1.1 CONFIGURACIÓN FIREWALL: 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. Un firewall de *iptables* no es como un servidor que se inicia o detiene o que pueda caer por un error de programación: iptables está integrado con el kernel, es parte del sistema operativo.

Funciona mediante la aplicación de reglas. Para ello se ejecuta el comando *iptables*, con el que se añaden, borran, o modifican reglas. Por ello un firewall de iptables no es más que un simple script de Shell en el que se van ejecutando las reglas de firewall.

Si desea conocer las reglas activas (IPv4) ejecute:

```
# iptables -L
```

Y para obtener las reglas de IPv6:

```
# ip6tables -L
```

Se pueden utilizar scripts Bash para llevar a cabo cambios en las reglas del firewall. Una de las posibles acciones (poco recomendable) es eliminar todas las reglas, con lo que el firewall queda desactivado, permitiendo todo acceso a la red (Ejemplo ¡Error! **No hay texto con el estilo especificado en el documento.**.1).

Ejemplo ¡Error! No hay texto con el estilo especificado en el documento.**.1 Flush IPTables (IPv4)**

```
#!/bin/bash
echo "Desactivando Firewall..."
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
```

Igualmente se puede hacer para las reglas de IPv6 (Ejemplo ¡Error! **No hay texto con el estilo especificado en el documento.**.2).

Ejemplo ¡Error! No hay texto con el estilo especificado en el documento.**.2 Flush IPTables (IPv6)**

```
#!/bin/bash
IPT6="/sbin/ip6tables"
echo "Desactivando Firewall IPv6..."
$IPT6 -F
$IPT6 -X
$IPT6 -Z
for table in $(</proc/net/ip6_tables_names)
do
        $IPT6 -t $table -F
        $IPT6 -t $table -X
        $IPT6 -t $table -Z
done
$IPT6 -P INPUT ACCEPT
$IPT6 -P OUTPUT ACCEPT
$IPT6 -P FORWARD ACCEPT
```

1.1.1 Scripts de administración

Un administrador debe tener sumo cuidado con las reglas que aplica, por lo que debe tener un amplio conocimiento de Iptables.

Si no dispone de un gran conocimiento sobre IPTables, el Ejemplo ¡Error! **No hay texto con el estilo especificado en el documento.**.3 puede ayudarle a manejarlas de un modo algo más sencillo.

Ejemplo ¡Error! No hay texto con el estilo especificado en el documento.**.3 Manejar Iptables**

```
#!/bin/bash
# Manejo simple de iptables

complete_rule=iptables
part_rule=

function _aplicar
{
    echo -e "Aplicar $part_rule a $complete_rule? (s/n)"
    read a
    if [[ $a = "s" ]]
    then
        complete_rule=$(echo $complete_rule $part_rule)
        echo regla:$complete_rule
    else
        echo regla:$complete_rule
    fi
}

function asistente # asistente: Ejecuta un asistente guiado para añadir una
regla
{
    clear
    echo "añadir o quitar norma"
    echo "añadir | quitar [INPUT|OUTPUT|FORWARD]"
    echo -e "qué hacer)\c"
    read aa
    $aa

    echo "protocolos admitidos: tcp udp icmp all"
    echo -e "protocolo)\c"
    read pro
    protocolo "$pro"

    echo "seleccione política"
    echo "acciones: aceptar | rechazar | ignorar "
    echo -e "acción)\c"
    read ac
    $ac

    echo "dirección: origen | destino [IP o host]"
    echo -e "dirección)\c"
    read di
    $di

    if  [ "$pro" = "tcp" ] -o [ "$pro" = "udp" ]
    then
        echo "puerto al que aplicar la norma"
        echo -e "puerto)\c"
        read pu
        puerto $pu
    fi

    echo "Puede Aplicar la norma ahora o añadir"
    echo "más propiedades en éste mismo CLI"
    echo -e "Aplicar la norma ahora (s/n)?\c"
    read ap
    if [[ $ap = "s" ]]
    then
        aa=$(echo $aa | cut -d " " -f 1)
        if [[ $aa = "añadir" ]]
        then
```

```
            ejecutar
        else
            echo -e "Desea borrar ésta norma?? (s/n)?\c"
            read bbb
            if [[ $bbb = "s" ]]
            then
                ejecutar
            else
                echo "No Borrada"
            fi
        fi
    else
        echo "Use la ayuda para ver que propiedades"
        echo "tiene disponibles"
        echo " "
    fi

}

function añadir # añadir [INPUT|OUTPUT|FORWARD]: Añade al tipo de Norma
{
    echo "-A"
    part_rule="-A $1"
    _aplicar
}

function quitar # quitar [INPUT|OUTPUT|FORWARD]
{
    echo "-D"
    part_rule="-D $1"
    _aplicar
}

function protocolo # protocolo [tcp|udp|icmp|all]: Añade un tipo de protocolo
{
    echo "-p $1"
    part_rule="-p $1"
    _aplicar
}

function puerto # puerto [n]: Define el puerto de la norma
{
    echo Puerto: $1
    part_rule="--dport $1"
    _aplicar
}

function aceptar # aceptar: Aplica la norma ACCEPT a los paquetes
{
    echo "-j ACCEPT"
    part_rule="-j ACCEPT"
    _aplicar
}

function rechazar # rechazar: Aplica la norma REJECT a los paquetes
{
    echo "-j REJECT"
    part_rule="-j REJECT"
    _aplicar
}

function ignorar # ignorar: Aplica la norma DROP a los paquetes
{
    echo "-j DROP"
    part_rule="-j DROP"
    _aplicar
}

function interfaz # interfaz [ethn] Define el interfaz para la norma
{
    echo "-i $1"
    part_rule="-i $1"
    _aplicar
}

function numero # numero [n]: Define el número de norma (útil para borrar)
{
```

```
    echo "-I $1"
    part_rule="-I $1"
    _aplicar
}

function origen # origen [IP o HOST]: Define el origen del paquete para la
norma
{
    echo "-s $1"
    part_rule="-s $1"
    _aplicar
}

function destino # destino [IP o HOST] Define el destino del paquete de la
norma
{
    echo "-d $1"
    part_rule="-d $1"
    _aplicar
}

function normas # normas: Hace un listado de las normas existentes
{
    echo "Norma que está siendo editada"
    echo $complete_rule
    echo "Última parte aplicada o no a la norma"
    echo $part_rule
    echo " "
    echo " "
    i=$(whoami)
    if [[ $i != "root" ]]
    then
        echo "No es root, no se puede ejecutar iptables"
    else
        iptables -L -v
    fi

}

function nueva # nueva: Empieza a hacer una norma nueva
{
    complete_rule="iptables"
    part_rule=""
    echo "Nueva norma iniciada"
}


function ejecutar # ejecutar: Aplica la norma
{
    echo "Aplicando norma: $complete_rule"
    $complete_rule
    echo "Hecho"
}

function guardar # guardar [fichero] Vuelca en un archivo las normas
{
    echo "Guardar normas en fichero"
    echo "nombre del fichero)\c"
    read nom
    iptables-save > $nom
    echo "Guardado como $nom"
}
    # exit 0 : sale del script

while [[ $? ]]
do
    i=$(whoami)
    if [[ $i != "root" ]]
    then
        echo "No es root, no se aplicará ningún cambio"
    fi
    echo "Configurador de iptables."
    echo -e "root)\c"
    read rul
    $rul
done
```

El funcionamiento del scrit anterior es muy sencillo, sobre todo a través del asistente. A continuación se demuestra como habilitar el tráfico de la red local (192.168.1.0/24) que se dirija hacia el puerto 80 (http):

```
[root@localhost ~]# ./iptables.sh
Configurador de iptables.
root)asistente

añadir o quitar norma
añadir | quitar [INPUT|OUTPUT|FORWARD]
qué hacer)añadir FORWARD
-A
Aplicar -A FORWARD a iptables? (s/n)
s
regla:iptables -A FORWARD
protocolos admitidos: tcp udp icmp all
protocolo)tcp
-p tcp
Aplicar -p tcp a iptables -A FORWARD? (s/n)
s
regla:iptables -A FORWARD -p tcp
seleccione política
acciones: aceptar | rechazar | ignorar
acción)aceptar
-j ACCEPT
Aplicar -j ACCEPT a iptables -A FORWARD -p tcp? (s/n)
s
regla:iptables -A FORWARD -p tcp -j ACCEPT
dirección: origen | destino [IP o host]
dirección)origen 192.168.1.0/24
-s 192.168.1.0/24
Aplicar -s 192.168.1.0/24 a iptables -A FORWARD -p tcp -j ACCEPT? (s/n)
s
regla:iptables -A FORWARD -p tcp -j ACCEPT -s 192.168.1.0/24
puerto al que aplicar la norma
puerto)80
Puerto: 80
Aplicar --dport 80 a iptables -A FORWARD -p tcp -j ACCEPT -s 192.168.1.0/24?
(s/n)
s
regla:iptables -A FORWARD -p tcp -j ACCEPT -s 192.168.1.0/24 --dport 80
Puede Aplicar la norma ahora o añadir
más propiedades en éste mismo CLI
Aplicar la norma ahora (s/n)?s
Aplicando norma: iptables -A FORWARD -p tcp -j ACCEPT -s 192.168.1.0/24 --
dport 80
Hecho
Configurador de iptables.
root)normas
Norma que está siendo editada
iptables -A FORWARD -p tcp -j ACCEPT -s 192.168.1.0/24 --dport 80
Última parte aplicada o no a la norma
--dport 80


Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
destination
 2600  161K RH-Firewall-1-INPUT  all  --  any    any     anywhere
anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
destination
    0     0 REJECT     all  --  any    any     anywhere             anywhere
reject-with icmp-host-prohibited
    0     0 ACCEPT     tcp  --  any    any     192.168.1.0/24       anywhere
tcp dpt:http

Chain OUTPUT (policy ACCEPT 2346 packets, 121K bytes)
 pkts bytes target     prot opt in     out     source
destination

Chain RH-Firewall-1-INPUT (1 references)
```

```
 pkts bytes target     prot opt in     out     source
destination
 2292  115K ACCEPT     all  --  lo     any     anywhere             anywhere
  308 46645 ACCEPT     all  --  eth1   any     anywhere             anywhere
    0     0 ACCEPT     icmp --  any    any     anywhere             anywhere
icmp any
    0     0 ACCEPT     esp  --  any    any     anywhere             anywhere
    0     0 ACCEPT     ah   --  any    any     anywhere             anywhere
    0     0 ACCEPT     udp  --  any    any     anywhere
224.0.0.251         udp dpt:mdns
    0     0 ACCEPT     udp  --  any    any     anywhere             anywhere
udp dpt:ipp
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
tcp dpt:ipp
    0     0 ACCEPT     all  --  any    any     anywhere             anywhere
state RELATED,ESTABLISHED
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
state NEW tcp dpt:ssh
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
state NEW tcp dpt:http
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
state NEW tcp dpt:domain
    0     0 ACCEPT     udp  --  any    any     anywhere             anywhere
state NEW udp dpt:domain
    0     0 REJECT     all  --  any    any     anywhere             anywhere
reject-with icmp-host-prohibited
Configurador de iptables.
root)exit
[root@localhost ~]#
```

El Ejemplo ¡Error! **No hay texto con el estilo especificado en el documento.**.4 presenta una versión más completa y elegante para trabajar con iptables.

Ejemplo ¡Error! No hay texto con el estilo especificado en el documento.**.4 Manejar Iptables II**

```
#!/bin/bash
# Manejo de iptables

echo -e "****************BIENVENIDO*************"

################################ DECLARACIÓN DE FUNCIONES
################################
checkstatus()
 {
  opt_checkstatus=1
  while [ $opt_checkstatus != 7 ]
  do
       clear
       #echo -e "\nSeleccione una opción:\n
       echo -e "\n\t*****Importante: Guarde sus iptables antes de
parar/reiniciar el servicio iptables*****\n"
       echo -e "
       1. Guardar las iptables\n
       2. Estado de Iptables\n
       3. Inicializar el servicio iptables\n
       4. Parar el servicio iptables\n
       5. Reiniciar el servicio iptables\n
       6. Flush iptables (**Úselo bajo su responsabilidad**)\n
       7. Volver al menú principal"

  read opt_checkstatus

  case $opt_checkstatus in
   1) echo -e "*********************************************************\n"
              /etc/init.d/iptables save
              echo -e
"\n*****************************************************\n"
              echo -e "Presione Intro para continuar..."
              read temp;;

   2) echo -e "*********************************************************\n"
               /etc/init.d/iptables status
      echo -e "*********************************************************"
```

```
               echo -e "Presione Intro para continuar..."
               read temp;;

   3) echo -e "*********************************************************\n"
               /etc/init.d/iptables start
      echo -e "*********************************************************\n"
               echo -e "Presione Intro para continuar..."
               read temp;;

   4) echo -e "*********************************************************\n"
               /etc/init.d/iptables stop
      echo -e "*********************************************************\n"
               echo -e "Presione Intro para continuar..."
               read temp;;

   5) echo -e "*********************************************************\n"
               /etc/init.d/iptables restart
      echo -e "*********************************************************\n"
               echo -e "Presione Intro para continuar..."
               read temp;;

   6) iptables -F
               echo -e
"*****************************************************"
               echo -e "Todas las reglas han sido eliminadas!!"
               echo -e
"*****************************************************\n"
               echo -e "Presione Intro para continuar..."
               read temp;;
   7) main;;
   *) echo -e "Opción incorrecta!"
  esac
 done
 }

################################ CONSTRUCCIÓN DE REGLAS
################################
buildfirewall()
 {
  ############### Añadir regla ############
  echo -e "Desea añadir una nueva regla?\n
  1. INPUT
  2. OUTPUT
  3. Forward"
  read opt_ch
  case $opt_ch in
   1) chain="INPUT" ;;
   2) chain="OUTPUT" ;;
   3) chain="FORWARD" ;;
   *) echo -e "Opción incorrecta!"
  esac

  ######### Obtención de la dirección IP ##########

  echo -e "
  1. Firewall: regla para IP individual\n
  2. Firewall: regla para la subred\n
  3. Firewall: regla para todos las fuentes\n"
  read opt_ip

  case $opt_ip in
   1) echo -e "\nIntroduzca la dirección IP de origen"
      read ip_source ;;
   2) echo -e "\nIntroduzca la subred de origen (e.g 192.168.10.0/24)"
      read ip_source ;;
   3) ip_source="0/0" ;; #cualquier fuente
   #4) ip_source = "NULL" ;;
   *) echo -e "Opción incorrecta!"
  esac

  ######### Obtención de la IP de destino ##########
  echo -e "
  1. Firewall: regla para  IP de destino única\n
  2. Firewall: regla para una subred de destino\n
  3. Firewall: regla para todos los destinos\n"
  read opt_ip
```

```
  case $opt_ip in
   1) echo -e "\nIntroduzca la IP de destino"
      read ip_dest ;;
   2) echo -e "\nIntroduzca la subred de destino (e.g 192.168.10.0/24)"
      read ip_dest ;;
   3) ip_dest="0/0" ;; #todos los destinos
   *) echo -e "Opción incorrecta!"
  esac

  ################ Obtención del protocolo #############
  echo -e "
  1. Boquear todo el tráfico TCP
  2. Bloquear un servicio TCP específico
  3. Bloquear un puerto específico
  4. No usar protocolo"
  read proto_ch

  case $proto_ch in
   1) proto=TCP ;;
   2) echo -e "Introduzca el nombre del servicio TCP: (USE MAYÚSCULAS)"
      read proto ;;
   3) echo -e "Introduzca el nombre del puerto: (USE MAYÚSCULAS)"
      read proto ;;
   4) proto="NULL" ;;
   *) echo -e "Opción incorrecta!"
  esac

  ############## Función de la regla #############
  echo -e "Qué debe hacer la regla?
  1. Aceptar el paquete
  2. Rechazar el paquete
  3. Reenviar el paquete
  4. Crear Log"
  read rule_ch

  case $rule_ch in
   1) rule="ACCEPT" ;;
   2) rule="REJECT" ;;
   3) rule="DROP" ;;
   4) rule="LOG" ;;
  esac

  ################### Generación de la regla ####################
  echo -e "\n\tPresione Intro para generar la regla!"
  read temp
  echo -e "Regla generada: \n"

  if [ $proto == "NULL" ]; then
    echo -e "\niptables -A $chain -s $ip_source -d $ip_dest -j $rule\n"
    gen=1
  else
    echo -e "\niptables -A $chain -s $ip_source -d $ip_dest -p $proto -j
$rule\n"
    gen=2
  fi

  echo -e "\n\tDesea almacenar permanentemente la regla en IPTABLES? Si=1 ,
No=2"
  read sino

  if [ $sino == 1 ] && [ $gen == 1 ]; then
    iptables -A $chain -s $ip_source -d $ip_dest -j $rule
  else if [ $sino == 1 ] && [ $gen == 2 ]; then
    iptables -A $chain -s $ip_source -d $ip_dest -p $proto -j $rule
  else if [ $sino == 2 ]; then
    main
  fi
  fi
  fi
}

main()
{
 ROOT_UID=0
 if [ $UID == $ROOT_UID ];
 then
  clear
```

```
  opt_main=1
  while [ $opt_main != 3 ]
  do
  echo -e "/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\n"

  echo -e "\t*****Menú Principal*****\n
  1. Comprobar los servicios de iptables\n
  2. Construir su firewall con iptables\n
  3. Exit"
  read opt_main

  case $opt_main in
       1) checkstatus ;;
       2) buildfirewall ;;
       3) exit 0 ;;
       *) echo -e "Opción incorrecta!"
  esac
  done
  else
       echo -e "Debe ejecutar $0 como root!"
  fi
}

main
exit 0
```

El siguiente ejemplo demuesta como utilizar el script para comprobar las reglas del firewall y posteriormente crear una nueva regla que de acceso al firewall desde la red local (192.168.1.0/24).

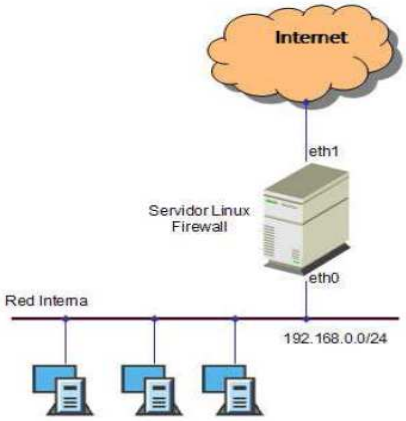
```
[root@localhost ~]# ./iptables2.sh
*****************BIENVENIDO*************
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\

        *****Menú Principal*****

  1. Comprobar los servicios de iptables

  2. Construir su firewall con iptables

  3. Exit
1
        *****Importante: Guarde sus iptables antes de  parar/reiniciar el
servicio iptables*****


        1. Guardar las iptables

        2. Estado de Iptables

        3. Inicializar el servicio iptables

        4. Parar el servicio iptables

        5. Reiniciar el servicio iptables

        6. Flush iptables (**Úselo bajo su responsabilidad**)

        7. Volver al menú principal
2
***************************************************

Tabla: filter
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  0.0.0.0/0           0.0.0.0/0

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     all  --  0.0.0.0/0           0.0.0.0/0           reject-with
icmp-host-prohibited
2    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:80

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

```
Chain RH-Firewall-1-INPUT (1 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
2    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
3    ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type
255
4    ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
5    ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
6    ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
7    ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
8    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
9    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state
RELATED,ESTABLISHED
10   ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW
tcp dpt:22
11   ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW
tcp dpt:80
12   ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW
tcp dpt:53
13   ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           state NEW
udp dpt:53
14   REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with
icmp-host-prohibited

**************************************************
Presione Intro para continuar...

        *****Importante: Guarde sus iptables antes de  parar/reiniciar el
servicio iptables*****


        1. Guardar las iptables

        2. Estado de Iptables

        3. Inicializar el servicio iptables

        4. Parar el servicio iptables

        5. Reiniciar el servicio iptables

        6. Flush iptables (**Úselo bajo su responsabilidad**)

        7. Volver al menú principal
7
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/

        *****Menú Principal*****

  1. Comprobar los servicios de iptables

  2. Construir su firewall con iptables

  3. Exit
2
Desea añadir una nueva regla?

  1. INPUT
  2. OUTPUT
  3. Forward
1

  1. Firewall: regla para IP individual

  2. Firewall: regla para la subred

  3. Firewall: regla para todos las fuentes

2

Introduzca la subred de origen (e.g 192.168.10.0/24)
192.168.1.0/24

  1. Firewall: regla para  IP de destino única

  2. Firewall: regla para una subred de destino
```

```
  3. Firewall: regla para todos los destinos

3

  1. Boquear todo el tráfico TCP
  2. Bloquear un servicio TCP específico
  3. Bloquear un puerto específico
  4. No usar protocolo
3
Introduzca el nombre del puerto: (USE MAYÚSCULAS)
TCP
Qué debe hacer la regla?
  1. Aceptar el paquete
  2. Rechazar el paquete
  3. Reenviar el paquete
  4. Crear Log
1

        Presione Intro para generar la regla!

Regla generada:


iptables -A INPUT -s 192.168.1.0/24 -d 0/0 -p TCP -j ACCEPT


        Desea almacenar permanentemente la regla en IPTABLES? Si=1 , No=2
1
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/

        *****Menú Principal*****

  1. Comprobar los servicios de iptables

  2. Construir su firewall con iptables

  3. Exit

3
```

EJERCICIOS PROPUESTOS

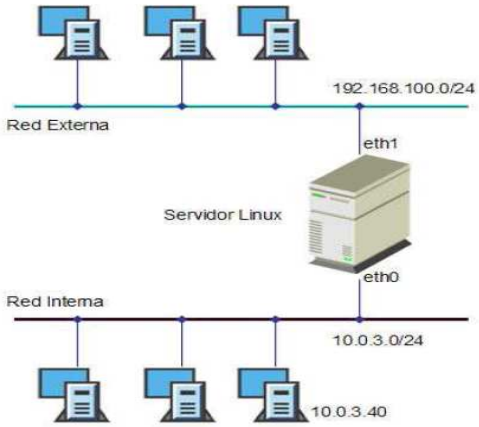
1. Suponiendo el siguiente esquema de red:



Desarrolle un script para configurar el firewall del servidor para que realice las siguientes tareas:

- Permita todos los accesos al servidor Linux desde la red interna.
- Desde internet permitir sólo las conexiones a los servicios HTTP, HTTPS Y SMTP.

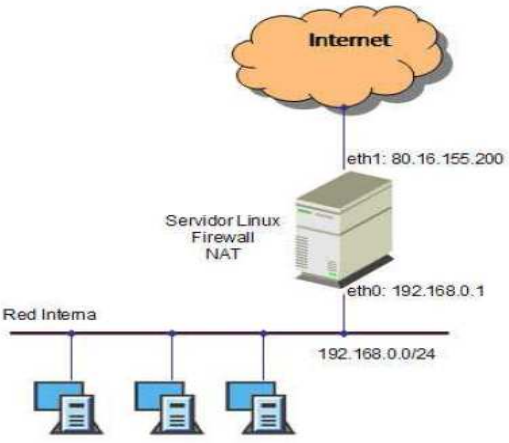
2. Suponga que dispone de dos redes locales con varios equipos y un servidor Linux haciendo funciones de routing entre ambas redes, tal y como muestra el siguiente esquema:



Desarrolle un script para configurar el firewall del servidor Linux teniendo en cuenta que debe realizar las siguientes tareas:

- Realizar un enmascaramiento de la red interna.
- Redireccionar las peticiones Web a la máquina 10.0.3.40, permitiendo el tráfico específico en el firewall.
- Permitir sólo el tráfico de salida de tipo WEB y DNS, y el resto se rechaza.
- Realizar un registro (Log) de intentos de acceso desde la red externa al firewall y a los equipos internos.
- El firewall sólo admitirá conexiones SSH desde la red interna.
- Limitar el tráfico de control ICMP para evitar ataques DoS (hasta un máximo de cinco peticiones por segundo)

3. Suponga que dispone de una red local con varios equipos y un servidor Linux haciendo las funciones de routing de dicha red local e Internet, tal y como muestra el siguiente esquema:



Desarrolle un script de modo que el firewall cumpla las siguientes restricciones:

- Proporcione acceso a Internet a los equipos internos a través de SNAT con la dirección del servidor Linux.
- No debe permitir a los equipos internos:
  - Hacer ping a un equipo externo.
  - Acceder por FTP a un equipo externo.
  - Acceder por SSH al servidor Linux.
- No debe permitir al servidor Linux:
  - Recibir ninguna petición a los puertos privilegiados (1-1024) desde la red local.
- Se permite todo lo demás (política ACCEPT).

4.	Realice un script de elimine toda la configuración establecida por el ejercicio anterior.