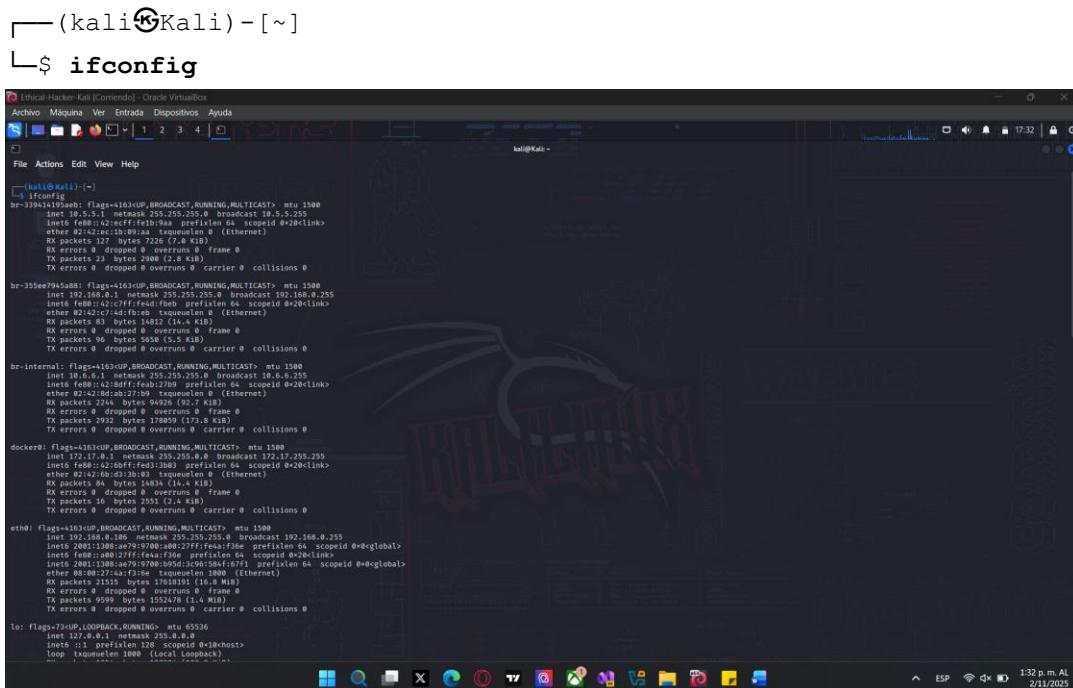
Las respuestas variarán según la configuración de la red del host físico.

Parte 2: Capture y guarde el tráfico de red.

En esta parte utilizarán **tcpdump** para capturar el contenido del tráfico HTTP. Utilizarán opciones de comando para guardar el tráfico en un archivo de captura de paquetes (pcap). Estos registros se pueden analizar posteriormente con diferentes aplicaciones que leen archivos pcap, incluida Wireshark.

Paso 1: Abrir un terminal e iniciar tcpdump

- Abran una aplicación del terminal e introduzcan el comando **ifconfig**.



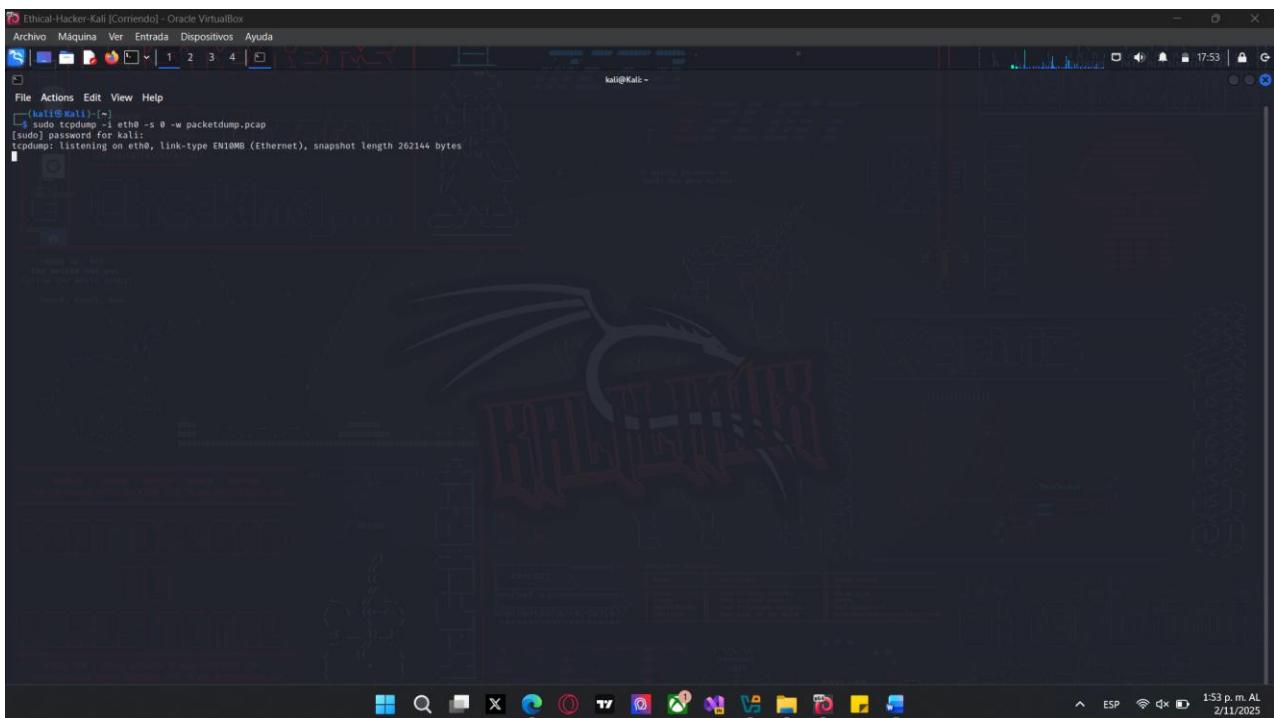
- b. En la salida de **ifconfig**, busque el nombre de la interfaz que corresponda al adaptador Ethernet (generalmente eth0). Haga clic con el botón derecho en el nombre de la interfaz y seleccione **Copiar selección**.
- c. Ingrese el comando **sudo tcpdump** como se muestra. Utilice el nombre de la interfaz que copió en el paso anterior, como en el ejemplo siguiente. Este comando requiere acceso de usuario raíz, así que ingrese **kali** como contraseña si se le solicita.

```
└── (kali㉿Kali)-[~]
└─$ sudo tcpdump -i eth0 -s 0 -w packetdump.pcap
```

La opción de comando **-i** les permite especificar la interfaz. Si no se la especifica, tcpdump capturará todo el tráfico en todas las interfaces.

La opción de comando **-s** especifica la longitud de la instantánea correspondiente a cada paquete. Al establecer esta opción en 0, se establece el valor predeterminado de 262144.

La opción de comando **-w** se utiliza para escribir el resultado del comando **tcpdump** en un archivo. Si se agrega la extensión **.pcap**, se garantiza que los sistemas operativos y las aplicaciones podrán leer el archivo. Todo el tráfico registrado se imprimirá al archivo **httpsdump.pcap**, en el directorio de inicio del usuario.



Paso 2: Paso 2: Generar tráfico de red mediante un navegador web.

- a. Para capturar una solicitud y una respuesta HTTP, abra un navegador web en el escritorio de Kali. Navegue a **Google.com**. No inicie sesión ni busque.
- b. Abra una segunda pestaña en el navegador, ingrese **skillsforall.com** en la barra de inicio. Cuando aparezca la página, haga clic en el icono de usuario en la parte superior derecha de la página. Inicie sesión con su información de inicio de sesión de skillsforall.
- c. Regrese a la ventana de terminal que ejecuta la utilidad **tcpdump** e ingrese **CTRL-C** para completar la captura de paquetes.

Práctica de laboratorio: Detección de redes con Wireshark

Ethical-Hacker-Kali [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

File Actions Edit View Help

```
[kali㉿kali: ~]# tcpreplay -i eth0 -s 0 -w packetdump.pcap
[sudo] password for kali:
tcpreplay: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
[tcpreplay] 0 packets transmitted, 0 captured
838 packets received, 0 filtered
0 packets dropped by kernel
[kali㉿kali: ~]
```

- d. La utilidad tcpdump guardó la salida en un archivo denominado **packetdump.pcap**. Este archivo debe guardarse en el directorio de inicio predeterminado. Verifique que el archivo exista en el directorio con el comando **ls**.

```
└─(kali㉿Kali)-[~]
└─$ ls packetdump.pcap
packetdump.pcap
```

A screenshot of a Kali Linux desktop environment. The terminal window at the bottom left shows the command 'ls /root/packetdump.pcap' being run, listing files like 'packetdump.pcap'. The main window is titled 'Wireshark - Network Minus' and displays a packet capture from 'kali@kali: ~'. The interface includes a toolbar, a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help', and a status bar at the bottom right showing the date and time.

Parte 3: Ver y analizar la captura de paquetes.

En esta parte de la práctica de laboratorio, utilizará Wireshark para analizar el archivo de captura de paquetes que creó en la parte anterior.

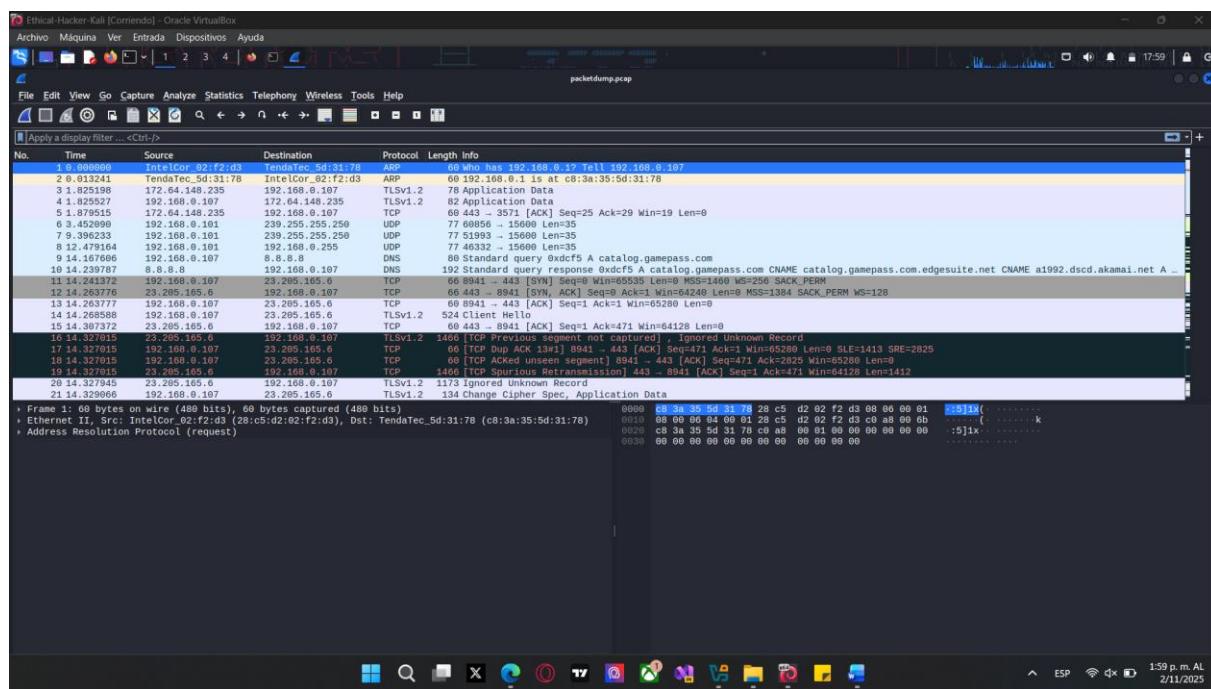
Paso 1: Abra la aplicación Wireshark para ver la captura de paquetes.

- e. Utilice Wireshark para ver los paquetes capturados. Inicie la aplicación gráfica Wireshark escribiendo **wireshark** en el indicador de la terminal.

```
└── (kali㉿Kali)-[~]
└─$ wireshark
```

La aplicación Wireshark debería abrirse en una ventana diferente. Expanda la ventana de Wireshark a pantalla completa.

- f. Utilice la opción de menú **File -> Open** y busque el archivo pcap. Haga clic en **Open**. Debe abrirse una pantalla que muestre el contenido del archivo **packetdump.pcap**.

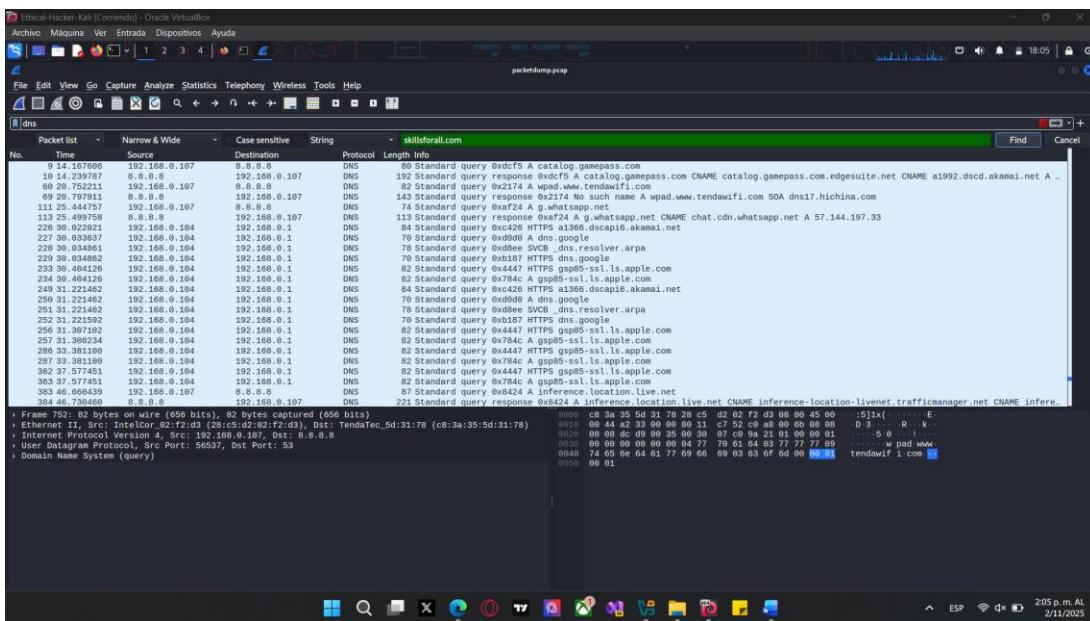


Paso 2: Analizar el tráfico DNS.

Cuando se escribe la URL de un sitio web en el navegador, la PC realiza una consulta de DNS a la dirección IP del servidor DNS. La observación de las consultas y respuestas del DNS proporciona los nombres (URL) y las direcciones IP de los sitios que visita un usuario. Conocer los sitios web que los usuarios visitan comúnmente puede ser valioso al formular ataques de ingeniería social.

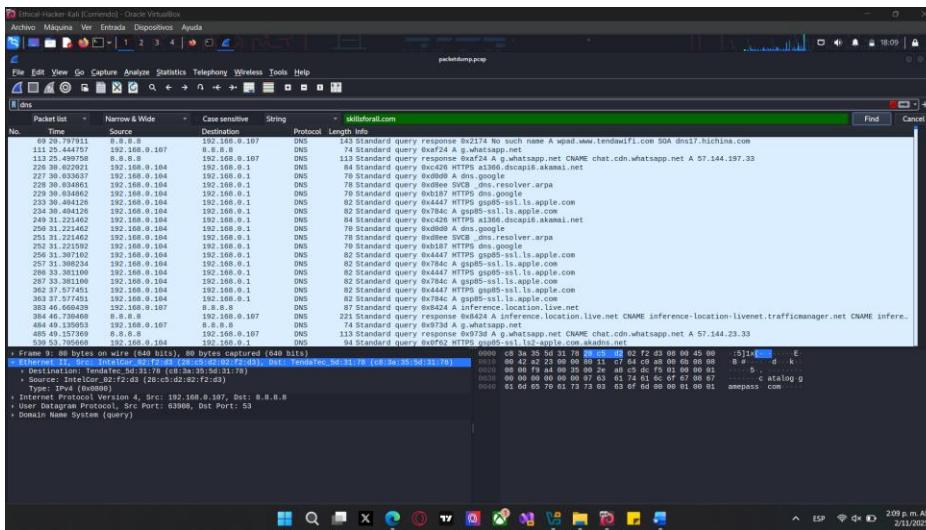
- a. Filtre el tráfico capturado para mostrar solo las consultas y respuestas de DNS. Introduzca **dns** en el campo Filtro de la pantalla principal de Wireshark. Notará que, además del sitio web **Skills for All** que solicitó, se muestran otras búsquedas de DNS. Corresponden a enlaces contenidos en las páginas de inicio de Skills For All y Google.
- b. Haga clic en el icono de búsqueda de la lupa o seleccione **Find Packet** en el menú **Edit**. Busque el nombre de host de **skillsforall.com**. Seleccione **String** en el cuadro desplegable Mostrar filtro e ingrese **skillsforall** en el cuadro de búsqueda. Haga clic en **Find** (Buscar).

Práctica de laboratorio: Detección de redes con Wireshark



- c. Seleccione la primera consulta estándar para el sitio web **skillsforall.com**. Expanda el panel de detalles de la consulta debajo de la lista de paquetes para ver el contenido del paquete de consulta.
- d. Expanda la información de **Ethernet II** para mostrar los datos del encabezado de capa 2 contenidos en el paquete. La dirección MAC de origen es la MAC de la interfaz del dispositivo de envío, en este caso la VM Kali, y la dirección MAC de destino es la MAC de la puerta de enlace predeterminada porque el servidor DNS no está en la misma red de capa 2.

¿La dirección MAC de origen coincide con la dirección que registró en la parte 1?



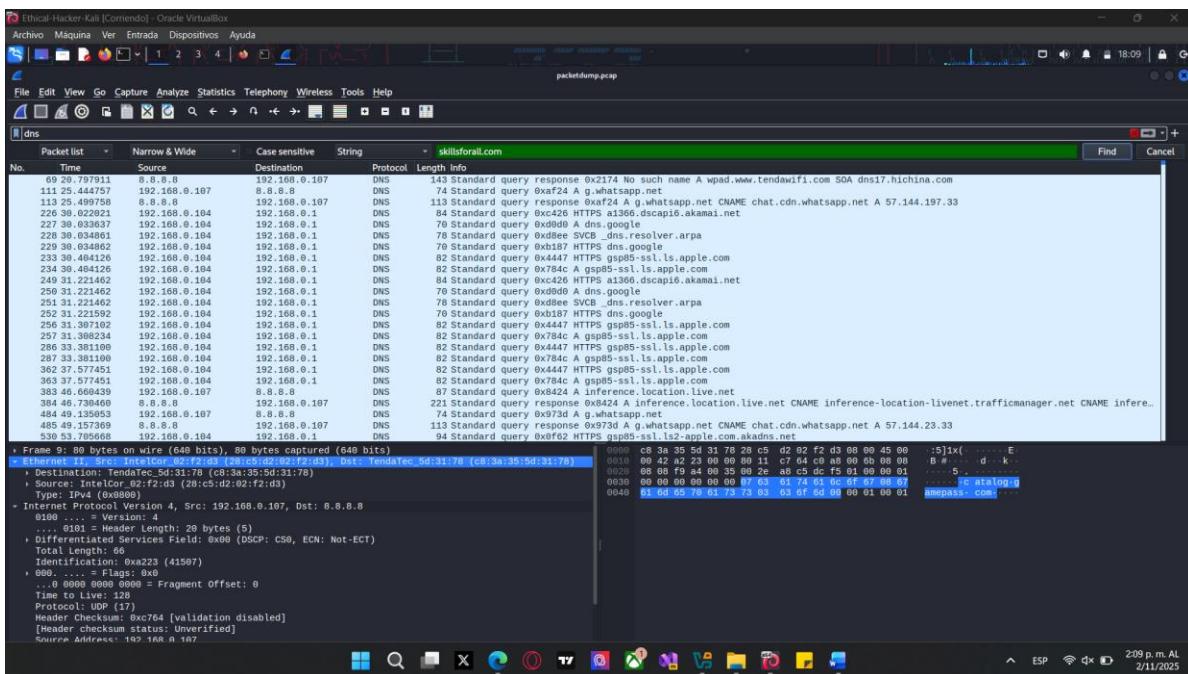
Sí.

MOSTRAR CAPTURA DE PANTALLA

- e. Expanda la sección **Domain Name System (query)** para ver los detalles de lo que se envía al servidor DNS. También indica la línea que contiene el paquete de respuesta que se recibió en respuesta a la consulta. Haga doble clic en el enlace a la respuesta. Se muestran los detalles del paquete de respuesta a la consulta estándar.

¿Qué direcciones IP están asociadas con la URL skillsforall.com?

Práctica de laboratorio: Detección de redes con Wireshark



Las respuestas pueden variar, por ejemplo, 13.33.21.125, 13.33.21.5, 13.33.21.122, 13.33.21.30

- f. Cierre Wireshark para volver al indicador de la CLI.

Paso 3: Analizar una sesión HTTP

En este paso, capturará y analizará una solicitud y una respuesta web. Utilizará Wireshark para capturar el tráfico y analizar los mensajes intercambiados entre el servidor web y el cliente. El servidor del sitio web es un servidor de VM que se ejecuta en un contenedor de Docker en la VM Kali Linux.

- a. Utilice **ifconfig** para determinar qué interfaz de la VM Kali Linux está configurada en la red 10.6.6.0/24.

```
└─(kali㉿Kali)-[~]
└─$ ifconfig
```

¿Cuál es el nombre de la interfaz conectada a la red 10.6.6.0/24?

```
└─(kali㉿Kali)-[~]
└─$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0C:29:1A:0D:01
          inet 10.6.6.1 brd 10.6.6.255 netmask 255.255.255.0 broadcast 10.6.6.255
              ether 00:0c:29:1a:0d:01 brd ff:ff:ff:ff:ff:ff
          BROADCAST,MULTICAST,UP,LOWER_UP
          RX packets 2636 bytes 3126 (3.8 KB)
          RX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          TX packets 2636 bytes 3126 (3.8 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

br-355ee7a5e45885  Link encap:Ethernet HWaddr 02:00:00:00:00:00
          inet 10.6.6.1 brd 10.6.6.255 netmask 255.255.255.0 broadcast 10.6.6.255
              ether 02:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
          BROADCAST,MULTICAST,UP,BROADCAST,NOARP
          RX packets 2636 bytes 3126 (3.8 KB)
          RX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          TX packets 2636 bytes 3126 (3.8 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1      Link encap:Ethernet HWaddr 00:0C:29:1A:0D:02
          inet 10.6.6.2 brd 10.6.6.255 netmask 255.255.255.0 broadcast 10.6.6.255
              ether 00:0c:29:1a:0d:02 brd ff:ff:ff:ff:ff:ff
          BROADCAST,MULTICAST,UP,LOWER_UP
          RX packets 2636 bytes 3126 (3.8 KB)
          RX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          TX packets 2636 bytes 3126 (3.8 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth2      Link encap:Ethernet HWaddr 00:0C:29:1A:0D:03
          inet 10.6.6.3 brd 10.6.6.255 netmask 255.255.255.0 broadcast 10.6.6.255
              ether 00:0c:29:1a:0d:03 brd ff:ff:ff:ff:ff:ff
          BROADCAST,MULTICAST,UP,LOWER_UP
          RX packets 2636 bytes 3126 (3.8 KB)
          RX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          TX packets 2636 bytes 3126 (3.8 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth3      Link encap:Ethernet HWaddr 00:0C:29:1A:0D:04
          inet 10.6.6.4 brd 10.6.6.255 netmask 255.255.255.0 broadcast 10.6.6.255
              ether 00:0c:29:1a:0d:04 brd ff:ff:ff:ff:ff:ff
          BROADCAST,MULTICAST,UP,LOWER_UP
          RX packets 2636 bytes 3126 (3.8 KB)
          RX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          TX packets 2636 bytes 3126 (3.8 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo        Link encap:Local Loopback
          inet 127.0.0.1 brd 127.0.0.1 netmask 255.255.255.0 broadcast 127.0.0.1
              loop brd 127.0.0.1 mtu 1500
              ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
          BROADCAST,NOARP,LOOPBACK,UP,LOWER_UP
          RX packets 0 bytes 0 (0.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 0 bytes 0 (0.0 B)
          TX errors 0 dropped 0 overruns 0 frame 0
```

Las respuestas pueden variar según la configuración. En la VM de Kali, el nombre de la interfaz se configura como br-internal.

¿Cuál es la dirección IP asignada a esa interfaz?

```

root@kali:~# ifconfig
br-0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.5.2 brd 255.255.255.255 broadcast 192.168.5.255
        link layer brd 00:0c:29:1e:00:02 brd 00:0c:29:1e:00:02
        ether 00:0c:29:1e:00:02 txqueuelen 0 (Ethernet)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

br-355ee94a5000: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.1 brd 255.255.255.255 broadcast 192.168.0.255
        link layer brd 00:0c:29:1e:00:01 brd 00:0c:29:1e:00:01
        ether 00:0c:29:1e:00:01 txqueuelen 0 (Ethernet)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

br-internal: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    link layer brd 00:0c:29:1e:00:03 brd 00:0c:29:1e:00:03
        ether 00:0c:29:1e:00:03 txqueuelen 0 (Ethernet)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 brd 255.255.255.255 broadcast 172.17.0.255
        link layer brd ff:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
        ether ff:ff:ff:ff:ff:ff txqueuelen 0 (Ethernet)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    link layer brd 00:0c:29:1e:00:04 brd 00:0c:29:1e:00:04
        ether 00:0c:29:1e:00:04 txqueuelen 1000 (Ethernet)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    link layer brd 00:00:00:00:00:00
    inet 127.0.0.1 brd 0.0.0.0 scope 127.0.0.1
        link layer brd 00:00:00:00:00:00
        ether 00:00:00:00:00:00 txqueuelen 0 (Ethernet)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

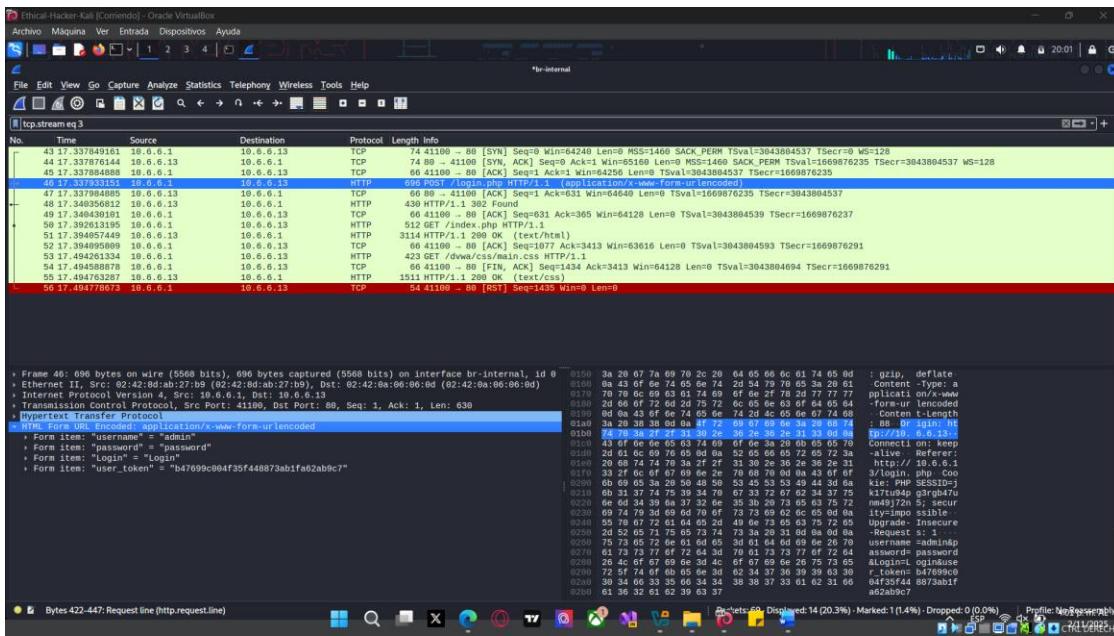
```

10.6.6.1

- Abra Wireshark escribiendo **wireshark** en la línea de comandos. Wireshark se abrirá en una nueva ventana, expandirá la ventana a pantalla completa. En el centro de la pantalla principal de Wireshark habrá una lista de nombres de interfaz para capturar el tráfico. ¿Cuál es el nombre de la interfaz conectada a la red 10.6.6.0/24? Esto iniciará la captura de paquetes.
 - Abra una ventana del navegador y navegue hasta la dirección IP 10.6.6.13. Aparece una pantalla de inicio de sesión para el servidor web DVWA. Escriba **admin** como nombre de **usuario** y la contraseña.
- Username: **admin**
Contraseña: **password**
- Cuando aparezca la página principal de DVWA, haga clic en el botón **Instructions** en la parte superior del menú en el lado izquierdo de la pantalla. Cuando aparezca la página de instrucciones, cierre la ventana del navegador.
 - Regrese a la ventana de Wireshark. Detenga la captura con el **ícono cuadrado rojo** en la barra de menús. El servidor web DVWA utiliza HTTP, no HTTPS. Utilice el **ícono de búsqueda** para encontrar la cadena **POST** en los paquetes capturados. Los mensajes POST transfieren datos del formulario del cliente al servidor, en este caso la información de inicio de sesión.
 - Haga doble clic en el primer paquete POST para ver los detalles del paquete en una ventana separada. Expanda la sección titulada **HTML Form URL Encoded**:

¿Qué información contiene esta sección?

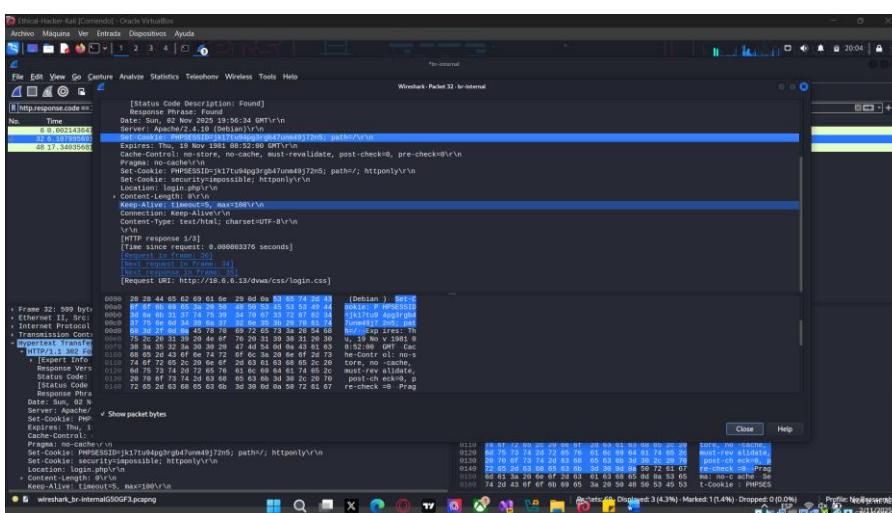
Práctica de laboratorio: Detección de redes con Wireshark



Esta sección contiene el nombre de usuario y la contraseña ingresados por el usuario. También se muestra un user_token.

- g. Las cookies se utilizan para diversos fines. Con mayor frecuencia, se utilizan para guardar información sobre la sesión de un usuario. Las cookies pueden ser secuestradas y utilizadas en ataques de secuestro de sesiones. La cookie inicial para una sesión se envía desde el servidor web al cliente con el valor **Set-Cookie** en una respuesta HTTP. Utilice **Ctrl-Inicio** para volver a la primera línea de la captura de paquetes. Utilice el ícono de búsqueda para encontrar la cadena **302 Found** en el panel de paquetes. Haga doble clic en el primer paquete que se encontró y expanda la sección **Hypertext Transport Protocol**.

¿Qué valor se establece en la cookie que se envía desde el servidor web al cliente Kali?



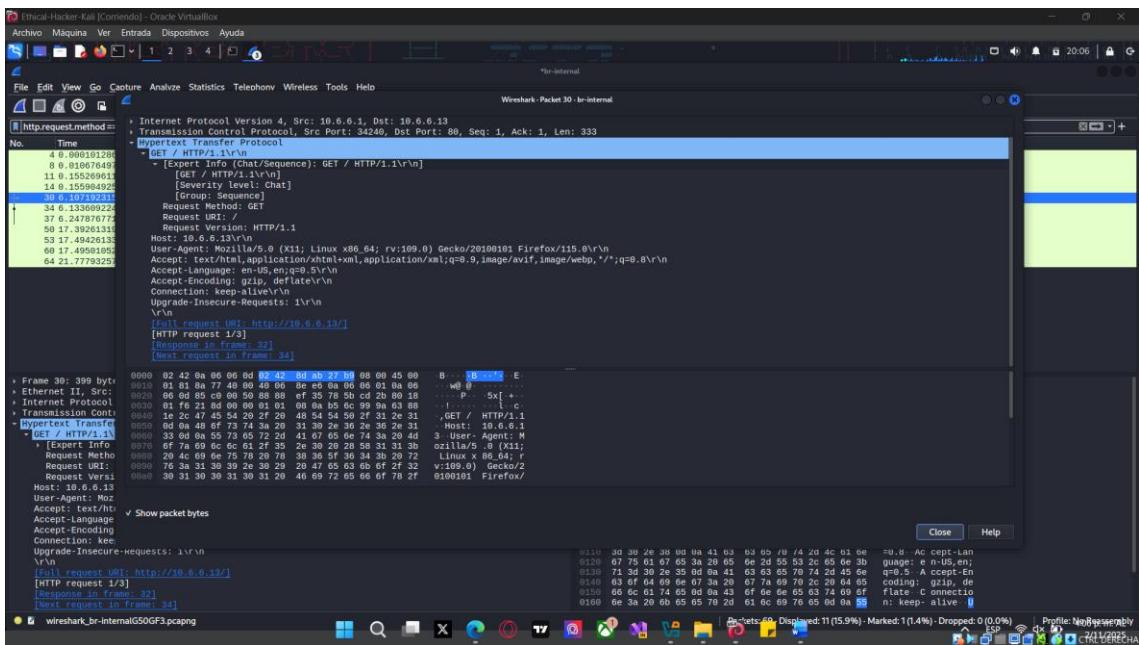
PHPSESSID

- h. Examine el siguiente paquete GET que se envía desde el navegador del cliente Kali después de recibir la información de la cookie. Expanda la sección Hypertext Transport Protocol. Busque los valores de cookie que se envían en el paquete.

¿El PHPSESSID que se envía de vuelta al servidor en la solicitud GET es el mismo que el enviado desde el servidor en la respuesta anterior?

Práctica de laboratorio: Detección de redes con Wireshark

Sí.



- i. Cierre Wireshark. Tendrá la opción de guardar el archivo .pcap que contiene la captura o salir sin guardar. El archivo .pcap se guardará en el directorio de trabajo actual a menos que se especifique lo contrario.

