

TEMA 3



DESARROLLO DE APLICACIONES WEB

Alex Domínguez Domínguez

CONTENIDO

INSTALACIÓN Y CONFIGURACIÓN DE COMPONENTES

.....¡ERROR! MARCADOR NO DEFINIDO.

Explica y enumera protocolos asociados a la transferencia de archivos a un servidor remoto: TFTP, FTP, FTPS, FTPES, SFTP, FXP. iii

SFTP v

 EL INTERCAMBIO DE DATOS..... v

 SEGURIDAD v

 FIREWALL v

FTP v

 EL INTERCAMBIO DE DATOS..... v

 SEGURIDAD vi

Scriptsvi

Alta usuarios.....vi

Baja Usuarios..... vii

alta Sitios web vii

Baja Sitios web viii

Script automatización de tareas viii

EXPLICA Y ENUMERA PROTOCOLOS ASOCIADOS A LA TRANSFERENCIA DE ARCHIVOS A UN SERVIDOR REMOTO: TFTP, FTP, FTPS, FTPES, SFTP, FXP.

TFTP: Es un protocolo de transferencia muy simple semejante a una versión básica de FTP. TFTP a menudo se utiliza para transferir pequeños archivos entre computadoras en una red, como cuando un terminal X Window o cualquier otro cliente ligero arranca desde un servidor de red.

Algunos detalles del TFTP:

- Utiliza UDP (en el puerto 69) como protocolo de transporte (a diferencia de FTP que utiliza los puertos 20 y 21 TCP).
- No puede listar el contenido de los directorios.
- No existen mecanismos de autenticación o cifrado.
- Se utiliza para leer o escribir archivos de un servidor remoto.
- Soporta tres modos diferentes de transferencia, "netascii", "octet" y "mail", de los que los dos primeros corresponden a los modos "ascii" e "imagen" (binario) del protocolo FTP.

FTP: (puerto 21)

FTP simple, sin cifrar, que se establece de manera predeterminada en el puerto 21. La mayoría de los navegadores web admiten FTP básico.

Ejemplo para FileZilla, para Host: ftp //: SERVER-IP

FTPS: (puerto 990)

FTP cifrado SSL / TLS implícito que funciona igual que HTTPS. La seguridad se habilita con SSL tan pronto como se inicia la conexión. El puerto FTPS predeterminado es 990. Este protocolo fue la primera versión de FTP encriptado disponible, y aunque se considera obsoleto, todavía se usa ampliamente. Ninguno de los principales navegadores web admite FTPS.

Ejemplo para FileZilla, para Host: ftps //: SERVER-IP

FTPES: (puerto 21)

FTP explícito sobre SSL / TLS. Esto comienza como FTP simple a través del puerto 21, pero a través de comandos FTP especiales se actualiza al cifrado TLS / SSL. Esta actualización generalmente ocurre antes de que las

credenciales de usuario se envíen a través de la conexión. FTPES es una forma algo más nueva de FTP encriptado (aunque todavía tiene más de una década), y se considera la forma preferida de establecer conexiones encriptadas porque puede ser más amigable con el firewall. Ninguno de los principales navegadores web admite FTPES.

SFTP: (puerto 22 o puerto ssh personalizado)

SFTP (Protocolo seguro de transferencia de archivos), SFTP se ejecuta sobre SSH, pero debido a que SFTP se ejecuta sobre SSH también tiene un problema que no puede bloquear a los usuarios en la carpeta de inicio y por eso, nosotros no lo recomendamos para usuarios habituales (los usuarios pueden explorar archivos / carpetas del sistema). Le recomendamos que permita SFTP solo a través del shell `"/usr/libexec/openssh/sftp-server"` o si tiene un cloudlinux con cage-fs, debe usar `/bin/bash`.

FXP: File eXchange Protocol (FXP) es un método de transferencia de datos, a través del cual los datos se envían de un servidor FTP a otro sin pasar por un cliente intermedio. La comunicación convencional FTP consiste en un solo servidor y un solo cliente. Toda la transferencia de datos se realiza entre los dos. Durante una sesión FXP, un cliente mantiene conexiones estándares con dos servidores, dirigiendo cualquiera de los dos servidores que se conecte al otro para iniciar una transferencia de datos. Este método permite a un cliente con poco ancho de banda intercambiar datos entre dos servidores con más ancho de banda sin el retraso asociado con la comunicación convencional FTP. A lo largo de este proceso, sólo el cliente es capaz de acceder a los recursos de los dos servidores.

COMPARACIÓN ENTRE LOS PROTOCOLOS SFTP VS FTP

SFTP

SFTP se confunde a menudo con FTPS y viceversa, aunque estos protocolos no comparten nada en común, excepto su capacidad para transferir archivos de forma segura. SFTP se basa en realidad en el protocolo SSH (Secure Shell), que es más conocido por su uso para proporcionar acceso seguro a cuentas shell en servidores remotos.

EL INTERCAMBIO DE DATOS

A diferencia de FTP/S, SFTP no utiliza comandos y canales de datos separados. Tanto los datos como los comandos se transfieren en paquetes especialmente formateados a través de una sola conexión.

SEGURIDAD

Todos los datos enviados entre el cliente y el servidor se cifran utilizando un cifrado de cifrado acordado. Las sesiones SFTP también pueden protegerse aún más mediante el uso de claves públicas y privadas, que ofrecen una forma alternativa de autenticación conocida como autenticación de clave pública. Esto se puede usar como una alternativa o junto con la forma tradicional de autenticación de nombres de usuario y contraseñas.

FIREWALL

Servidor – Permitir conexiones entrantes en el puerto 22.

Cliente: permite las conexiones salientes al puerto 22.

FTP

El protocolo FTP (Protocolo de transferencia de archivos) ha existido durante bastante tiempo. Se propuso por primera vez en RFC 114, hace más de 40 años, y finalmente evolucionó a RFC 959, que es el estándar que los clientes y servidores de FTP siguen hoy.

EL INTERCAMBIO DE DATOS

El protocolo FTP intercambia datos usando dos canales separados conocidos como el canal de comando y el canal de datos.

El canal de comando normalmente se ejecuta en el puerto del servidor 21 y es responsable de aceptar las conexiones del cliente y de manejar el intercambio de comandos simples entre un cliente FTP y un servidor. Los comandos USER y PASS utilizados para autenticar a un usuario de FTP son ejemplos de comandos que se intercambian en el canal de comandos. El canal de comando permanece abierto hasta

que el cliente envíe el comando QUIT para desconectarse, o el servidor desconecte al cliente debido a inactividad u otra razón.

El canal de datos se ejecuta utilizando puertos temporales a pedido en el servidor (modo pasivo) o en el cliente (modo activo) y es responsable del intercambio de datos en forma de listas de directorios y transferencias de archivos. Los comandos LIST, STOR y RETR utilizados para obtener una lista de directorios del servidor, cargar un archivo y descargar un archivo son ejemplos de comandos (enviados mediante el canal de comandos) que abren un canal de datos. A diferencia del canal de comando que permanece abierto durante toda la sesión de FTP, el canal de datos se cierra una vez que se completa la transferencia de datos. Para manejar transferencias de archivos concurrentes o listados de directorios, se debe utilizar un rango de puertos de canales de datos.

SEGURIDAD

Usando FTP, tanto el comando como los canales de datos no están encriptados. Cualquier dato enviado a través de estos canales puede ser interceptado y leído. Una vulnerabilidad común que aprovecha esta vulnerabilidad en particular es el ataque de hombre en medio (mitm o man in the middle) que usa el envenenamiento ARP y un detector de paquetes.

SCRIPTS

ALTA USUARIOS

```
#!/bin/bash
DIR_APACHE="/var/www/"
GRUPO_SFTP="ftpushers"
SUFIJO_USUARIOS="DAW"
VALOR_INICIAL=201
VALOR_FINAL=220
PASSWORD="paso"

for ((NUM=VALOR_INICIAL;NUM<=VALOR_FINAL;NUM++))
do
    USUARIO=$SUFIJO_USUARIOS$NUM
    echo "Se creará el usuario $USUARIO"
    useradd -G "$GRUPO_SFTP" -m -d "$DIR_APACHE$USUARIO" -g www-data -p "paso" $USUARIO
    echo $USUARIO:$PASSWORD | chpasswd
    chown root:root "$DIR_APACHE$USUARIO"
    #Eliminar permisos de escritura
    chmod 555 "$DIR_APACHE$USUARIO"
    #Crear el directorio public_html
    mkdir "$DIR_APACHE$USUARIO"/public_html
    #ASignar permisos al directorio public_html
    chown "$USUARIO" "$DIR_APACHE$USUARIO"/public_html
    chgrp "www-data" "$DIR_APACHE$USUARIO"/public_html
done
```

BAJA USUARIOS

```
#!/bin/bash
DIR_APACHE="/var/www/"
SUFJO_USUARIOS="DAW"

VALOR_INICIAL=201
VALOR_FINAL=220

for ((NUM=VALOR_INICIAL;NUM<=VALOR_FINAL;NUM++))
do
    USUARIO=$SUFJO_USUARIOS$NUM
    echo "Se borrará el usuario "$USUARIO
    userdel "$USUARIO"
    rm -R "$DIR_APACHE$USUARIO"
done
```

ALTA SITIOS WEB

```
#!/bin/bash
DIR_APACHE="/var/www/"
DOMINIO="alex.local"
SUFJO_USUARIO="DAW"
VALOR_INICIAL=201
VALOR_FINAL=220
PASSWORD="paso"

for ((NUM=VALOR_INICIAL;NUM<=VALOR_FINAL;NUM++))
do
    USUARIO=$SUFJO_USUARIO$NUM
    echo "Se creará el dominio para el usuario $USUARIO"

    printf "%s\n\"
    "<VirtualHost *:80>\"
    "    ServerName $USUARIO.$DOMINIO\"
    "    ServerAlias www.$USUARIO.com\"
    "    ServerAdmin $USUARIO@gmail.com\"
    "    DocumentRoot $DIR_APACHE$USUARIO\"
    "    ErrorLog /var/log/apache2/error$USUARIO.log\"
    "    CustomLog /var/log/apache2/access$USUARIO.log combined\"
    "    ErrorLog $DIR_APACHE$USUARIO/public_html/error.log\"
    "    CustomLog $DIR_APACHE$USUARIO/public_html/access.log combined\"
    "</VirtualHost>\"
    > /etc/apache2/sites-available/"$USUARIO".conf
    a2ensite "$USUARIO".conf
done

service apache2 restart
```

BAJA SITIOS WEB

```
#!/bin/bash
DIR_APACHE="/var/www/"
DOMINIO="alex.local"
SUFJO_USUARIO="DAW"
VALOR_INICIAL=201
VALOR_FINAL=220

for ((NUM=VALOR_INICIAL;NUM<=VALOR_FINAL;NUM++))
do
    USUARIO=$SUFJO_USUARIO$NUM
    echo "Se borrará el dominio para el usuario $USUARIO"
    a2dissite "$USUARIO".conf
    rm /etc/apache2/sites-available/"$USUARIO".conf
    rm /var/log/apache2/access$USUARIO.log
    rm /var/log/apache2/error$USUARIO.log
done
service apache2
restart
```

SCRIPT AUTOMATIZACIÓN DE TAREAS

```
#!/bin/bash
opcion=-1
while [ $opcion -ne 0 ]
do
clear
    echo "*****"
    echo "*****  GESTIÓN DE USUARIOS WEB  *****"
    echo "*****"
    echo "*** 1. Crear usuarios ftp *****"
    echo "*** 2. Borrar usuarios ftp *****"
    echo "*** 3. Crear sitio web *****"
    echo "*** 4. Borrar sitio web *****"
    echo "*** 5. Ver usuarios ftp *****"
    echo "*** 6. Ver sitios activos *****"
    echo "*** 0. Salir *****"
    echo "*****"
    read -p "Introduzca una opcion: " opcion

    case $opcion in
        1)
            echo "Se crearán los usuarios ftp"
            source crearUsuarios.sh
            read -p "Se han creado los usuarios. Pulsa intro para continuar..."
            ;;
        2)
            echo "Se borrarán los usuarios ftp"
            source borrarUsuarios.sh
            read -p "Se han borrado los usuarios. Pulsa intro para continuar..."
```



```
3)      ;;
        echo "Se crearán los sitios web"
        source altaSitio.sh
        read -p "Se han creado los sitios web. Pulsa intro para continuar..."
        ;;
4)      echo "Se borrarán los sitios web"
        source bajaSitio.sh
        read -p "Se han borrado los sitios web. Pulsa intro para continuar..."
        ;;
5)      echo "Estos son todos los usuarios ftp:"
        cat /etc/group | grep ftpusers | cut -d ":" -f 4
        read -p "Pulsa intro para continuar"
        ;;
6)      echo "Estos son los sitios activos:"
        a2ensite
        read -p "Pulsa intro para continuar"

esac
done
```