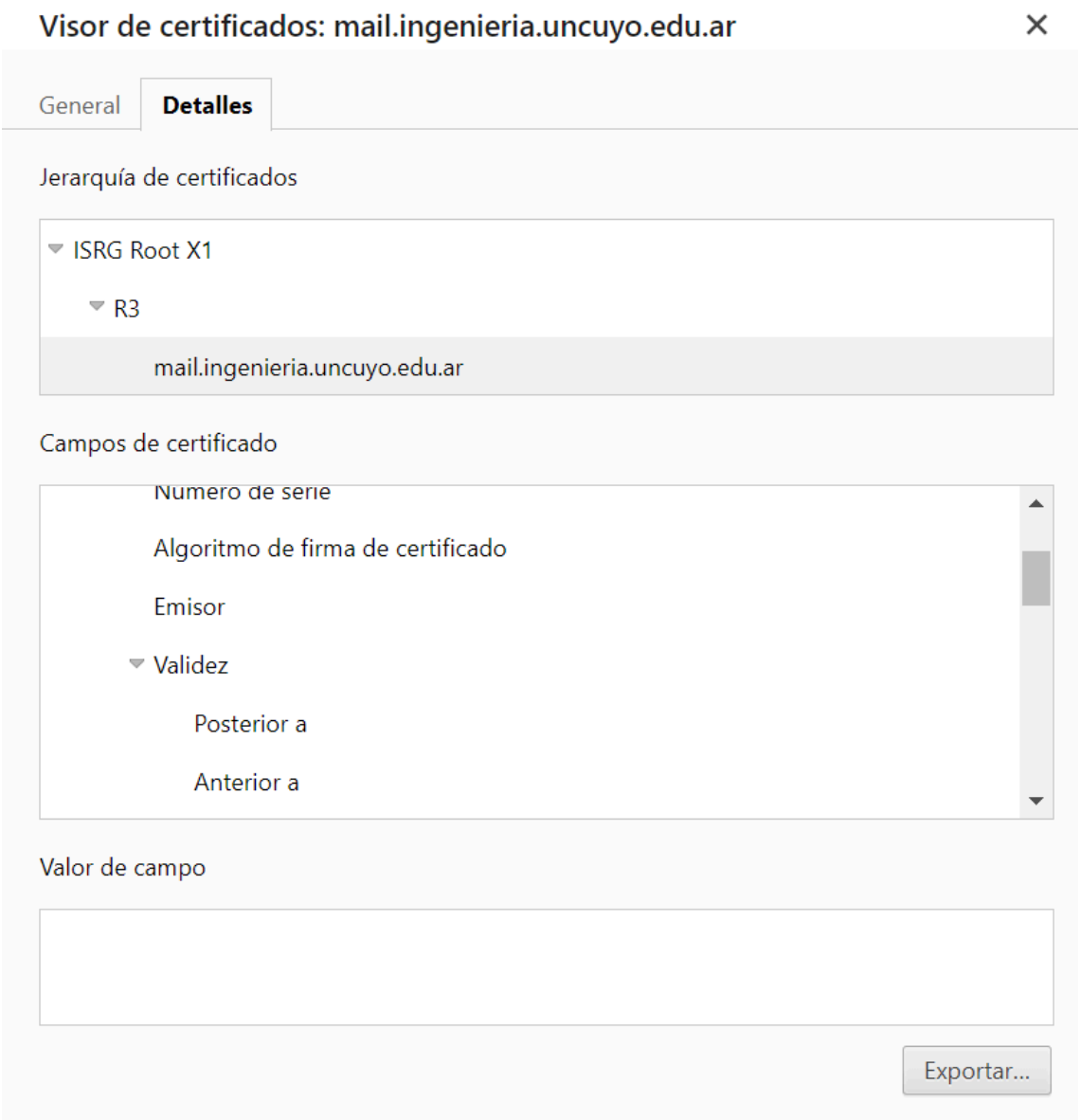
Informe trabajo práctico Nº 7

Integrantes: Perea Trinidad, Valerio Perla, Suden Paulina

Actividad 1:

En esta actividad, se lleva a cabo un análisis de los certificados y la encriptación de varias páginas web. Esto implica revisar los certificados de seguridad para asegurar que las conexiones son seguras y están correctamente encriptadas. Se verifica el algoritmo de firma del certificado, la autoridad de certificación, el algoritmo de encriptación de clave pública, entre otros. La información se obtiene del visor de certificados que vemos en la imagen.



Actividad 2:

En esta actividad utilizamos la herramienta webhttrack para clonar la página del banco patagonia.

Una vez clonada, realizamos lo siguiente:

1- Modificamos el link que hacía referencia hacia la página de login, y pusimos que vaya a la página login clonada

2- Luego modificamos el código de manera que cuando el usuario ingrese su nombre de usuario y contraseña y presione ingresar se invoquen a los procedimientos que realizamos en php, los cuales fueron:

Almacenar en un archivo llamado registros.txt los usuarios y contraseñas ingresados, y al tocar ingresar sea direccionado a la página login real del banco patagonia, de modo que parezca que ha habido algún fallo.

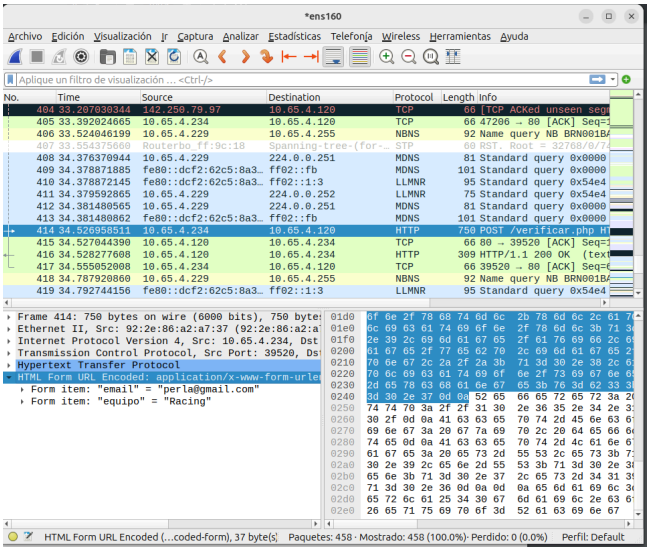
Actividad 3:

En esta actividad analizamos si la información que maneja la pagina web creada para el TP5 estaba encriptada para prevenir robos de datos. Para ello utilizamos wireshark para capturar el tráfico de red:

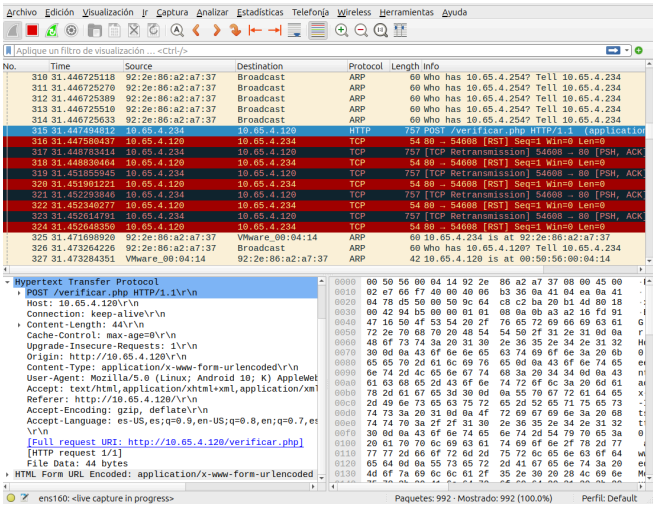
- Accedimos a la página web desde otro dispositivo, ingresamos los datos que solicitaba en el formulario y los enviamos.
- filtramos los paquetes HTTP por la IP del dispositivo cliente y buscamos las peticiones POST para verificar se podía ver la información enviada
- Como se podían ver los datos, la página no estaba encriptando la información por lo que no era segura

Para mejorar la seguridad agregamos un certificado SSL autofirmado, para ello instalamos OpenSSL para Apache2 y habilitamos los módulos SSL. Luego creamos un certificado y una clave privada e indicamos a Apache la ubicación del certificado en el archivo de configuración correspondiente. Finalmente, reiniciamos Apache y accedimos a la página web usando https en lugar de http y verificamos nuevamente con Wireshark para comprobar si la información estaba encriptada.

Antes:



Después:



Actividad 4:

Sistema operativo usado: Linux Kali

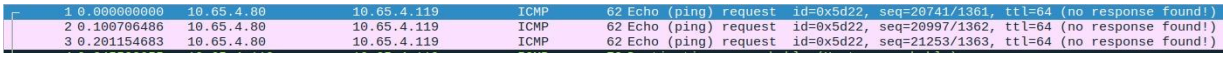
4.1 ARP Spoofing con Nping

- Elegimos dos computadoras, una de ellas para ejecutar los comandos y la otra sería la “víctima”, de esta última anotamos la dirección MAC asignada al gateway, su dirección IP y la IP del router
- Con los datos anteriores ejecutamos el comando: sudo nping --arp --count 100000 -arp-type ARP-reply --rate 1000 --arp-sender-mac --arp-sender-ip <IP del router> <IP victima>
- Luego, intentamos acceder a internet desde la IP atacada
- Observamos que la dirección MAC que aparecía en la tabla ARP de la computadora había cambiado

```
estudiante@ubuntu:~$ arp -n
Dirección                TipoHW  DirecciónHW         Indic Máscara         Interfaz
10.65.4.120              ether   00:50:56:00:04:14   C                     ens160
10.65.4.211              ether   a2:b2:b9:46:27:89   C                     ens160
10.65.4.228              ether   38:d5:7a:48:a9:4b   C                     ens160
10.65.4.254              ether   74:4d:28:c1:57:70   C                     ens160
estudiante@ubuntu:~$ arp -n
Dirección                TipoHW  DirecciónHW         Indic Máscara         Interfaz
10.65.4.120              ether   00:50:56:00:04:14   C                     ens160
10.65.4.211              ether   a2:b2:b9:46:27:89   C                     ens160
10.65.4.228              ether   38:d5:7a:48:a9:4b   C                     ens160
10.65.4.254              ether   74:4d:28:c1:57:77   C                     ens160
```

4.2 DoS con hping3

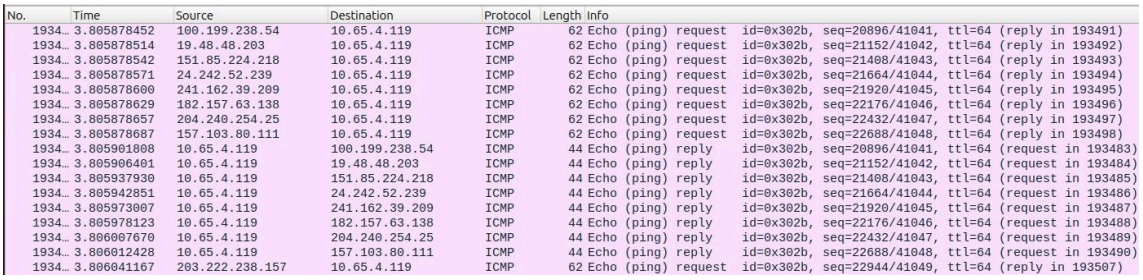
- Utilizamos dos computadoras y en una de ellas realizamos una suplantación de IP -> IP a suplantar: 10.65.4.80; IP destino: 10.65.4.119
- Utilizamos wireshark para analizar los paquetes



1	0.000000000	10.65.4.80	10.65.4.119	ICMP	62	Echo (ping) request id=0x5d22, seq=20741/1361, ttl=64 (no response found!)
2	0.100706486	10.65.4.80	10.65.4.119	ICMP	62	Echo (ping) request id=0x5d22, seq=20997/1362, ttl=64 (no response found!)
3	0.201154683	10.65.4.80	10.65.4.119	ICMP	62	Echo (ping) request id=0x5d22, seq=21253/1363, ttl=64 (no response found!)

<u>Ataque DoS por inundación con hping3:</u>

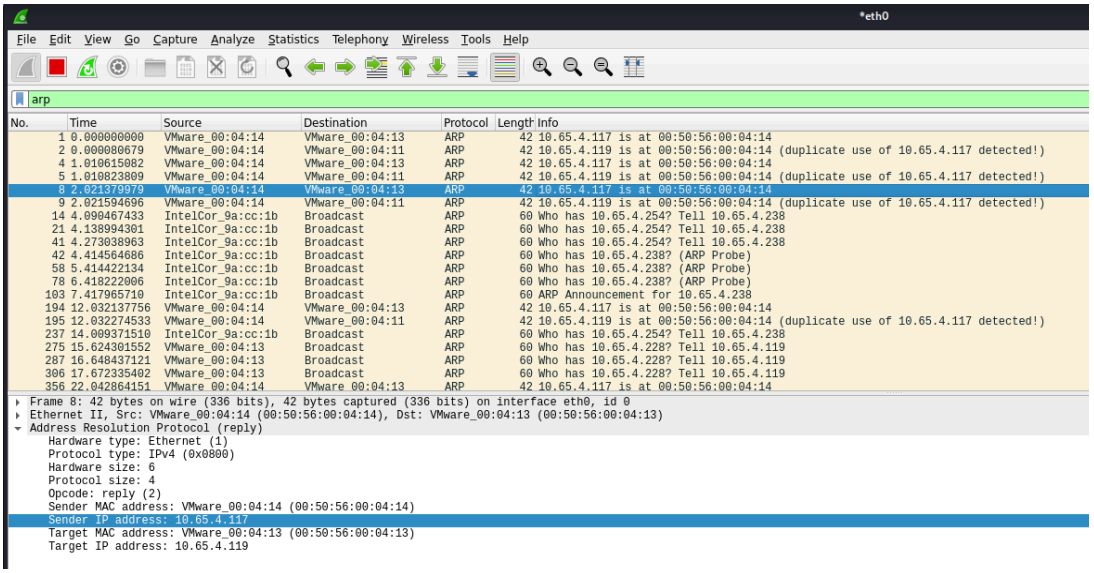
- No se puede navegar adecuadamente desde la máquina atacada
- Al verificar con wireshark en la máquina de la víctima los paquetes que recibe obtenemos:



No.	Time	Source	Destination	Protocol	Length	Info
1934…	3.805878452	100.199.238.54	10.65.4.119	ICMP	62	Echo (ping) request id=0x302b, seq=20896/41041, ttl=64 (reply in 193491)
1934…	3.805878514	19.48.48.203	10.65.4.119	ICMP	62	Echo (ping) request id=0x302b, seq=21152/41042, ttl=64 (reply in 193492)
1934…	3.805878542	151.85.224.218	10.65.4.119	ICMP	62	Echo (ping) request id=0x302b, seq=21408/41043, ttl=64 (reply in 193493)
1934…	3.805878571	24.242.52.239	10.65.4.119	ICMP	62	Echo (ping) request id=0x302b, seq=21664/41044, ttl=64 (reply in 193494)
1934…	3.805878600	241.162.39.209	10.65.4.119	ICMP	62	Echo (ping) request id=0x302b, seq=21920/41045, ttl=64 (reply in 193495)
1934…	3.805878629	182.157.63.138	10.65.4.119	ICMP	62	Echo (ping) request id=0x302b, seq=22176/41046, ttl=64 (reply in 193496)
1934…	3.805878657	204.240.254.25	10.65.4.119	ICMP	62	Echo (ping) request id=0x302b, seq=22432/41047, ttl=64 (reply in 193497)
1934…	3.805878687	157.103.80.111	10.65.4.119	ICMP	62	Echo (ping) request id=0x302b, seq=22688/41048, ttl=64 (reply in 193498)
1934…	3.805901808	10.65.4.119	100.199.238.54	ICMP	44	Echo (ping) reply id=0x302b, seq=20896/41041, ttl=64 (request in 193483)
1934…	3.805906401	10.65.4.119	19.48.48.203	ICMP	44	Echo (ping) reply id=0x302b, seq=21152/41042, ttl=64 (request in 193484)
1934…	3.805937930	10.65.4.119	151.85.224.218	ICMP	44	Echo (ping) reply id=0x302b, seq=21408/41043, ttl=64 (request in 193485)
1934…	3.805942851	10.65.4.119	24.242.52.239	ICMP	44	Echo (ping) reply id=0x302b, seq=21664/41044, ttl=64 (request in 193486)
1934…	3.805973007	10.65.4.119	241.162.39.209	ICMP	44	Echo (ping) reply id=0x302b, seq=21920/41045, ttl=64 (request in 193487)
1934…	3.805978123	10.65.4.119	182.157.63.138	ICMP	44	Echo (ping) reply id=0x302b, seq=22176/41046, ttl=64 (request in 193488)
1934…	3.806007670	10.65.4.119	204.240.254.25	ICMP	44	Echo (ping) reply id=0x302b, seq=22432/41047, ttl=64 (request in 193489)
1934…	3.806012428	10.65.4.119	157.103.80.111	ICMP	44	Echo (ping) reply id=0x302b, seq=22688/41048, ttl=64 (request in 193490)
1934…	3.806041167	203.222.238.157	10.65.4.119	ICMP	62	Echo (ping) request id=0x302b, seq=22944/41049, ttl=64 (reply in 193507)

4.3 MITM

- Realizamos un ataque MITM envenenando las tablas ARP de dos computadoras con la aplicación ettercap
- Ejecutamos el comando: ettercap -T --mitm arp /10.65.4.117// /10.65.4.119// y visualizamos con Wireshark si se puede ver la información intercambiada por las máquinas atacadas



Actividad 5:

Para esta actividad usamos Gufw (para configurar las reglas de ufw a través de una interfaz gráfica) y ufw (uncomplicated firewall) para configurar por linea de comandos el Firewall incluido en el núcleo de Linux

- En Gfuw realizamos las siguientes acciones:
    1. Intentamos acceder a la página web con la configuración inicial:
        a. Estado: Habilitado
        b. Entrante: denegar
        c. Saliente: permitir
        - Resultado: no se pudo acceder a la página
    2. Agregamos la regla avanzada que se indica en el TP y permitimos que una IP específica pudiera acceder
        - Resultado: la máquina pudo acceder a la página
    3. Agregamos una regla para impedir conectarse a la ip de Facebook
        - Resultado: la máquina pudo acceder a la página porque Facebook tiene varias ips