



Instituto Politécnico Nacional

Escuela Superior de Computo



Administración de Servicios en Red

Prof. Rangel González Josué

Nombre:

Ortega Hernández Jorge

Grupo:

4CM1

Pregunta 7.5:

¿Puede una configuración global implementar cambios en todos los dispositivos en menos de 24 horas? (como el CERT o el PSIRT)

Tabla de contenido

Introducción	3
Red	4
Script	5
Comandos para el control de acceso	6
Comandos de parámetros para la transferencia de datos	7
Comandos para la solicitud de servicios	8
Desempeño	10
Conclusión	15
Bibliografía	15

Introducción

Una red necesitara protección dado que se necesita siempre estar en constante uso con ella por distintas actividades que podemos realizar, como lo es consultar el estado de un evento, investigar para dar a conocer un tema, jugar en línea de ves en cundo, etc. Este constante uso hace que sea necesario protegerla de cualquier tipo de amenaza que pude surgir de maneras infinitas en el mundo digital.

Por lo tanto, necesitamos de alguna forma poder hacer cambios en la red de forma global o particular según sea el caso. Pero para esto tenemos que saber acerca de la red ya que no es lo mismo cambiar una red para una empresa pequeña; que para una empresa grande o para una casa. Lo que necesitamos es información especializada de la red y que amenazas le pueden afectar o no.

También debemos saber qué tipo de amenazas existen en el momento que surgen, pero para ello tendríamos que hacer un grupo de expertos en el descubrimiento de amenazas para que ellos informaran que amenazas surgen y por donde se pueden infiltrar, dado estos requerimientos nos pondríamos a pensar en que costo tendría hacer este grupo y mantener un grupo así, pero para nuestra suerte existen grupos ya hechos que nos pueden prestar este servicio según sus términos como el CERT (Equipo de respuesta ante emergencias informáticas) de la UNAM. Este CERT nos da información de amenazas que podrían llegar a afectar nuestra red de alguna forma y nos dan un periodo de tiempo para corregir nuestra red antes de que ellos hagan públicos las vulnerabilidades que podría tener cualquier red.

Este manual solo ayuda a hacer cambios en los enrutadores RCP100 de una red de forma "global" por decirlo de algún modo, pero esto no afecta de ninguna forma la configuración que se pude tener en una maquina ya que para ello tendríamos que saber qué tipo de servicio se está brindando, si es que está brindando alguno, o tener información especializada para poder cambiar la configuración de la red de forma especializada. Para ello se recomienda tener un grupo como el PSIRT (equipo de respuesta a incidentes de seguridad de productos) que ayuda a responder ante amenazas de red en productos como cisco, pero este PSIRT es de cisco por lo que no ayudaría en este caso por el tipo de producto que estamos ocupando.

Red

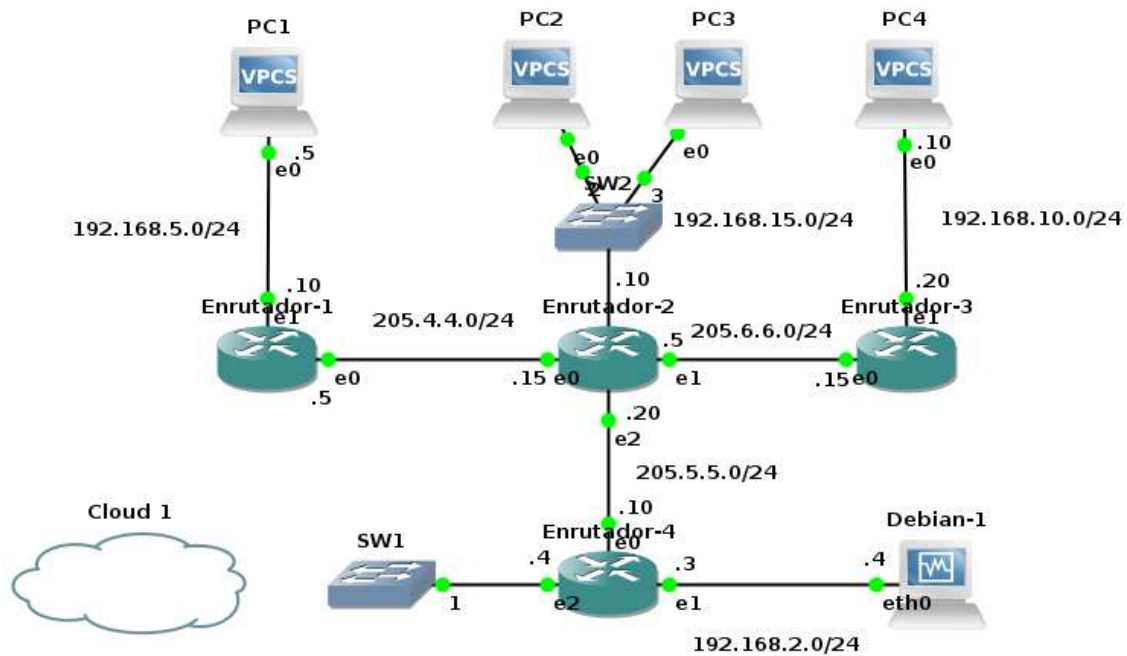


Ilustración 1

Para efectos de este manual, necesitaremos armar la red que se muestra en la Ilustración 1. Se puede hacer la conexión de la red de cualquier forma, para este caso yo la hice rip.

Script

El planteamiento de la solución es mediante un script que puede ejecutarse en cualquier dispositivo que tenga un intérprete Python.

```
1  #!/usr/bin/python
2  from ftplib import *
3  import os
4
5  def obtenStartup(host, usuario, contra):
6      try:
7          ftp = FTP(host, usuario, contra)
8          ftp.retrbinary('RETR startup-config', open('startup-config', 'wb').write)
9      except:
10         print "Error al establecer comunicacion"; exit(0)
11     #print ftp.retrlines("RETR startup-config")
12     return ftp
13
14 def cambiaStartup(listaCambios):
15     startup=open('startup-config', 'r')
16     archivo=startup.read()
17     for cambio in listaCambios:
18         if cambio[0] == 505:
19             archivo=archivo.replace(cambio[1], cambio[2])
20             #archivo=archivo.replace(cambio[1], cambio[2], numOcorrencias)
21         elif cambio[0] == 448:
22             acumulador = ''
23             for campo in archivo.split("!"):
24                 if campo.find(cambio[1]) != -1:
25                     if campo.find(cambio[2]) == -1
26                         campo=campo+cambio[2]+"\\n"
27                     acumulador=acumulador+campo+"!"
28             archivo=acumulador
29         elif cambio[0] == 83:
30             archivo=archivo.replace(cambio[1], '')
31         else:
32             print "No se pudo aplicar el cambio: "+cambio
33     newStartup=open('startup-config', 'w')
34     newStartup.write(archivo)
35
36 def enviaStartup(ftp):
37     #ftp.storbinary('STOR startup-config', open('startup-config', 'rb').read)
38     ftp.storbinary('STOR startup-config', open('startup-config', 'r'))
39
40 def ejecutaComandos(ftp, listaComandos):
41     for comando in listaComandos:
42         resp = ftp.sendcmd(comando)
43         #print resp
44
45 def cambios(path, listaComandos, listaCambios):
46     fRouter = open(path, "r")
47     while True:
48         router=fRouter.readline()
49         if not router:break
50         host,usr,pas=router.split(" ")
51         pas = pas.replace("\\n", '')
52         ftp = obtenStartup(host,usr,pas)
53         print("Se obtuvo el archivo startup-config de ", host)
54         ejecutaComandos(ftp, listaComandos)
55         cambiaStartup(listaCambios)
56         enviaStartup(ftp)
57         ftp.quit()
58     os.remove('startup-config')
59
60 #main
61 cambios("datosServer.data", ["ACCT"],
62        [[505, "network 192.168.2.0/24", "network 192.168.123.0/24"],
63         [448, "service", "service dhcp"],
64         [83, "service tftp"]])
```

Ilustración 2

El script de la Ilustración 2 se anexa con este manual a continuación se explicará cada una de sus funciones.

La línea 2 importa la biblioteca de ftplib que nos ayudara a poder hacer conexiones ftp desde Python. Esta biblioteca es estándar por lo que no se necesita instalarla.

La línea 3 importa un objeto que a través del podremos ejecutar comandos de Linux desde Python.

Función `obtenStartup`: recibe con los argumentos `host` usuario y `contra` se entablará la comunicación ftp y se obtendrá el `startup-config` de ese host. La función nos devuelve el objeto ftp creado.

Función `cambiaStartup`: recibe una lista de cambios para hacerlos de forma directa en el `startup-config` de los cuales hay tres tipos de cambios que se pueden realizar:

- 1) Un remplazo directo: donde se encuentre la línea completa ahí se cambiará por otra. Se necesita: La línea de código que se remplazará y la línea de código suplente.
- 2) Una inserción: donde este la palabra clave, ahí se insertará la línea que se indique. Se necesita la palabra clave y la línea que se insertará.
- 3) Una eliminación: donde se encuentre la línea de código completa, esta se eliminará las veces que aparezca. Se necesita la línea que se eliminara.

Función `enviaStartup`: envía, con ayuda del argumento `ftp`, el archivo `startup-config` y se remplacea en el enrutador.

Función `ejecutaComandos`: envía la lista de comandos ftp al servidor. En la actualidad el protocolo de transferencia de archivos (protocolo FTP) incluye más de treinta comandos que se pueden emplear para manejar los procedimientos de transmisión de archivos. Cada comando FTP se clasifica en una de tres categorías: control de acceso, parámetros para la transferencia de datos, y solicitud de servicios.

Comandos para el control de acceso

Los comandos para el control de acceso identifican al usuario con el servidor FTP o le indican a este a que directorios se desea acceder.

- `USER` (nombre de usuario). Este comando solicita un parámetro que permita identificar al usuario con el servidor.
- `PASS` (contraseña). Después de especificar un nombre de usuario debe especificar una contraseña. Este comando requiere de un parámetro que es la contraseña del usuario.
- `ACCT` (cuenta). Este comando se acompaña de un parámetro que identifique la cuenta del usuario permitiendo así que las cuentas de usuario mantengan un registro contable. Por ejemplo, los empleados de una empresa pueden trabajar en varias áreas, se les puede pedir a los usuarios que escriban un número de cuenta cuando

inicien sesión y usar esta información contable para dar seguimiento a la duración del trabajo.

- CWD (cambiar de directorio). Este comando permite al usuario ir a un directorio diferente del que se encuentre. Se acompaña de un parámetro que especifica la ruta a la cual se accederá.
- CDUP (cambiar a directorio superior). Este comando cambia del directorio actual al siguiente nivel superior.
- SMNT (montaje de estructura). Permite que un usuario monte una estructura de datos para el sistema de archivos. Se acompaña de un parámetro que especifique una ruta al directorio o a algún otro asignador de grupos de archivos.
- REIN (reinicializar). Este comando regresa al cliente al estado que sigue inmediatamente al establecimiento de la conexión de control. Se puede usar este comando para transferir archivos para varios usuarios sin tener que cerrar y reabrir una conexión para cada uno de ellos.
- QUIT (terminar sesión). Se usa para cerrar la sesión FTP.

Comandos de parámetros para la transferencia de datos

En FTP se pueden especificar tipos, formatos y estructuras de archivos, así como modos de transmisión. Estos comandos permiten que el cliente defina estas opciones de FTP por el servidor.

- PORT (puerto de datos). Especifica el puerto que se usará en la transferencia de archivos. Este comando necesita de un parámetro que especifique un puerto de protocolo que se pueda usar en la conexión de datos. Este parámetro es la combinación de una dirección IP y una de puerto TCP de 16 bits. El cliente debe dividir esta información de direcciones en campos de 8 bits y transmitirlos separados por, (coma) y como número decimal.
- PASV (pasivo). Pide al proceso de transferencia de datos del servidor que atienda en un puerto de información que no es su puerto de información preestablecido, y que espere una conexión.
- TYPE (tipo de representación). Indica cómo representar un archivo durante una operación de transferencia de archivos. Hay cuatro tipos de archivos: local, imagen, EBCDIC y ASCII.
- STRU (estructura de archivo). Hay tres tipos de estructuras: archivo, registro y página. Este comando especifica qué estructura utilizar para las operaciones de transferencia de archivos. Requiere un sólo carácter como parámetro: F (archivo, sin estructura), R (estructura de registro) y P (estructura de página).
- MODE (modo de transferencia). Hay tres tipos de modos de transferencia de archivos. Con este comando se especifica qué modo utilizar. Se necesita un sólo carácter como parámetro: S (flujo), B (bloque) y C (comprimido).

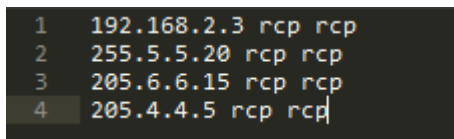
Comandos para la solicitud de servicios

Estos comandos especifican las operaciones de transferencia de archivos que el usuario tiene que ejecutar. El parámetro para un comando de servicio por lo general es una ruta de acceso, las cuales deben adecuarse a las convenciones del servidor FTP.

- RETR (obtener). Se indica al servidor que envíe un archivo a otra computadora anfitrión (generalmente, aunque no de manera necesaria la computadora local del cliente).
- STOR (almacenar). Permite al usuario transmitir un archivo al servidor FTP. Si el archivo destino existe se sobrescribe.
- STOU (almacenar único). Se comporta como STOR excepto que el archivo que crea está en el directorio en uso bajo un nombre único para ese directorio. El código de respuesta del servidor número 250 (que indica que empezó la transferencia) incluye el nombre generado por el servidor.
- APPE (anexar -con crear-). Se comporta como STOR excepto que no sobrescribe un archivo existente. Si el archivo especificado existe en el servidor, entonces éste anexa los datos a ese archivo, si no existe aún entonces lo crea.
- ALLO (asignar). Le indica al servidor FTP que reserve el espacio según el número de bytes indicado para el almacenamiento (con STOR o APPE) de un archivo.
- REST (reiniciar). Cuando se detiene temporalmente una transferencia de archivo, con este comando se le indica al servidor que la reanude.
- RNFR (renombrar desde). Cambia nombre de archivos existentes en el servidor. RNFR especifica el nombre antiguo del archivo (archivo actual) que el usuario quiere cambiar, posteriormente debe usar el comando RNTD.
- RNTD (renombrar a). Es posterior al comando RNFR, especifica el nombre nuevo de un archivo existente.
- ABOR (abortar). Indica al servidor que aborte el comando de servicio anterior y cualquier transferencia de datos asociada que esté en curso.
- DELE (borrar). Se borra el archivo especificado en el parámetro.
- RMD (eliminar directorio). Borra el directorio especificado.
- MKD (crear directorio). Crea el directorio especificado.
- PWD (imprimir directorio de trabajo). Devuelva el nombre del directorio actual.
- LIST (listar). Se imprime una lista de archivos con información de archivo según el directorio indicado.
- NLIST (lista de nombres). Es similar a LIST, el parámetro que se le indique debe especificar un directorio u otro descriptor de grupos de archivos específicos. El servidor sólo devuelve los nombres de archivos y nada más.
- SITE (parámetros del sitio). El servidor FTP muestra sus servicios (comandos) personalizados.
- SYST (sistema). Determina el tipo de sistema operativo del anfitrión remoto.

- STAT (estado). Hace que el servidor envíe una respuesta del estado en el que se encuentra.
- HELP (ayuda). Hace que el servidor FTP envíe información de ayuda de un comando específico.
- NOOP (sin operación). Este comando no afecta ningún parámetro o comandos ejecutados antes, no especifica ninguna acción, sólo que el servidor envíe un OK por respuesta.

Función cambios: es la función principal la cual recibe una lista de comandos, una lista de cambios y una ruta (path) la cual tiene los datos necesarios para acceder vía ftp como se muestra en la Ilustración 3.



```

1 192.168.2.3 rcp rcp
2 255.5.5.20 rcp rcp
3 205.6.6.15 rcp rcp
4 205.4.4.5 rcp rcp

```

Ilustración 3

Por último, en las líneas 61 a 64 tenemos la llamada a la función principal con el archivo datosServer.data que contiene lo que se puede ver en la Ilustración 3, también una lista con un comando ftp para saber el estado de la comunicación ftp y por último hacemos tres cambios. El primero es un cambio tipo 1 donde la red 192.168.2.0 se cambia por 192.168.123.0, el segundo cambio es de tipo 2 donde se inserta la línea "service dhcp" en la parte del documento donde este la palabra clave "service" y el tercer cambio es la eliminación de un servicio tftp. Estos cambios se llevarán a cabo en todos los hosts que existen en el archivo datosServer.data.

Desempeño

```
root@Debian-1:~# ifconfig eth0 192.168.2.4/24 up
root@Debian-1:~# route add default gw 192.168.2.3
root@Debian-1:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1a:1a:9c
          inet addr:192.168.2.4  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1a:1a9c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:210 (210.0 B)  TX bytes:818 (818.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@Debian-1:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.2.3    0.0.0.0         UG    0      0        0 eth0
192.168.2.0     *               255.255.255.0   U     0      0        0 eth0
```

Ilustración 4

Bueno ya tendiendo el script en la maquina debían 1 y esta tenga conexión con toda la red lo que necesitamos es un ejemplo del antes y después de un archivo startup-config. En este caso tomamos el archivo del Enrutador-4 el cual se muestra en la Ilustración 5.

```

service telnet
service http encrypted password VJLWNTK32f63955c9c99393574b4f6cbb4219a312
service ftp
|
administrator rcp encrypted password UYTKGCONsbTpwAyuF0TKz29k6aRdh0
|
|
|
router rip
network 205.5.5.0/24
network 192.168.2.0/24
network 192.168.1.0/24
|
interface loopback lo
ip address 127.0.0.1/8
ip mtu 15436
|
interface ethernet eth0
ip address 209.5.5.10/24
ip mtu 1500
no shutdown
|
interface ethernet eth1
ip address 192.168.2.3/24
ip mtu 1500
no shutdown
|
interface ethernet eth2
ip address 192.168.1.4/24
ip mtu 1500
no shutdown
|
interface ethernet eth3
ip mtu 1500
shutdown
|
interface ethernet eth4
ip mtu 1500
shutdown
|
interface ethernet eth5
ip mtu 1500
shutdown
|
interface ethernet eth6
ip mtu 1500

```

Ilustración 5

Como podemos observar tenemos la red 192.168.2.0.

```
root@Debian-1:~# ./script.py
('Se obtuvo el archivo startup-config de ', '192.168.2.3')
('Se obtuvo el archivo startup-config de ', '205.5.5.20')
('Se obtuvo el archivo startup-config de ', '205.6.6.15')
('Se obtuvo el archivo startup-config de ', '205.4.4.5')
root@Debian-1:~#
```

Ilustración 6

Ejecutamos el script como se indica en la Ilustración 6. Para hacerlo antes se tuvo que modificar sus permisos de ejecución con el comando `chmod +x`.

```

hostname rcp
|
service telnet
service http encrypted password VJLWANTK$2f83565c9c99393674b4fccb4219a312
service ftp
service dhcp
|
administrator rcp encrypted password UYYKWGON$bTpuYNgvFQTKz29k6sAdh0
|
router rip
|
network 205.5.5.0/24
network 192.168.123.0/24
network 192.168.1.0/24
|
interface loopback lo
ip address 127.0.0.1/8
ip mtu 16436
|
interface ethernet eth0
ip address 205.5.5.10/24
ip mtu 1500
no shutdown
|
interface ethernet eth1
ip address 192.168.2.3/24
ip mtu 1500
no shutdown
|
interface ethernet eth2
ip address 192.168.1.4/24
ip mtu 1500
no shutdown
|
interface ethernet eth3
ip mtu 1500
shutdown
|
interface ethernet eth4
ip mtu 1500
shutdown
|
interface ethernet eth5
ip mtu 1500
shutdown

```

Ilustración 7

Después de ejecutar el script notaremos los cambios que se muestran en la Ilustración 7 los cuales fueron dos, en la sección de servicios se agregó el servicio dhcp y la red 192.68.2.0 se reemplazó por 192.168.123.0.

```

hostname rcp
!
service telnet
service http encrypted password VJLMANTK$2f83565c9c99393674b4fccb4219a3
service ftp
!
administrator rcp encrypted password UYYKUGON$btPuyNjvFQTKz29k6sAdh0
!
service dhcp
ip dhcp server

router rip
 network 205.5.5.0/24
 network 192.168.123.0/24
 network 192.168.1.0/24
!
interface loopback lo
 ip address 127.0.0.1/8
 ip mtu 15436
!
interface ethernet eth0
 ip address 205.5.5.16/24
 ip mtu 1500
 no shutdown
!
interface ethernet eth1
 ip address 192.168.2.3/24
 ip mtu 1500
 no shutdown
!
interface ethernet eth2
 ip address 192.168.1.4/24
 ip mtu 1500
 no shutdown
!
interface ethernet eth3
 ip mtu 1500
 shutdown
!
interface ethernet eth4
 ip mtu 1500
 shutdown
!
interface ethernet eth5
 ip mtu 1500
 shutdown

```

Ilustración 8

Después si hacemos un reinicio en el Enrutador-4 para que surtan efecto los cambios, tendremos como resultado de la Ilustración 8 donde el servicio dhcp desapareció y en su lugar apareció un nuevo campo señalado en la ilustración 8. Esto se debe a operaciones que realiza el enrutador y que pueden afectar de alguna forma la manipulación de los datos de forma general. Por esta razón no se hizo por campo los cambios ya que esto puede variar dependiendo de la configuración de cada uno de los enrutadores.


```

#configure rcp
#
service telnet
service http
service ftp
service dhcp
#
administrator rcp encrypted password EBJC.NH08h0d0714o4Lg40NljnBqj11
#
#
#
router rip
 network 192.168.5.0/24
 network 205.4.4.0/24
#
interface loopback lo
 ip address 127.0.0.1/8
 ip mtu 16436
#
interface ethernet eth0
 ip address 205.4.4.5/24
 ip mtu 1500
 no shutdown
#
interface ethernet eth1
 ip address 192.168.5.10/24
 ip mtu 1500
 no shutdown
#
interface ethernet eth2
 ip mtu 1500
 shutdown
#
interface ethernet eth3
 ip mtu 1500
 shutdown
#
interface ethernet eth4
 ip mtu 1500
 shutdown
#
interface ethernet eth5
 ip mtu 1500
 shutdown
#
interface ethernet eth6
 ip mtu 1500
 shutdown
#

```

Ilustración 9

Por último, veamos qué cambios se realizaron en otro enrutador, en este caso el enrutador-1 que tiene el archivo de configuración presentado en la Ilustración 9 después de haber ejecutado el script por lo que se agregó el servicio dhcp, pero no se cambió la red 192.168.2.0 ya que no existe y tampoco se eliminó el servicio tftp porque tampoco existía.

```
!
service telnet
service http
service ftp
administrator rcp encrypted password EBJCJNHQ9rh0dDYI4o4Lg40HuJnDqj1!
!
!
service dhcp
ip dhcp server
!
router rip
network 192.168.5.0/24
network 205.4.4.0/24
!
interface loopback lo
 ip address 127.0.0.1/8
 ip mtu 15436
!
interface ethernet eth0
 ip address 205.4.4.5/24
 ip mtu 1500
 no shutdown
!
interface ethernet eth1
 ip address 192.168.5.15/24
 ip mtu 1500
 no shutdown
      I
interface ethernet eth2
 ip mtu 1500
 shutdown
!
interface ethernet eth3
 ip mtu 1500
 shutdown
!
interface ethernet eth4
 ip mtu 1500
 shutdown
!
interface ethernet eth5
 ip mtu 1500
 shutdown
!
interface ethernet eth6
```

Ilustración 10

En la Ilustración 10 nos muestra cómo queda el archivo de configuración una vez reiniciado el enrutador, el cual también tiene el cambio en el servicio dhcp el cual pasaba en el enrutador-4.

Conclusión

Dado que no siempre todas las redes tienen la misma topología y tampoco tienen la misma configuración cada uno de los enrutadores, entonces la solución presentada tendría que cambiarse un poco para funcionar en la red que se necesite cambiar, además de que se tiene que reiniciar de forma manual cada uno de los enrutadores porque no existe alguna forma de reiniciarlos de forma automática por ftp.

Además, si la red es muy grande, se tendría que implementar de forma distribuida el script y de forma no invasiva, etc.

Bibliografía

<http://www.secayo.com/blog/2013/comandos-ftp-y-sus-definiciones.asp>