AEC3 – Redes de Ordenadores

ISMAEL HERNÁNDEZ CLEMENTE

Descripción

Objetivo:
Instalar un servidor WEB y un servidor DNS en la misma máquina servidora, realizar una navegación WEB desde una máquina cliente mediante Firefox y analizar, con Wireshark, las cabeceras TCP, UDP e IP de las consultas DNS y del tráfico WEB.

Requisitos previos:

- Dos máquinas virtuales con **Ubuntu** correctamente configuradas.
- Configuración de red de las máquinas virtuales como **adaptador puente**.

Pasos para la ejecución de la práctica:

1. **Asignar roles a las máquinas virtuales:**
    - Elegir una máquina como cliente (**MV_CLIENTE**) y otra como servidora (**MV_SERVIDOR**).
2. **Identificar las IPs:**
    - Obtener las direcciones IP de **MV_CLIENTE** y **MV_SERVIDOR**.
3. **Instalar paquetes en MV_SERVIDOR:**
    - **Apache2:** Usar la página web de prueba predeterminada.
    - **Bind9:** Configurar el servidor DNS con las siguientes características:
        - Dominio: internet.wrl.
        - Entrada en la base de datos:
            - Subdominio www.internet.wrl debe apuntar a la IP de **MV_SERVIDOR**.
        - Configuración de TTL en 5 segundos para forzar nuevas consultas DNS.
        - Usar como base los archivos proporcionados:
            - named.conf.options,
            - named.conf.local,
            - db.u-tad.uni.
            - Modificar adecuadamente para reflejar el dominio internet.wrl.
    - Verificar la sintaxis de los archivos y recargar la configuración del servidor DNS.
4. **Captura 1:**
    - Tomar un pantallazo del archivo db.internet.wrl.
5. **Configurar DNS en MV_CLIENTE:**
    - Establecer como DNS principal la IP de **MV_SERVIDOR**.

6. **Iniciar Wireshark y Firefox:**
    - En **MV_CLIENTE**, iniciar Wireshark y navegar a http://www.internet.wrl en Firefox.
    - **Captura 2:** Tomar un pantallazo mostrando la URL ingresada en el navegador.
7. **Análisis del tráfico DNS con Wireshark:**
    - Filtrar el tráfico DNS y recargar varias veces la página para observar el tráfico.
    - Analizar el paquete de consulta DNS y registrar:
        - **Nivel UDP:**
            - Puerto Origen.
            - Puerto Destino.
            - Protocolo de transporte.
        - **Nivel de aplicación:**
            - Query DNS.
    - **Captura 3:** Incluir los pantallazos correspondientes.
8. **Análisis del tráfico HTTP con Wireshark:**
    - Filtrar el tráfico HTTP y analizar el paquete de conexión inicial (**SYN**).
    - Registrar:
        - **Nivel TCP:**
            - Puerto Origen.
            - Puerto Destino.
            - SeqN, AckN, Checksum.
            - Flags activas.
        - **Nivel IP:**
            - IP Origen.
            - IP Destino.
            - TTL.
    - **Captura 4:** Incluir los pantallazos correspondientes.

Realización de la Actividad Practica

1. Preparación inicial

- Inicialmente igual que en la AEC2 definí qué máquina virtual será cliente (MV_CLIENTE) y cuál será servidor (MV_SERVIDOR). Asegúrando de que ambas máquinas están configuradas como Adaptador puente en las opciones de red para que se conecten a la red local y verifiqué las IPs de ambas máquinas utilizando el comando:

ip addr

2. Configuración de MV_SERVIDOR

Instalación de Apache2

Instale Apache2 en MV_SERVIDOR:

sudo apt update

sudo apt install apache2 -y

Verifique que Apache esté funcionando accediendo a la página de prueba:

curl http://localhost

Instalación de Bind9

1. Instale Bind9:

   sudo apt install bind9 -y

2. Configure los siguientes archivos para Bind9:

   Archivo: named.conf.options

   sudo nano /etc/bind/named.conf.options

Actualice o confirme que tenga las siguientes configuraciones:

```
options {
    directory "/var/cache/bind";
    listen-on { any; };
    listen-on-v6 { none; };
    forwarders {
        8.8.8.8; // DNS de Google
    };
    dnssec-validation auto;
    auth-nxdomain no;
    allow-query { any; };
```

};

Archivo: named.conf.local

sudo nano /etc/bind/named.conf.local

Incluí la definición del dominio:

```
zone "internet.wrl" {
   type master;
   file "/etc/bind/db.internet.wrl";
};
```

Archivo: db.internet.wrl

Cree el archivo y lo configuré con:

sudo nano /etc/bind/db.internet.wrl

Contenido del archivo (reemplacé 192.168.x.x con la IP de MV_SERVIDOR):

```
$TTL    5
@       IN      SOA     ns.internet.wrl. admin.internet.wrl. (
                  1 ; Serial
                  604800 ; Refresh
                  86400 ; Retry
                  2419200 ; Expire
                  5 ) ; TTL
@       IN      NS      ns.internet.wrl.
ns      IN      A       192.168.1.138
www     IN      A       192.168.1.138
```

3. Verificación de configuración

- Comprobé la sintaxis de los archivos mediante:

sudo named-checkconf

sudo named-checkzone internet.wrl /etc/bind/db.internet.wrl

- Reinicié Bind9:

sudo systemctl restart bind9

4. Configuración de MV_CLIENTE

- Configuré MV_CLIENTE para usar el DNS de MV_SERVIDOR:

sudo nano /etc/resolv.conf

Añadí la línea:

nameserver 192.168.x.x

(Sustituyendo 192.168.x.x por la IP de MV_SERVIDOR).

5. Análisis de tráfico DNS

1. Inicié **Wireshark** en MV_CLIENTE y empecé a capturar tráfico en la interfaz activa y en Firefox, accedí a:

http://www.internet.wrl

Filtros en Wireshark para capturar tráfico DNS:

udp.port == 53

3. Identifiqué y analicé los paquetes DNS:
    - **UDP**: Puerto origen, puerto destino, protocolo.
    - **Aplicación**: Query DNS.

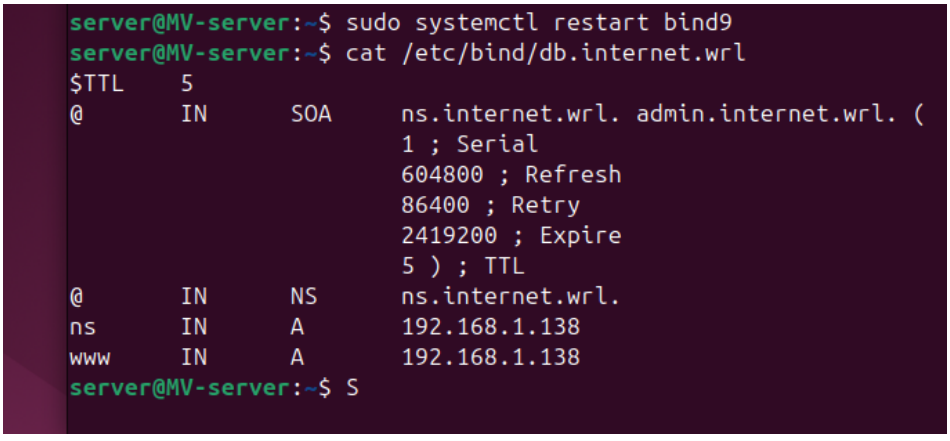
6. Análisis de tráfico HTTP

1. En Wireshark, apliqué el filtro para tráfico HTTP:

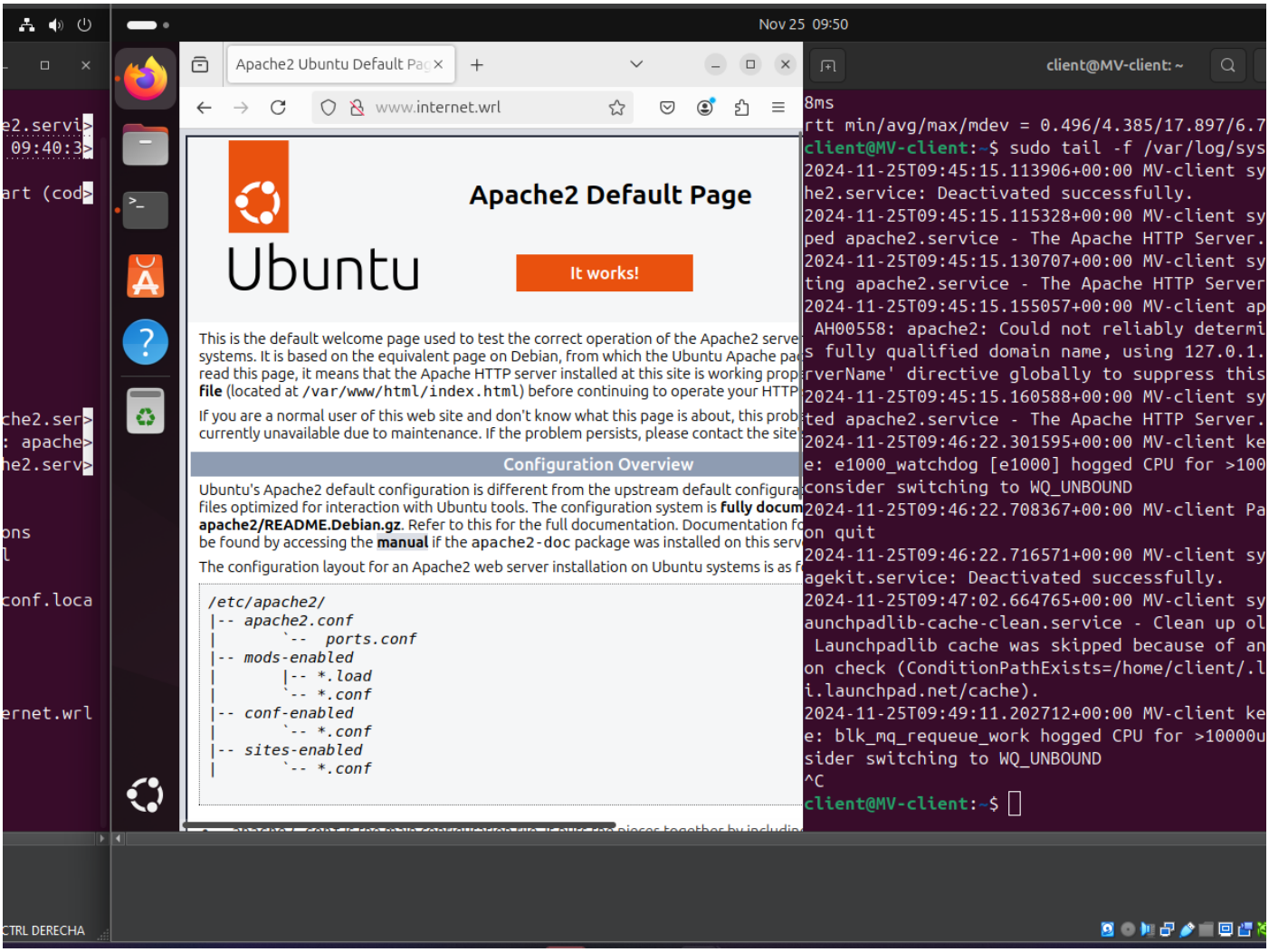
tcp.port == 80

2. Recargué la página en Firefox.
3. Por último tuve que analizar el paquete SYN inicial:
    - **TCP**: Puerto origen, puerto destino, SeqN, AckN, Checksum, flags activas.
    - **IP**: IP origen, IP destino, TTL.

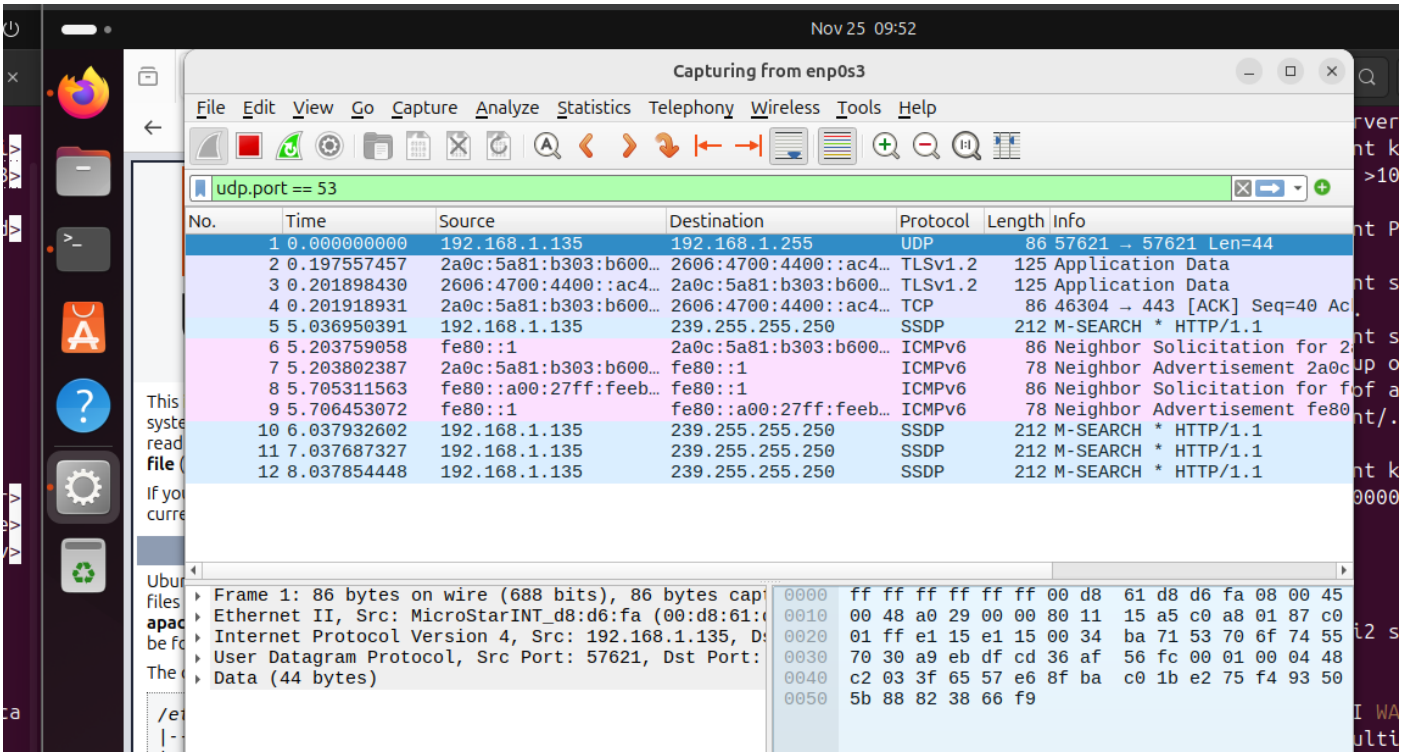
CAPTURA_1



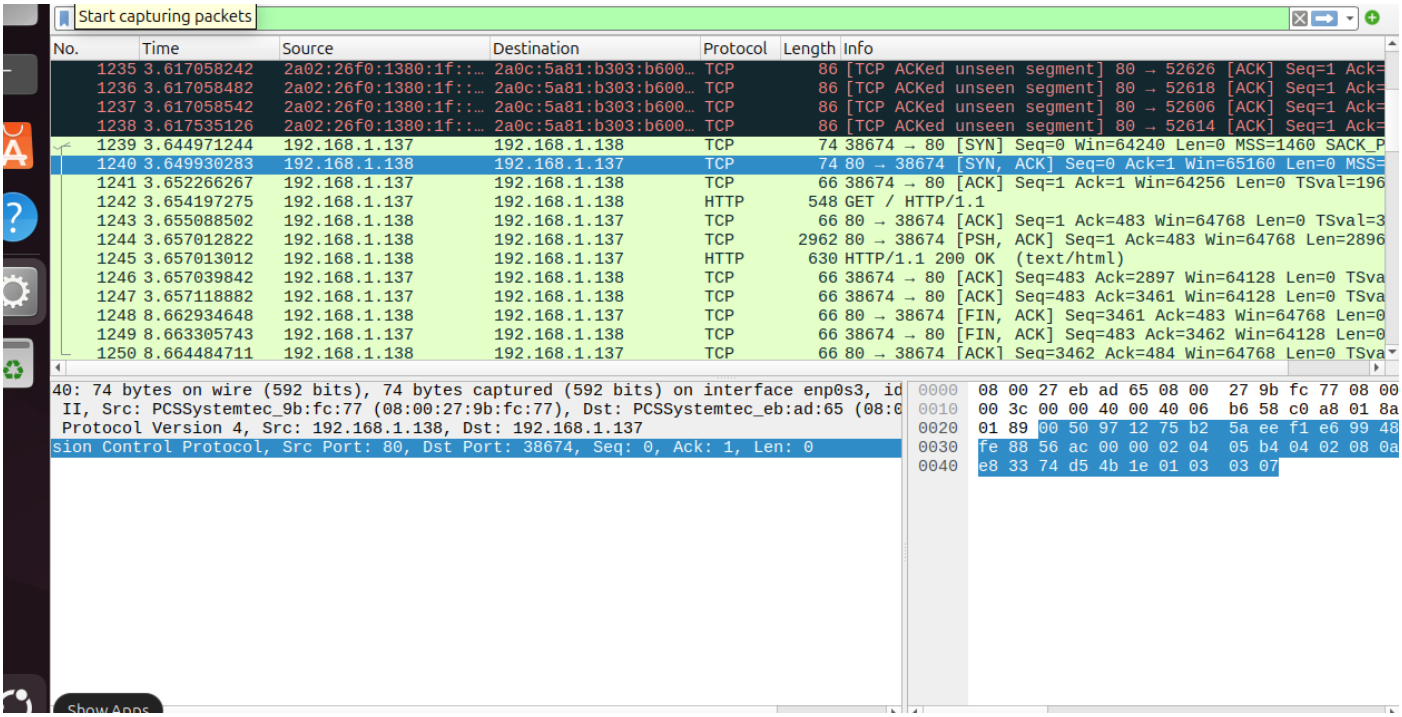
CAPTURA_2



CAPTURA_3



CAPTURA_4


Start capturing packets
No. Time Source Destination Protocol Length Info
1235 3.617058242 2a02:26f0:1380:1f::… 2a0c:5a81:b303:b600… TCP 86 [TCP ACKed unseen segment] 80 → 52626 [ACK] Seq=1 Ack=
1236 3.617058482 2a02:26f0:1380:1f::… 2a0c:5a81:b303:b600… TCP 86 [TCP ACKed unseen segment] 80 → 52618 [ACK] Seq=1 Ack=
1237 3.617058542 2a02:26f0:1380:1f::… 2a0c:5a81:b303:b600… TCP 86 [TCP ACKed unseen segment] 80 → 52606 [ACK] Seq=1 Ack=
1238 3.617535126 2a02:26f0:1380:1f::… 2a0c:5a81:b303:b600… TCP 86 [TCP ACKed unseen segment] 80 → 52614 [ACK] Seq=1 Ack=
1239 3.644971244 192.168.1.137 192.168.1.138 TCP 74 38674 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_P
1240 3.649930283 192.168.1.138 192.168.1.137 TCP 74 80 → 38674 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=
1241 3.652266267 192.168.1.137 192.168.1.138 TCP 66 38674 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=196
1242 3.654197275 192.168.1.137 192.168.1.138 HTTP 548 GET / HTTP/1.1
1243 3.655088502 192.168.1.138 192.168.1.137 TCP 66 80 → 38674 [ACK] Seq=1 Ack=483 Win=64768 Len=0 TSval=3
1244 3.657012822 192.168.1.138 192.168.1.137 TCP 2962 80 → 38674 [PSH, ACK] Seq=1 Ack=483 Win=64768 Len=2896
1245 3.657013012 192.168.1.138 192.168.1.137 HTTP 630 HTTP/1.1 200 OK (text/html)
1246 3.657039842 192.168.1.137 192.168.1.138 TCP 66 38674 → 80 [ACK] Seq=483 Ack=2897 Win=64128 Len=0 TSva
1247 3.657118882 192.168.1.137 192.168.1.138 TCP 66 38674 → 80 [ACK] Seq=483 Ack=3461 Win=64128 Len=0 TSva
1248 8.662934648 192.168.1.138 192.168.1.137 TCP 66 80 → 38674 [FIN, ACK] Seq=3461 Ack=483 Win=64768 Len=0
1249 8.663305743 192.168.1.137 192.168.1.138 TCP 66 38674 → 80 [FIN, ACK] Seq=483 Ack=3462 Win=64128 Len=0
1250 8.664484711 192.168.1.138 192.168.1.137 TCP 66 80 → 38674 [ACK] Seq=3462 Ack=484 Win=64768 Len=0 TSva
40: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp0s3, id
II, Src: PCSSystemtec_9b:fc:77 (08:00:27:9b:fc:77), Dst: PCSSystemtec_eb:ad:65 (08:0
Protocol Version 4, Src: 192.168.1.138, Dst: 192.168.1.137
sion Control Protocol, Src Port: 80, Dst Port: 38674, Seq: 0, Ack: 1, Len: 0
0000 08 00 27 eb ad 65 08 00 27 9b fc 77 08 00
0010 00 3c 00 00 40 00 40 06 b6 58 c0 a8 01 8a
0020 01 89 00 50 97 12 75 b2 5a ee f1 e6 99 48
0030 fe 88 56 ac 00 00 02 04 05 b4 04 02 08 0a
0040 e8 33 74 d5 4b 1e 01 03 03 07
Show Apps