Práctica 1. Gestión de Redes

Parte 1. Primeros pasos

Se pretende crear una configuración de dos máquinas virtuales conectadas entre ellas en la red 172.16.0.0/16. Una de ellas corresponde con el agente (172.16.1.3) y la otra con el NMS con Pandora (172.16.1.2).

Instalación de la VM con PandoraFMS

Descargar la imagen de Máquina Virtual (VM) de PandoraFMS preinstalada para la práctica. Es un operativo CentOS/Linux con el servidor pandora preinstalado. La contraseña del usuario Linux root es 'pandora'.

Instalación:

1) Iniciar VirtualBox.
2) Configurar VirtualBox (File → Preferences) para que emplee como directorio de VM por defecto: /opt/gprs/practica/1/<grupo>/vms/pandora-fms
3) Importar la imagen del NMS desde /opt/gprs
4) Fichero → Importar. Seleccionar el archivo descriptor de la VM que se encontrará en el directorio anterior. **No seleccionar "Reiniciar la dirección MAC de todas las tarjetas de red"**.
5) Comprobar la configuración de la interfaz de red para la VM:
    a) Configuración → Red
    b) Adaptador 1 como NAT
    c) Habilitar adaptador 2 como Red anfitrión
6) Arrancar la VM
7) Comprobar la configuración del adaptador 2 (eth1) desde el operativo:
    a) Sistema → Preferencias → Red
    b) Adaptador 2 con la configuración IP/Máscara/gateway adecuados.

Instalación de la VM con el agente SNMP

Descargar la imagen del agente preinstalado para la práctica. Se trata de un operativo tipo Debian GNU/Linux preinstalado.

Instalación:

1) Iniciar VirtualBox.
2) Importar la imagen desde /opt/gprs
3) Comprobar la configuración de interfaces de red:
    a) Adaptador 1 como NAT
    b) Adaptador 2 como Red anfitrión
4) Arrancar la VM e iniciar sesión con usuario pandora (contraseña pandora).
5) Ejecutar comando ifconfig y comprobar.
6) La VM trae instalados los paquetes snmp (herramientas), snmpd (servicio), smitools (herramientas para compilar MIBs), libsmi2-common (definiciones adicionales) y snmp-mibs-downloader.
7) Ejecutar startx para acceder al sistemas de ventanas. La VM trae instalado el editor de textos nedit.

Ejercicio 1. Polling SNMP

El uso de comandos es muy útil a la hora de configurar escenarios y comprobar el correcto funcionamiento de los mismos de una forma fácil ya que permite testear cada una de las funcionalidades por separado. En este ejercicio se deberá realizar la comprobación del funcionamiento de sondeo SNMP (polling) utilizando los comandos snmpget, snmpwalk y snmpd.

Para realizar algunas comprobaciones tipo polling emplear los comandos snmpget y snmpwalk desde el NMS. Formato orientativo:

snmpget -v 2c -c comunidad agente OID
snmpwalk -v 2c -c comunidad agente [OID]

Se pueden obtener las definiciones MIB de bases de datos públicas como www.oid-info.com, mibdepot.com y alvestrand.no. Se recomienda buscar el OID para las interfaces de red.

Se ha configurado un router Cisco del laboratorio para servir SNMPv1. Su IP es la 10.3.16.1 y el nombre de comunidad es “publicTLM” sin comillas. Probar los comandos anteriores con dicho router.

A continuación configurar el servicio snmpd en el agente Linux para que de servicio SNMPv2 con una comunidad de nombre arbitrario.

Pista: El archivo que hay que configurar es /etc/snmp/snmpd.conf. Se puede emplear el asistente *snmpconf* o editar la configuración a mano. Las dos directivas más interesantes son *rocommunity* y *rwcommunity*. Su uso y sintaxis se pueden consultar con el comando *man snmpd.conf*. Una vez modificado recordar que es necesario reiniciar el servicio para recargar la

configuración. Idea: Se puede parar el servicio *service snmpd stop* y ejecutar el demonio en modo depuración:

service snmpd stop
snmpd -r -f -L

Comprobar qué es lo que pasa al ejecutar sondeos desde el NMS.

Volver a arrancar el servicio. Para ello se puede reiniciar o ejecutar, por ejemplo, el comando *service snmpd start*.

Comprobar que el agente Linux sigue operativo.

Finalmente, se pueden comprobar las tramas intercambiadas entre Agente y NMS empleando wireshark. ¿Qué tramas se observan?

Ejercicio 2. Traps y notificaciones

En este ejercicio se deberá realizar la comprobación del funcionamiento de modo asíncrono de SNMP (traps y notificaciones) utilizando los comandos snmptrap y snmptrapd en el agente Linux y el NMS, respectivamente.

Es necesario que las nuevas definiciones de MIB se hayan instalado tanto en el NMS como en el agente. Para este ejercicio se ha preparado e instalado un módulo de nombre UPNA-TEST-SMI en el directorio /usr/share/snmp/mibs/ con la definición de un trap y una notificación. La definición es la siguiente:

 # cd /usr/share/snmp/mibs/
 # cat > UPNA-TEST-SMI.txt
UPNA-TEST-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY FROM SNMPv2-SMI
    enterprises FROM SNMPv2-SMI;

upna MODULE-IDENTITY
    LAST-UPDATED "201507050000Z"
    ORGANIZATION "Public University of Navarra"
    CONTACT-INFO
  "This mib is not assigned by IANA."
    DESCRIPTION
  "Test module to use in GPRS"

```
        REVISION "201507050000Z"
        DESCRIPTION
    "Initial proposal"
        ::= { enterprises 11073 }

demonotifs OBJECT IDENTIFIER ::= { upna 991 }

demo-notif NOTIFICATION-TYPE
        STATUS current
        OBJECTS { sysLocation }
        DESCRIPTION "Just a test notification"
        ::= { demonotifs 17 }

demotraps OBJECT IDENTIFIER ::= { upna 990 }

demo-trap TRAP-TYPE
        STATUS current
        ENTERPRISE demotraps
        VARIABLES { sysLocation }
        DESCRIPTION "This is just a demo"
        ::= 17

END
```

Desde el NMS, ejecutar el demonio snmptrapd en modo depuración en un terminal para poder ver los traps:

snmptrapd -f -Lo -d

Enviar traps al NMS (snmptrap o snmpinform):

snmptrap -v1 -c <comunidad> <nms> UPNA-TEST-MIB::demotraps localhost 6 17 "" SNMPv2-MIB::sysLocation.0 s "Laboratorio TLM"

Enviar un trap SNMPv2 (notificación):

snmptrap -v 2c -c <comunidad> <nms> "" UPNA-TEST-MIB::demo-notif SNMPv2-MIB::sysLocation.0 s "Laboratorio TLM (v2)"

Se debe ver como los traps llegan del agente Linux al NMS. Analizar las trazas intercambiadas empleando wireshark.

Parte 2. Gestión SNMP empleando un NMS

El objetivo de esta parte es familiarizarse con gestión de red basándose en software de gestión de nivel profesional. En este caso se ha propuesto emplear PandoraFMS pero hay muchas otras alternativas que tienen diferentes ventajas e inconvenientes.

Arrancar el NMS y abrir la consola de PandoraFMS desde el icono del escritorio. El usuario es admin y la contraseña pandora.

Las opciones que se pueden experimentar son las siguientes:

- Configuración de la monitorización:
    - ICMP
    - SNMP (GET, SET y Traps)
    - TCP (i.e. HTTP)
- Configuración de alertas
- Informes y consola visual
- Acceso remoto
- Reconocimiento de redes (recon server)

Se pueden realizar tanto contra el agente virtual como el router habilitado para ello.

Nota: Para recoger traps con PandoraFMS es necesario modificar el siguiente valor de configuración del archivo /etc/pandora/pandora_server.conf:

snmpconsole 1

Y luego reiniciar el servidor para que recargue la configuración:

service pandora_server restart

Parte 3. Desarrollo de un agente de emisión de traps simple

Se trata de desarrollar un servicio que se ejecute en el agente y que compruebe activamente cuándo se produce una condición dada. En ese momento el servicio debe lanzar un trap al

NMS que notifique dicho evento. Se debe especificar el trap con un MIB como el mostrado en la Parte 1. Se recomienda emplear el comando *smilint* para comprobar la corrección de la definición. Se puede emplear cualquier lenguaje de programación para realizarlo. Se debe demostrar su funcionamiento en el laboratorio. Se valorará la utilidad que se encuentre al agente y la definición de la MIB para la trap.

Nota: Pensar un buen esquema de testeo que evite pender tiempo copiando archivos de un sitio a otro. Por ejemplo se puede probar a ejecutar snmptrap y snmptrapd en la misma máquina en un primer lugar. Una vez funcione localmente probar en despliegue real.

Entregables:

- **Memoria (extensión máxima de 1000 palabras)**
    - **Evaluación del sistema**
    - **Checklist con las tareas realizadas**
    - **Gráficas obtenidas y análisis**
    - **Descripción breve de la aplicación del agente de la Parte 3**
- **MIB definido en la Parte 3**
- **Código fuente desarrollado en la Parte 3 (solo para comprobar que funciona)**