1x Router: CISCO 7200 12.4(24)T5

1x Switch: CISCO IOS-XE 15.4.1T - IOU L3

2x PCs cliente: Debian 12

1x Server: FreeRADIUS 3.2.8

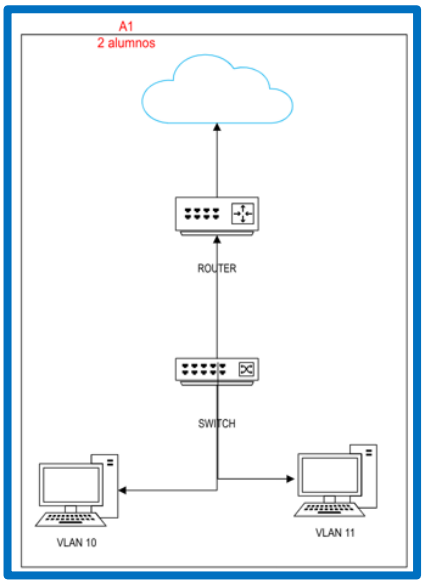
Imagen Docker de FreeRADIUS

Tabla de Direccionamiento realitzada con draw.io

GNS3 Graphical Network Simulator

Entrega - "ACT 1 – VLAN"

Tema 1 : Perímetro de la red



Explicar como hacer una configuración básica para crear una red segura con clientes, switch, router con VLANs en GNS-3 (Graphical Network Simulator)

Joan Puig, Albert Rodríguez,
1r Curso Grado de Ciber – ENTI-UB

ÍNDEX

1. Introducción

En este documento, presentaremos como realizar un conjunto de configuraciones básicas para crear una red segura muy simple. De hecho, bastará solo con añadir 4 máquinas virtuales a nuestro escenario de GNS3 junto a los siguientes dispositivos con sus modelos e imágenes.

Dispositivo	Modelo	Imagen IOS
1x Router	Cisco 7200 124-24.T5	c7200-adventerprisek9-mz.124-24.T5.image
1x Switch	Cisco IOU Switch L2	i86bi-linux-l2-adventerprisek9-15.2d.bin
2x Cliente Debian 12.6	Debian 12.6	debian.gns3a

1.1 Diagrama de la topología inicial

Seguidamente, procederemos a crear con draw.io un diagrama de la topología de red junto a su plan de direccionamiento. En este, se reflejarán cada una de las VLANs que debemos crear (VLAN 10 y 11) juntos a sus credenciales de red (ID y Nombre de la VLAN, IP de la red, Subnet Mask, Gateway y Broadcast), además del rango de direccionamiento de la DHCP Pool de cada VLAN, y finalmente: las credenciales de red que irán asignadas a cada una de las tarjetas de red de cada dispositivo.

¡IMPORTANTE! Se enfatiza mucho en el esquema un servidor FreeRadius para implementar VLANs dinámicas a los clientes de nuestra red. Sin embargo, esta funcionalidad no se ha podido implementar debido a las limitaciones de la imagen usada en el Switch. Mas info en el apartado Incidencias.

1.2 Configuración básica e incidencias técnicas

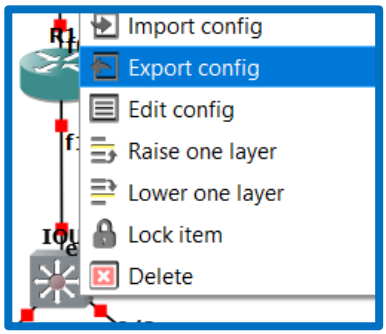
Además, redactaremos paso por paso de principio a fin como llevar a cabo cada una de las configuraciones necesarias en los dispositivos de GNS3: para que los clientes dentro de su VLAN tengan acceso a Internet a través de su IP privada recibida por DHCP.

Después, nos aseguraremos de añadir medidas de seguridad en el Switch como el Port Security (permitir solo una dirección MAC en ese puerto) y el DHCP Snooping (evitar servidores DHCP falsos).

Cabe resaltar, que también redactaremos todas aquellas configuraciones que no hemos podido llevar a cabo con éxito, como la VLAN dinámica a través de FreeRadius (asignar la VLAN a través de la dirección MAC del dispositivo), y la incompatibilidad del router CISCO de ser un servidor FreeRadius.

1.3 Archivos de configuraciones de los dispositivos

Finalmente, subiremos a la entrega de esta tarea este documento en formato PDF, el proyecto GNS3 en modo portable (extensión ".gns3") y los ficheros de configuración del router y switch CISCO.



Para ello, iremos a uno de estos dispositivos (exceptuando los clientes) y haremos click derecho > Export Config > Guardar.

Versión GNS3: 2.2.54

2. Vocabulario

A continuación, definiremos un conjunto de vocabulario básico esencial para entender con mayor profundidad el escenario de red que montaremos en GNS3.

2.1 ¿Qué es una VLAN (Virtual Local Area Network)?

Una VLAN (Virtual LAN) permite dividir una red física en varias redes lógicas / virtuales dentro de los switches. En otras palabras, a pesar de que todos los equipos estén conectados al mismo switch, una VLAN permite que los dispositivos se vean entre si a pesar de formar parte de redes diferentes.

¡RECUERDA! Una VLAN se configura principalmente en los switches y afecta a cómo se mueve el tráfico de Capa 2 (tramas) dentro de nuestra infraestructura de red.

2.2 VLAN de acceso VS VLAN troncal

Cuando un puerto del switch circula una sola VLAN, este recibe el nombre de “Access VLAN”, mientras que si el tráfico circula una o varias VLAN a la vez: se usa un enlace troncal que etiqueta el tráfico para distinguir a qué VLAN pertenece cada trama.

En este caso, en nuestro escenario usamos 2 VLANs de Acceso, ya que cada cliente tiene su propia VLAN (10 o 11), y solo nos basta dedicar un puerto a cada VLAN para conectarlo a un único PC.

2.3 ¿Qué es subnetting?

El subnetting consiste en dividir una red IP en subredes más pequeñas. Es un concepto de Capa 3 (direccionamiento IP). Por ejemplo, separar un /24 en varios /26 para asignar rangos IP distintos a diferentes áreas.

2.4 Subnetting VS VLAN

La VLAN separa dominios de broadcast en Capa 2 mientras que el subnetting separa y organiza direccionamiento en Capa 3. En la práctica, cada VLAN suele asociarse a una subred IP distinta para mantener orden y segmentación completa. Sin embargo, VLAN y Subnetting no son lo mismo, ya que podemos tener una VLAN sin cambiar el direccionamiento (no recomendable en entornos reales), y podemos hacer subnetting incluso sin VLAN (si toda la red estuviera en un único dominio de Capa 2).

¡RECUERDA! VLAN “decide por dónde circulan las tramas” en el switch; subnetting “decide cómo se direccionan y enrutan los paquetes” en IP.

2.5 Que es y cuando se usa un ·Router on a stick·

Un Router-on-a-stick (ROAS) es una técnica que permite enrutar entre varias VLAN utilizando un único enlace físico entre el router y el switch. Para que esto funcione, ese enlace se configura como troncal (trunk), de manera que puede transportar tráfico de múltiples VLAN.

En otras palabras, un “Router-on-a-stick” utiliza un solo puerto físico del router simula más varios puertos (uno por VLAN) gracias a la creación de subinterfaces y al etiquetado del enlace troncal.

De este modo, el switch empieza etiquetando las tramas por VLAN en el enlace troncal. Luego, el router recibe el tráfico etiquetado y lo procesa en la subinterfaz correspondiente. Si el destino está en otra VLAN, el router lo enruta y lo devuelve por el mismo enlace troncal, ya etiquetado con la VLAN de destino para que el switch lo entregue únicamente a los puertos que pertenecen a esa VLAN.”

¡IMPORTANTE! Router on a stick es útil en escenarios pequeños o de laboratorio, pero puede convertirse en cuello de botella si el volumen de tráfico inter-VLAN crece, porque todo el enrutamiento pasa por un único enlace y por la capacidad del router.

2.6 Switch de capa 2 vs Switch de capa 3

Un switch de Capa 2 toma decisiones basadas en direcciones MAC y mueve tramas dentro de la misma VLAN. Es decir, se usa este tipo de Switch para la conmutación local, segmentación con VLAN y control del dominio de broadcast, pero no enruta entre VLAN.

En cambio, un switch de Capa 3 (además de poder conmutar en Capa 2), puede enrutar también en Capa 3, lo que significa que toma decisiones basadas en direcciones IP y puede mover tráfico entre redes distintas. Para ello, puede crear interfaces virtuales (SVI), una por cada VLAN, asignándoles una dirección IP que actuará como gateway de esa VLAN.

De este modo, el enrutamiento entre VLAN (inter-VLAN routing) se realiza dentro del propio switch, para no depender tanto de routers externos a la hora de trabajar con VLANs.

2.7 ¿Qué es Port Security?

Port Security es una funcionalidad de seguridad en los switches que limita qué dispositivos pueden usar un puerto a través de la identificación y registro de la dirección MAC del dispositivo permitido.

Su objetivo es claro, garantizar que únicamente se conecta con ese puerto del Switch el dispositivo que tenga esa dirección MAC en su tarjeta de red. Si no es así, el dispositivo no autorizado no tendrá conectividad con ese puerto, denegando de este modo su conexión a la red.

2.8 ¿Qué es DHCP Snooping?

DHCP Snooping es una función de seguridad que protege contra servidores DHCP falsos (rogue DHCP). Esto se debe a que un servidor DHCP falso puede dar credenciales de red (IPs, SubnetMasks, Gateways, DNS…) maliciosas para dejar sin red a los clientes, o ejecutar un man-in-the-middle.

El DHCP Snooping es importante saberlo ya que el switch clasifica puertos como “confiables” (trusted) o “no confiables” (untrusted). De este modo, solo desde los puertos confiables se permiten respuestas DHCP válidas (paquetes “ACK”).

¡IMPORTANTE! Si marcamos como confiable un puerto incorrecto, abrimos la puerta a un DHCP malicioso; y si olvidamos marcar confiable el uplink hacia el DHCP real: los clientes no obtendrán IP.

2.9 VLAN dinámica y MAC Authentication Bypass (MAB)

Una VLAN dinámica asigna la VLAN al dispositivo en función de una política en lugar de dejarla fija en el puerto. Esto permite que el mismo puerto cambie de VLAN segun quién o qué se conecte.

2.10 Qué es MAB y por qué existe

MAB (MAC Authentication Bypass) se usa cuando no podemos hacer autenticación fuerte con 802.1X (control de acceso a redes basado en puertos).

En vez de autenticar un usuario, se "identifica" el dispositivo por su MAC y se consulta a un servidor de control de acceso para decidir si asignarle una VLAN concreta, o en caso contrario: denegar acceso.

¡RECUERDA! MAB es útil, pero es menos robusto que 802.1X porque la MAC se puede suplantar, siendo MAB de este modo un plan B para dispositivos que no soportan autenticación completa.

2.11 ¿Que es FreeRADIUS?

FreeRADIUS es un servidor RADIUS open source, muy usado para AAA (autenticación, autorización y contabilidad) centralizado a través de nuestra red. Este tipo de servidores se suele integrar con switches, y VPNs para decidir si un usuario o dispositivo puede entrar y con qué permisos.

Por ejemplo, en un escenario típico, el switch actúa como "cliente RADIUS" y pregunta a FreeRADIUS si el acceso está permitido. Si la política lo decide, FreeRADIUS puede devolver atributos que indiquen, por ejemplo, qué VLAN asignar dinámicamente, o qué tipo de acceso aplicar.

En cambio, con MAB: la "identidad" que se consulta suele ser la MAC del dispositivo, y la respuesta puede meterlo en una VLAN de invitados, de cuarentena o de producción según tus reglas.

¡IMPORTANTE! La seguridad del sistema depende tanto de la política como de la calidad de los identificadores: con MAB hay que asumir el riesgo de spoofing y compensarlo con segmentación, monitorización y controles adicionales.

2.12 ¿Qué es Docker?

Docker es una plataforma de contenedores que permite empaquetar aplicaciones y sus dependencias en unidades aislada. De hecho, es muy común usar Docker en redes y sistemas para desplegar servicios como FreeRADIUS, servidores web, bases de datos o herramientas de monitorización sin la necesidad de dedicarle un servidor físico o una MV.

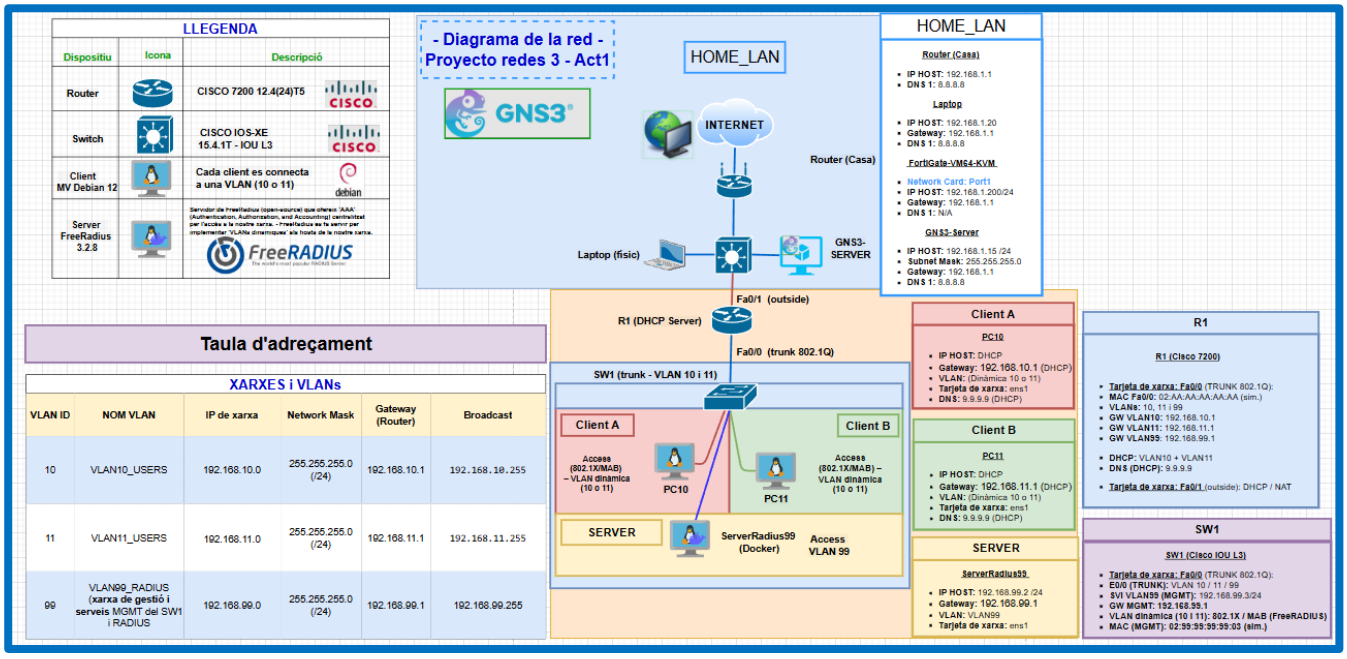
Es más, Docker crea redes virtuales para que los contenedores se comuniquen. Puedes tener redes tipo bridge (para comunicación interna en un host), host (el contenedor comparte la red del host) u overlays (en clusters). En laboratorios de redes, Docker es especialmente cómodo para simular servicios como RADIUS o DHCP y conectarlos a tu equipamiento o a un entorno virtual.

¡RECUERDA! Docker facilita el despliegue pero no elimina la necesidad de entender puertos, rutas, firewalls y segmentación. Es decir, el tráfico sigue existiendo, solo que dentro de una topología virtual.

3. Diagrama de topología inicial

Si queremos acceder al diagrama de la topología inicial, podemos acceder a él a través de esta URL.

https://app.diagrams.net/#G1E1sBBJKPGhXsnZgbV_wlVQ_m8ko8uLoZ#%7B%22pageId%22%3A%22fRRjgxpdgqEocGk969YJ%22%7D



DHCP Pools

VLAN ID	NOM VLAN	Rango DHCP (clientes)	Excluidas (Reservadas)	DNS (principal / mestre)
10	VLAN10_DHCP	192.168.10.21 – 192.168.10.254	192.168.10.1 – 192.168.10.20	9.9.9.9
11	VLAN11_DHCP	192.168.11.21 – 192.168.11.254	192.168.11.1 – 192.168.11.20	9.9.9.9

No hi ha DHCP Pool per la sala de servidors.
Totes les les IPs dels hosts s'afegiran manualment respectant el rang 192.168.99.2 -192.168.254 /24

Direccions dels dispositius

Dispositivo	ID VLAN	Interficie de xarxa	IP	Subnet Mask	Gateway	DNS	MAC Adress
Router R1	-	Fa0/0 (física)	N/A	N/A	N/A	N/A	02:AA:AA:AA:AA:AA
Router R1	10	Fa0/0.10	192.168.10.1	255.255.255.0 (/24)	N/A	N/A	Subinterficie -> MAC heretadada de R1 fa0/0
Router R1	11	Fa0/0.11	192.168.11.1	255.255.255.0 (/24)	N/A	N/A	Subinterficie -> MAC heretadada de R1 fa0/0
Router R1	99	Fa0/0.99	192.168.99.1	255.255.255.0 (/24)	N/A	N/A	Subinterficie -> MAC heretadada de R1 fa0/0
SW1 (trunk - VLAN 10 - 11)	99	SVI Vlan99 (Switch Virtual Interface)	192.168.99.3	255.255.255.0 (/24)	192.168.99.1	N/A	02:99:99:99:99:03
ServerRadius (Docker)	99	eth0	192.168.99.2	255.255.255.0 (/24)	192.168.99.1	N/A	MAC real del docker)
PC10 (Debian)	Dinàmica (10 o 11)	ens0	DHCP	DHCP	VLAN 10: 192.168.10.1 VLAN 11: 192.168.11.1 VLAN dinàmica amb FreeRadius	9.9.9.9 (DHCP)	02:10:10:10:10:10
PC11 (Debian)	Dinàmica (10 o 11)	ens1	DHCP	DHCP	VLAN 10: 192.168.10.1 VLAN 11: 192.168.11.1 VLAN dinàmica amb FreeRadius	9.9.9.9 (DHCP)	**02:11:11:11:11:11**

3.1 Introducción: Implementar configuración inicial en Switch IOU1 y R1

Antes de nada, lo primero que haremos será aplicar la configuración base del laboratorio mediante los dos scripts adjuntos, uno para IOU1 y otro para R1. La intención es que, al ejecutarlos, dejemos preparado en pocos minutos todo lo esencial del escenario: la creación de VLAN 10 y VLAN 11 en el switch, la activación del enlace trunk 802.1Q hacia el router, la asignación de los puertos de acceso para los clientes, y las medidas de control como DHCP Snooping en IOU1.

En paralelo, el script de R1 levanta la interfaz WAN hacia el NAT de GNS3, crea las subinterfaces dot1Q que actúan como puerta de enlace de cada VLAN, habilita el servicio DHCP para que los clientes obtengan IP automáticamente y completa la salida a Internet con la ruta por defecto y la traducción NAT/PAT, aplicando además las ACLs que controlan el tráfico y evitan comportamientos no deseados.

Durante esta fase conviene ejecutar los scripts desde consola en modo de configuración y, si el terminal de GNS3 recorta el contenido al pegar, aplicarlos en bloques pequeños manteniendo el orden original. Una vez guardada la configuración con write memory y con los dispositivos reiniciados si fuera necesario, ya tendremos el entorno listo para pasar a la verificación detallada, donde confirmaremos paso a paso que cada componente del escenario está operativo.

3.2 Verificación de la implementación en el Switch IOU1

En este apartado dejamos constancia de que el escenario está correctamente levantado en GNS3, siguiendo una verificación lógica: primero comprobamos que el switch IOU1 separa el tráfico en VLAN 10 y VLAN 11 y que el enlace hacia el router transporta esas VLAN etiquetadas. Después, validamos que R1 actúa como router-on-a-stick con sus subinterfaces levantadas; y finalmente confirmamos que existe salida real a Internet gracias a NAT/PAT y que las ACLs están aplicándose de forma efectiva.

3.2.1 VLANs operativas y puertos de acceso bien asignados

En la captura de show vlan brief se observa que VLAN10_CLIENT_A está activa y asociada al puerto Ethernet0/1, mientras que VLAN11_CLIENT_B está activa y asociada al puerto Ethernet0/2. Esto confirma que cada cliente queda encapsulado en su dominio de broadcast correspondiente y que la separación en capa 2 se está aplicando tal y como se diseñó.

```
IOU1#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Et0/3, Et1/0, Et1/1, Et1/2
                                                Et1/3, Et2/0, Et2/1, Et2/2
                                                Et2/3, Et3/0, Et3/1, Et3/2
                                                Et3/3
10   VLAN10_CLIENT_A                  active    Et0/1
11   VLAN11_CLIENT_B                  active    Et0/2
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
IOU1#
```

3.2.2 Trunk 802.1Q funcionando hacia el router

En la salida de show interfaces trunk se ve que Ethernet0/0 está en estado trunking con 802.1Q y que las VLAN 10 y 11 aparecen permitidas y activas sobre el troncal. Con esto queda validado el elemento clave del router-on-a-stick: el switch está enviando tráfico etiquetado hacia R1 y no hay bloqueo por configuración de trunk.

```
IOU1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Et0/0       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Et0/0       1-4094

Port        Vlans allowed and active in management domain
Et0/0       1,10-11

Port        Vlans in spanning tree forwarding state and not pruned
Et0/0       1,10-11
IOU1#
```

3.2.3 DHCP Snooping activo, trunk de confianza y Option 82 desactivado

En la captura de show ip dhcp snooping se confirma que DHCP Snooping está habilitado y operativo en las VLAN 10-11, que el puerto Ethernet0/0 figura como trusted y que la inserción de Option 82 está desactivada. Con esto mantenemos la protección contra servidores DHCP no autorizados en los puertos de usuario, pero evitamos añadir Option 82 para no introducir complejidad extra en el laboratorio. Entonces, el punto crítico aquí es que el trunk hacia el router es de confianza, lo que permite que las respuestas DHCP legítimas provenientes de R1 no queden bloqueadas.

```
IOU1#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10-11
DHCP snooping is operational on following VLANs:
10-11
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is disabled
   circuit-id default format: vlan-mod-port
   remote-id: aabb.cc00.0100 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                  Trusted    Allow option    Rate limit (pps)
-----------------------    -------    ------------    ----------------
Ethernet0/0                yes        yes             unlimited
  Custom circuit-ids:
IOU1#
```

3.3 Verificación de la implementación en el R1

3.3.1 Subinterfaces levantadas y WAN con IP por DHCP

En la salida de show ip interface brief se aprecia que FastEthernet0/0 está up/up con IP obtenida por DHCP en la red 192.168.122.0/24, lo que valida la conectividad de R1 con el NAT de GNS3. A la vez, las subinterfaces FastEthernet1/0.10 y FastEthernet1/0.11 están up/up con 192.168.10.1 y 192.168.11.1, confirmando que el router está proporcionando el gateway de cada VLAN y que el router-on-a-stick está activo.

```
R1#show ip interface brief
Any interface listed with OK? value "NO" does not have a valid configuration

Interface                  IP-Address      OK? Method Status                Prot
ocol
FastEthernet0/0            192.168.122.178 YES DHCP   up                    up

FastEthernet1/0            unassigned      YES manual up                    up

FastEthernet1/0.10         192.168.10.1    YES manual up                    up

FastEthernet1/0.11         192.168.11.1    YES manual up                    up

NVI0                       unassigned      NO  unset  up                    up

R1#
*Feb 10 22:23:18.683: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

3.3.2 Evidencia definitiva: NAT/PAT y ACLs en funcionamiento

En la salida donde aparece show ip nat translations se ve una traducción activa desde una IP interna (inside local, por ejemplo 192.168.10.21) hacia la IP WAN del router (inside global, 192.168.122.178) con puertos distintos. Esto demuestra tráfico real saliendo desde la VLAN hacia Internet y siendo traducido por NAT overload (PAT), que era el objetivo del enlace NAT1.

```
R1#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 192.168.122.178:56498 192.168.10.21:56498 92.113.12.77:123 92.113.12.77:123
R1#show access-lists
Standard IP access list NAT_INSIDE
    10 permit 192.168.10.0, wildcard bits 0.0.0.255 (2 matches)
    20 permit 192.168.11.0, wildcard bits 0.0.0.255
Extended IP access list VLAN10_IN
    10 permit udp any eq bootpc any eq bootps
    20 deny ip 192.168.10.0 0.0.0.255 192.168.11.0 0.0.0.255
    30 permit ip 192.168.10.0 0.0.0.255 any (3 matches)
Extended IP access list VLAN11_IN
    10 permit udp any eq bootpc any eq bootps
    20 deny ip 192.168.11.0 0.0.0.255 192.168.10.0 0.0.0.255
    30 permit ip 192.168.11.0 0.0.0.255 any
R1#
```

En esa misma evidencia, show access-lists muestra coincidencias en la ACL estándar NAT_INSIDE y en las ACLs extendidas de entrada por VLAN, lo que confirma que no están solo “definidas”, sino que están filtrando y permitiendo tráfico de verdad. Aunque show ip dhcp binding salga vacío en ese instante, la existencia de traducciones NAT desde 192.168.10.21 prueba que el host tiene IP, gateway y conectividad efectiva, y que el camino completo hasta Internet está operativo.

3.3.3 Tabla de rutas de R1 y confirmación de la salida a Internet

En la captura correspondiente al comando show ip route se observa que R1 tiene aprendidas y activas tres rutas conectadas, junto con una ruta por defecto que actúa como “gateway of last resort”. En primer lugar aparecen como redes directamente conectadas la 192.168.122.0/24 asociada a FastEthernet0/0, que es el enlace WAN hacia el NAT de GNS3, y las redes 192.168.10.0/24 y 192.168.11.0/24 asociadas a las subinterfaces FastEthernet1/0.10 y FastEthernet1/0.11, que representan las VLAN internas del laboratorio.

A continuación, la línea que indica “Gateway of last resort is 192.168.122.1 to network 0.0.0.0” y la entrada S* 0.0.0.0/0 via 192.168.122.1 confirman que R1 dispone de una ruta por defecto válida hacia el exterior. Esta es la pieza imprescindible para que el tráfico que no pertenezca a las redes internas (por ejemplo, destinos en Internet) no se descarte, sino que se reenvíe hacia el NAT de GNS3, permitiendo completar el camino de salida que después será traducido mediante NAT/PAT.

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.122.1 to network 0.0.0.0

C    192.168.122.0/24 is directly connected, FastEthernet0/0
C    192.168.10.0/24 is directly connected, FastEthernet1/0.10
C    192.168.11.0/24 is directly connected, FastEthernet1/0.11
S*   0.0.0.0/0 [1/0] via 192.168.122.1
R1#
```

3.4 Verificar que los clientes tienen acceso a Internet (Cliente A & B)

Una vez tenemos nuestro Switch (IOU1) y nuestro router correctamente configurado, ya podemos encender nuestras dos maquinas cliente con Debian. De este modo, a ambos los conectaremos en una VLAN distinta (10 -> Cliente A y 11 -> Cliente B) y tendrán acceso a Internet gracias a recibir una IP por DHCP a través del Router1. Es más, este también les brinda Internet al aplicar políticas NAT.

Cabe resaltar, que para diferenciar ambas maquinas, es mejor canviarles el hostname y luego reiniciar la MV para aplicar dichos cambios en los clientes.

```
debian@debian:~$ sudo hostnamectl set-hostname clienteA
debian@debian:~$ sudo nano /etc/hosts
```

CLIENT-A - PuTTY

```
  GNU nano 7.2                                /etc/hosts
127.0.0.1       localhost
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

127.0.1.1       clienteA
```

3.4.1 Obtención automática de dirección IP mediante DHCP + Gateway

En la primera captura se aprecia la ejecución de sudo dhclient -r ens4 seguida de sudo dhclient -v ens4, lo que fuerza al cliente a liberar cualquier concesión previa y solicitar una nueva. La salida confirma el ciclo completo del protocolo: el cliente envía un DHCPDISCOVER, recibe un DHCPOFFER desde 192.168.10.1, realiza el DHCPREQUEST y finalmente obtiene el DHCPACK. El resultado es que el cliente queda configurado con la dirección 192.168.10.21, validando que el servidor DHCP en R1 está operativo y que el tráfico DHCP atraviesa correctamente el switch y el trunk.

```
debian@clienteA:~$ sudo dhclient -r ens4
debian@clienteA:~$ sudo dhclient -v ens4
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/ens4/0c:36:ae:33:00:00
Sending on   LPF/ens4/0c:36:ae:33:00:00
Sending on   Socket/fallback
DHCPDISCOVER on ens4 to 255.255.255.255 port 67 interval 8
DHCPOFFER of 192.168.10.21 from 192.168.10.1
DHCPREQUEST for 192.168.10.21 on ens4 to 255.255.255.255 port 67
DHCPACK of 192.168.10.21 from 192.168.10.1
bound to 192.168.10.21 -- renewal in 38966 seconds.
debian@clienteA:~$
```

```
debian@clienteA:~$ ping -c 3 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=255 time=11.8 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=255 time=9.43 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=255 time=2.96 ms

--- 192.168.10.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 2.959/8.060/11.789/3.733 ms
debian@clienteA:~$ ip route
default via 192.168.10.1 dev ens4
192.168.10.0/24 dev ens4 proto kernel scope link src 192.168.10.21
```

Es mas, en la 2na captura, donde se ejecuta ping -c 3 192.168.10.1 se observan respuestas correctas con 0% de pérdida. Esto confirma que el cliente alcanza su puerta de enlace dentro de la VLAN 10 y que el enlace físico, la asignación a la VLAN y el encaminamiento local hacia la subinterfaz del router funcionan con normalidad.

3.4.2 Ruta por defecto aprendida y salida a Internet por IP y DNS

En la salida de ip route se observa la existencia de una ruta por defecto default via 192.168.10.1 dev ens4, lo que confirma que el cliente utiliza R1 como salida fuera de su red local. A continuación, el ping -c 3 8.8.8.8 devuelve respuesta sin pérdida, demostrando conectividad real hacia Internet por dirección IP y, por extensión, que el enrutamiento y la traducción NAT/PAT están operativos.

```
debian@clienteA:~$ ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=24.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=24.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=26.7 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 24.747/25.433/26.657/0.867 ms
```

En las capturas donde se ejecuta ping -c 3 google.com se confirma que el nombre se resuelve a una IP pública y que se reciben respuestas con normalidad.

Esto valida que el cliente no solo llega a Internet, sino que además tiene servidores DNS funcionales entregados por DHCP y puede acceder a destinos externos usando nombres de dominio.

```
debian@clienteA:~$ ping -c google.com
ping: invalid argument: 'google.com'
debian@clienteA:~$ ping -c3 google.com
PING google.com (142.250.181.110) 56(84) bytes of data.
64 bytes from fjr04s08-in-f14.1e100.net (142.250.181.110): icmp_seq=1
me=40.9 ms
64 bytes from fjr04s08-in-f14.1e100.net (142.250.181.110): icmp_seq=2
me=24.9 ms
64 bytes from fjr04s08-in-f14.1e100.net (142.250.181.110): icmp_seq=3
me=38.4 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 24.870/34.697/40.857/7.022 ms
debian@clienteA:~$
```

```
debian@clienteB:~$ ping -c3 192.168.11.1
PING 192.168.11.1 (192.168.11.1) 56(84) bytes of data.
64 bytes from 192.168.11.1: icmp_seq=1 ttl=255 time=16.6 ms
64 bytes from 192.168.11.1: icmp_seq=2 ttl=255 time=8.63 ms
64 bytes from 192.168.11.1: icmp_seq=3 ttl=255 time=3.10 ms

--- 192.168.11.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 3.101/9.427/16.555/5.521 ms
debian@clienteB:~$ ping cloudflare.com
PING cloudflare.com (104.16.133.229) 56(84) bytes of data.
64 bytes from 104.16.133.229 (104.16.133.229): icmp_seq=1 ttl=53 time=41.2 ms
64 bytes from 104.16.133.229 (104.16.133.229): icmp_seq=2 ttl=53 time=29.6 ms
```

3.5 Port-Security en IOU1 y demostración del bloqueo por cambio de MAC

Port-Security es una medida de seguridad de capa 2 que limita qué direcciones MAC pueden utilizar un puerto. En este laboratorio lo aplicamos en los puertos de acceso para que cada puerto acepte solo la MAC del cliente esperado; si aparece otra MAC distinta (por cambio de equipo o suplantación), el switch detecta la violación y, en modo “shutdown”, bloquea el puerto dejando constancia del evento.

3.5.1 Por qué activamos “logging synchronous” en consola

Cuando ocurre una violación, el switch genera mensajes de log en tiempo real con el puerto afectado y la MAC causante. Sin sincronización, esos mensajes pueden mezclarse con lo que estamos escribiendo y la captura queda confusa. Con logging synchronous, los logs se muestran ordenados y el prompt se mantiene limpio, facilitando ver y capturar el momento exacto del bloqueo.

Aplicamos este comando a IOU1:

```
enable
conf t
line con 0
 logging synchronous
end
```

3.5.2 Estado inicial de los puertos y preparación de la prueba

Antes de provocar la violación, verificamos en IOU1 el estado de Port-Security en los puertos de acceso con show port-security interface ethernet0/1 y show port-security interface ethernet0/2. En las capturas iniciales se observa “Port Security: Enabled” y “Port Status: Secure-up”, junto con “Violation Mode: Shutdown”.

Esto confirma que la política está activa y que, si el switch detecta una MAC no autorizada en un puerto que ya tiene una MAC segura, aplicará la acción más estricta: poner el puerto en err-disable. En este mismo punto también vemos que todavía no hay direcciones sticky aprendidas (“Sticky MAC Addresses: 0”), lo cual es coherente cuando acabamos de limpiar el aprendizaje o aún no se ha generado tráfico suficiente para fijar la MAC.

```
IOU1#show port-security interface ethernet0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0
```

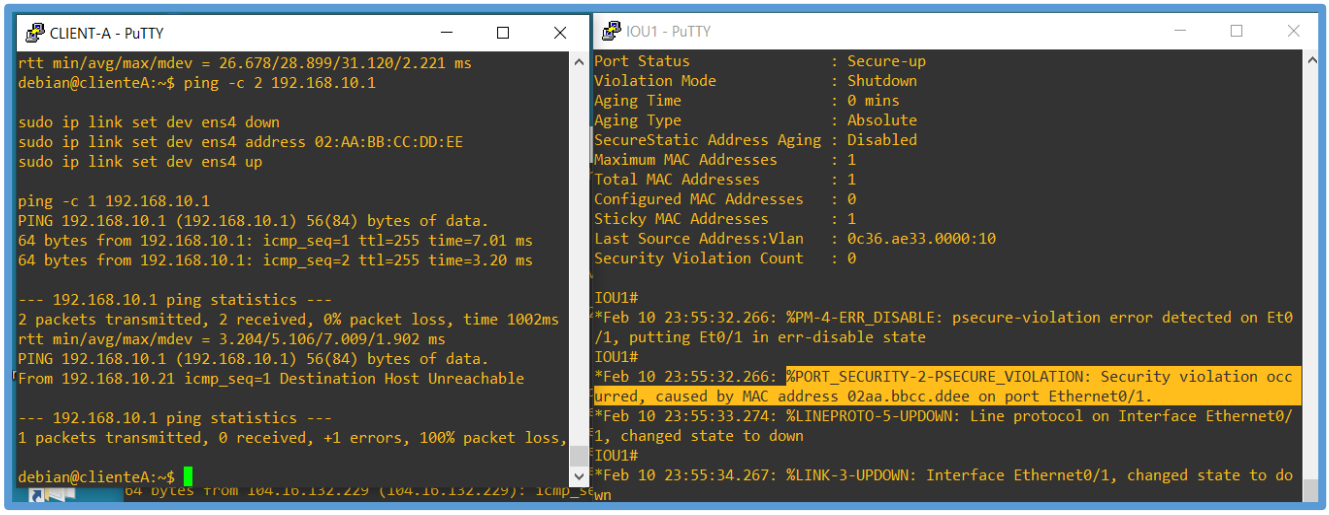
```
IOU1#show port-security interface ethernet0/2
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0

IOU1#show port-security address
              Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                          Ports   Remaining Age
                                                                    (mins)
----    -----------       ----                          -----   -------------
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 4096
IOU1#
```

3.5.3 suplantación desde el Cliente A de su MAC & Evidencial del bloqueo (err-disable)

Con el puerto en estado seguro, desde el Cliente A generamos conectividad normal hacia el gateway y, a continuación, forzamos un cambio de MAC en la interfaz ens4 mediante ip link set. En la captura del Cliente A se aprecia que la MAC activa pasa a ser 02:aa:bb:cc:dd:ee, mientras que la dirección original del adaptador permanece identificada como “permaddr”.

Este cambio es la base de la demostración: el puerto está configurado para aceptar una sola MAC, y la aparición de una MAC distinta en el mismo puerto debe considerarse una violación. El efecto práctico se ve en la conectividad: el ping al gateway deja de responder en el momento en que el switch aplica la política y bloquea el puerto.

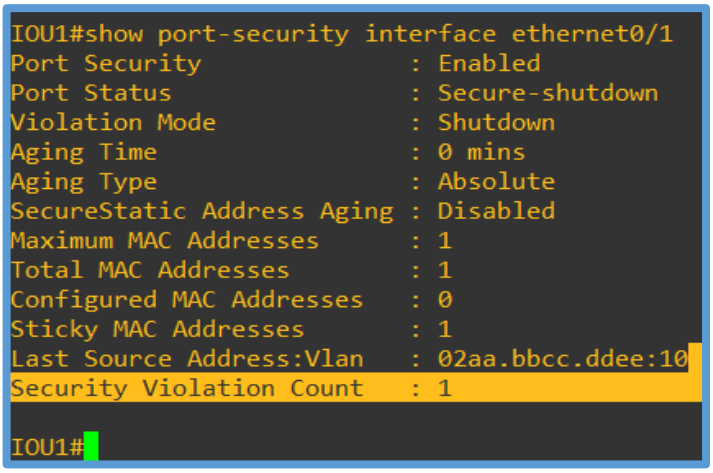


La evidencia más concluyente aparece cuando en la consola de IOU1 se registran los mensajes del sistema. En la captura se ve PORT_SECURITY-2-PSECURE_VIOLATION, indicando que ha ocurrido una violación y mostrando la MAC causante y el puerto afectado.

Justo después aparece PM-4-ERR_DISABLE: psecure-violation ... putting Et0/1 in err-disable state, confirmando que el switch ejecuta la acción asociada al modo “Shutdown” y coloca el puerto en err-disable. Los mensajes LINEPROTO-5-UPDOWN y LINK-3-UPDOWN completan la prueba, porque reflejan el cambio de estado real del enlace a down, lo que explica directamente por qué el cliente deja de alcanzar su gateway.

3.5.4 Confirmación persistente: estado del puerto y contador de violaciones

Después del evento, validamos el resultado con show port-security interface ethernet0/1. En la captura se aprecia que el “Port Status” pasa a “Secure-shutdown”, que equivale a puerto bloqueado por violación en modo shutdown. También se observa que “Last Source Address” coincide con la MAC alterada 02aa.bbcc.ddee en VLAN 10 y que el “Security Violation Count” se incrementa a 1.



.

3.5.5 Recuperación controlada del servicio y retorno a la MAC original

Una vez documentada el incidente, restauramos el funcionamiento normal. En el Cliente A devolvemos la MAC a la dirección original indicada por “permaddr” (0c:36:ae:33:00:00) y lo confirmamos con ip link show ens4.

En IOU1 recuperamos el puerto aplicando shutdown y no shutdown sobre Ethernet0/1 y, cuando es necesario, limpiamos el aprendizaje sticky para que el puerto vuelva a aprender la MAC correcta.

Las capturas finales reflejan el retorno a la operatividad: el puerto vuelve a estar disponible y el ping al gateway vuelve a responder, demostrando que Port-Security bloquea únicamente ante una MAC no autorizada y permite el tráfico cuando el cliente vuelve a presentar la identidad esperada.

```
debian@clienteA:~$ ip link show ens4
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_co
del state UP mode DEFAULT group default qlen 1000
    link/ether 02:aa:bb:cc:dd:ee brd ff:ff:ff:ff:ff:ff permaddr
 0c:36:ae:33:00:00
    altname enp0s4
debian@clienteA:~$
```

```
debian@clienteA:~$ sudo ip link set dev ens4 down
sudo ip link set dev ens4 address 0c:36:ae:33:00:00
sudo ip link set dev ens4 up
debian@clienteA:~$ ip link show ens4
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
 group default qlen 1000
    link/ether 0c:36:ae:33:00:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
debian@clienteA:~$
```

```
IOU1#enable
IOU1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
IOU1(config)#interface ethernet0/1
IOU1(config-if)# shutdown
IOU1(config-if)# !
IOU1(config-if)# ! Netejar l'aprenentatge sticky del port
IOU1(config-if)# no switchport port-security mac-address sticky
IOU1(config-if)# switchport port-security mac-address sticky
IOU1(config-if)# !
IOU1(config-if)# no shutdown
IOU1(config-if)#end
IOU1#wr mem
Building configuration...
Compressed configuration from 2367 bytes to 1444 bytes[OK]
IOU1#
*Feb 11 00:05:18.539: %SYS-5-CONFIG_I: Configured from console by console
IOU1#
*Feb 11 00:05:20.543: %LINK-3-UPDOWN: Interface Ethernet0/1, changed state to up
*Feb 11 00:05:21.549: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/1
, changed state to up
IOU1#
```

```
debian@clienteA:~$ ping -c 2 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
From 192.168.10.21 icmp_seq=1 Destination Host Unreachable
From 192.168.10.21 icmp_seq=2 Destination Host Unreachable

--- 192.168.10.1 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1017ms
pipe 2
debian@clienteA:~$ ping -c 2 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=255 time=14.9 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=255 time=10.9 ms
```

4. Limitaciones

4.1 Incompatibilidad del Router como Servidor RADIUS

Se intentó configurar el Router Gateway (Cisco c7200, IOS 12.4-24.T5) para actuar como servidor RADIUS local utilizando el comando radius-server local.

El CLI del router rechazó el comando con el error ***% Invalid input detected at '^' marker***, indicando que esta imagen específica de IOS (Advanced Enterprise) no tiene compilada la característica de servidor local AAA.

```
R1(config)#radius-server local
                           ^
% Invalid input detected at '^' marker.
```

4.2 Asignación Dinámica de VLANs (MAB) en Cisco IOU

Se intentó implementar la funcionalidad de **MAC Authentication Bypass (MAB)** para asignar dinámicamente a los clientes a las VLANs 10 u 11 basándose en su dirección MAC mediante un servidor RADIUS externo (Docker FreeRADIUS). Sin embargo, esta característica no pudo ponerse en producción debido a limitaciones inherentes a la imagen de emulación **Cisco IOU L2 (15.1a)** utilizada en el laboratorio:

- Durante las pruebas de depuración (debug mab all), el switch devolvió repetidamente el error interno Invalid EVT 9 from EAP. Este es un error documentado en ciertas versiones emuladas de IOU donde el proceso de autenticación entra en un bucle de reinicio al intentar procesar la dirección MAC, impidiendo el envío de la solicitud al servidor RADIUS.

```
Radius server fail-over debugging is off
Radius elog debugging debugging is off
IOU1#debug mab all
All MAC Authentication Bypass debugging is on
IOU1#debug dot1x all
*Feb  5 21:16:44.008: %SYS-5-CONFIG_I: Configured from console by console
IOU1#debug dot1x all
All Dot1x debugging is on
IOU1#
*Feb  5 21:17:55.418: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/2, changed state to up
IOU1#
*Feb  5 21:17:55.418: mab-ev(Et0/2): Received MAB context create from AuthMgr
*Feb  5 21:17:55.418: mab-ev(Et0/2): Created MAB client context 0x62000001
*Feb  5 21:17:55.418:     mab : initial state mab_initialize has enter
*Feb  5 21:17:55.418: mab-ev(Et0/2): Sending create new context event to EAP from MAB for 0x62000001 (0000.0000.0000)
*Feb  5 21:17:55.419: mab-ev: Invalid EVT 9 from EAP
*Feb  5 21:17:55.419: mab-sm(Et0/2): Received event 'MAB_START' on handle 0x62000001
*Feb  5 21:17:55.419:     mab : during state mab_initialize, got event 4(mabStart)
*Feb  5 21:17:55.419: @@@ mab : mab_initialize -> mab_acquiring
IOU1#
```

- El puerto bloqueaba el tráfico entrante esperando autenticación, pero el cliente (Debian) no enviaba tráfico al no detectar enlace activo. El comando de mitigación estándar authentication control-direction in no tuvo el efecto esperado en esta versión virtualizada, resultando en un puerto inoperativo.