



Septiembre de 2022

Laboratorio 2

Monitoreo y supervisión de redes

Datos Personales:

Nombre:

Número estudiante:

Fecha:

INTRODUCCIÓN

Topología

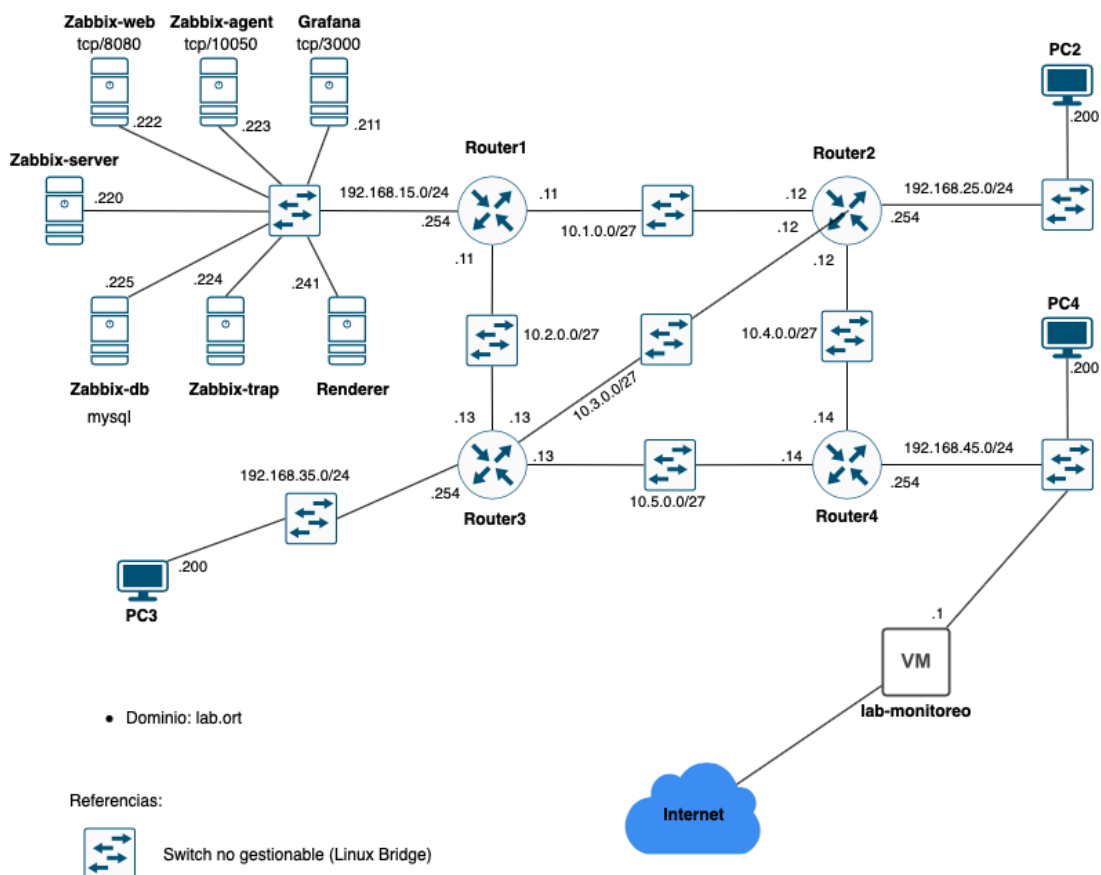


Imagen 1: Topología de red

La conexión se realiza mediante SSH a la máquina virtual lab-monitoreo. Por defecto se presenta en el puerto tcp/2222 de la interface de red de la computadora personal del estudiante

Usuario	estudiante
Password	estudiante
Puerto	2222

PRÁCTICOS

Práctico 1

El objetivo de este práctico es ver la configuración remota de dispositivos con SNMP

1. Para comenzar ejecutemos, desde la consola de lab-monitoreo
 - *lab2*
2. Para configurar una variable específica de un host utilizaremos el comando `snmpset`
Ejemplo: `snmpset -v2c -c micomunidad router1.lab.ort sysContact.0 s sysadmin@lab.ort`
 - (a) `snmpset`: es el comando que utilizaremos para configurar variables de un host
 - (b) `-v`: define la versión de SNMP en la que queremos trabajar
 - (c) `2c`: versión 2c (puede ser 1, 2c, 3)
 - (d) `-c`: indica la comunidad a utilizar
 - (e) `micomunidad`: nombre de la comunidad (debe coincidir con lo configurado en el agente SNMP)
 - (f) `router1.lab.ort`: nombre del host al que intentaremos configurar
 - (g) `sysContact.0`: nombre de una variable
 - (h) `s`: indica que el valor es un string. Hay diferentes opciones según el tipo de variable (por ejemplo, `i`: entero, `a`: dirección IP, `d`: decimal, etc.)
 - (i) `sysadmin@lab.ort`: valor a configurar en la variable indicada
3. Definamos una comunidad de escritura/lectura llamada "micomunidad" para `router4.lab.ort`
 - En el host `router4.lab.ort` editemos el archivo `/etc/snmp/snmpd.conf`, en la sección `ACCESS CONTROL` agreguemos la siguiente línea de configuración

Código 1: `/etc/snmp/snmpd.conf`

```
1 #####
2 #
3 # ACCESS CONTROL
4 #
5
6 rwcommunity micomunidad default
```

4. Reiniciemos el servicio `snmpd`
 - `service snmpd restart`
5. Antes de empezar veremos como obtener algunos datos de interfaces de red

- Lo primero a saber es que el sistema operativo asigna un número de índice para las interfaces, se pueden consultar estos valores con la variable ifIndex

snmpwalk -v2c -c micomunidad router4.lab.ort ifIndex

Código 2: snmpwalk -v2c -c micomunidad router4.lab.ort ifIndex

```

1 (ansible -2.9.0) estudiante@lab-monitoreo:~/lab_monitoreo$
  snmpwalk -v2c -c micomunidad router4.lab.ort ifIndex
2 IF-MIB::ifIndex.1 = INTEGER: 1
3 IF-MIB::ifIndex.17 = INTEGER: 17
4 IF-MIB::ifIndex.25 = INTEGER: 25
5 IF-MIB::ifIndex.33 = INTEGER: 33

```

Con este resultado podemos asumir que el dispositivo consultado tiene 4 interfaces de red

- Verifiquemos ahora los nombres de estas interfaces con la variable ifName

snmpwalk -v2c -c micomunidad router4.lab.ort ifName

Código 3: snmpwalk -v2c -c micomunidad router4.lab.ort ifName

```

1 (ansible -2.9.0) estudiante@lab-monitoreo:~/lab_monitoreo$
  snmpwalk -v2c -c micomunidad router4.lab.ort ifName
2 IF-MIB::ifName.1 = STRING: lo
3 IF-MIB::ifName.17 = STRING: eth1
4 IF-MIB::ifName.25 = STRING: eth2
5 IF-MIB::ifName.33 = STRING: eth0

```

Notemos que por ejemplo ifName.17 corresponde a eth1. Esto quiere decir que el número de índice 17 es el que debemos usar de aquí en adelante para consultar o modificar parametros de la interface eth1

- Consultemos ahora la dirección MAC de la interface eth1

snmpwalk -v2c -c micomunidad router4.lab.ort ifPhysAddress.17

Código 4: snmpwalk -v2c -c micomunidad router4.lab.ort ifPhysAddress.17

```

1 (ansible -2.9.0) estudiante@lab-monitoreo:~/lab_monitoreo$
  snmpwalk -v2c -c micomunidad router4.lab.ort ifPhysAddress
  .17
2 IF-MIB::ifPhysAddress.17 = STRING: 2:42:a:4:0:e

```

6. **Luego de repasar la lógica utilizada para operar sobre interfaces de red y sabiendo que existe una variable ifAlias que permite definir una descripción. Se solicita verificar como se conecta cada interface de router4.lab.ort y asignar una descripción respetando el siguiente criterio:**

- **lo:** Interface loopback
- **eth0:** Conexion con {Nombre de dispositivo}
- **eth1:** Conexion con {Nombre de dispositivo}
- **eth2:** Conexion con {Nombre de dispositivo}

7. **Realice consultas SNMP para verificar los valores de ifAlias que acaba de configurar y guarde el resultado:**

Práctico 2

Aplicaremos en esta instancia lo visto hasta el momento de SNMP

1. Realice la siguiente configuración para router3.lab.ort:
 - (a) sysContact devuelva su mail personal
 - (b) sysContact sea editable por operación SNMP SET
 - (c) sysLocation retorne Universidad ORT - Laboratorio
 - (d) sysLocation NO sea editable por SNMP SET
 - (e) Defina una comunidad "public" con las siguientes características:
 - Solo lectura
 - Que permita consultar únicamente los valores de sysDescr, sysContact, sysName y sysLocation
 - Que se pueda consultar desde cualquier IP
 - (f) Defina una comunidad "comunidadORT" con las siguientes características:
 - Acceso de lectura y escritura
 - Que permita consultar cualquier valor
 - Que se pueda consultar solo desde el prefijo 192.168.45.0/24
2. Luego de realizada la configuración, validemos el correcto funcionamiento
 - (a) Desde lab-monitoreo.lab.ort
 - *snmpwalk -v2c -c public router3.lab.ort*

Código 5: snmpwalk -v2c -c public router3.lab.ort

```
1 (ansible -2.9.0) estudiante@lab-monitoreo:~$ snmpwalk -v2c
   -c public router3.lab.ort
2 SNMPv2-MIB::sysDescr.0 = STRING: Linux router3 4.15.0-20-
   generic #21-Ubuntu SMP Tue Apr 24 06:16:15 UTC 2018
   x86_64
3 SNMPv2-MIB::sysContact.0 = STRING: jg269703@fi365.ort.edu.
   uy
4 SNMPv2-MIB::sysName.0 = STRING: router3
5 SNMPv2-MIB::sysLocation.0 = STRING: Universidad ORT -
   Laboratorio
```

```
6 SNMPv2-MIB::sysLocation.0 = No more variables left in this
   MIB View (It is past the end of the MIB tree)
```

- *snmpwalk -v2c -c comunidadORT router3.lab.ort ifDescr*

Código 6: snmpwalk -v2c -c comunidadORT router3.lab.ort ifDescr

```
1 (ansible -2.9.0) estudiante@lab-monitoreo:~$ snmpwalk -v2c
   -c comunidadORT router3.lab.ort ifDescr
2 IF-MIB::ifDescr.1 = STRING: lo
3 IF-MIB::ifDescr.19 = STRING: eth2
4 IF-MIB::ifDescr.31 = STRING: eth3
5 IF-MIB::ifDescr.39 = STRING: eth0
6 IF-MIB::ifDescr.49 = STRING: eth1
```

- *snmpset -v2c -c public router3.lab.ort sysLocation.0 s Laboratorio*

Código 7: snmpwalk -v2c -c comunidadORT router3.lab.ort ifDescr

```
1 (ansible -2.9.0) estudiante@lab-monitoreo:~$ snmpset -v2c -
   c public router3.lab.ort sysLocation.0 s Laboratorio
2 Error in packet.
3 Reason: noAccess
4 Failed object: SNMPv2-MIB::sysLocation.0
```

- *snmpset -v2c -c public router3.lab.ort sysContact.0 s sysadmin@lab.ort*

Código 8: snmpset -v2c -c public router3.lab.ort sysContact.0 s sysadmin@lab.ort

```
1 (ansible -2.9.0) estudiante@lab-monitoreo:~$ snmpset -v2c -
   c public router3.lab.ort sysContact.0 s sysadmin@lab.
   ort
2 Error in packet.
3 Reason: noAccess
4 Failed object: SNMPv2-MIB::sysContact.0
```

- *snmpset -v2c -c comunidadORT router3.lab.ort sysLocation.0 s Laboratorio*

Código 9: snmpset -v2c -c comunidadORT router3.lab.ort sysLocation.0 s Laboratorio

```
1 (ansible -2.9.0) estudiante@lab-monitoreo:~$ snmpset -v2c -
   c comunidadORT router3.lab.ort sysLocation.0 s
   Laboratorio
2 Error in packet.
3 Reason: notWritable (That object does not support
   modification)
4 Failed object: SNMPv2-MIB::sysLocation.0
```

- *snmpset -v2c -c comunidadORT router3.lab.ort sysContact.0 s sysadmin@lab.ort*

Código 10: `snmpset -v2c -c comunidadORT router3.lab.ort sysContact.0 s sysadmin@lab.ort`

```

1 (ansible -2.9.0) estudiante@lab-monitoreo:~$ snmpset -v2c -
  c comunidadORT router3.lab.ort sysContact.0 s
  sysadmin@lab.ort
2 SNMPv2-MIB::sysContact.0 = STRING: sysadmin@lab.ort

```

(b) Desde `zabbix-server.lab.ort`

- `snmpwalk -v2c -c public router3.lab.ort`

Código 11: `snmpwalk -v2c -c public router3.lab.ort`

```

1 zabbix-server:~# snmpwalk -v2c -c public router3.lab.ort
2 SNMPv2-MIB::sysDescr.0 = STRING: Linux router3 4.15.0-20-
  generic #21-Ubuntu SMP Tue Apr 24 06:16:15 UTC 2018
  x86_64
3 SNMPv2-MIB::sysContact.0 = STRING: jg269703@fi365.ort.edu.
  uy
4 SNMPv2-MIB::sysName.0 = STRING: router3
5 SNMPv2-MIB::sysLocation.0 = STRING: Universidad ORT -
  Laboratorio
6 SNMPv2-MIB::sysLocation.0 = No more variables left in this
  MIB View (It is past the end of the MIB tree)

```

- `snmpwalk -v2c -c comunidadORT router3.lab.ort ifDescr`

Código 12: `snmpwalk -v2c -c comunidadORT router3.lab.ort ifDescr`

```

1 zabbix-server:~# snmpwalk -v2c -c comunidadORT router3.lab
  .ort ifDescr
2 Timeout: No Response from router3.lab.ort

```

- `snmpset -v2c -c public router3.lab.ort sysLocation.0 s Laboratorio`

Código 13: `snmpset -v2c -c public router3.lab.ort sysLocation.0 s Laboratorio`

```

1 zabbix-server:~# snmpset -v2c -c public router3.lab.ort
  sysLocation.0 s Laboratorio
2 Error in packet.
3 Reason: noAccess
4 Failed object: SNMPv2-MIB::sysLocation.0

```

- `snmpset -v2c -c public router3.lab.ort sysContact.0 s sysadmin@lab.ort`

Código 14: `snmpset -v2c -c public router3.lab.ort sysContact.0 s sysadmin@lab.ort`

```

1 zabbix-server:~# snmpset -v2c -c public router3.lab.ort
  sysContact.0 s sysadmin@lab.ort
2 Error in packet.
3 Reason: noAccess
4 Failed object: SNMPv2-MIB::sysContact.0

```

- `snmpset -v2c -c comunidadORT router3.lab.ort sysLocation.0 s Laboratorio`

Código 15: `snmpset -v2c -c comunidadORT router3.lab.ort sysLocation.0 s Laboratorio`

```

1 zabbix-server:~# snmpset -v2c -c comunidadORT router3.lab.
  ort sysLocation.0 s Laboratorio
2 Timeout: No Response from router3.lab.ort

```

- `snmpset -v2c -c comunidadORT router3.lab.ort sysContact.0 s sysadmin@lab.ort`

Código 16: `snmpset -v2c -c comunidadORT router3.lab.ort sysContact.0 s sysadmin@lab.ort`

```

1 zabbix-server:~# snmpset -v2c -c comunidadORT router3.lab.
  ort sysContact.0 s sysadmin@lab.ort
2 Timeout: No Response from router3.lab.ort

```

Práctico 3

Veremos en este práctico como trabajar con las operaciones de tipo TRAP (agente->servidor)

1. Trabajaremos sobre router4.lab.ort. Primero que nada editaremos el archivo `/etc/snmp/snmpd.conf`
`nano /etc/snmp/snmpd.conf`

Código 17: `/etc/snmp/snmpd.conf`

```

1 #####
2 #
3 #  ACTIVE MONITORING
4 #
5
6                                     #  send SNMPv1  traps
7 # trapsink      localhost public
8                                     #  send SNMPv2c traps
9 #trap2sink     localhost public
10                                    #  send SNMPv2c INFORMs
11 #informsink    localhost public

```

2. En la sección ACTIVE MONITORING definiremos las siguientes líneas de configuración:

Código 18: `/etc/snmp/snmpd.conf`

```

1 #####
2 #
3 #  ACTIVE MONITORING
4 #
5
6 trap2sink      zabbix-trap.lab.ort    comunidadTRAP
7 authtrapenable 1

```


- Donde la siguiente línea significa:

trap2sink zabbix-trap.lab.ort comunidadTRAP

- (a) trap2sink: establece que se configurará un gestor de red al que enviar traps SNMP versión 2
 - (b) zabbix-trap.lab.ort: hostname del gestor, puede definirse por dirección IP también
 - (c) comunidadTRAP: comunidad SNMP
- La línea:

authtrapenable 1

- (a) authtrapenable: determina que se configurará el envío de traps ante intentos de consulta SNMP con parámetros de autenticación incorrectos
 - (b) 1: se habilita el envío de traps
3. Reiniciemos el servicio snmpd para aplicar los cambios
 - *service snmpd restart*
 4. Para verificar que los traps estén llegando al destino esperado (zabbix-trap.lab.ort) utilizaremos tcpdump, lo veremos en detalle en el siguiente práctico

Práctico 4

En este práctico veremos como analizar tráfico entrante/saliente de un dispositivo, con herramientas que serán de utilidad en adelante para la resolución de problemas

1. Con el objetivo de verificar la configuración realizada en el práctico anterior analizaremos el tráfico que ingresa en el dispositivo zabbix-trap.lab.ort, utilizaremos para esto la aplicación tcpdump. Herramienta muy popular para captura de tráfico, sobre todo en ambientes donde no hay entorno gráfico instalado
2. Ingreseemos a zabbix-trap.lab.ort
 - *ssh zabbix-trap.lab.ort*
3. La forma general de ejecutar tcpdump es sencilla:
 - *tcpdump {opciones} {expresion}*

Veremos de todas formas que se pueden lograr configuraciones complejas que nos simplifiquen la tarea de análisis

4. Alguna de las opciones son:
 - -i: sirve para definir la interface por donde pasa el tráfico que queremos capturar, puede ser una específica o "any". IMPORTANTE: si no se define este parámetro toma la interface con número menor
 - -e: despliega datos de capa 2
 - -w: guarda en el archivo indicado la captura, en formato .pcap

- -n: muestra información de IP y protocolos en formato numérico
- -direction: in, out, inout. Permite indicar el sentido del tráfico que queremos capturar
- -X: muestra el payload de los paquetes en hexadecimal

5. Los parámetros más comunes para desarrollar una expresión son:

- host: puede ser tanto un nombre DNS, una dirección IP, como el comienzo de una IP
- net: prefijo de red
- icmp: solo tráfico icmp
- tcp: solo paquetes tcp
- udp: solo paquetes udp
- dst: filtra cuando coincide en el destino
- src: filtra cuando coincide en los datos de origen
- port: número de puerto
- La expresión puede usar operadores AND, OR y NOT

6. Para detener la aplicación utilizaremos la combinación de teclas Ctrl+C

7. Probemos a capturar tráfico

- `tcpdump -i any`

¿Por qué cree hay tanto tráfico?

8. Para reducir la cantidad de paquetes verifiquemos por cuál interface ingresará el tráfico que estamos queriendo capturar y apliquemos un filtro mejor

- `tcpdump -i ethX`

9. Para mejorar la expresión necesitamos conocer el tipo de tráfico que estamos deseando capturar

¿Qué datos conoce del tráfico que espera esté llegando al host zabbix-trap.lab.ort?

¿Qué otra opción y/o expresión intentaría utilizar para que se logren capturar menos paquetes, pero que incluyan los que esperamos recibir cuando se genere una trap SNMP?

10. Dejemos corriendo la captura en una consola

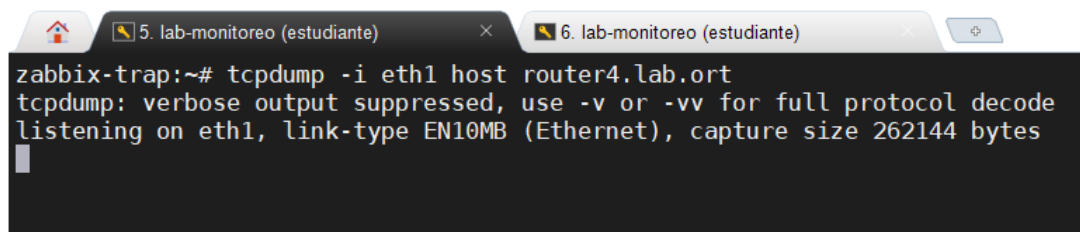


Imagen 2: tcpdump¹

11. En otra ventana ingresemos a lab-monitoreo.lab.ort y realicemos una consulta snmp a router4.lab.ort con datos de comunidad inválidos, por ejemplo

- `snmpwalk -v2c -c comunidadequivocada router4.lab.ort`

12. En zabbix-trap.lab.ort detenga la aplicación tcpdump presionando Ctrl+C
Sería esperable encontrarnos con una salida en pantalla similar a:

Código 19: tcpdump

```

1 zabbix-trap:~# tcpdump -i any host router4.lab.ort
2 tcpdump: verbose output suppressed, use -v or -vv for full
  protocol decode
3 listening on any, link-type LINUX_SLL (Linux cooked v1), capture
  size 262144 bytes
4 03:28:29.435736 IP router4.lab.ort.48400 > zabbix-trap.lab.ort
  .162: C="comunidadTRAP" V2Trap(82) system.sysUpTime.0=342030
  S:1.1.4.1.0=S:1.1.5.5 S:1.1.4.3.0=E:8072.3.2.10
5 03:28:29.435793 IP zabbix-trap.lab.ort > router4.lab.ort: ICMP
  zabbix-trap.lab.ort udp port 162 unreachable, length 140
6 03:28:30.437744 IP router4.lab.ort.48400 > zabbix-trap.lab.ort
  .162: C="comunidadTRAP" V2Trap(82) system.sysUpTime.0=342130
  S:1.1.4.1.0=S:1.1.5.5 S:1.1.4.3.0=E:8072.3.2.10
7 03:28:30.437792 IP zabbix-trap.lab.ort > router4.lab.ort: ICMP
  zabbix-trap.lab.ort udp port 162 unreachable, length 140

```

13. Realicemos una nueva captura y guardemos en un archivo con la opción -w captura.pcap, puede usar la expresión que haya pensado anteriormente, de lo contrario la del siguiente ejemplo:

- `tcpdump -i any host router4.lab.ort -w captura.pcap`

14. Sin detener la captura, genere un evento que provoque el envío de la TRAP, desde cualquier dispositivo de la red:

- `snmpwalk -v2c -c comunidadequivocada router4.lab.ort`

15. Detenga la captura utilizando la combinación Ctrl+C

16. Para ver el contenido en un formato más amigable copiaremos el archivo captura.pcap a la computadora personal. Por las características de la red primero deberá copiar el archivo a un directorio del host lab-monitoreo.

(a) Verifique en que directorio de zabbix-trap.lab.ort se encuentra el archivo captura.pcap

- `pwd`

Código 20: `pwd`

```
1 zabbix-trap:~# pwd
2 /root
```

(b) Vuelva a posicionarse sobre el host lab-monitoreo, ejecute el siguiente comando para copiar desde zabbix-trap (asumiendo que el archivo se encuentra en el directorio /root/) a lab-monitoreo

- `scp zabbix-trap:/root/captura.pcap ./`

(c) Para copiar el archivo desde lab-monitoreo a su computadora personal puede usar MobaXterm

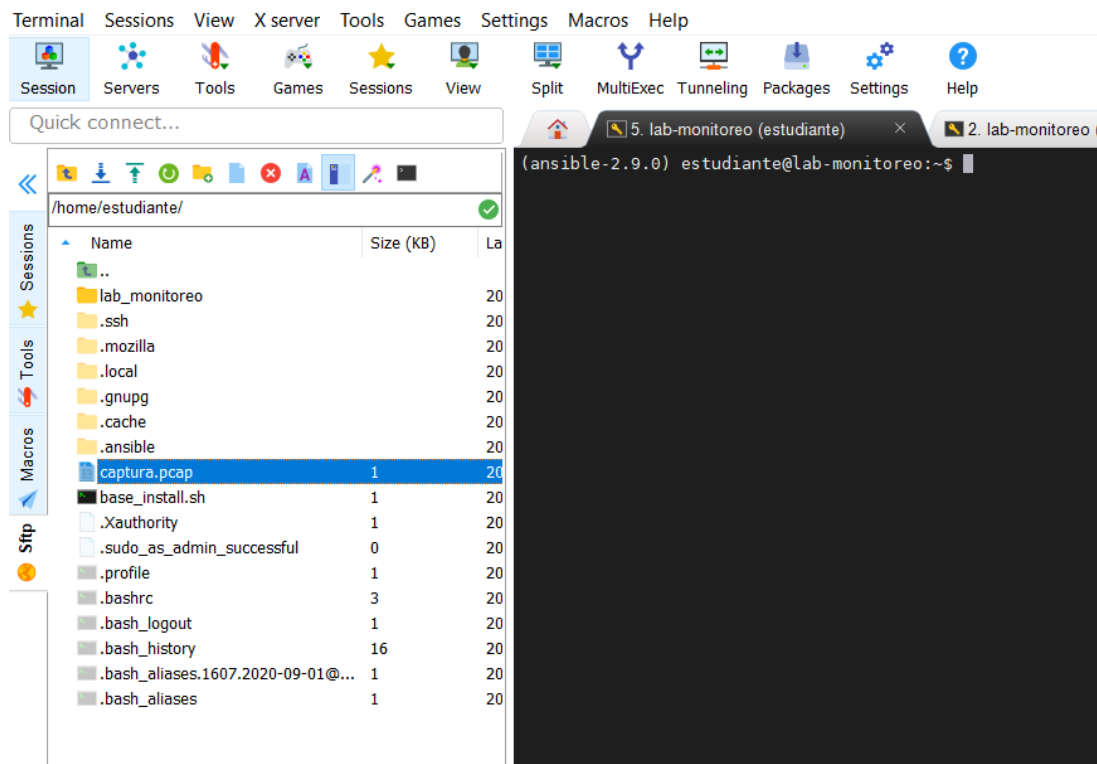
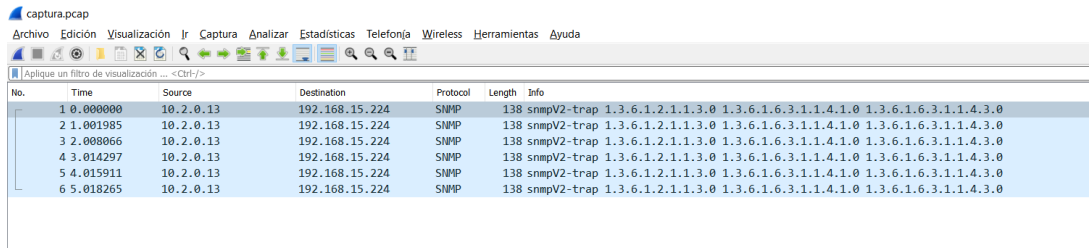


Imagen 3: Ejemplo en MobaXterm²

Puede pasarlo a su computadora con la opción de descarga

- (d) Para abrir el archivo debermos instalar Wireshark (<https://www.wireshark.org/download.html>)
- (e) Abrir archivo captura.pcap con Wireshark



captura.pcap

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

Aplique un filtro de visualización ... <Ctrl-F>

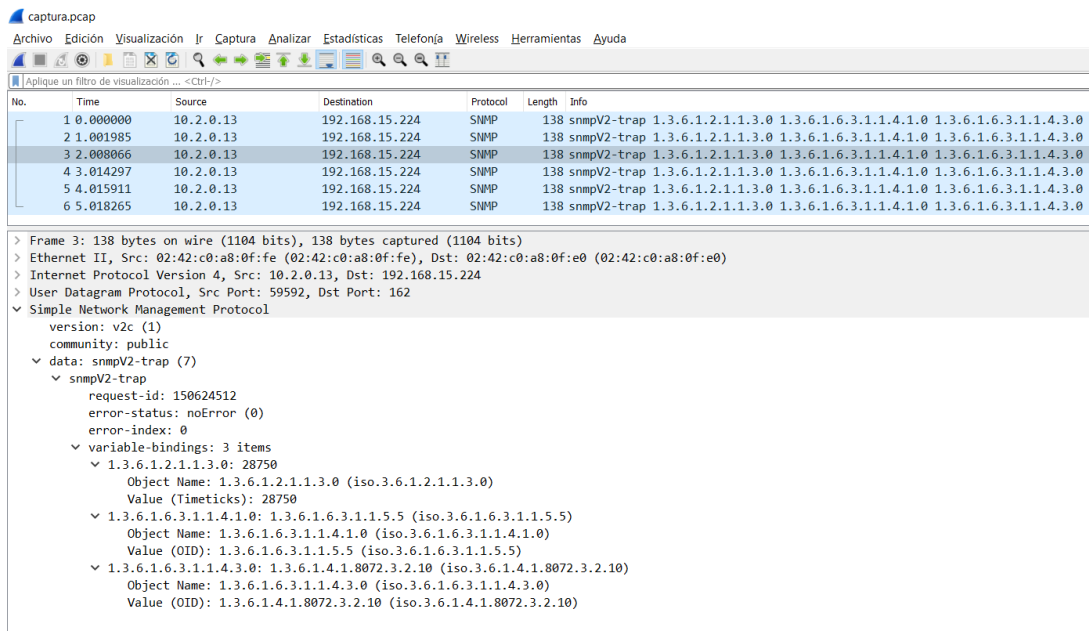
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0
2	1.001985	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0
3	2.008066	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0
4	3.014297	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0
5	4.015911	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0
6	5.018265	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0

Imagen 4: Captura de tráfico - Wireshark³

Seleccione un paquete que cumpla con el criterio esperado:

- Source: 10.2.0.13
- Destination: 192.168.15.224
- Protocol: SNMP
- Info: snmpV2-trap

¿Por qué supone hay tantos paquetes similares?



captura.pcap

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

Aplique un filtro de visualización ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0
2	1.001985	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0
3	2.008066	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0
4	3.014297	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0
5	4.015911	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0
6	5.018265	10.2.0.13	192.168.15.224	SNMP	138	snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.6.3.1.1.4.1.0 1.3.6.1.6.3.1.1.4.3.0

> Frame 3: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits)
 > Ethernet II, Src: 02:42:c0:a8:0f:fe (02:42:c0:a8:0f:fe), Dst: 02:42:c0:a8:0f:e0 (02:42:c0:a8:0f:e0)
 > Internet Protocol Version 4, Src: 10.2.0.13, Dst: 192.168.15.224
 > User Datagram Protocol, Src Port: 59592, Dst Port: 162
 > Simple Network Management Protocol
 version: v2c (1)
 community: public
 data: snmpV2-trap (7)
 snmpV2-trap
 request-id: 150624512
 error-status: noError (0)
 error-index: 0
 variable-bindings: 3 items
 1.3.6.1.2.1.1.3.0: 28750
 Object Name: 1.3.6.1.2.1.1.3.0 (iso.3.6.1.2.1.1.3.0)
 Value (Timeticks): 28750
 1.3.6.1.6.3.1.1.4.1.0: 1.3.6.1.6.3.1.1.5.5 (iso.3.6.1.6.3.1.1.5.5)
 Object Name: 1.3.6.1.6.3.1.1.4.1.0 (iso.3.6.1.6.3.1.1.4.1.0)
 Value (OID): 1.3.6.1.6.3.1.1.5.5 (iso.3.6.1.6.3.1.1.5.5)
 1.3.6.1.6.3.1.1.4.3.0: 1.3.6.1.4.1.8072.3.2.10 (iso.3.6.1.4.1.8072.3.2.10)
 Object Name: 1.3.6.1.6.3.1.1.4.3.0 (iso.3.6.1.6.3.1.1.4.3.0)
 Value (OID): 1.3.6.1.4.1.8072.3.2.10 (iso.3.6.1.4.1.8072.3.2.10)

Imagen 5: Captura de tráfico - Wireshark⁴

Luego de seleccionado en la sección inferior podrá observar datos específicos de ese paquete:

- Frame: Datos genericos de la trama
- Ethernet II: Información del protocolo ethernet
- Internet Protocol Version 4: Datos específicos de IPv4

- User Datagram Protocol: Parámetros de protocolo UDP
- Simple Network Management Protocol: Carga útil del protocolo SNMP

```
▼ Simple Network Management Protocol
  version: v2c (1)
  community: public
  ▼ data: snmpV2-trap (7)
    ▼ snmpV2-trap
      request-id: 150624512
      error-status: noError (0)
      error-index: 0
      ▼ variable-bindings: 3 items
        ▼ 1.3.6.1.2.1.1.3.0: 28750
          Object Name: 1.3.6.1.2.1.1.3.0 (iso.3.6.1.2.1.1.3.0)
          Value (Timeticks): 28750
        ▼ 1.3.6.1.6.3.1.1.4.1.0: 1.3.6.1.6.3.1.1.5.5 (iso.3.6.1.6.3.1.1.5.5)
          Object Name: 1.3.6.1.6.3.1.1.4.1.0 (iso.3.6.1.6.3.1.1.4.1.0)
          Value (OID): 1.3.6.1.6.3.1.1.5.5 (iso.3.6.1.6.3.1.1.5.5)
        ▼ 1.3.6.1.6.3.1.1.4.3.0: 1.3.6.1.4.1.8072.3.2.10 (iso.3.6.1.4.1.8072.3.2.10)
          Object Name: 1.3.6.1.6.3.1.1.4.3.0 (iso.3.6.1.6.3.1.1.4.3.0)
          Value (OID): 1.3.6.1.4.1.8072.3.2.10 (iso.3.6.1.4.1.8072.3.2.10)
```

Imagen 6: Carga útil SNMP - Wireshark⁵

Dentro de la carga útil SNMP observemos los valores principales:

- version
- community
- data
- variable-bindings

Verifique utilizando un MIB browser la correspondencia de OID que se muestra en la sección variable-bindings

¿Qué información puede obtener de la TRAP?