HARDENING - Configuración Segura de Sistemas Operativos

Características VM Hardening

Usuarios

user: soporte pass: opticpass	user: root pass: opticroot

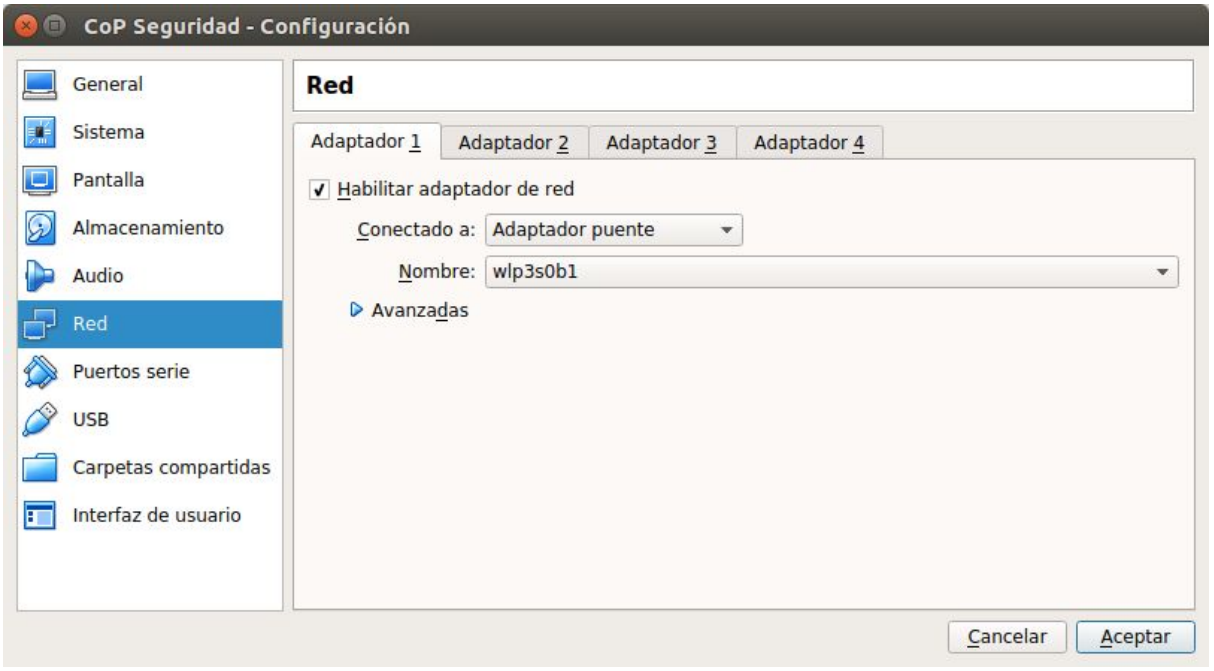
Servicios

- SSH
- mysql-server
- Bind9
- Apache2
- Fail2ban

Herramientas instaladas

- net-tools
- dnsutils

La configuración de la placa de red en la configuración de Virtualbox debe ser “Adaptador Puente”.



Antes de empezar deberemos saber qué dirección IP obtuvo la VM y configurar el servicio de DNS.

Conocer la dirección IP

```
soporte@hardening:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 08:00:27:f5:04:4c brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.116/24 brd 192.168.0.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fef5:44c/64 scope link
       valid_lft forever preferred_lft forever
```

Configuramos el servicio de DNS con la dirección IP que obtuvo la VM:

```
# nano /etc/bind/db.neuquen.gov.ar

;
; BIND zone file for neuquen.gov.ar
;

$TTL 3D
@ IN SOA ns1.neuquen.gov.ar. admin.neuquen.gov.ar. (
        1       ; Serial for updates
        10800   ; Refresh after 3 hours
        3600    ; Retry after 1 hours
        604800  ; Expire after 1 week
        86400)  ; Minimum TTL of 1 week

@               IN      NS      ns1.neuquen.gov.ar.
@               IN      MX      10      mail.neuquen.gov.ar.
ns1             IN      A       192.168.0.34  -> CAMBIAR IP
mail            IN      A       192.168.0.254
database        IN      CNAME   ns1
csirt-nqn       IN      CNAME   ns1
server          IN      CNAME   ns1
dhcp            IN      CNAME   ns1
```

Restarteamos el servicio

```
# /etc/init.d/bind9 restart
```

En nuestra máquina física configurar el servidor de dns

En Linux se debe modificar el archivo **/etc/resolv.conf** con la siguiente configuración

```
# nano /etc/resolv.conf

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver X.X.X.X -> dirección IP del servidor de DNS
nameserver 127.0.1.1
```

Para probar si quedo bien configurado, probar en la máquina física utilizando el comando host o dig

```
# host ns1.neuquen.gov.ar
ns1.neuquen.gov.ar has address X.X.X.X

# dig NS neuquen.gov.ar
; <<>> DiG 9.10.3-P4-Ubuntu <<>> NS neuquen.gov.ar
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35083
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;neuquen.gov.ar.                IN      NS

;; ANSWER SECTION:
neuquen.gov.ar.         259200  IN      NS      ns1.neuquen.gov.ar.

;; ADDITIONAL SECTION:
ns1.neuquen.gov.ar.   259200    IN      A       X.X.X.X

;; Query time: 0 msec
;; SERVER: X.X.X.X#53(X.X.X.X)
;; WHEN: Wed Oct 24 09:24:13 -03 2018
;; MSG SIZE  rcvd: 77
```

Ver servicios corriendo en la VM

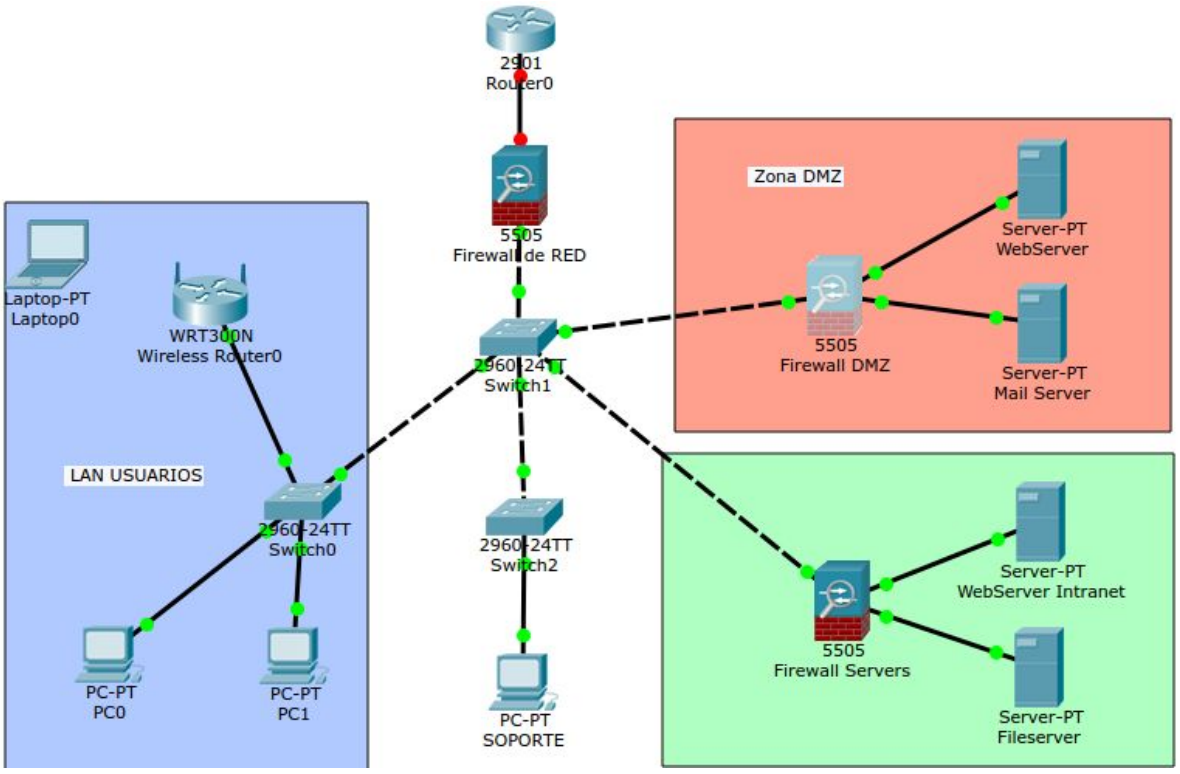
```
# netstat -natup
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN      2404/named
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      1574/mysqld
tcp        0      0 192.168.0.34:53         0.0.0.0:*               LISTEN      2404/named
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      2404/named
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      320/sshd
tcp6       0      0 ::1:953                 :::*                    LISTEN      2404/named
tcp6       0      0 :::80                   :::*                    LISTEN      2885/apache2
tcp6       0      0 :::53                   :::*                    LISTEN      2404/named
tcp6       0      0 :::22                   :::*                    LISTEN      320/sshd
udp        0      0 192.168.0.34:53         0.0.0.0:*                           2404/named
udp        0      0 127.0.0.1:53            0.0.0.0:*                           2404/named
udp        0      0 0.0.0.0:68              0.0.0.0:*                           300/dhclient
udp6       0      0 :::53                   :::*                                2404/named
```

Listado de puertos más comunes

- 21 > FTP
- 22 > SSH
- 23 > Telnet
- 25 > SMTP
- 43 > WHOIS
- 53 > Nameservers (sistema DNS)
- 80 > HTTP (servidor web, ya sea Apache, Nginx u otro)
- 110 > Protocolo POP utilizado para mail.
- 143 > Protocolo IMAP utilizado para mail.
- 161 > SNMP (Simple Network Management Protocol)
- 443 > HTTP seguro (utilizado por certificados SSL a nivel web)
- 953 > rndc
- 993 > IMAP bajo SSL
- 995 > Protocolo POP con SSL

Los números de puerto del 0 al 1023 están reservados para servicios bien conocidos. la mayor parte de los servicios usan bien TCP bien UDP pero hay algunos que pueden comunicar con los dos protocolos. Para cada uno tenemos 65535 puertos.

IPTABLES



IPtables es un sistema de firewall vinculado al kernel de linux que se ha extendido enormemente a partir del kernel 2.4 de este sistema operativo.
IPtables es una herramienta que permite filtrar, redireccionar, rechazar, encolar paquetes según unas características como pueden ser ip origen o destino, puerto, mac, entre otros.
IPtables nos permite definir reglas de distintos tipos como pueden ser de filtrado (default), de nat (para la intranet), de mangle (para manipular paquetes) y de raw (para excepciones)

Tabla de Filtros

- Input: Indica paquetes recibidos
- Output: Indica paquetes salientes
- Forward: Indican paquetes que se reciben pero que no son para nosotros sino que se enrutan de nuevo.

Políticas

- ACCEPT: El firewall aceptara el paquete
- DROP: El firewall rechazará el paquete

Ver tabla de Filtros de IPTABLES

```
# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Vamos a trabajar con un script armado para crear reglas con IPTABLES que se encuentra en **/etc/init.d/firewall**.

```
# nano /etc/init.d/firewall
```

El script puede recibir los siguientes parámetros

```
# /etc/init.d/firewall
Usege (start | stop | restart | status | start4 | stop4 | status4 | start6 | stop6 | status6)
```

Para iniciar el script

```
# /etc/init.d/firewall start
```

Reglas de ejemplo

```
# Permitir que nuestra IP de nuestra máquina física tenga acceso a todo
iptables -A INPUT -s <IP máquina física> -j ACCEPT

# Permitir que una subred tenga acceso a todo
iptables -A INPUT -s X.X.X.X/XX -j ACCEPT

# Permitir que nuestra IP de nuestra máquina física tenga acceso solo al servicio de SSH
iptables -A INPUT -s <IP máquina física> -p tcp --dport 22 -j ACCEPT

# Permitir que cualquier IP tenga acceso al servicio web HTTP
iptables -A INPUT -p tcp --dport 80 -j ACCEPT

# Permitir que nuestra IP de nuestra máquina física tenga acceso solo al servicio de SSH
iptables -A INPUT -s <IP máquina física> -p tcp --dport 22 -j ACCEPT

# Permitir que nuestra IP de nuestra máquina física tenga acceso solo al servicio de MYSQL
iptables -A INPUT -s <IP máquina física> -p tcp --dport 3306 -j ACCEPT

# Permitir que nuestra IP de nuestra máquina física tenga acceso solo al servicio de MYSQL
iptables -A INPUT -s <IP máquina física> -p tcp --dport 3306 -j ACCEPT

# Bloquear y prevenir ataques de DDoS
iptables -A INPUT -p tcp --dport 80 -m limit --limit 1/minute --limit-burst 100 -j ACCEPT
```

Ejercicio

Cambiemos el puerto por default donde corre el servicio de SSH y agregamos la nueva regla al firewall

Archivo de configuración de SSH **/etc/ssh/sshd_config**

```
# nano /etc/ssh/sshd_config

#       $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

#Port 22  → Modificar por el puerto 2200 y descomentamos
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Una vez que se modifica el puerto donde corre el servicio de SSH, restartear el servicio

```
# /etc/init.d/ssh restart
```

Desconectarse de la máquina y tratar de conectarse por ssh a la VM.

Cómo filtrar escaneos a nuestra red

Un comando muy utilizado para realizar escaneos es NMAP

Ejemplos NMAP

Escaneo simple

```
# nmap <dirección ip>
# nmap 192.168.10.37
Starting Nmap 7.01 ( https://nmap.org ) at 2018-10-25 08:41 -03
Nmap scan report for 192.168.10.37
Host is up (0.0015s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
53/tcp open  domain
80/tcp open  http
MAC Address: 08:00:27:F5:04:4C (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.64 seconds
```

Probemos con otras parametros

```
# nmap -sV -Pn -O -vv -o archivo.txt 192.168.10.37
```

El anterior comando ejecuta un barrido (scan) de puertos sobre la IP seleccionada, evita que se ejecute Ping sobre la máquina, además de esto intenta detectar el sistema operativo, para cada puerto según las cabeceras que se retornan se detecten los servicios ejecutándose y la información se dejará en el archivo.txt

Fail2ban

Fail2ban es una herramienta de seguridad escrita en Python fundamental para cualquier servidor que preste servicios públicos.
Su principal función es securizar un servidor del siguiente modo:

1. **Evitando accesos indeseados** a nuestro equipo o servidor.
2. **Evitando ataques de fuerza bruta** para que un tercero averigüe nuestra contraseña o nos deje sin servicio.

Iniciar Fail2ban

```
# /etc/init.d/fail2ban start
[ ok ] Starting fail2ban (via systemctl): fail2ban.service.
root@hardening:/home/soporte# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
f2b-sshd   tcp  --  0.0.0.0/0            0.0.0.0/0            multiport dports 22

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain f2b-sshd (1 references)
target     prot opt source               destination
RETURN     all  --  0.0.0.0/0            0.0.0.0/0
```

Archivo de configuración de Fail2ban **/etc/fail2ban/jail.conf**

```
# nano /etc/fail2ban/jail.conf
```

```
[DEFAULT]

#
# MISCELLANEOUS OPTIONS
#

# "ignoreip" can be an IP address, a CIDR mask or a DNS host. Fail2ban will not
# ban a host which matches an address in this list. Several addresses can be
```

```
# defined using space (and/or comma) separator.
ignoreip = 127.0.0.1/8

# External command that will take an tagged arguments to ignore, e.g. <ip>,
# and return true if the IP is to be ignored. False otherwise.
#
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned.
bantime  = 600

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime  = 600

# "maxretry" is the number of failures before a host get banned.
maxretry = 5
```

Por un lado fail2ban está monitorizando las autenticaciones que una determinada IP hace a un determinado/s puerto/s y servicio/s.
Para ello fail2ban está consultando permanente los logs de autenticación de nuestro sistema como por ejemplo el **/var/log/auth.log**.
Si fail2ban detecta un número determinado de intentos de conexión fallidos bloqueará la IP que está intentando acceder a nuestro equipo o el servicio. La forma de bloquear la IP será introduciendo una regla en el Firewall de nuestro equipo o servidor durante un tiempo determinado que por ejemplo pueden ser 600 segundos.
Una vez transcurridos los 600 segundos, o el tiempo que nosotros queramos, se borrará la regla del firewall. Por lo tanto la IP que fue bloqueada podrá intentar conectar de nuevo a nuestro servidor.

Fail2ban también trabaja con otros servicios, para ver los filtros que tiene se puede acceder a **/etc/fail2ban/filter.d/**

```
# ls /etc/fail2ban/filter.d/
3proxy.conf               freeswitch.conf          postfix-rbl.conf
apache-auth.conf          froxlor-auth.conf        postfix-sasl.conf
apache-badbots.conf       groupoffice.conf         proftpd.conf
apache-botsearch.conf     gssftpd.conf             pure-ftpd.conf
apache-common.conf        guacamole.conf           qmail.conf
apache-fakegooglebot.conf haproxy-http-auth.conf   recidive.conf
apache-modsecurity.conf   horde.conf               roundcube-auth.conf
apache-nohome.conf        ignorecommands           screensharingd.conf
apache-noscript.conf      kerio.conf               selinux-common.conf
apache-overflows.conf     lighttpd-auth.conf       selinux-ssh.conf
```

```
apache-pass.conf          mongodb-auth.conf     sendmail-auth.conf
apache-shellshock.conf    monit.conf            sendmail-reject.conf
assp.conf                 murmur.conf           sieve.conf
asterisk.conf             mysqld-auth.conf      slapd.conf
botsearch-common.conf     nagios.conf           sogo-auth.conf
common.conf               named-refused.conf    solid-pop3d.conf
counter-strike.conf       nginx-botsearch.conf  squid.conf
courier-auth.conf         nginx-http-auth.conf  squirrelmail.conf
courier-smtp.conf         nginx-limit-req.conf  sshd.conf
cyrus-imap.conf           nsd.conf              sshd-ddos.conf
directadmin.conf          openhab.conf          stunnel.conf
dovecot.conf              openwebmail.conf      suhosin.conf
dropbear.conf             oracleims.conf        tine20.conf
drupal-auth.conf          pam-generic.conf      uwimap-auth.conf
ejabberd-auth.conf        perdition.conf        vsftpd.conf
exim-common.conf          php-url-fopen.conf    webmin-auth.conf
exim.conf                 portsentry.conf       wuftpd.conf
exim-spam.conf            postfix.conf          xinetd-fail.conf
```

Quitar “baneos” de fail2ban

El primer paso es identificar la línea de la IP que queremos desbanear dentro de la cadena chain f2b-ssh2.

```
# iptables -L --line-number
```

Una vez que tenemos identificada la línea, quitamos la regla

```
# iptables -D f2b-sshd <número>
```

Más Información

- IPTABLES https://wiki.archlinux.org/index.php/Iptables_(Espa%C3%B1ol)
- Estados de IPtables http://geneura.ugr.es/~gustavo/cortafuegos/state.html
- Fail2ban https://www.fail2ban.org/wiki/index.php/HOWTO_fail2ban_spanish
- Fail2ban Instalación https://geekland.eu/instalar-configurar-y-usar-fail2ban-para-evitar-ataques-de-fuerza-bruta/

Mis datos:

Pacheco Veliz Sebastian Exequiel
spacheco@neuquen.gov.ar