Construcción de una Red Empresarial

Equipo 2

Instalación de servidor de syslog

Para tener un almacenamiento de bitácoras en Linux y de recepción de logs de los equipos cisco, fue necesario instalar un servidor de syslog en nuestros servidores de Ubuntu. Para ello, debido a que existen un par de versiones ya obsoletas, elegimos "rsyslog", una versión vigente y un poco más sencilla de usar que algunas anteriores.

El procedimiento para la instalación es muy sencillo. Sólo debemos ingresar lo siguiente en la línea de comandos y el proceso de instalación dará inicio:

>sudo apt-get install rsyslog

```
root@a01114070:~# sudo apt-get install rsyslog
Reading package lists... Done
Building dependency tree
Reading state information... Done
rsyslog is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

Posterior a la instalación es necesario configurar nuestros servidor de syslog; para ello, hay que editar el archivo en /etc/rsyslog.conf (pero esto será explicado más adelante).

Instalación servicio de netflow

Netflow es un potente protocolo de red desarrollado por Cisco para recolectar información acerca del tráfico dentro de una red, su importancia en el mercado lo ha llevado a convertirse en estándar y distintas empresas se han dedicado a desarrollar software que trabaje de la mano con este protocolo para monitorear el tráfico dentro de la misma.

Como se mencionó, existen distintos programas que trabajan con Netflow; al principio habíamos elegido SolarWinds, pero debido a incopatibilidades con el sistema operativo, optamos por usar Netflow Analyzer de Manage Engine.

Netflow Analyzer es muy sencillo de utilizar; una vez instalado desde su página oficial, es cuestión de configurar el puerto de escucha y conectarnos a la red para comenzar a monitorear el tráfico.

Instalación servidor de tftp

La instalación de un servidor de tftp es de gran ayuda para guardar las configuraciones de los equipos cisco y tener un respaldo en caso de que algún inconveniente suceda.

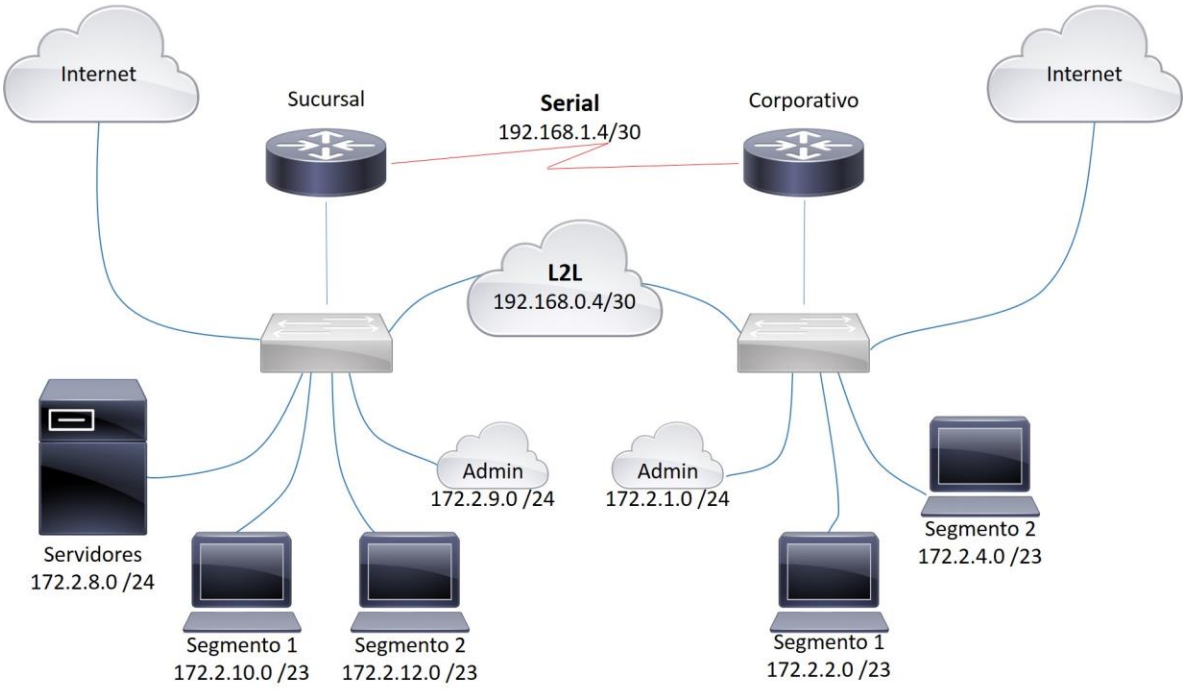
Para su instalación es necesario ingresar el siguiente comando en nuestro servidor:

>sudo apt-get install tftpd-hpa

```
root@a01114070:~# sudo apt-get install tftpd-hpa
Reading package lists... Done
Building dependency tree
Reading state information... Done
tftpd-hpa is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

Tras haber finalizado la instalación es necesario comenzar con la configuración, pero esto será explicado más adelante.

Explicación y direccionamiento topología.



Para el diseño de nuestra red empresarial se consideraron dos localidades (Corporativo y Sucursal) interconectadas por medio de un enlace L2L con respaldo por un enlace Serial.

Cada una de las localidades cuenta con dos segmentos de 500 usuarios y un segmento para administración de dispositivos; también se cuenta con salida a Internet localmente, que en caso de fallos debe tener redundancia por el enlace con la otra localidad.

Además, se cuenta con un segmento de servidores en la localidad de Sucursal, brindando a ambas localidades servicio de DNS y DDNS, entrega de direcciones por DHCP, respaldo de configuraciones por TFTP, almacenamiento de bitácoras y recepción de logs con SYSLOG, servicio de páginas web con APACHE2 y monitoreo de la red para detección de anomalías por medio de NETFLOW.

Resolución de problemas, ¿qué ajustes y diagnósticos se hicieron para que funcionara el escenario?

- El contrab visualiza los logs del sistema en la web, sin embargo, también imprimía cuando el crontab era ejecutado y esto no debía suceder. Resolvimos nuestro problema al modificar la configuración del rsyslog en la ruta /etc/rsyslog.d/50-default.conf, ahí editamos una línea que estaba como: *.*;auth,authpriv.none -/var/log/syslog y lo cambiamos por: *.*;auth,authpriv.none;cron.none -/var/log/syslog, lo que nos permitió tener la visualización correcta de los logs del sistema.

- Salida a internet: quisimos conectarnos a internet por medio de los cables de ethernet de las computadoras del laboratorio de redes conectándolos a los sockets de la mesa y utilizando el reflejo para conectarlo al switch. Sin embargo, no sabíamos que estos cables tienen configurado un port security que al detectar multiples direcciones MAC cierra la conexión. Tuvimos que esperar a que el profesor nos asignara un reflejo de internet en los racks sin esta configuración.

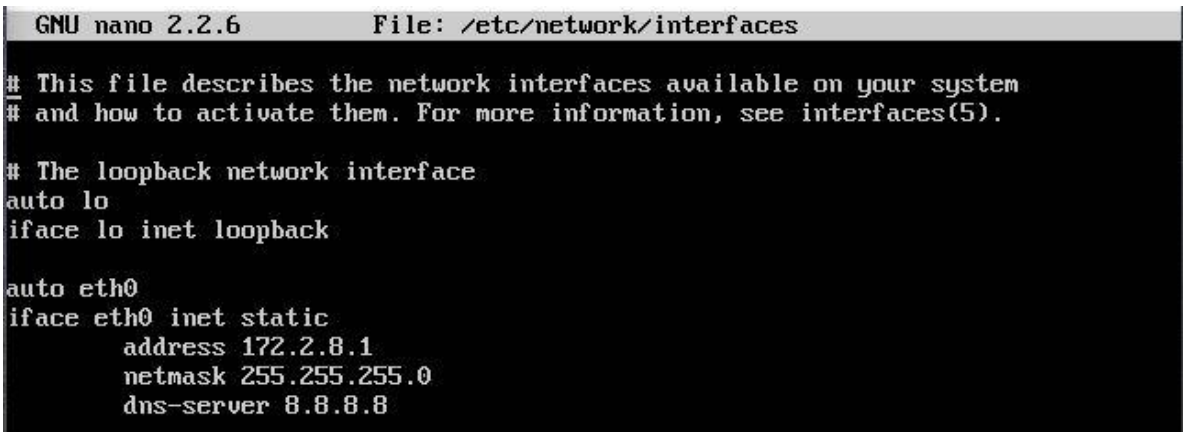
- Salida a internet: no se había asignado la configuración de acceso a la vlan 7(salida a internet) en el switch, por lo que el switch no le tenía ningun puerto mapeado a la vlan 7.

Apartado de configuraciones:

Configuración del syslog

El primer punto a considerar es la configuración de una dirección estática en nuestro servidor, esto para que su ip no esté cambiando y, así, tener control absoluto sobre el mismo:

>nano /etc/network/interfaces



Una vez que nos aseguramos que tenemos una dirección ip estática y, además, que tenemos comunicación con el resto de la red, procedemos con la configuración de nuestro servidor de tftp.

Para habilitar la recepción de mensajes de syslog en nuestro servidor es necesario modificar el siguiente archivo, así como descomentar las líneas que hacen referencia a TCP y UDP, que habilitan los puertos 514 en escucha.

>nano /etc/rsyslog.conf

```
 GNU nano 2.2.6          File: /etc/rsyslog.conf

#  /etc/rsyslog.conf    Configuration file for rsyslog.
#
#                       For more information see
#                       /usr/share/doc/rsyslog-doc/html/rsyslog_conf.html
#
#  Default logging rules can be found in /etc/rsyslog.d/50-default.conf


##################
#### MODULES ####
##################

$ModLoad imuxsock # provides support for local system logging
$ModLoad imklog   # provides kernel logging support
#$ModLoad immark  # provides --MARK-- message capability

# provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514

# provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514
```

Una vez que los cambios se han llevado a cabo, sólo tenemos que reiniciar el servicio y el servidor estará listo para recibir logs:

```
root@a01114070:~# service rsyslog restart
rsyslog stop/waiting
rsyslog start/running, process 1186
```

Ahora el servidor ya está listo para recibir logs. El siguiente paso es configurar los dispositivos CISCO para que manden los logs a nuestro servidor, los comandos aplican de igual manera en Switches y Routers.

Router(config)#logging on host 172.2.8.1

Router(config)#logging trap 7

Si hacemos un "show running-config" desde el dispositivo, la parte correspondiente a la configuración de los logs debe lucir como a continuación:

```
logging trap debugging
logging 172.2.8.1
```

Aquí hemos finalizado, nuestro servidor ya escucha y guarda los logs de todos nuestros dispositivos CISCO.

En nuestro trabajo, también creamos páginas web mediante nuestro servidor de APACHE para mostrar dichos logs; para ello, agregamos cierta configuración por medio de CRONTAB para que los logs se actualicen cada minuto.

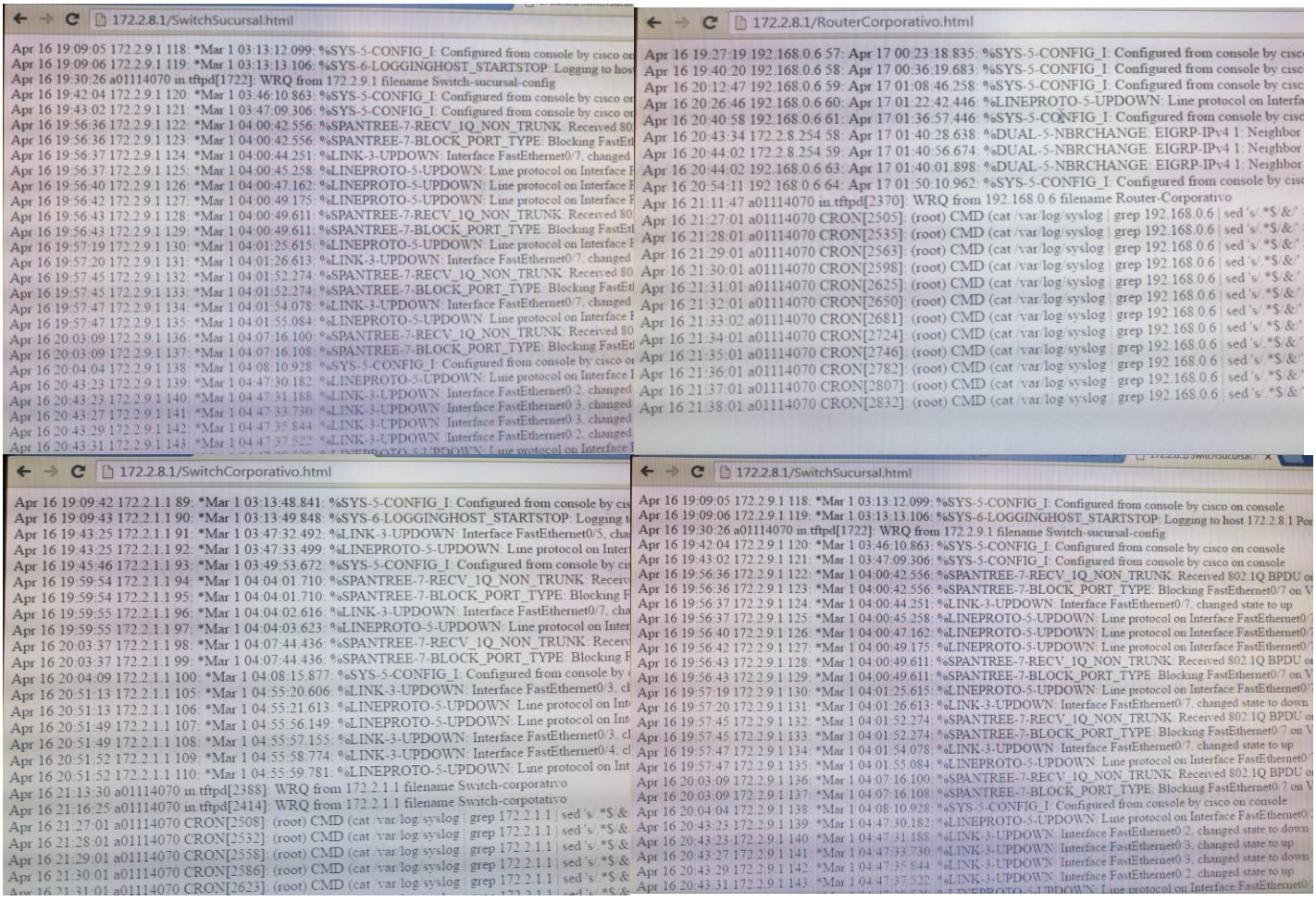
El archivo de configuración de CRONTAB luce así:

```
* * * * * cat /var/log/syslog | grep 172.2.9.1 | sed 's/.*$/&<br\/>/' > /var/ww$
* * * * * cat /var/log/syslog | grep 172.2.1.1 | sed 's/.*$/&<br\/>/' > /var/ww$
* * * * * cat /var/log/syslog | grep 172.2.8.254 | sed 's/.*$/&<br\/>/' > /var/$
* * * * * cat /var/log/syslog | grep 192.168.0.6 | sed 's/.*$/&<br\/>/' > /var/$
```

En la imagen no se alcanzan a apreciar todos los datos, por lo que sólo aparece la ruta hacia la página de nuestro APACHE, donde se guarda toda la información.

El comando es muy sencillo, los asteriscos significa que se actualice cada minuto, de cada hora, de cada semana, de cada mes, de cada año; y lo siguiente es listar los logs, pero como este archivo es demasiado grande, es necesario filtrarlos mediante el comando grep, viendo solo las ip que nos interesan, es decir, las ip de nuestros Routers y Switches CISCO.

Las páginas que muestran los logs de los Routers y Switches lucen así:



Configuración del netflow.

La configuración por parte del servidor es muy sencilla. Al ser instalado en Windows tenemos interface gráfica, y al correrlo por primera vez nos pregunta por cuál puerto deseamos ponerlo en escucha; el programa viene por default con el puerto número 9996 habilitado, y fue así como lo dejamos nosotros para nuestra red.

Por parte de los dispositivos CISCO, es necesario ingresar cierta configuración para permitir mandar el flujo del tráfico hacia nuestro servidor. La configuración se realiza de la siguiente manera:

Router(config)#ip flow-export source GigabitEthernet 0/0.5

//donde la interface es el lugar donde se encuentra nuestro servidor, así se evita el procesamiento de buscarla

Router(config)#ip flow-export version 5

Router(config)#ip flow-export destination 172.2.8.2 9996

//donde 172.2.8.1 es la dirección de nuestro servidor y 9996 el puerto configurado para escuchar.

Además, es necesario ingresar, en cada una de las interfaces que deseamos monitorear, el siguiente comando:

Router(config-if)#ip flow ingress

En nuestro caso, debido a que contamos con VLANS, podemos ingresar el siguiente comando sólo en la interface física, y se configurará automáticamente en cada una de las interfaces virtuales:

Router(config-if)#ip route-cache flow

Si vemos nuestra running-config debe lucir algo así:

```
ip flow-export source GigabitEthernet0/0.5
ip flow-export version 5
ip flow-export destination 172.2.8.2 9996
```

También, cada una de nuestras interfaces por monitorear debe mostrar la siguiente información así:

```
 bandwidth 20000
 encapsulation dot1Q 2
 ip address 192.168.0.5 255.255.255.252
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 mykey
 ip flow ingress
 ip nat inside
 ip virtual-reassembly in
!
interface GigabitEthernet0/0.3
 description SEGMENTO 1
 encapsulation dot1Q 3
 ip address 172.2.11.254 255.255.254.0
 ip helper-address 172.2.8.1
 ip flow ingress
 ip nat inside
 ip virtual-reassembly in
!
interface GigabitEthernet0/0.4
 description SEGMENTO 2
 encapsulation dot1Q 4
 ip address 172.2.13.254 255.255.254.0
 ip helper-address 172.2.8.1
 ip flow ingress
 ip nat inside
 ip virtual-reassembly in
!
```

Configuración de tftp

Para configurar nuestro servidor de tftp, es necesario modificar el siguiente archivo:

>nano /etc/default/tftpd-hpa

Después de haber modificado el directorio donde deseamos guardar las configuraciones de nuestros dispositivos CISCO, el archivo aparecerá de la siguiente forma:

```
  GNU nano 2.2.6               File: tftpd-hpa

# /etc/default/tftpd-hpa

TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/var/tftpboot/cisco"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure --create -v"
```

Es necesario crear la siguiente carpeta y, además, brindarle permisos de administrador:

```
mkdir /var/tftpboot/cisco

chmod 777 /car/tftpboot/cisco
```

Una vez realizado esto, nuestro servidor se encuentra en las condiciones correctas para guardar configuraciones.

Si deseamos guardar la configuración, desde nuestro dispositivo CISCO, el comando es el siguiente:

```
Router#copy running-config tftp:
```

Posteriormente, preguntará la dirección ip destino y el nombre con el que se desea guardar el archivo de configuración.

Para cargar la configuración, el proceso es muy parecido:

```
Router#copy tftp: running-config
```

De igual manera, preguntará la dirección ip fuente y el nombre del archivo a cargar.

Configuración de routers y switches (Para más detalles revisar Anexos)

Protocolos de Ruteo

Para nuestra red empresarial, configuramos dos protocolos de ruteo conviviendo: EIGRP y OSPF.

Para la parte de EIGRP se declara un Sistema Autónomo entre 1 y 65535. Este Sistema tiene que coincidir en ambos ruteadores para lograr la mutua comunicación.

Posteriormente, se declaran redes comunes y redes que deseamos compartir (si se omite la Wildcard, el protocolo toma la clasful inmediata).

Esta es la configuración básica para que el protocolo entre en funcionamiento, sin embargo, para añadir seguridad a nuestra red, se configuraron también interfaces pasivas, lo que deja como no pasivas las interfaces por donde deseamos compartir información, y así evitamos que lleguen paquetes con información de ruteo en lugares no deseados.

Otro punto importante fue la configuración de llaves, lo que solo permite a los ruteadores con llave comunicarse y compartir información de ruteo.

Para configurar una llave los comandos son los siguientes:

Router(config)#key chain mykey

Router(config-keychain)#key

Router(config-keychain-key)#key-string password

La llave lucirá así en la configuración:

```
key chain mykey
 key 1
  key-string 7 1404071B0916272A2A
crypto pki token default removal timeout 0
!
```

Una vez creada la llave, será necesario aplicarla en las interfaces por donde se va a hablar eigrp, para ello se ingresamos los siguientes comandos:

Router(config)#ip authentication mode eigrp 1 md5

Router(config)#ip authentication key-chain eigrp 1 mykey

La configuración de OSPF es muy parecida. Una de las diferencias es que, en lugar de declarar un Sistema Administrativo, se declara un solo process ID. Contrariamente a EIGRP, donde ahí sí puede haber más de un Sistema Administrativo configurado, en OSPF sólo puede existir un process ID configurado, puesto que la división se hace por medio de áreas, y el área 0 es el backbone. En nuestra pequeña red, ambos ruteadores serán configurados en el área 0.

La declaración de interfaces pasivas, redes comunes, redes a compartir y llaves es muy similar.

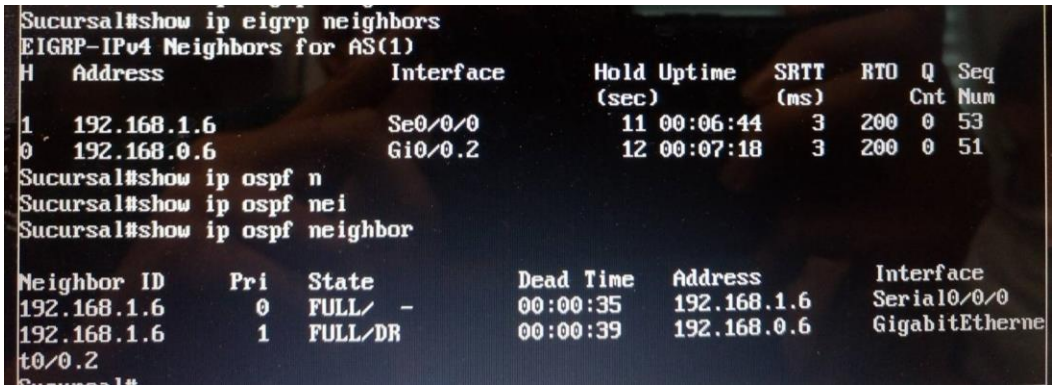
La configuración es la siguiente:

```
!
router eigrp 1
 network 172.2.0.0
 network 192.168.0.0
 network 192.168.1.0
 passive-interface default
 no passive-interface GigabitEthernet0/0.2
 no passive-interface Serial0/0/0
!
router ospf 1
 passive-interface default
 no passive-interface GigabitEthernet0/0.2
 no passive-interface Serial0/0/0
 network 172.2.0.0 0.0.255.255 area 0
 network 192.168.0.4 0.0.0.3 area 0
 network 192.168.1.4 0.0.0.3 area 0
 area 0 autenthication message-digest
!
```

Y las interfaces por donde se hablarán los protocolos muestran datos de la siguiente manera:

```
interface GigabitEthernet0/0.2
 description L2L
 bandwidth 20000
 encapsulation dot1Q 2
 ip address 192.168.0.5 255.255.255.252
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 mykey
 ip ospf authentication
 ip ospf authentication-key password
 ip flow ingress
 ip nat inside
 ip virtual-reassembly in
!
```

Si mostramos los vecinos de los protocolos aparecerá algo como lo siguiente:



Debido a que EIGRP tiene menor distancia administrativa, este será el protocolo que operará en nuestra red. OSPF no entrará en funcionamiento a menos de que el protocolo de EIGRP haya caído o se encuentre inhabilitado.

Salida a Internet

Los pasos para salir a Internet son los siguientes:

1-Asignar las direcciones IP a las interfaces (la interface que tiene salida a Internet se le deberá asignar por DHCP).

```
interface GigabitEthernet0/0.7
 description INTERNET
 encapsulation dot1Q 7
 ip address dhcp
 ip flow ingress
 ip nat outside
 ip virtual-reassembly in
!
```

2-Crear un ACL (Access Control List) en donde seleccionamos el rango de direcciones IP que saldrán a Internet.

```
access-list 2 permit 172.2.0.0 0.0.255.255
access-list 2 permit 192.168.1.4 0.0.0.3
access-list 2 permit 192.168.0.4 0.0.0.3
!
```

3-Crear regla de NAT (Network Address Translation) que permitirá la traducción de direcciones de la lista que seleccionamos en el ACL a una IP válida asignada por nuestro ISP (Internet service provider)

Además, debemos hacer NAT a nuestras páginas de Internet para que puedan ser accedidas desde fuera. También se debe hacer PAT en la red del campus para poder ser vistos desde Internet.

```
ip nat inside source list 2 interface GigabitEthernet0/0.7 overload
ip nat inside source static tcp 172.2.8.1 80 interface GigabitEthernet0/0.7 80
ip nat inside source static tcp 172.2.8.1 8080 interface GigabitEthernet0/0.7 8
80
ip nat inside source static tcp 172.2.8.2 80 interface GigabitEthernet0/0.7 81
```

4-Aplicar las reglas de nateo en cada interfaz. Se debe aplicar el siguiente comando en cada una de las interfaces donde queramos Internet

Router(config-if)#ip nat inside

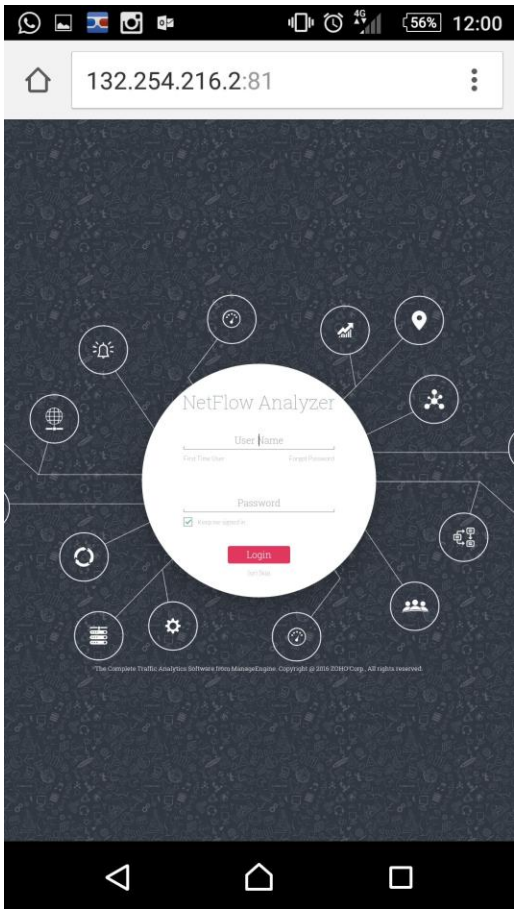
También, en la interface que sale a Iinternet se debe aplicar el siguiente comando:

Router(config-if)#ip nat outside

5-Crear una ruta por defecto para la salida a Internet y redundancia, esto para el caso en que el Internet local caiga y se deba salir a Internet por el enlace L2L.

```
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0.7
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0.2 10
```

Si todo se hizo correctamente, tendremos Internet en las interfaces donde se configuró el NAT inside, además de que nuestras páginas deben poder ser accedidas desde la red del campus o desde una red pública, por ejemplo, desde nuestro celular usando datos móviles:



Seguridad

Vlans(no cdp): utilizamos vlans en nuestra red empresarial para separar los clusters sensibles del resto de la red, así pues tenemos segmentos de red dedicados a administración (vlan 6), servidores (vlan 5), usuarios segmentos 1 y 2 (vlan 3 y 4) y salida a internet (vlan 7). Los segmentos de usuarios tienen configurado el comando no cdp enable para evitar compartir la información sobre otros equipos Cisco directamente conectados.

Listas de acceso (telnet, virus, netflow): Se configuraron listas de acceso para permitir la conexión por medio de telnet unicamente a los usuarios que se encuentren en el segmento de red de servidores(Vlan 5), también se creó una lista de acceso diseñada para bloquear una dirección ip especifica por la que el profesor intentó atacar nuestra red.

```
Extended IP access list 100
    10 deny udp host 172.2.4.21 host 1.1.1.1 eq domain (19364 matches)
    20 permit ip any any (10660 matches)
```

Como se aprecia en la imagen, si mostramos la access list aplicada podemos observar el número de match que ha hecho la línea que bloquea a la ip del host que está intentando sobrecargar la red.

SrcIf	SrcIPaddress	DstIf	DstIPaddress	Pr	SrcP	DstP	Pkts
Total:	6434	0.4	83 1136 35.9		12.3		13.9
SrcIf	SrcIPaddress	DstIf	DstIPaddress	Pr	SrcP	DstP	Pkts
Gi0/0.2	172.2.8.2	Gi0/0.7	65.52.108.212	06	1CDC	01BB	2
Gi0/0.2	172.2.8.1	Local	172.2.0.254	06	839E	0017	64
Gi0/0.4	172.2.4.21	Null	1.1.1.1	11	0035	0035	72K
Gi0/0.4	172.2.4.21	Null	1.1.1.1	11	0000	0000	528K
Se0/0/0	192.168.1.5	Null	224.0.0.5	59	0000	0000	59
Gi0/0.2	192.168.0.5	Null	224.0.0.10	58	0000	0000	24
Gi0/0.4	172.2.4.21	Gi0/0.7	172.17.16.4	11	C000	00A1	3
Gi0/0.4	172.2.4.21	Gi0/0.7	172.16.0.2	11	C000	00A1	3
Gi0/0.4	172.2.4.21	Gi0/0.7	172.18.16.3	11	C000	00A1	3
Se0/0/0	192.168.1.5	Null	224.0.0.10	58	0000	0000	123
Gi0/0.2	192.168.0.5	Null	224.0.0.5	59	0000	0000	10
Gi0/0.7	65.52.108.212	Gi0/0.2	10.40.72.84	06	01BB	1CDC	1
Gi0/0.4	172.2.4.21	Gi0/0.7	10.40.19.65	11	C000	00A1	3

Mandando a la ruta NULL todo lo proveniente de la IP detectada y configurada en la access list.

Llaves eigrp y ospf: los protocolos de ruteo solo comparten información con equipos que tienen configurada la misma llave de autenticación, mykey en el protocolo eigrp y superman en el caso de ospf.

Configuración de dhcpd.conf, DNS y DDNS

Para la configuración de la entrega de direcciones de DHCP con nuestro servidor LINUX es necesario editar el archivo de configuración en /etc/dhcpd.conf.

Se creó un pool de direcciones para cada segmento de hosts en los que se desea dar servicio, además se debe crear un pool para la red en la que se encuentra nuestro servidor aún cuando no se desee dar direcciones en esta red.

Una buena costumbre al declarar pools de DHCP es dejar cierto rango disponible de direcciones para futuro uso, es por ello que para nuestro servidor de DHCP se dejaron 20 direcciones apartadas en cada pool.

La configuración es la siguiente:

```
  GNU nano 2.2.6            File: /etc/dhcpd.conf

ddns-update-style none;
ddns-updates off;
default-lease-time 300;
max-lease-time 300;
server-name "equipo2.dhcp1.mx";

option netbios-node-type 8;
option netbios-name-servers 192.168.1.5;
option domain-name "dhcp1.mx";
option domain-name-servers 8.8.8.8 , 8.8.4.4;

shared-network Sucursal_Server{
        subnet 172.2.8.0 netmask 255.255.255.0 {
                option broadcast-address 172.2.8.255;
                option routers 172.2.8.254;
        }
}

shared-network Sucursal_Segmento1{
        subnet 172.2.10.0 netmask 255.255.254.0 {
                range 172.2.10.20 172.2.11.253;
                option broadcast-address 172.2.11.255;
                option routers 172.2.11.254;
        }
}
```

```
 GNU nano 2.2.6            File: /etc/dhcpd.conf

shared-network Sucursal_Segmento2{
        subnet 172.2.12.0 netmask 255.255.254.0 {
                range 172.2.12.20 172.2.13.253;
                option broadcast-address 172.2.13.255;
                option routers 172.2.13.254;
        }
}

shared-network Corporativo_segmento2{
        subnet 172.2.4.0 netmask 255.255.254.0{
                range 172.2.4.20 172.2.5.253;
                option broadcast-address 172.2.5.253;
                option routers 172.2.5.254;
        }
}
```

Una vez configurado nuestro servidor de DHCP, también es necesario configurar el Router. En el Router se debe colocar un helper address con la dirección ip de nuestro servidor en cada una de las interfaces que van conectadas a los segmentos de host, así, cuando un equipo dentro de esta red realice su proceso de DORA, sabrá a dónde dirigirse (servidor de DHCP) y entonces obtener una dirección ip.

Para poder instalar nuestro servidor de DNS, comenzaremos por conseguir el paquete de instalación. Para ello tendremos que realizar un apt-get install bind9:

```
root@cande:~# apt-get install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
bind9 is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 66 not upgraded.
```

Seguido de esto, tenemos que editar y crear una nueva DB para la zona de dominio en la ruta /etc/bind/named.conf.local:

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "equipo2.com" {
        type master;
        file "/etc/bind/zones/db.equipo2.com";
};
```

Más tarde, debemos crear un directorio donde estarán las zonas de dominio. Esto lo lograremos por medio de un mkdir /etc/bind/zones, para asignar los nombres de las zonas y poner registros A y CNAME en la dirección /etc/bind/zones/db.equipo2.com:

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     equipo2.com. root.equipo2.com. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;

        NS      dns.equipo2.com.
@       A       172.2.0.1
dns     A       172.2.0.1
equipo2   A       172.2.0.3
equipo22    A       172.16.0.4
www     CNAME   dns
redes CNAME   equipo2
```

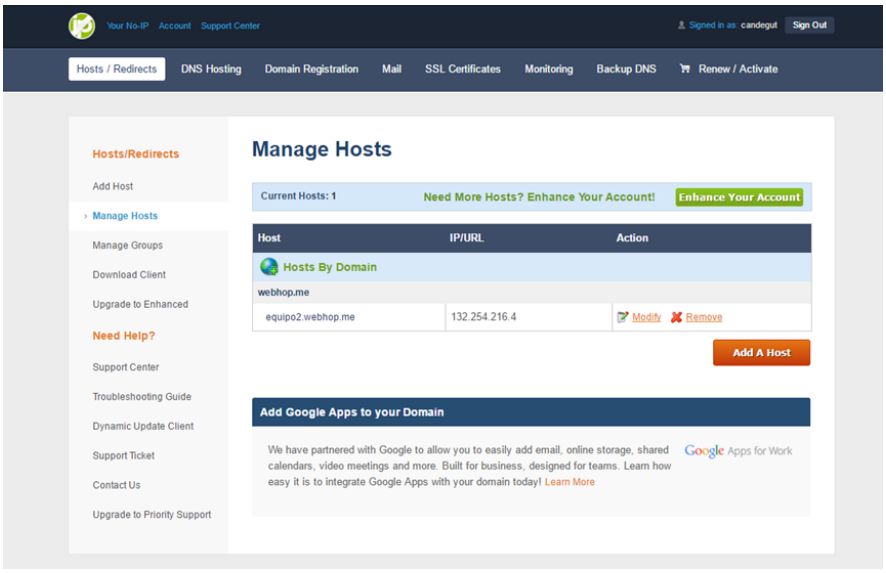
De nuestros últimos pasos será editar y agregar al final de nuestra configuración de la zona de dominio, en el archivo /etc/bind/named.conf.options. Esto para que así puedan hacer consulta los clientes, y así, si no se tiene la tabla, se mandará como default la recursión a los root servers:

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

         forwarders {
                10.40.42.47;
         };

        //========================================================================$
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================$
        dnssec-validation auto;

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };


allow-query     { any; };
allow-transfer  { none; };
allow-recursion { any; };
recursion yes;



};
```

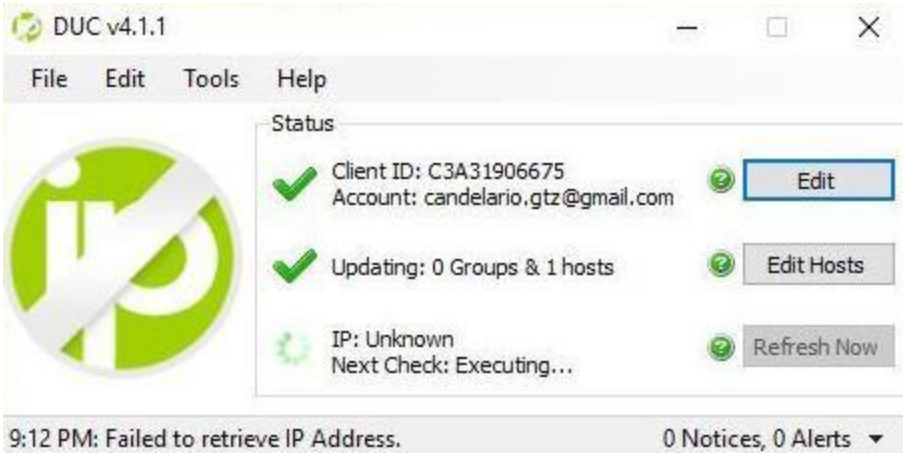
Por último, tenemos que reiniciar nuestro servicio de DNS:

```
root@cande:/etc/bind/zones# service bind9 restart
 * Stopping domain name service... bind9
waiting for pid 3353 to die

 * Starting domain name service... bind9
```

Para que los usuarios fuera de la red puedan ver nuestro contenido por medio de una dirección fácil de recordar, es necesario hacer la configuración de un DDNS, este servicio lo hicimos posible realizar gracias a un tecero llamado NO-IP, el cual brinda los servidores necesarios para que podamos crear nuestro propio host, y así, al ingresar nuestra dirección, todos los que estén afuera pueden ver lo que tenemos por los puertos 80 y 81 (página web y netflow):

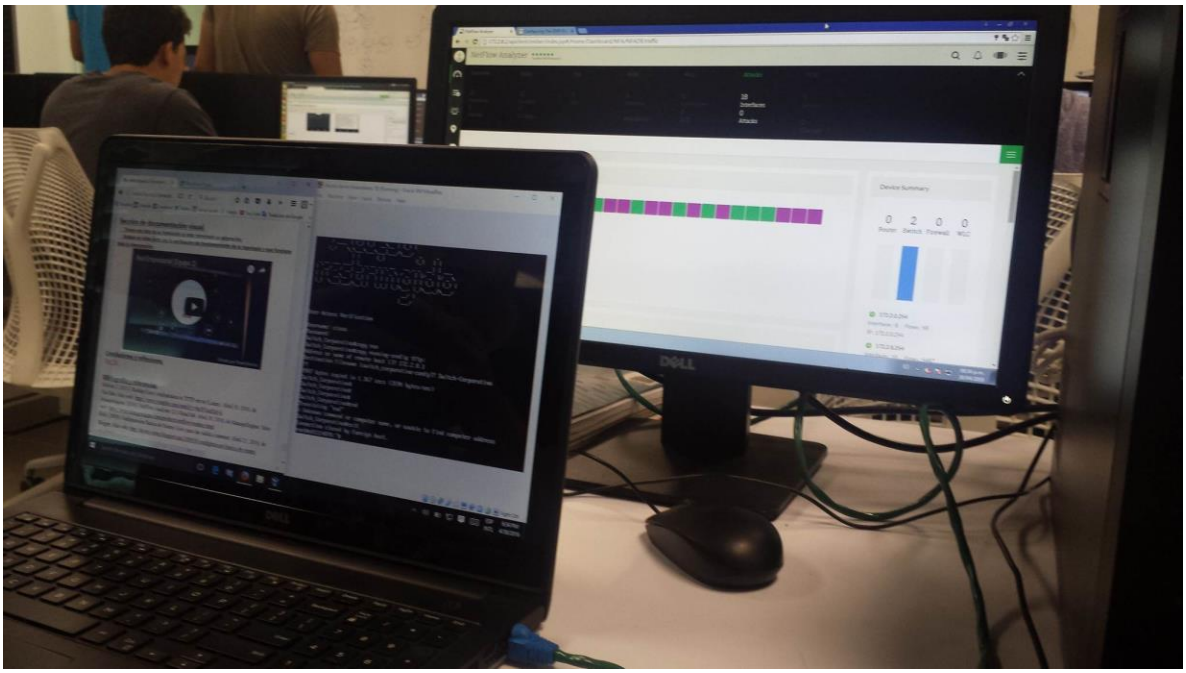


No obstante, como nuestra IP cambiaba constantemente, fue necesario instalar un cliente para actualizar la IP pública y realizara una sincronización entre el host (los servidores de NO-IP) y la IP pública para la salida a Internet. Con esto lograremos que cada vez que se actualizara la IP no tendremos preocupaciones por si los usuarios no pudieran ingresar a ver nuestro contenido.



Sección de documentación visual.

Tomen una foto de su instalación al estar terminado su asignación.



Graben un video corto con la verificación del funcionamiento de su topología y que funciona toda la interconexión.

https://www.youtube.com/watch?v=_6D8WANUMMc

Conclusiones y reflexiones.

Gracias al presente trabajo hemos podido aplicar de manera constructiva los conocimientos aprendidos a lo largo del semestre, llevando a la práctica el desarrollo de una red empresarial, que aunque aún le faltan muchas cosas por mejorar, es una red muy bien diseñada para alumnos que cursan sexto semestre de su carrera.

Durante el desarrollo de la actividad se nos presentaron distintos contratiempos, pero con un poco de paciencia y esfuerzo, logramos salir adelante para entregar un proyecto bastante completo que incluye, a grandes rasgos, comunicación entre distintas localidades, salida a internet y seguridad de los dispositivos dentro de la LAN.

Para poder armar una arquitectura como esta recordamos todo lo que vimos en la materia, para dejar una red con las características suficientes para que brinde un buen funcionamiento. Fue necesario hechar un vistazo a los temas que ya habíamos visto e intregarle todo lo nuevo. Del proyecto pudimos analizar y comprender sobre:

- Vlans, esto es cómo poder dividir una red en subredes para poder tener un mejor control de las diferentes zonas que se desean dividir.
- DNS, esto es una tabla ruteo para la salida a internet y poder buscar la dirección deseada.
- DDNS, esto es un servicio para la actualización en tiempo real de la información sobre los nombres de dominio que está en un servidor de nombres.
- EIGRP, es un protocólo de ruteo para poder darle el mejor encaminamiento a un router, para que los routers puedan compartir la información de todos los caminos posibles y así seleccionar el mejor en base a la métrica.
- Llaves, esto es implementar una llave en un router para poder evitar que otros routers sin la llave puedan acceder al mismo.
- OSPF, es un protocólo de ruteo para poder comunicar dispositivos de una red por el camino más corto e idóneo.

Todos estos temas nos sirvieron de mucho para poder dejar funcionando un sistema compacto de una red que puede ser demasiado escalable, por que todo lo que le configuramos comúnmente son para redes que están muy desarolladas.

Bibliografía y referencias.

Bolivar, J. (2015). Backup Cisco configuration to TFTP server (Linux). Abril 30, 2016, de YouTube. Sitio web: https://www.youtube.com/watch?v=buWSuzHa9-k

ManageEngine. (2016). NetFlow Analyzer 12.0 Read Me. Abril 30, 2016, de ManageEngine. Sitio web: https://www.manageengine.com/products/netflow/readme.html

Rolo. (2009). Configuración Básica de Router Cisco para dar salida a internet. Abril 25, 2016, de Blogger, Sitio web: http://doctor-redes.blogspot.mx/2009/08/configuracion-basica-de-router-cisco.html

RSYSLOG (2016). Installation. Abril 26, 2016, de RSYSLOG, sitio web: http://www.rsyslog.com/doc/v8-stable/installation/index.html#installation

shgnInc. (2014). how to prevent crontab logs from registring in syslog?. Abril 22, 2016, de ServerFault. Sitio web: http://serverfault.com/questions/31334/how-can-i-prevent-cron-from-filling-up-my-syslog

Vexxhost. (2014). How to Setup Remote System Logging with rsyslog on Ubuntu 14.04 LTS. Abril 28, 2016, de Vexxhost. Sitio web: https://vexxhost.com/resources/tutorials/how-to-setup-remote-system-logging-with-rsyslog-on-ubuntu-14-04-lts/

Configuring the OSPF router-ID.(2014).CCNA workbook. https://www.freeccnaworkbook.com/workbooks/ccna/configuring-the-ospf-router-id

Vexxhost. (2014). How to Setup Remote System Logging with rsyslog on Ubuntu 14.04 LTS. abril 28, 2016, de Vexxhost Sitio web: https://vexxhost.com/resources/tutorials/how-to-setup-remote-system-logging-with-rsyslog-on-ubuntu-14-04-lts/

Rolo. (2009). Configuración Básica de Router Cisco para dar salida a internet. Abril 25, 2016, de Blogger, sitio web: http://doctor-redes.blogspot.mx/2009/08/configuracion-basica-de-router-cisco.html