

LABORATORIO: VPN (GRE – IPSEC) / PPP (PAP – CHAP)



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

LUIS MIGUEL POLO – 20182020158

NICOLÁS DAVID SABOGAL – 20202020008

LUIS SEBASTIAN MARTINEZ GUERRERO – 20191005153

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

INGENIERÍA DE SISTEMAS

REDES DE COMUNICACIONES III

PAULO ALONSO GAONA GARCIA

INTRODUCCIÓN

El presente laboratorio se centra en la implementación y evaluación de tecnologías y protocolos clave que aseguran la seguridad y confiabilidad en redes de comunicaciones, con un enfoque en el despliegue de configuraciones avanzadas, como el establecimiento de VPN utilizando el Protocolo GRE e IPSec y la seguridad de los canales de comunicación en enlaces PPP a través de los métodos de Autenticación de Contraseña de Acceso (PAP) y el Protocolo de Autenticación de Desafío de Contraseña (CHAP).

El objetivo principal de esta práctica es configurar mecanismos de seguridad en el contexto de una VPN, implementando protocolos de autenticación y cifrado para garantizar la privacidad y autenticidad de los datos transmitidos. Además, se busca asegurar la integridad y confiabilidad de los canales de comunicación en enlaces PPP, proporcionando una conectividad segura mediante autenticación y encriptación como medidas esenciales.

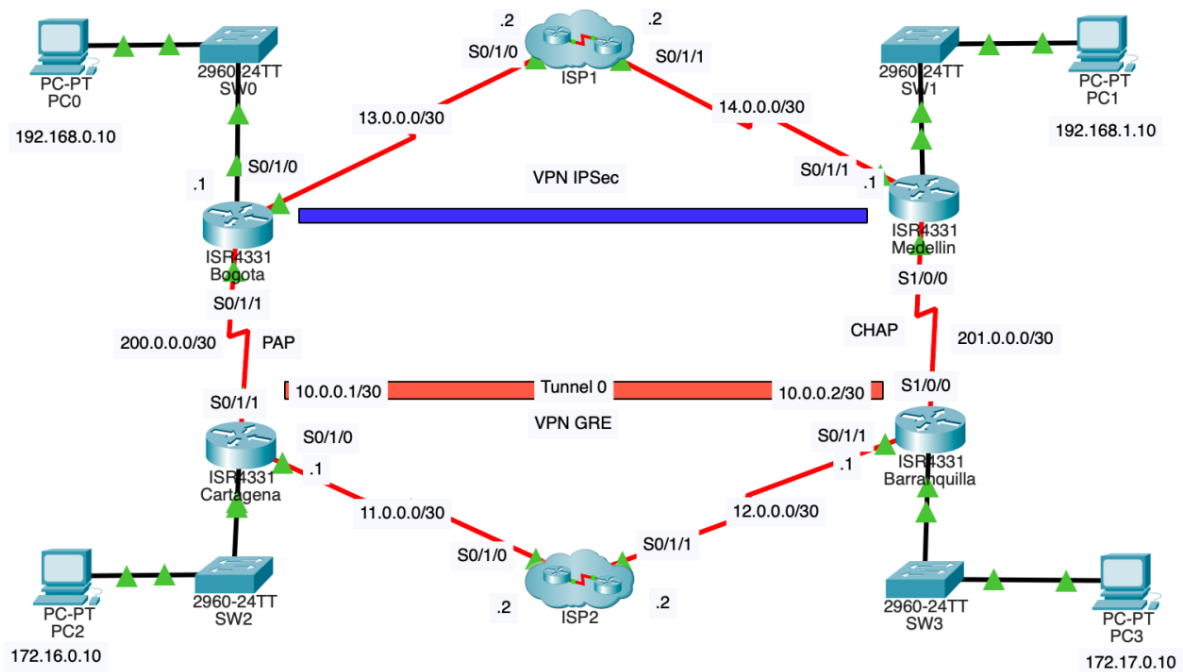
DESCRIPCIÓN GENERAL

Llevar a cabo despliegue de configuración VPN mediante especificaciones GRE, y IPSec así como aseguramiento de canales de comunicación mediante PPP a través de PAP y CHAP, y despliegue de accesos a través de ACL.

OBJETIVOS

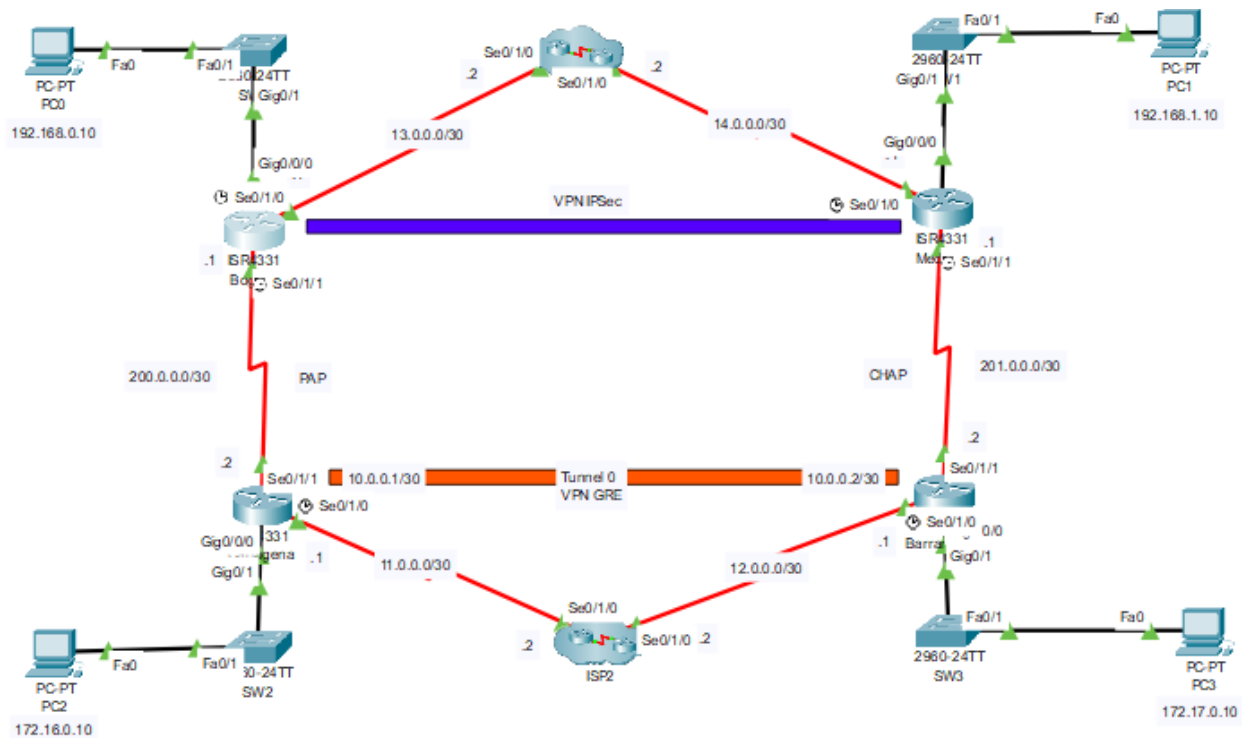
- Configuración de mecanismos de seguridad bajo VPN.
- Asegurar canales de comunicación enlaces PPP.
- Despliegue de ACL.

TOPOLOGÍA DE TRABAJO



DESARROLLO

En primer lugar, se armó la topología en Packet Tracer, la cual se muestra a continuación:

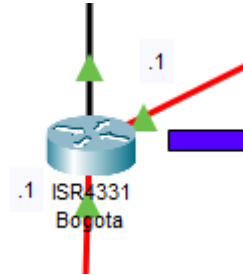


Configuración Básica de Equipos

Se realizó la configuración básica asignando nombres para cada dispositivo, direcciones IP públicas y privadas, y la configuración de enrutamiento OSPF para la topología de trabajo.

Routers

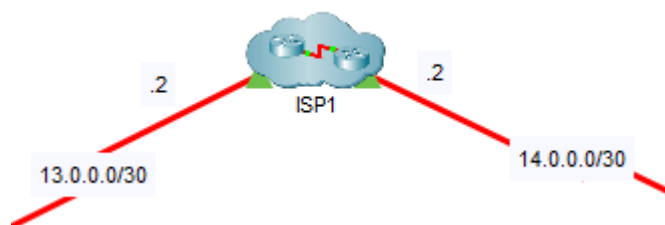
- **Bogotá**



```
enable
configure terminal
hostname BOGOTA
```

```
interface GigabitEthernet 0/0/0
ip address 192.168.0.1 255.255.255.0
no shutdown
```

```
router ospf 1
network 13.0.0.0 0.0.0.3 area 1
network 192.168.0.0 0.0.0.255 area 1
network 200.0.0.0 0.0.0.3 area 1
```



- **ISP1-1**

```
enable
configure terminal
```

```
hostname ISP1-1
interface Serial 0/1/0
ip address 13.0.0.2 255.255.255.252
no shutdown
```

```
interface Serial 0/1/1
ip address 8.0.0.1 255.255.255.252
no shutdown
```

```
router ospf 1
network 8.0.0.0 0.0.0.3 area 1
network 13.0.0.0 0.0.0.3 area 1
```

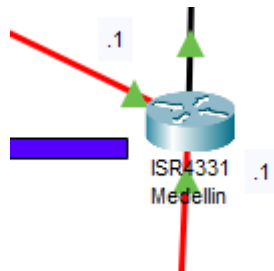
- **ISP1-2**

```
enable
configure terminal
hostname ISP1-2
interface Serial 0/1/0
ip address 14.0.0.2 255.255.255.252
no shutdown
```

```
interface Serial 0/1/1
ip address 8.0.0.2 255.255.255.252
no shutdown
```

```
router ospf 1
network 8.0.0.0 0.0.0.3 area 1
network 14.0.0.0 0.0.0.3 area 1
```

- **Medellin**



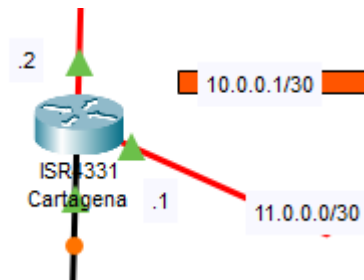
```
enable
configure terminal
hostname MEDELLIN
```

```
interface GigabitEthernet 0/0/0
```

```
ip address 192.168.1.1 255.255.255.0
no shutdown
```

```
router ospf 1
network 14.0.0.0 0.0.0.3 area 1
network 192.168.1.0 0.0.0.255 area 1
network 201.0.0.0 0.0.0.3 area 1
```

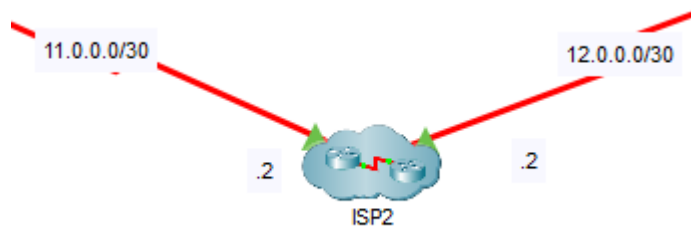
- **Cartagena**



```
enable
configure terminal
hostname CARTAGENA
```

```
interface GigabitEthernet 0/0/0
ip address 172.16.0.1 255.255.255.0
no shutdown
```

```
interface Serial 0/1/0
ip address 11.0.0.1 255.255.255.252
no shutdown
```



- **ISP2-1**

```
enable
configure terminal
```

```
hostname ISP2-1
interface Serial 0/1/0
ip address 11.0.0.2 255.255.255.252
no shutdown
```

```
interface Serial 0/1/1
ip address 9.0.0.1 255.255.255.252
no shutdown
```

```
router ospf 1
network 9.0.0.0 0.0.0.3 area 1
network 11.0.0.0 0.0.0.3 area 1
```

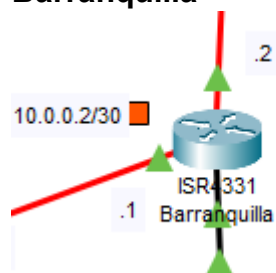
- **ISP2-2**

```
enable
configure terminal
hostname ISP2-2
interface Serial 0/1/0
ip address 12.0.0.2 255.255.255.252
no shutdown
```

```
interface Serial 0/1/1
ip address 9.0.0.2 255.255.255.252
no shutdown
```

```
router ospf 1
network 9.0.0.0 0.0.0.3 area 1
network 12.0.0.0 0.0.0.3 area 1
```

- **Barranquilla**



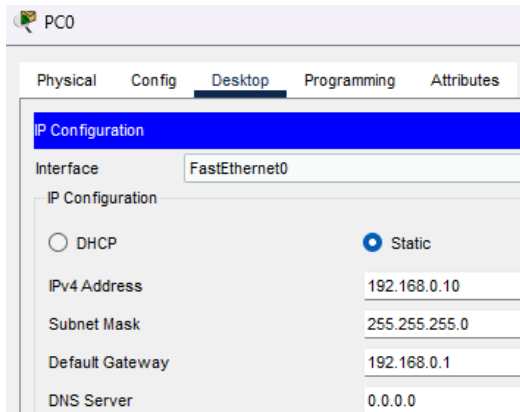
```
enable
configure terminal
hostname BARRANQUILLA
```

```
interface GigabitEthernet 0/0/0
ip address 172.17.0.1 255.255.255.0
no shutdown
```

```
interface Serial 0/1/0
ip address 12.0.0.1 255.255.255.252
no shutdown
```

Equipos

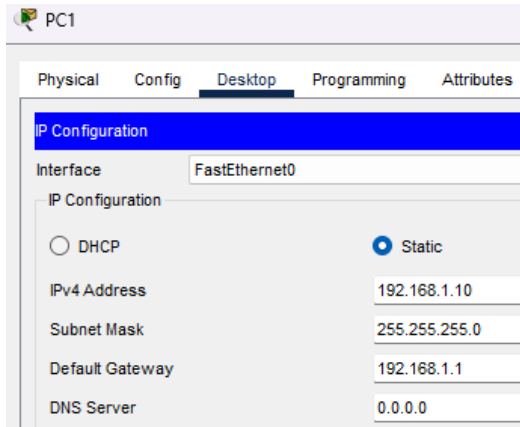
- **PC0**



The screenshot shows the configuration window for PC0. The 'Desktop' tab is selected. Under 'IP Configuration', the 'Interface' is 'FastEthernet0'. The 'Static' radio button is selected for IP Configuration. The fields are filled with the following values:

Field	Value
IPv4 Address	192.168.0.10
Subnet Mask	255.255.255.0
Default Gateway	192.168.0.1
DNS Server	0.0.0.0

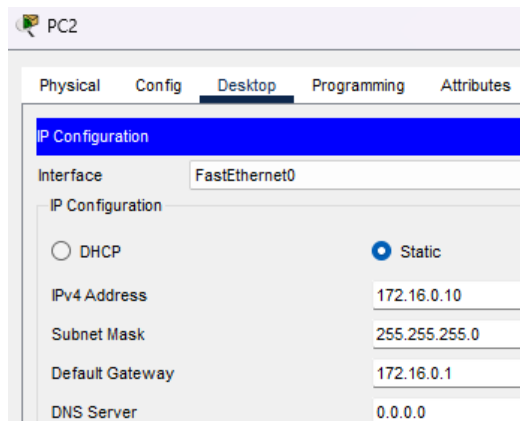
- **PC1**



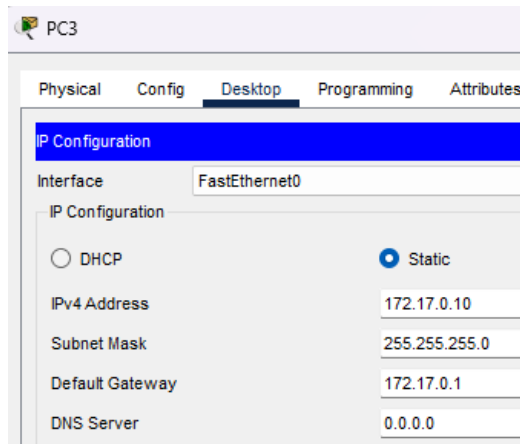
The screenshot shows the configuration window for PC1. The 'Desktop' tab is selected. Under 'IP Configuration', the 'Interface' is 'FastEthernet0'. The 'Static' radio button is selected for IP Configuration. The fields are filled with the following values:

Field	Value
IPv4 Address	192.168.1.10
Subnet Mask	255.255.255.0
Default Gateway	192.168.1.1
DNS Server	0.0.0.0

- **PC2**



- **PC3**



VPN GRE

El Protocolo de Encapsulación de Datagramas en Enrutamiento (GRE) es una tecnología que permite establecer túneles virtuales de comunicación en redes, facilitando la transferencia de paquetes de datos entre dos puntos de la red a través de una infraestructura pública o compartida, como Internet, simulando una conexión directa. El funcionamiento de una VPN basada en GRE incluye los siguientes pasos:

Establecimiento del Túnel: Se configura un túnel GRE en ambos extremos de la conexión, es decir, en el punto de inicio y en el punto final del enlace.

Encapsulación de Datos: Los paquetes de datos que se desean transmitir desde el extremo de origen se encapsulan en paquetes GRE. Esto consiste en envolver los datos originales con un encabezado GRE que incluye información sobre el origen y el destino.

Transmisión por el Túnel: Los paquetes encapsulados se envían a través de la red pública o compartida. Como la infraestructura de la red solo reconoce los paquetes GRE, no tiene acceso al contenido original de los datos.

Desencapsulación en el Destino: En el extremo final, se elimina el encabezado GRE, recuperando los datos originales encapsulados.

Entrega de los Datos: Los datos originales se entregan al dispositivo receptor como si hubieran sido transmitidos de manera directa, a pesar de haber atravesado una red pública o compartida.

Comandos configuración VPN GRE

La configuración descrita a continuación se aplicó a los routers ubicados en Cartagena y Barranquilla. Para implementar el tunneling GRE, se establece un túnel, que corresponde a una interfaz lógica, asignándole un identificador específico. Durante la configuración del túnel, se define la dirección IP que utilizará el router en dicha interfaz, así como el origen y el destino de los mensajes. Los comandos empleados son los siguientes:

```
Router(config)# interface tunnel 0
```

Con el primer comando, se crea la interfaz lógica del túnel y se le asigna un identificador, que en este caso es 0.

```
Router(config-if)# ip address <dirección ipv4> <máscara de red>
```

El segundo comando permite asignar al router, dentro de la interfaz lógica del túnel, una dirección IP específica, utilizando una dirección IPv4.

```
Router(config-if)# tunnel source serial <id interfaz>
```

El tercer comando define la fuente de los mensajes del túnel, que corresponde a la interfaz serial física del router.

```
Router(config-if)# tunnel destination <dirección ipv4 pública>
```

El cuarto comando especifica la dirección IP de destino de los mensajes, que es la dirección IPv4 pública configurada en la interfaz serial del router opuesto.

```
Router(config-if)# tunnel mode gre ip
```

El quinto comando establece el modo de funcionamiento del túnel, que en este caso es GRE sobre IP.

Finalmente, se configura tanto en Router Cartagena como en Router Barranquilla el enrutamiento OSPF para que el tráfico fluya a través del túnel. Es necesario declarar previamente las redes locales en OSPF para que ambos routers las compartan y las reconozcan mutuamente.

- **Cartagena**

```
interface Tunnel 0
ip address 10.0.0.1 255.255.255.252
tunnel source Serial 0/1/0
tunnel destination 12.0.0.1
tunnel mode gre ip
```

```
router ospf 1
network 10.0.0.0 0.0.0.3 area 1
network 11.0.0.0 0.0.0.3 area 1
network 172.16.0.0 0.0.0.255 area 1
network 200.0.0.0 0.0.0.3 area 1
```

```
!
interface Tunnel0
ip address 10.0.0.1 255.255.255.252
mtu 1476
tunnel source Serial0/1/0
tunnel destination 12.0.0.1
!
!
```

- **Barranquilla**

```
interface Tunnel 0
ip address 10.0.0.2 255.255.255.252
tunnel source Serial 0/1/0
tunnel destination 11.0.0.1
tunnel mode gre ip
```

```
router ospf 1
network 10.0.0.0 0.0.0.3 area 1
network 12.0.0.0 0.0.0.3 area 1
```

```
network 172.17.0.0 0.0.0.255 area 1
network 201.0.0.0 0.0.0.3 area 1
```

```
!
interface Tunnel0
 ip address 10.0.0.2 255.255.255.252
 mtu 1476
 tunnel source Serial0/1/0
 tunnel destination 11.0.0.1
!
!
```

Verificación de conexiones

Para la verificación de las conexiones, se realiza un ping desde un router al PC de la red local del otro router, así como un ping desde el PC en la red local de un router al PC en la red local del otro router.

Ping desde Router de Cartagena hacia PC3 (172.17.0.10)

```
CARTAGENA#ping 172.17.0.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.17.0.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/18/23 ms
```

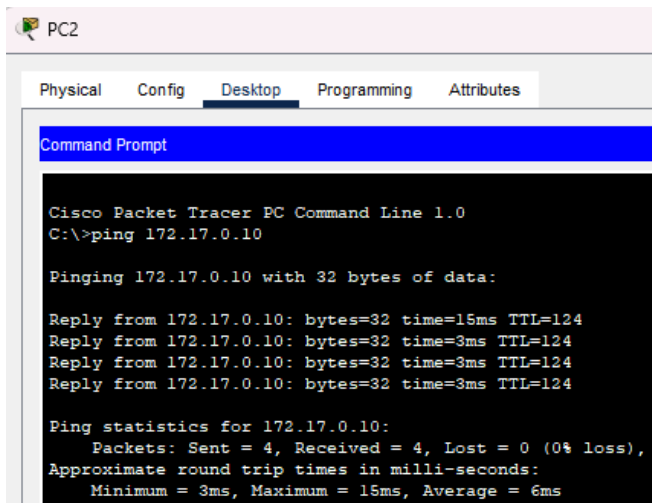
Ping desde Router de Barranquilla hacia PC2 (172.16.0.10)

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.10, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 15/17/24 ms
```

```
BARRANQUILLA#
BARRANQUILLA#ping 172.16.0.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/14/21 ms
```

Ping desde PC2 (172.16.0.10) hacia PC3 (172.17.0.10)



PC2

Physical Config Desktop Programming Attributes

Command Prompt

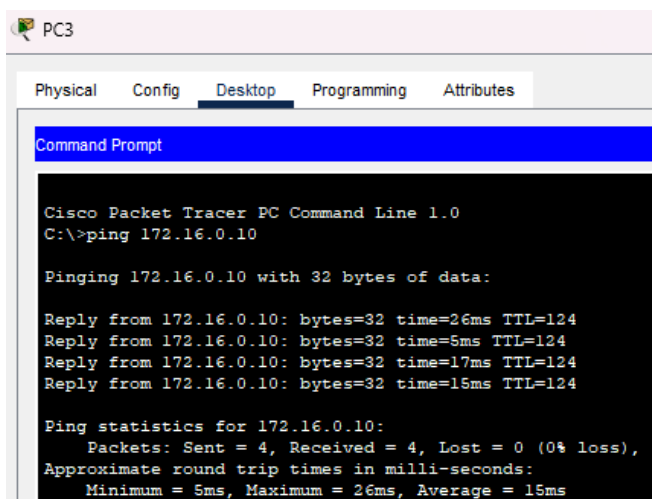
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 172.17.0.10

Pinging 172.17.0.10 with 32 bytes of data:

Reply from 172.17.0.10: bytes=32 time=15ms TTL=124
Reply from 172.17.0.10: bytes=32 time=3ms TTL=124
Reply from 172.17.0.10: bytes=32 time=3ms TTL=124
Reply from 172.17.0.10: bytes=32 time=3ms TTL=124

Ping statistics for 172.17.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 15ms, Average = 6ms
```

Ping desde PC3 (172.17.0.10) hacia PC2 (172.16.0.10)



PC3

Physical Config Desktop Programming Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 172.16.0.10

Pinging 172.16.0.10 with 32 bytes of data:

Reply from 172.16.0.10: bytes=32 time=26ms TTL=124
Reply from 172.16.0.10: bytes=32 time=5ms TTL=124
Reply from 172.16.0.10: bytes=32 time=17ms TTL=124
Reply from 172.16.0.10: bytes=32 time=15ms TTL=124

Ping statistics for 172.16.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 26ms, Average = 15ms
```

VPN IPSec

IPSec es una solución ampliamente adoptada para establecer conexiones seguras a través de redes públicas o compartidas, como Internet. Su propósito principal es proteger la confidencialidad, integridad y autenticidad de los datos transmitidos. A continuación, se detalla el proceso de funcionamiento de una VPN mediante IPSec:

Negociación de parámetros de seguridad: Antes de iniciar la transmisión de datos, los extremos de la conexión acuerdan los detalles de seguridad, como algoritmos de cifrado,

métodos de autenticación y otros aspectos necesarios para garantizar una comunicación segura.

Autenticación de los extremos: Ambos extremos de la conexión verifican su identidad para garantizar que están comunicándose con la entidad correcta. Este paso puede incluir el uso de certificados digitales, contraseñas u otros mecanismos de autenticación.

Creación del túnel VPN: Tras completar la negociación de parámetros y la autenticación, se establece un túnel seguro que encapsula todo el tráfico de datos que se transmitirá entre los extremos de la VPN.

Cifrado de datos: Los datos enviados a través del túnel son cifrados utilizando los algoritmos acordados previamente, asegurando que permanezcan ilegibles para cualquier tercero no autorizado que intente interceptarlos.

Autenticación y verificación de integridad: Además del cifrado, IPSec aplica autenticación y protección de integridad a los datos transmitidos. Los paquetes se firman digitalmente, lo que permite verificar que no hayan sido alterados y que provienen de una fuente confiable.

Transmisión segura: Los paquetes cifrados se envían a través de la red pública hacia su destino final. Gracias al cifrado, la infraestructura de la red no tiene acceso al contenido de los paquetes.

Desencriptación y entrega: En el extremo receptor, los paquetes se desencriptan y se autentican para verificar su origen y garantizar que no han sido modificados. Finalmente, los datos originales se entregan de manera segura al dispositivo de destino.

Comandos configuración VPN IPSec

Se establece una política de seguridad en los enrutadores de Bogotá y Medellín que incorpora las propiedades de la VPN IPSec, incluyendo algoritmos de cifrado y autenticación.

Establecimiento de prioridad en la política:

```
Router(config)# crypto isakmp policy 20
```

Se define el tipo de autenticación, el algoritmo de cifrado (3DES), la función hash (MD5), el grupo para intercambio de claves (grupo 1) y la duración de la fase ISAKMP (3600 segundos).

```
Router(config-isakmp)# authentication pre-share
```

```
Router(config-isakmp)# encryption 3des
```

```
Router(config-isakmp)# hash md5
```

```
Router(config-isakmp)# group 1
```

```
Router(config-isakmp)# lifetime 3600
```

En Medellín y Bogotá se configuran las claves de autenticación:

```
Medellin(config)# crypto isakmp key redes3 address 13.0.0.1
```

```
Bogota(config)# crypto isakmp key redes3 address 14.0.0.1
```

Se configura el conjunto de transformación que define los algoritmos de cifrado y autenticación para IPSec:

```
Bogota(config)# crypto ipsec transform-set sistemas esp-3des esp-md5-hmac
```

```
Medellin(config)# crypto ipsec transform-set sistemas esp-3des esp-md5-hmac
```

Configuramos la lista de control de acceso definiendo la dirección de origen y la de destino con su wildcard:

```
Bogota(config)# access-list 100 permit ip 192.168.0.0 0.0.0.255  
192.168.1.0 0.0.0.255
```

```
Medellin(config)# access-list 100 permit ip 192.168.1.0 0.0.0.255  
192.168.0.0 0.0.0.255
```

Se define el mapa de direcciones que se utilizará para IPSec:

```
Router(config)# crypto map mimap 20 ipsec-isakmp
```

```
Medellin(config-crypto-map)# set peer 13.0.0.1
```

```
Bogota(config-crypto-map)# set peer 14.0.0.1
```

```
Router(config-crypto-map)# set transform-set sistemas
```

```
Router(config-crypto-map)# match address 100
```

En las interfaces correspondientes de los enrutadores, se aplica el mapa criptográfico:

```
Bogota(config)# interface serial 0/1/0
```

```
Medellin(config)# interface serial 0/1/0
```

```
Router(config-if)# crypto map mimap
```

Con esta configuración, los enrutadores quedan preparados para operar con IPSec, garantizando la seguridad en las comunicaciones entre las dos sedes.

- **Bogotá**

```
crypto isakmp policy 20
authentication pre-share
encryption 3des
hash md5
group 1
lifetime 3600
crypto isakmp key redes3 address 14.0.0.1

crypto ipsec transform-set sistemas esp-3des esp-md5-hmac

access-list 100 permit ip 192.168.0.0 0.0.0.255 192.168.1.0
0.0.0.255

crypto map mimap 20 ipsec-isakmp
set peer 14.0.0.1
set transform-set sistemas
match address 100
interface Serial 0/1/0
crypto map mimap
ip address 13.0.0.1 255.255.255.252
no shutdown
```



```
Bogota
Physical Config CLI Attributes
IOS Command Line Interface
!
!
crypto isakmp policy 20
  encr 3des
  hash md5
  authentication pre-share
  lifetime 3600
!
crypto isakmp key redes3 address 14.0.0.1
!
!
!
crypto ipsec transform-set sistemas esp-3des esp-md5-hmac
!
crypto map mimap 20 ipsec-isakmp
  set peer 14.0.0.1
  set transform-set sistemas
  match address 100
!

!
access-list 100 permit ip 192.168.0.0 0.0.0.255 192.168.1.0 0.0.0.255
!
!

BOGOTA#show crypto map
Crypto Map mimap 20 ipsec-isakmp
  Peer = 14.0.0.1
  Extended IP access list 100
    access-list 100 permit ip 192.168.0.0 0.0.0.255 192.168.1.0 0.0.0.255
  Current peer: 14.0.0.1
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={
    sistemas,
  }
  Interfaces using crypto map mimap:
    Serial0/1/0
```

- **Medellín**

```
crypto isakmp policy 20
authentication pre-share
encryption 3des
hash md5
group 1
lifetime 3600
crypto isakmp key redes3 address 13.0.0.1

crypto ipsec transform-set sistemas esp-3des esp-md5-hmac

access-list 100 permit ip 192.168.1.0 0.0.0.255 192.168.0.0
0.0.0.255

crypto map mimap 20 ipsec-isakmp
```

```

set peer 13.0.0.1
set transform-set sistemas
match address 100

interface Serial 0/1/0
crypto map mimap
ip address 14.0.0.1 255.255.255.252
no shutdown

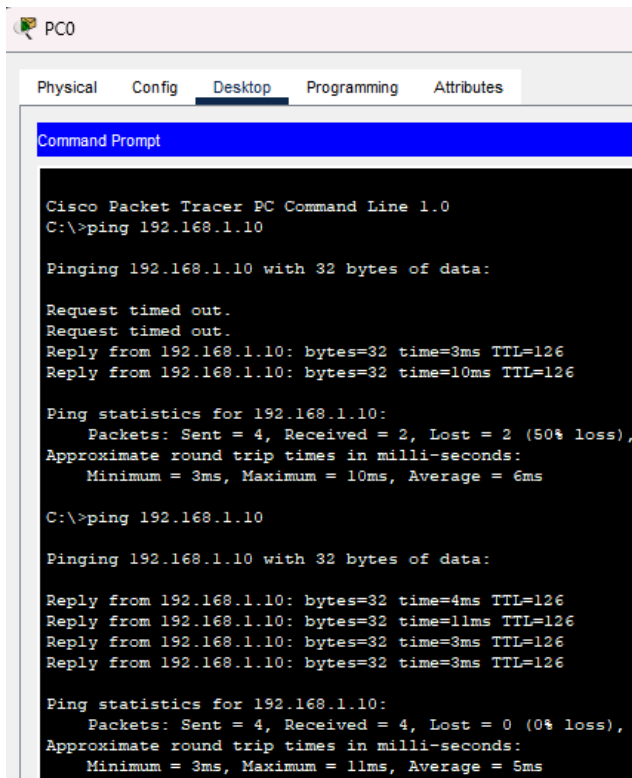
!
crypto isakmp policy 20
  encr 3des
  hash md5
  authentication pre-share
  lifetime 3600
!
crypto isakmp key redes3 address 13.0.0.1
!
!
!
crypto ipsec transform-set sistemas esp-3des esp-md5-hmac
!
crypto map mimap 20 ipsec-isakmp
  set peer 13.0.0.1
  set transform-set sistemas
  match address 100
!

!
access-list 100 permit ip 192.168.1.0 0.0.0.255 192.168.0.0 0.0.0.255
!
!

MEDELLIN#show crypto map
Crypto Map mimap 20 ipsec-isakmp
  Peer = 13.0.0.1
  Extended IP access list 100
    access-list 100 permit ip 192.168.1.0 0.0.0.255 192.168.0.0 0.0.0.255
  Current peer: 13.0.0.1
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={
    sistemas,
  }
  Interfaces using crypto map mimap:
    Serial0/1/0

```

Verificación de conexión



```
PC0
Physical Config Desktop Programming Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Request timed out.
Request timed out.
Reply from 192.168.1.10: bytes=32 time=3ms TTL=126
Reply from 192.168.1.10: bytes=32 time=10ms TTL=126

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 10ms, Average = 6ms

C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 192.168.1.10: bytes=32 time=4ms TTL=126
Reply from 192.168.1.10: bytes=32 time=11ms TTL=126
Reply from 192.168.1.10: bytes=32 time=3ms TTL=126
Reply from 192.168.1.10: bytes=32 time=3ms TTL=126

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 11ms, Average = 5ms
```

Configuración Autenticación PPP

PAP es un protocolo de autenticación sencillo utilizado en redes para verificar la identidad de un cliente o dispositivo al establecer una conexión. Se caracteriza por su simplicidad y facilidad de implementación, ya que utiliza un enfoque directo en el que el cliente envía sus credenciales (nombre de usuario y contraseña) al servidor para su validación. PAP puede ser configurado dentro de PPP para verificar la identidad del dispositivo remoto. Sin embargo, este método presenta vulnerabilidades significativas en términos de seguridad, por lo que su uso se limita a entornos controlados o cuando no se dispone de alternativas más seguras.

Comandos configuración PAP

```
Router(config)# interface serial <id_interfaz>
```

```
Router(config-if)# ip address <dirección ipv4> <máscara de red>
```

```
Router(config-if)# encapsulation ppp
```

Accede a la interfaz serial que se va a modificar, asigna una dirección IP a la interfaz serial, define su máscara de subred y cambia el encapsulamiento de la interfaz a PPP

```
Router(config-if)# ppp authentication pap
```

Configurar el protocolo PPP para usar PAP como método de autenticación.

```
Router(config-if)# ppp pap sent-username bogota password redes3
```

Definir el nombre de usuario y la contraseña que este router enviará al otro extremo del enlace para autenticarse mediante PAP en router Cartagena.

```
Router(config-if)# no keepalive
```

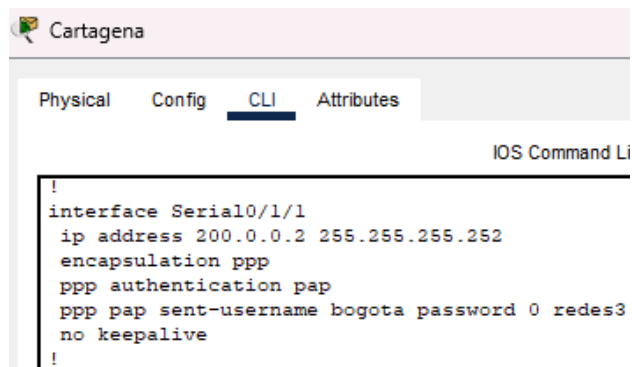
```
Router(config-if)# no shutdown
```

```
Router(config-if)# username cartagena password redes3
```

Crea un usuario en el router con el nombre cartagena y la contraseña redes3. Este usuario será utilizado por el router remoto para autenticarse en este dispositivo.

- **Cartagena**

```
interface Serial 0/1/1
ip address 200.0.0.2 255.255.255.252
encapsulation ppp
ppp authentication pap
ppp pap sent-username bogota password redes3
no keepalive
no shutdown
username cartagena password redes3
```

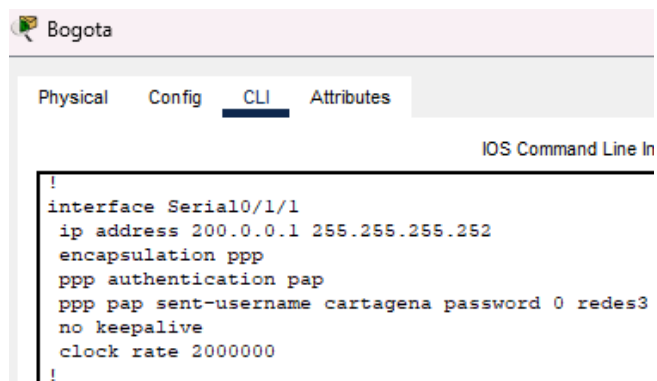


La configuración resultante se puede observar con el comando `show interface serial <id interfaz>`.

```
CARTAGENA#show interface serial 0/1/1
Serial0/1/1 is up, line protocol is up (connected)
  Hardware is HD64570
  Internet address is 200.0.0.2/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set, keepalive not set
  LCP Open
  Open: IPCP, CDPCP
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/0/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 1158 kilobits/sec
  5 minute input rate 54 bits/sec, 0 packets/sec
  5 minute output rate 54 bits/sec, 0 packets/sec
  1329 packets input, 92668 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  1314 packets output, 92116 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up
```

- **Bogotá**

```
interface Serial 0/1/1
ip address 200.0.0.1 255.255.255.252
encapsulation ppp
ppp authentication pap
ppp pap sent-username cartagena password redes3
no keepalive
no shutdown
username bogota password redes3
```







```

BOGOTA#show interface serial 0/1/1
Serial0/1/1 is up, line protocol is up (connected)
Hardware is HD64570
Internet address is 200.0.0.1/30
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, loopback not set, keepalive not set
LCP Open
Open: IPCP, CDPCP
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
Conversations 0/0/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
Available Bandwidth 1158 kilobits/sec
5 minute input rate 37 bits/sec, 0 packets/sec
5 minute output rate 40 bits/sec, 0 packets/sec
    29 packets input, 2880 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    30 packets output, 2772 bytes, 0 underruns

```

Para verificar la configuración, se enviará un mensaje del router de Cartagena hasta el router de Bogotá y viceversa.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	Carta...	Bogota	ICMP		0.000	N	0	(edit)
	Successful	Bogota	Cartagena	ICMP		0.000	N	1	(edit)

Configuración CHAP

CHAP es un protocolo de autenticación ampliamente utilizado en redes para verificar de forma segura la identidad de un cliente o dispositivo antes y durante una conexión. Su diseño se basa en un proceso de desafío-respuesta, que garantiza que las credenciales no se transmitan directamente a través de la red, minimizando el riesgo de exposición a ataques. Es común en enlaces configurados con el protocolo PPP y en aplicaciones que requieren autenticación segura.

Una de las principales características de CHAP es su capacidad para realizar autenticaciones periódicas durante toda la duración de la conexión. Esto lo diferencia de métodos como PAP, que verifica la autenticidad solo al inicio de la sesión.

Con CHAP, el servidor puede enviar desafíos aleatorios al cliente en cualquier momento, obligándolo a autenticarse nuevamente. Este mecanismo proporciona una capa adicional de seguridad frente a interrupciones o intentos de suplantación de identidad.

Explicación comandos CHAP

```
Router(config)# username <username> secret <password>
```

Define un nombre de usuario y una contraseña para la autenticación CHAP.

Router(config)# interface serial <id_interfaz>

Entra en la configuración de la interfaz serial específica donde se desea aplicar el protocolo PPP y la autenticación CHAP.

Router(config-if)# ip address <dirección ipv4> <máscara de red>

Asigna una dirección IP a la interfaz serial y define su máscara de subred.

Router(config-if)# encapsulation ppp

Configura el protocolo de encapsulación PPP en la interfaz.

Router(config-if)# ppp authentication chap

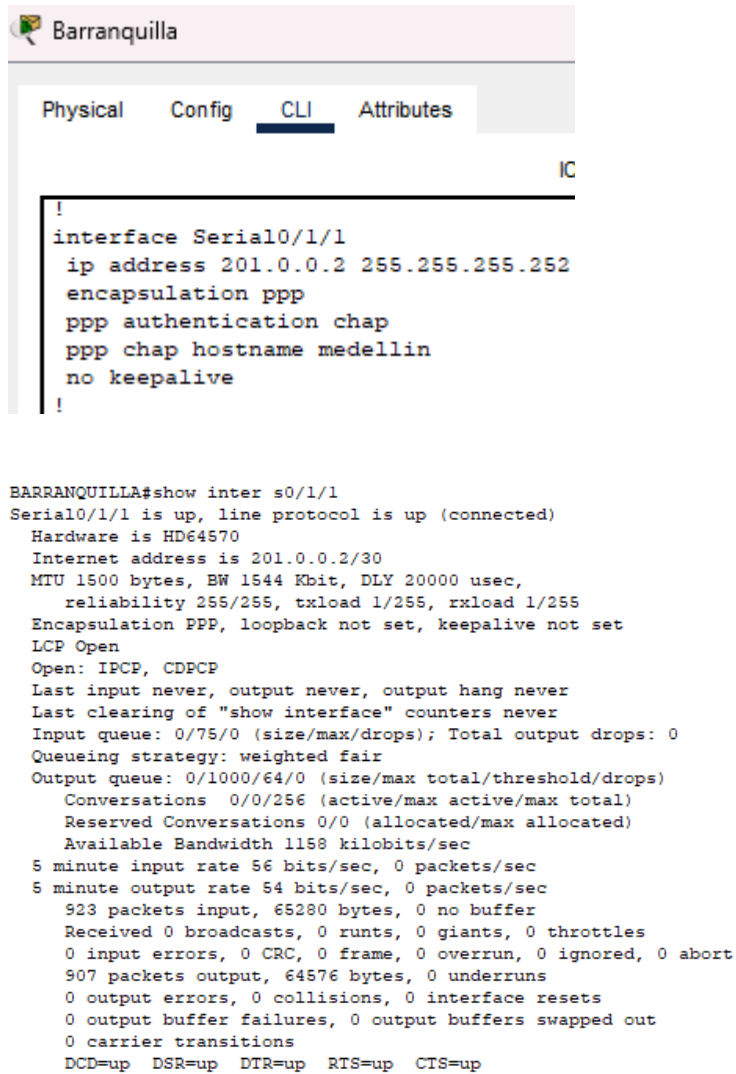
Activa CHAP como el método de autenticación para PPP en la interfaz.

Router(config-if)# ppp chap hostname <router_remoto>

Define el nombre de usuario y la contraseña que se utilizarán para autenticar al router remoto.

- **Barranquilla**

```
username barranquilla secret redes3
interface Serial 0/1/1
ip address 201.0.0.2 255.255.255.252
encapsulation ppp
ppp authentication chap
ppp chap hostname medellin
no keepalive
no shutdown
```

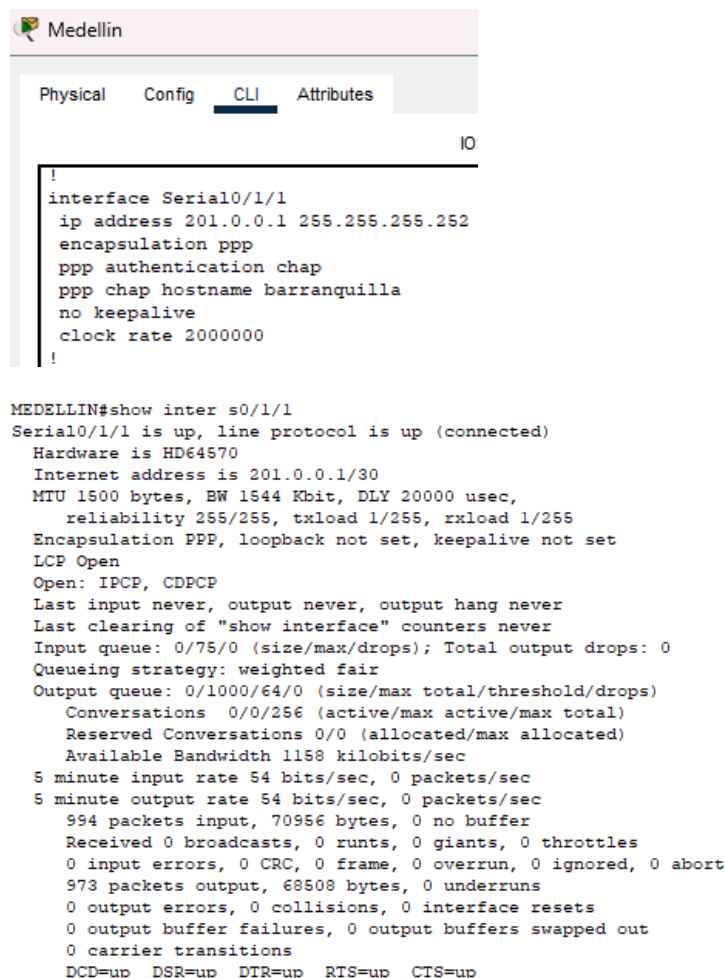


```
Barranquilla
Physical  Config  CLI  Attributes
IC
!
interface Serial0/1/1
 ip address 201.0.0.2 255.255.255.252
 encapsulation ppp
 ppp authentication chap
 ppp chap hostname medellin
 no keepalive
!

BARRANQUILLA#show inter s0/1/1
Serial0/1/1 is up, line protocol is up (connected)
  Hardware is HD64570
  Internet address is 201.0.0.2/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set, keepalive not set
  LCP Open
  Open: IPCP, CDPCP
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations  0/0/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 1158 kilobits/sec
  5 minute input rate 56 bits/sec, 0 packets/sec
  5 minute output rate 54 bits/sec, 0 packets/sec
    923 packets input, 65280 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  907 packets output, 64576 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up
```

- **Medellín**

```
username medellin secret redes3
interface Serial 0/1/1
 ip address 201.0.0.1 255.255.255.252
 encapsulation ppp
 ppp authentication chap
 ppp chap hostname barranquilla
 no keepalive
 no shutdown
```





```

!
interface Serial0/1/1
 ip address 201.0.0.1 255.255.255.252
 encapsulation ppp
 ppp authentication chap
 ppp chap hostname barranquilla
 no keepalive
 clock rate 2000000
!

MEDELLIN#show inter s0/1/1
Serial0/1/1 is up, line protocol is up (connected)
Hardware is HD64570
Internet address is 201.0.0.1/30
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, loopback not set, keepalive not set
LCP Open
Open: IPCP, CDPCP
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
  Conversations  0/0/256 (active/max active/max total)
  Reserved Conversations 0/0 (allocated/max allocated)
  Available Bandwidth 1158 kilobits/sec
5 minute input rate 54 bits/sec, 0 packets/sec
5 minute output rate 54 bits/sec, 0 packets/sec
  994 packets input, 70956 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  973 packets output, 68508 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up

```

Para verificar la configuración, se enviará un mensaje del router de Barranquilla hasta el router de Medellin y viceversa.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	Barra...	Medellin	ICMP		0.000	N	0	(edit)
	Successful	Medellin	Barranquilla	ICMP		0.000	N	1	(edit)

Características y principales diferencias VPN GRE y VPN mediante IPSec

VPN mediante GRE (Protocolo de Encapsulación de Datagramas en Enrutamiento):

El protocolo GRE se utiliza para encapsular datos dentro de otro protocolo, como IP, creando un túnel virtual que facilita el transporte de múltiples protocolos a través de una red, lo que resulta ideal para conexiones punto a punto. Sin embargo, presenta limitaciones en cuanto a la seguridad, ya que no incluye medidas de protección integradas.

Por esta razón, se complementa con herramientas como cortafuegos o IPSec para garantizar la seguridad de las comunicaciones. Además, GRE ofrece soporte multiprotocolo, permitiendo encapsular no solo IPv4 e IPv6, sino también otros protocolos de red, lo que lo convierte en una solución flexible y adaptable a diversas necesidades de comunicación.

VPN mediante IPSec (Protocolo de Seguridad de Internet):

IPSec está diseñado para ofrecer una seguridad robusta en conexiones VPN, proporcionando autenticación, cifrado y verificación de la integridad de los datos, lo que asegura tanto la confidencialidad como la autenticidad de la información. Además, es compatible con diversos protocolos de comunicación, lo que facilita su implementación en distintos entornos.

IPSec también ofrece flexibilidad mediante sus modos de operación, permitiendo configurarse en modo túnel para cifrar todo el tráfico entre redes o en modo transporte para proteger únicamente los datos entre puntos finales. Su funcionamiento independiente de la capa de red le permite ser utilizado con protocolos como IPv4, IPv6 o cualquier otro, ampliando aún más sus posibilidades de aplicación.

Diferencias entre GRE e IPSec:

El propósito principal de GRE es crear túneles y encapsular datos, mientras que IPSec se enfoca en asegurar las comunicaciones de manera integral. En términos de la capa del modelo OSI, GRE opera principalmente en la capa de enlace (capa 2), permitiendo la transmisión de tráfico de capa 3, como IP, mientras que IPSec trabaja en la capa de red (capa 3), proporcionando seguridad mediante cifrado y encapsulación del tráfico IP. En cuanto a la versatilidad, GRE es flexible al soportar diversos protocolos de red, mientras que IPSec está especializado en proteger conexiones VPN.

En cuanto a los usos comunes, GRE es ideal para establecer túneles simples entre ubicaciones específicas, como conexiones entre sucursales o routers, mientras que IPSec se emplea principalmente en entornos empresariales para acceso remoto o VPN seguras, gracias a sus avanzadas capacidades de protección.

Conclusiones

La implementación de VPN mediante GRE e IPSec demuestra ser efectiva en la creación de túneles seguros que no solo encapsulan datos, sino que también aseguran su confidencialidad y autenticidad a través de robustos mecanismos de cifrado y autenticación, lo que es crucial en entornos empresariales donde la protección de la información es prioritaria.

La autenticación de los extremos mediante PAP y CHAP es un componente crítico en la seguridad de las conexiones PPP. Aunque PAP es más sencillo de implementar, su falta de cifrado lo hace vulnerable a ataques, lo que resalta la necesidad de utilizar CHAP en situaciones donde la seguridad es esencial, garantizando así una verificación más robusta de la identidad de los dispositivos.

La combinación de GRE e IPSec, junto con el uso de ACL, permite una configuración flexible y adaptable a diversas necesidades de comunicación. Esta integración no solo mejora la seguridad de las redes, sino que también facilita la implementación de soluciones multiprotocolo, lo que es fundamental para el crecimiento y la escalabilidad de las infraestructuras de red modernas.

Bibliografía

[1] "Túneles GRE: Características y Configuración - CCNA desde Cero". [Online] Available: <https://ccnadesdecero.es/tuneles-gre-caracteristicas-y-configuracion/>

[2] "¿Qué es IPsec? - Explicación del protocolo IPsec - AWS". [Online] Available: <https://aws.amazon.com/es/what-is/ipsec/>

[3] "Mejora la seguridad de tu VPN con el protocolo IPsec". RedesZone. [Online] Available: <https://www.redeszone.net/tutoriales/vpn/ipsec-que-es-como-funciona/>

[4] "Configuración y Autenticación de PPP - CCNA desde Cero". [Online] Available: <https://ccnadesdecero.es/configuracion-y-autenticacion-ppp/>

[5] "Configuración y Comprensión de la Autenticación CHAP PPP". Cisco. [Online] Available: https://www.cisco.com/c/es_mx/support/docs/wan/point-to-point-protocol-ppp/25647-understanding-ppp-chap.html

[6] "PPP authentication: PAP and CHAP - learncisco.net". [Online] Available: <https://www.learncisco.net/courses/icnd-2/wan-technologies/ppp-authentication-pap-and-chap.html>