**Bloque 1: Protección de redes de un host**

**Administración de zonas con firewalld**

Asegurarnos de deshabilitar ufw: sudo ufw disable y reiniciar la MV

Acceder al firewall: sudo firewall-config

Cambiar combobox de runtime a permanente

Si enp0s3 no está asignada a ninguna zona, usar

sudo firewall-cmd --zone=public --change-interface=enp0s3

**Bloqueo de direcciones IP y redes con firewalld**

Pillas la IP del contenedor (entrando en sesión con él y haciendo ifconfig)

Buscas la tarjeta en el firewall buscando su ip entre el resultado de hacer ifconfig en la mv

La metes en public

Entras en rich rules de public

Cree una nueva regla enriquecida que elimine (drop) todas las conexiones al servicio ssh desde la IP del contenedor de ataque 1. Comprueba además todas las opciones que tiene la creación de reglas y piensa en sus posibilidades, especialmente las opciones de limit, log y audit.

• Prueba que ahora ya no puedes conectarte a través de ssh

• Modifica la regla anterior para bloquear toda la red 192.168.72.0/24

Poner en source 192.168.72.0/24 y volver a probar que no nos deja conectarlos desde ssh

Interfaz de usuario gráfica

Descripción generada automáticamente

**Reenvío de puertos en firewalld**

Localizar la tarjeta de red que sirve al contenedor web\_sies (ip=192.168.7.0/16, que debería ser 192.168.7.3, a mi me ponía que era 192.168.7.1) esto puede que sea mejor hacerlo en la mv normal

Me sale que es: br-b3ad1e810dbc

Colocar esta tarjeta en dmz (con la interfaz)

Ahora vamos a comunicarlos mediante reenvío de puertos:

Vete a la zona "pública" en la GUI y busca la pestaña "Port Forwarding" (dale a la flecha derecha para ver más pestañas)

Crea una nueva regla de reenvío de puertos que redirija todas las conexiones al puerto 3345 al puerto 80 de la IP 192.168.7.3 (el contenedor lab07\_web\_sies). Si el firewall te pide que habilites el enmascaramiento, responde Yes

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Vuelve a cargar el firewall (Options – Reload firewall)

Para probar esto, ejecutar Firefox en la línea de comandos del contenedor lab07\_firefox\_client (solo hay que poner firefox en la terminal)

**Bloque 2: Protección de red en conexiones salientes y entrantes**

**Protección de conexiones salientes: Filtrado de DNS con PiHole**

sudo apt-get remove apache2

Ir a la carpeta pihole de los archivos de lab

Ejecutar script run\_pihole.sh

Ir al navegador de la mv y buscar 127.0.0.1/admin

La contraseña de administrador es test123...

Te recomendamos que navegues ahora a cualquier página web con publicidad (como www.elmundo.es) y dejes esa web abierta

Cambiar el DNS de la tarjeta eth0 a 127.0.0.1. Para hacer esto:

Texto

Descripción generada automáticamente

También tendrás que cambiarlo en el fichero /etc/resolv.conf

**Filtrado avanzado de conexiones salientes con PiHole**

Vamos a incorporar una lista de ips (estas https://blocklistproject.github.io/Lists/) a PiHole

Copia el enlace de la lista más completa (Everything) en formato compatible con PiHole (https://blocklistproject.github.io/Lists/everything.txt).

Agrega la URL a las listas de bloqueo de tu PiHole (Login - Adlists - Group Management -Pega la URL de la lista en el campo "Dirección", agrega un comentario - Haz clic en "Add")

Actualiza Gravity (Tools-Update Gravity-Click "Update") para incorporar las nuevas URL a la lista de bloqueo. El proceso lleva un tiempo, ¡por favor espera y no navegues fuera de esta página!

Navega de nuevo a cualquier web con la página web de estadísticas de PiHole abierta y observa cómo están ahora en funcionamiento listas de bloqueo mucho más grandes

**Protección de intentos de intrusión: Fail2Ban**

Hacerlo en la mv normal

sudo apt install fail2ban

Modifica la configuración de fail2ban (archivo /etc/fail2ban/fail2ban.conf). Es MUY recomendable copiar este archivo a fail2ban.local y editar este nuevo archivo en lugar del .conf original. En este archivo puedes configurar cosas como el nivel de detalle del log (loglevel), dónde escribir la información de log (logtarget), etc. Usaremos la configuración predeterminada en este laboratorio:

sudo cp /etc/fail2ban/fail2ban.conf /etc/fail2ban/fail2ban.local

• Modifica el archivo /etc/fail2ban/jail.local para habilitar o deshabilitar las jails. Para hacer eso, haz lo siguiente:

o Si el archivo jail.local no existe, créalo copiando jail.conf:

sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

o Al principio, solo la jail sshd está activa por defecto. El resto debe activarse manualmente agregando enabled = true después de cada nombre de jail que quieras activar (el que se encuentra entre []).

o Cuando se activa una jail, los parámetros bantime, findtime y maxretry definen cómo se desautoriza una IP. Un bantime negativo significa un bloqueo permanente. Un host es bloqueado por una jail si activa sus condiciones con éxito más de maxretry veces a lo largo de findtime segundos.

o Las jails disponibles se definen en una sección de este archivo. Como decíamos, sshd está habilitada por defecto, mientras que las demás deben activarse manualmente. Eso sí, esto solo se recomienda si realmente tenemos el servicio en marcha.

Para este laboratorio habilitaremos el modo agresivo para sshd y garantizar el máximo modo de protección SSH.

Según chat el modo agresivo es poner en el jail de ssh:

[sshd]

enabled = true

port = ssh

filter = sshd

logpath = /var/log/auth.log (esto no lo puse)

maxretry = 3

bantime = 86400 # 24 horas

findtime = 600 # 10 minutos

Entonces: sudo nano jail.local y añadir lo de arriba

Comprobación del estado de Fail2Ban

fail2ban-client puede usarse para gestionar fail2ban a través de las siguientes opciones:

• start: inicia fail2ban

• reload: Recarga la configuración de fail2ban

• reload JAIL: Reemplaza la configuración actual de la jail especificada por la que está en el archivo

que se pasa

• stop: Detiene fail2ban

• status: muestra el estado actual de fail2ban y las jails activas (incluidos sus nombres)

• status JAIL: muestra el estado de la jail especificada. Es una opción muy importante ya que

también muestra cuántas veces se ha activado y las IPs bloqueadas actualmente.

Cada vez que actives nuevas jails en el archivo anterior debes volver a cargar fail2ban y verificar cuántas

jails están activas.

**Protección de intentos de escaneo: PortSentry como honeypot**

sudo apt install portsentry

Comprueba que se está ejecutando: sudo service portsentry status

Entiende los dos modos de trabajo de la herramienta (valores de los parámetros TCP\_MODE y UDP\_MODE en el archivo /etc/default/portsentry)

o Modo stealth avanzado (valores: "atcp", "audp"): Usa un raw socket para detectar escaneos y así la herramienta no puede ser detectada fácilmente: los puertos parecen tener el servicio "normal" en él, pero es portsentry quien está escuchando y detectando actividad sospechosa.

• Comprender las posibles respuestas a un intento de escaneo (valores de BLOCK\_UDP y BLOCK\_TCP en el archivo /etc/portsentry/portsentry.conf)

o Solo hacer log del tráfico sospechoso en /var/log/syslog (valor: "0") (valor por defecto)

o Bloquear a la máquina “infractora” (valor: "1")

Para registrar y bloquear intentos de análisis

Lo primero que hay que hacer para habilitar portsentry es habilitar los siguientes servicios en la zona de firewall docker: SSH, FTP y Telnet.

Una vez hecho esto, abre el fichero /etc/default/portsentry y pon TCP\_MODE a "atcp" y UDP\_MODE a "audp" para habilitar el modo stealth avanzado.

Reinicia portsentry y haz un análisis nmap simple desde cualquiera de los contenedores de infraestructura de Lab 7 a la máquina host (nmap ).

Una vez finalizado el análisis, comprueba el contenido de /var/log/syslog para ver si se han detectado los intentos de análisis.

Que se haga log con éxito de los intentos de escaneo es la condición para terminar de configurar portsentry según lo previsto: Abre /etc/portsentry/portsentry.conf y cambia los valores de los parámetros BLOCK\_UDP y BLOCK\_TCP a "1". Reinicia portsentry y vuelve a realizar el mismo análisis.

Para revertir bloqueos

• Eliminar la IP de la máquina bloqueada de /etc/hosts.deny. Estar en este archivo impide cualquier comunicación desde esta IP a nuestra máquina.

• Eliminar la regla del firewall que rechaza (REJECT) todo el tráfico del cliente infractor: sudo route del -host reject

• Podemos usar netstat -rn para comprobar que se ha eliminado la regla.

**Bloque 3. Detección de intrusiones**

**Instalación y prueba del IDS Maltrail**

Cd maltrail

Sudo python3 sensor.py

(en otra terminal) -> sudo python3 server.py

En Firefox accedemos a 127.0.0.1:8338 y nos sale la interfaz

Si hacemos un nmap a la maquina debería salir 1 threat

**Wazuh y ataques de red**

Abrir la MV Wazuh, realizar muchos logins incorrectos a nuestra MV y un escaneo con nmap y ver como reacciona

**Bloque 4. Extremando la seguridad en las conexiones: VPNs**

**Uso de VPNs tradicionales comerciales: ProtonVPN**