ESCUELA SUPERIOR POLITECNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“IMPLEMENTACIÓN Y PRUEBAS DE MONITOREO EN UNA RED LAN, BASADOS EN SNMP V3”

TESINA DE SEMINARIO

Previa la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por

ELSA ELOÍSA OCHOA GÓMEZ
MARCELO PATRICIO VENEGAS ZÚNIGA

Guayaquil-Ecuador

2014

AGRADECIMIENTO

Nuestro sincero agradecimiento a todas las personas que de una u otra forma brindaron su colaboración para la realización de este proyecto, especialmente al Ing. Washington Medina, quien con su guía y enseñanza nos ha ayudado a terminar este trabajo.

DEDICATORIA

A DIOS, nuestros Padres, hermanos, demás familiares y amigos, por su constante apoyo, y por todo cuanto nos han brindado. También a los profesores, consejeros y guías que a lo largo de los años nos han formado académicamente y en valores, a todos ellos va dedicado este trabajo.

TRIBUNAL DE SUSTENTACIÓN

Ing. Washington Medina

PROFESOR DEL SEMINARIO DE GRADUACIÓN

Dr. Boris Ramos

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral"

Elsa Eloísa Ochoa Gómez

Marcelo Patricio Venegas Zúñiga

RESUMEN

El protocolo SNMP (Protocolo Simple de Administración de Redes) es el núcleo de las operaciones de administración generadas por cualquiera de las aplicaciones de uso libre o con licenciamiento que son instaladas en las redes corporativas de cualquier tipo para su monitoreo. En su gran mayoría son las primeras dos versiones de SNMP las que son configuradas en los equipos.

Sin embargo, aquellas versiones utilizan el esquema de comunidades el cual es vulnerable. Sólo por mencionar algunos ejemplos de ataques si alguien obtuviera de forma fraudulenta la comunidad de escritura, se puede enviar desde un servidor TFTP (Protocolo Trivial de Transferencia de Archivos) un nuevo archivo de configuración en ejecución que elimine cualquier contraseña, para así ingresar al router; o por el contrario, se puede enviar desde el router, la configuración de inicio al servidor TFTP; o si el router tiene habilitado el acceso web, se puede hacer una inserción de código para habilitar un password de acceso privilegiado e ingresar al router con él, lo que se demuestra en la sección 3.6 de este documento.

Con el fin de probar las seguridades de la última versión del protocolo como son la autenticación y privacidad, se diseñará una pequeña red LAN (red de área local) conformada por dos computadoras, un switch y un router, en la cual se configurarán los agentes SNMPv3 (Protocolo Simple de Gestión de Red versión 3). Luego, usando la aplicación WhatsUp Gold se descubrirán los dispositivos para su monitoreo y con la aplicación SNMP JManager se realizarán otras pruebas complementarias para demostrar cómo operan algunas seguridades del protocolo.

Finalmente, se analizarán los paquetes capturados con Wireshark del monitoreo para comparar desde el punto de vista de seguridad, los mensajes generados en un monitoreo basado en la versión 2 del protocolo y los mensajes generados usando la versión 3. Este proyecto pretende acercar a cualquier usuario a los beneficios de SNMPv3, el cual es muy sencillo de implementar en cualquiera de los dispositivos que forman parte de las redes hoy en día y que sin embargo, no está implementado en la gran mayoría de los entornos informáticos, precisamente por el desconocimiento de su implementación y operación.

ÍNDICE GENERAL

ABREVIATURAS Y SIMBOLOGÍA

AES: Advanced Encryption Standard

ASN: Abstract Syntax Notation

BER: Basic Encoding Rules

CBC: Cipher Block Chaining

CCITT: Consultative Committee for International Telegraphy and Telephony

CFB: Cipher Feedback

CIPH: CIPHertext

CLNS: ConnectionLess Network Service

CPU: Central Processing Unit

DDP: Datagram Delivery Protocol

DES: Data Encryption Standard

DNS: Domain Name System

EIGRP: Enhanced Interior Gateway Routing Protocol.

FTP: File Transfer Protocol

HDLC: High-Level Data Link Control

HMAC: Hashed Message Authentication Code

HTML: HyperText Markup Language

HTTP: HyperText Transfer Protocol

IANA: Internet Assigned Numbers Authority

ICMP: Internet Control Message Protocol

IETF: Internet Engineering Task Force

IGMP: Internet Group Management Protocol

IIS: Internet Information Services

IP: Internet Protocol

IPAD: Inner PADding

IPX: Internetwork Packet Exchange

IPv4: Internet Protocol version 4

IPv6: Internet Protocol version 6

ISO: International Standard Organization

LAN: Local Area Network

LCD: Local Communication Datastore

MAC: Message Authentication Code, si se refiere a los mensajes hash.

MAC: Media Access Control si se refiere a las direcciones de capa física MAC ADDRESS.

MD: Managed Device

MD5: Message-Digest Algorithm 5

MIB: Management Information Base

NAT: Network Address Translation

NMS: Network Management System

OID: Object Identifiers

OPAD: Outer PADding

OSI: Open System Interconnection

OSPF: Open Shortest Path First

PC: Personal Computer

PDU: Protocol Data Unit

RFC: Request For Comments

RPC : Remote Procedure Call

SHA: Secure Hash Algorithm

SMI: Structure of Management Information

SNMP: Simple Network Management Protocol

SNMPv2c: Simple Network Management Protocol community-based

SNMPv3: Simple Network Management Protocol version 3

SQL: Structured Query Language

SSH: Secure Shell

TCP: Trasmission Control Protocol

Telnet: TELecommunication NETwork

TFTP: Trivial File Transfer Protocol

TLS: Transport Layer Security

UDP: User Datagram Protocol

USM: User-based Security Model

VACM: View-based Access Control Model

VLAN: Virtual Local Area Network

VoIP: Voice over IP

WAN: Wide Area Network

WMI: Windows Management Instrumentation

XOR : logical Exclusive OR

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Es muy natural en toda empresa buscar una manera de mejorar y facilitar el manejo de su red informática, debido a esto y al avance de la tecnología que existe día a día, han surgido diferentes gestores que son usados por los administradores de red, para poderla monitorear y diagnosticar problemas teniendo así el control de ésta.

La administración de una red implicó desde sus inicios mecanismos de intercambio de información entre los diferentes dispositivos, lo cual era complicado, ya que los diferentes dispositivos de red eran de diferentes fabricantes, por ende, la manera de administrarlos era también privativa de cada empresa. Esta situación complicaba el trabajo de las estaciones de monitoreo de la redes heterogéneas; sin mencionar además que hace algunos años no era tan sencillo fabricar los diferentes dispositivos de networking de los que hace uso cualquier empresa en su infraestructura informática, por lo que su mercado era bastante restringido y con precios elevados.

Con el crecimiento de la popularidad de TCP (Protocolo de Control de Transmisión) sobre IP (Protocolo de Internet), se dieron las condiciones para que la IETF (Grupo de Trabajo de Ingeniería de Internet) propusiera un estándar de gestión, este estándar es el SNMP (Protocolo Simple de Gestión de Red), el cual ha ido desarrollándose en 3 diferentes versiones con el paso del tiempo, cada una de ellas con sus respectivas mejoras. Sin embargo la versión 3 de SNMP no ha sido mayoritariamente aceptada en la industria, y es aquí donde entra en marcha nuestro proyecto: “Implementación y pruebas de monitoreo en una red LAN, basados en SNMP v3”, donde utilizamos algunas de las tantas aplicaciones que han sido desarrolladas para el manejo de los mensajes de SNMP.

CAPÍTULO 1

1 GENERALIDADES

1.1 Identificación del problema

Toda organización, empresa mediana o grande necesita que la red de comunicaciones tenga una alta disponibilidad ya que ofrece servicios críticos tanto de forma interna como externa, y se logra monitoreando la red mediante herramientas que funcionen sobre diferentes protocolos tales como ICMP (Protocolo de Mensajes de Control de Internet), Telnet (Red de Telecomunicaciones), pero sobretodo el estándar SNMP cuyo alcance y funcionalidades son más amplios y pueden recabar más información sobre la salud de la red. Estas herramientas ayudan en el análisis de facilitar la tarea del administrador de red en resolver los fallos. Por ejemplo, si se reportan problemas de lentitud en acceder a la red internet en cualquier organización,

varias aplicaciones permiten demostrar a los usuarios que el ancho de banda es o no el adecuado, entre muchas otras prestaciones. En torno a todo esto, es importante reforzar la seguridad, en el proceso de comunicación del gestor y equipos monitoreados, ya que la seguridad en las primeras versiones de SNMP es vulnerable a posibles ataques internos, dado a que tanto la información como la contraseña no se cifran.

Los administradores de la red, ya sea de empresas o de un sistema académico, necesitan para su gestión tener permisos no sólo de lectura, sino también de escritura, para modificar valores de los objetos monitoreados cuando sea necesario, sin embargo el permiso de escritura se deshabilita como una medida de prevención para que no haya manipulación inadecuada de la información, lo que puede afectar a la red.

Las organizaciones, sea por desconocimiento de SNMPv3 que es una implementación relativamente nueva, o por miedo a que sea incompatible con ciertas aplicaciones, no la implementan en la red, lo que no permite aprovechar las mejoras del protocolo, sobre todo la seguridad.

1.2 Justificación

Este proyecto será de mucha utilidad para cualquier organización que tenga una red LAN de datos con servidores, dispositivos de networking y equipos finales, motivo suficiente como para preocuparse de la conectividad entre usuarios a los servicios que ofrece y más aún si estos servicios también se entregan hacia una red WAN (red de área amplia). La implementación de la versión 3 del protocolo SNMP no requiere de una inversión monetaria, sino solamente tener el conocimiento de cómo habilitarlo en los diferentes agentes que funcionan en dispositivos como routers, switches, servidores y computadores que ya son componentes de la red monitorizada.

Es más, el no planificar estrategias de seguridad para la red de mediano a largo plazo o incluso el no invertir una cierta cantidad de recursos para capacitar a los administradores de red sobre esta versión de SNMP por ejemplo, tiene más impacto económico negativo, ya que a la final se destinan más recursos en solucionar problemas derivados de vulneraciones y ataques a la red con la consecuente caída de servicios. Por último, sea que la red esté ya implementada o no; o que se cuente o no con un software libre o licenciado instalado en una máquina administradora, la funcionalidad de la autenticación y cifrado suceden en el agente SNMP monitoreado (llamado

también entidad sobre todo cuando nos referimos en SNMPv3), por lo que la aplicación no influye de mayor manera aunque la gran mayoría disponible para el uso tanto comercial como libre tienen habilitada la versión 3. Si no se dispone de una aplicación que pueda operar en la versión 3, se puede buscar una de las múltiples aplicaciones libres disponibles si es que no se quiere destinar dinero en alguna licenciada.

1.3 Objetivos

1.3.1 Objetivo general

Implementar las primitivas SNMPv3 en una red LAN.

1.3.2 Objetivos específicos

- Diseñar y descubrir una red LAN en el entorno de la aplicación de monitoreo en el NMS (Sistema Administrador de Red), en la cual se realizarán las pruebas de las primitivas.
- Probar el funcionamiento de solicitudes y respuestas primitivas con objetos críticos de una red (ancho de banda, equipos de la red, etc)
- Probar el funcionamiento de solicitudes a modificaciones de objetos de la MIB (Base de Datos de Información de Administración) vía comandos set.

- Identificar las alarmas (traps) más comunes en una red LAN y gestionar su tratamiento y solución.
- Probar la funcionalidad de seguridad (principalmente cifrado) en la mensajería gestor-agente.

1.4 Metodología

Lo primero es recolectar información sobre SNMP sobre todo la versión 3, y cómo habilitarla en las entidades de red; así como datos sobre las aplicaciones que usaremos para realizar las pruebas de las primitivas, cómo configurarlas y su respectivo uso. En nuestro caso se usará el SNMP JManager y el WhastUp Gold v16.1 para el monitoreo y el Wireshark para la captura y análisis de los mensajes SNMP.

Luego de esto, se diseñará una pequeña LAN en la cual haremos la prueba, en la cual se deberá definir si hay restricciones en cuanto a subredes, una o varias VLAN (red de área local virtual) o permisos, como sucede en cualquier red corporativa real, para poder ejecutar los softwares de la prueba y resolver si es que se da esta situación.

Se configurarán los diferentes agentes SNMPv3, se instalarán las aplicaciones en la computadora NMS y se comenzarán a hacer las pruebas. Primero se ejecutarán solicitudes SNMP con una versión que no implemente seguridades como la versión 2 y luego con la versión 3, donde se va a capturar en todo momento los paquetes SNMP para su posterior análisis con algún programa analizador de tráfico como el Wireshark. Luego en cambio, se harán modificaciones a objetos de la MIB mediante solicitudes set SNMP con una versión que no implemente seguridades como la versión 2 y luego con la versión 3, capturando en todo momento los paquetes SNMP para su posterior análisis. En el medio de estas dos clases de pruebas, de forma espontánea o provocando sucesos anómalas en la red, capturar los traps que circulan dentro de la misma con el programa Wireshark, tanto cuando estos viajan cifrados o no.

Finalmente, se analizarán todas las capturas, para poder tener las conclusiones con respecto a las aplicaciones usadas y poder comparar las versiones del protocolo en base a lo observado dentro de los paquetes.

1.5 Observaciones

- El agente del Sistema Operativo Windows no tiene compatibilidad, o no funciona correctamente con SNMPv3, por lo cual, se recurre a trabajar con el agente de Ubuntu, que sí es compatible con la versión 3.
- El protocolo SNMPv3 puede utilizar cualquier mecanismo de transporte, pero original y habitualmente trabaja sobre UDP (Protocolo de Datagrama de Usuario) a través del puerto 161, aunque en la actualidad también soporta CLNS (Servicio no Orientado a la Conexión) de OSI (Modelo de Interconexión de Sistemas Abiertos), AppleTalk DDP (Protocolo de Entrega de Datagramas), y Novell IPX (Intercambio de Paquetes de Internet).
- SNMP hace uso de una base de datos (MIB), estas MIB´s son capaces de clasificar la información sobre los recursos de una red a modo de lista de objetos de datos, y en la actualidad existen algunos tipos de MIB´s, lo que permite a la empresa escoger la cantidad y el tipo de MIB´s que requiera según sea su necesidad.
- SNMP es un protocolo que consume un considerable ancho de banda, lo cual limita su utilización en entornos de red muy extendidos.

1.6 Resultados esperados

- Dar a conocer el adecuado manejo de SNMPv3 y sus nuevas ventajas de seguridad.
- Informar sobre los tipos de seguridades de SNMPv3 y su respectivo funcionamiento.
- Conocer el desenvolvimiento de las primitivas SNMPv3 en la red y su funcionamiento.
- Mostrar el manejo y funcionamiento del agente SNMP Ubuntu.
- Dar a conocer las aplicaciones que se pueden utilizar para el desarrollo de la gestión de una red.
- Mostrar el funcionamiento de dichas aplicaciones y sus diferentes prestaciones.

1.7 Hardware y software a utilizar

Hardware

- Computadora con sistema operativo Ubuntu que funcione como gestor.
- Computadoras de cualquier sistema operativo que tengan sus agentes SNMP corriendo y que serán monitoreadas.
- Switches LAN y routers con servicio SNMP activo.

Software

- Aplicaciones de administración compatibles con SNMPv3, sean estas libres o propietarias, ejemplo: SNMP J Manager o WhatsUp respectivamente.
- Wireshark, para analizar el tráfico SNMP.

CAPÍTULO 2

2 MARCO TEÓRICO

En este capítulo mostraremos la parte teórica de nuestro proyecto, la explicación técnica de cómo funciona, y las respectivas definiciones de cada implemento utilizado en el desarrollo de nuestro proyecto, ya sea éste hardware o software, incluyendo las funcionalidades de cada componente.

2.1 Gestión de la red

La Gestión de Redes es el conjunto de actividades destinadas a garantizar el control, la supervisión y la administración de los diferentes elementos que constituyen una red para que la comunicación tenga lugar. La gestión de red toma la forma de seguimiento, coordinación y control de los recursos informáticos y de comunicaciones.

El objetivo de la gestión de redes es mantener los sistemas de una organización en un estado óptimo de funcionamiento el tiempo máximo posible, minimizando la pérdida que ocasionará si existe una parada del mismo.

De acuerdo con la clasificación establecida por la Organización Internacional de Estándares (ISO, International Standard Organization), las Áreas Funcionales de la Gestión de Red se engloban en cinco grandes grupos: Gestión de Fallos y Recuperación, Gestión de la Configuración, Gestión del Rendimiento, Gestión de la Contabilidad y Gestión de la Seguridad. [25]

2.2 Protocolo

2.2.1 Definición

Es el conjunto de reglas o estándares, implementados por hardware o software, que usan las computadoras o nodos de una red para poder comunicarse intercambiando mensajes de una forma ordenada, sincronizada y rigiéndose a una sintaxis y semántica adecuada.

Entre algunas de las características importantes que poseen los protocolos para su operación tenemos: enlace tres vías (three way handshaking) para dar inicio a una sesión de comunicación entre pares, formateo del mensaje o PDU (Unidad de Datos de Protocolo) en cada capa, detección y corrección de errores en los datos transmitidos, iniciación y finalización de las sesiones de comunicación, entre otras.

Para que la comunicación se pueda realizar, es necesario que los nodos “hablen el mismo lenguaje”, es decir que si dos computadoras usan protocolos de comunicación diferentes no se podrían comunicar. El estándar para comunicación en internet para diferentes tipos de quipos informáticos es el protocolo TCP/IP. [29]

2.2.2 Modelo de capas OSI

Creado por iniciativa de la Interconexión de Sistemas Abiertos (OSI, Open Systems Interconnection) es un estándar por medio del cual la arquitectura de la red se divide en 7 capas funcionales, cada una tiene funciones y características específicas, lo que permite que si un nuevo protocolo o

tecnología es desarrollada para una capa en especial, no afecte el desempeño ni la comunicación de las capas contiguas. La comunicación de datos es de forma vertical para lograr transmitir un mensaje desde el origen hacia el destino; una capa da servicios a la superior y a su vez, ésta recibe servicios de la inferior. Cuando los datos van bajando en el stack de capas, se van encapsulando con el encabezado de la capa en la que se encuentran, y tomando un nombre acorde a ésta la respectiva PDU.

Las 7 capas de modelo OSI son, desde sus capas superiores a las inferiores las siguientes: Aplicación, Presentación, Sesión, Transporte, Red, Enlace de Datos y Física. [4]

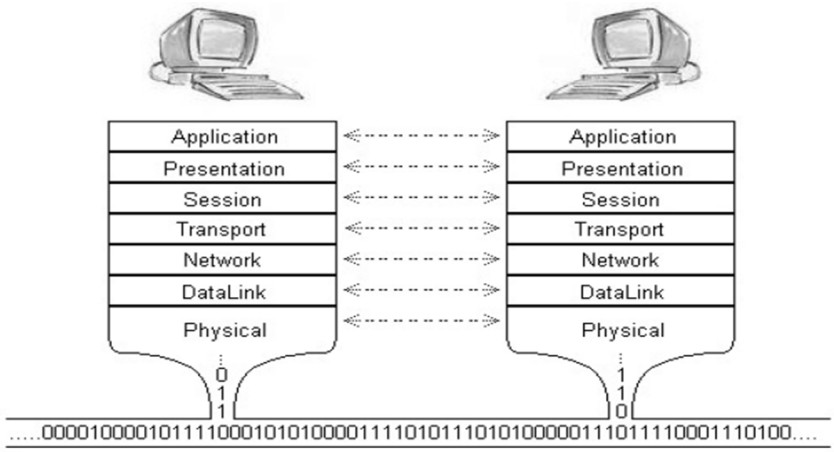


Figura 2.1 – Modelo de referencia OSI. [4]

Un breve resumen de las funciones de cada capa:

- **Capa 7: Aplicación**

  Esta capa interactúa tanto con el usuario final como con el software de aplicación que tiene un componente de comunicación. Principalmente, se encarga de busca un dispositivo par de comunicación con el cual intercambiará datos, una vez que lo encuentra debe administrar la sincronización para la compartición de los archivos si es que es una comunicación síncrona y además, verificar si existen los recursos de red adecuados, como ancho de banda, para que se pueda establecer dicha comunicación. Ejemplos de aplicaciones de esta capa son: FTP (Protocolo de Transferencia de Archivos), Telnet, entre otros.

  El PDU propio de esta capa son los mensajes generados por las aplicaciones. [4]

- **Capa 6: Presentación**

  Se encarga de interpretar las sintaxis y semánticas de los datos entregados por las aplicaciones superiores, sin importar cuales sean, y de transformarlos en un formato de red cuando se envían los datos a otro nodo; y pasar los datos de red a un formato entendible a las

aplicaciones cuando éstos han llegado al nodo destino. Los datos de esta capa son encapsulados en la PDU de Sesión o mensajes. [4]

- **Capa 5: Sesión**

  Se encarga de iniciar, mantener y terminar las sesiones establecidas de comunicación entre aplicaciones, en modo full dúplex, half dúplex o simplex. Esta capa es implementada en los ambientes de aplicación que usan llamadas de procedimiento remoto. El PDU propio de esta capa son los mensajes generados por las aplicaciones. [4]

- **Capa 4: Transporte**

  Esta capa permite un servicio de transferencia confiable de datos de las capas superiores entre usuarios finales, implementando detección y corrección de errores, control de flujo y segmentación / ensamblaje de los datos. El PDU de esta capa se denomina segmento.

Los dos protocolos más conocidos que operan en esta capa son el TCP que es orientado a la conexión y el UDP que no es orientado a la conexión. Orientado a la conexión quiere decir que se sigue el rastro de los segmentos enviados a la red de destino, de tal forma que si uno se pierde o llega con errores, se lo vuelve a enviar desde el origen; por otro lado, un servicio no orientado a la conexión es aquel que envía los segmentos sin preocuparse si llegan o no al destino. Un servicio no orientado a la conexión se usa cuando los datos transmitidos por la red requieren el mayor ancho de banda posible, como la videoconferencia, online streaming y similares ya que UDP no degrada el ancho de banda, cosa que con TCP suele suceder por la carga extra de tráfico de control. [4]

- **Capa 3: Red**

  Permite enviar los datos de un origen hacia un destino pasando por una o algunas redes. Esta capa ejecuta labores de enrutamiento y también fragmentación, re ensamblaje y reporte de errores, conservando la calidad de servicio en la transmisión de los datos solicitada por las capas superiores. El esquema de direccionamiento ejecutado por los routers es lógico, es decir, escogido por los

administradores de red, y jerárquico, pudiendo tener múltiples subredes de una red principal, dependiendo de las necesidades de la organización. El PDU de esta capa es el paquete. [4]

- **Capa 2: Enlace de datos**

  Detecta y permite el transporte de datos entre entidades de red y ejecuta una detección de errores que ocurren en la capa física para corregirlos. Básicamente, sea la conectividad punto a punto, punto a multipunto, dentro una LAN o WAN, el protocolo maneja las direcciones físicas (MAC Address) para conectar dos nodos. El PDU de esta capa es la trama. [4]

- **Capa 1: Física**

  Estándares y especificaciones eléctricas y físicas para los dispositivos de red, que describen la relación entre éstos y el medio; algunas de ellas la disposición de los pines, voltajes, tipos de cables o conectores y más. Define el inicio y terminación de las sesiones de conexión a un medio de comunicaciones que es compartido por algunos usuarios,

enviando las cadenas de bits por los medios físicos luego de haber realizado la transformación a la respectiva variable física (eléctrica, óptica, señal de radio, entre otras). El PDU de esta capa son los bits que se transmiten por los medios. [4]

2.2.3 Modelo de capas TCP/IP

Para que la comunicación entre dispositivos se pueda realizar, es necesario que los nodos “hablen el mismo lenguaje”, es decir que si dos computadoras usan protocolos de comunicación diferentes no se podrían comunicar. El estándar para comunicación en internet para diferentes tipos de equipos informáticos es la suite de protocolos TCP/IP. Este modelo es mantenido por la IETF y provee de reglas para la conectividad de extremo a extremo entre dispositivos en la red; especificando cómo los datos deben recibir formato, ser dirigidos, enrutados y conmutados hacia el destino final. [4]

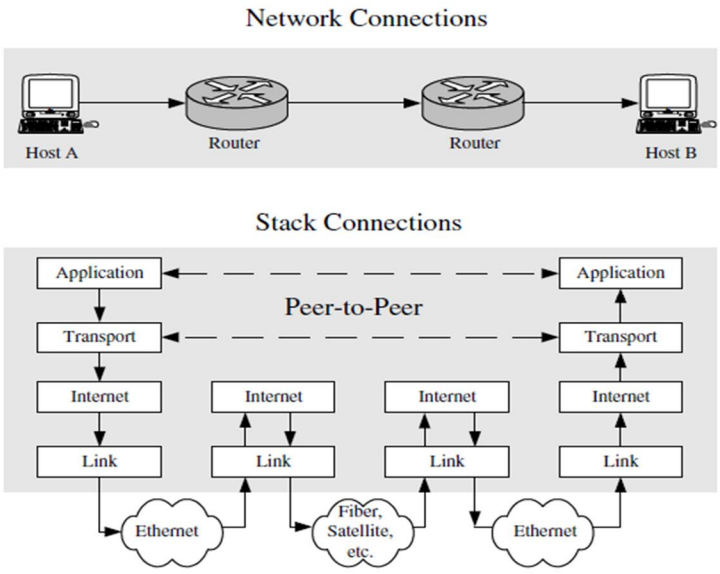


Figura 2.2 – Modelo de referencia TCP/IP. [4]

La suite tiene 4 capas funcionales, por lo que se la suele comparar con el estándar OSI. Son las siguientes:

- **Capa de aplicación**

  Contiene los protocolos de capa superior usados por las aplicaciones de usuario para la comunicación de datos en la red. Comparándolo con el modelo de referencia OSI, esta capa agrupa las tres primeras de aquel stack, es decir, también hace las funciones de presentación y sesión, gracias a librerías que permiten que las aplicaciones se comuniquen con protocolos suplementarios dentro de la capa. Luego, los datos codificados por las respectivas aplicaciones, son enviados a

la capa de transporte y encapsulados en segmentos TCP o datagramas UDP para ser enviados al destino. [4]

Con respecto a SNMP, la capa de aplicación provee de servicios al usuario final que desea obtener información sobre el estado de un equipo, por ejemplo si un puerto de un switch esta up/down. Cualquiera que sea la aplicación instalada en un equipo administrador, ésta es la que gestiona el envío de solicitudes, respuestas con los agentes SNMP de los equipos monitoreados y la que recibe los traps enviados por estos últimos. [3]

- **Capa de transporte**

  Como lo indica el nombre, esta capa se encarga de transportar los datos de las aplicaciones asegurándose que lleguen a su destino, principalmente definiendo los números de puertos de origen y destino. A no ser que haya una aplicación de capa superior que lo haga (como RPC, Llamada a Procedimiento Remoto), es ésta la primera capa del stack TCP/IP que introduce confiabilidad en el proceso de entrega de

datos hacia el destino, ya que la capa de Internet es de mejor esfuerzo. El protocolo HDLC (High-Level Data Link Control) provee también confiabilidad en la entrega de los datos en la capa de Enlace. Implementa al igual que la capa transporte de OSI, control de errores, control de flujo y fragmentación; y el servicio puede ser orientado (TCP) o no orientado a la conexión (UDP). [4]

Con respecto a SNMP el protocolo de la capa de transporte utilizado es UDP siendo así la transmisión de los datagramas poco confiable, por lo que le corresponde a la capa de aplicación verificar si se han perdido o no, para retransmitirlos de darse el primer caso. Esto se consigue mediante un tiempo de espera (timeout) asociado a una solicitud enviada por un NMS, si pasado el tiempo el NMS no recibe una respuesta satisfactoria, asume que el datagrama se ha perdido y lo vuelve a enviar. Tanto el tiempo de espera como las veces que se reenvían los datagramas son configurables por el usuario. Los puertos usados en esta capa son el 161 para enviar solicitudes y recibir respuestas SNMP y el 162 para recibir traps enviados por agentes remotos. [3]

- **Capa de Internet o de Red**

  Transmite los datos generados por los protocolos de capa superior encapsulados en paquetes, a través de la red, hacia el destino. Entre los protocolos IP más conocidos se encuentran el ICMP, que transmite información de diagnóstico del estado de la interconectividad entre dos nodos IP; y el IGMP (Protocolo de administración de grupos de Internet), el cual administra tráfico multicast. La interconectividad se centra básicamente en enrutar los mensajes en las direcciones IP de origen y destino, para lo cual se usan diferentes protocolos de enrutamiento en esta capa tales como EIGRP (Protocolo de enrutamiento de puerta de enlace interior mejorado), OSPF (Protocolo toma de ruta más corta), entre otros. [4]

Cuando se envía una consulta SNMP hacia un dispositivo, o desde un dispositivo monitoreado se desea enviar alertas a un equipo NMS, interviene esta capa al enviar los paquetes que contienen información SNMP a la dirección IP de destino. [3]

- **Capa de enlace**

  Esta capa transporta paquetes entre dos interfaces de la capa de red de dos diferentes hosts que usan el mismo link; cuyo proceso es controlado en mayor parte por el software del driver de la tarjeta de red. Las funciones de esta capa agrupan las de las capas de enlace de datos y física del modelo referencia OSI. [4]

  Con respecto a SNMP capa recibe los datos de las capas superiores para enviarlos dentro de las tramas a la dirección física del equipo remoto, o también, recibe las tramas desde la red física y las prepara para el proceso de desencapsulación hacia las capas superiores de tal forma que los datos sean procesados finalmente por la aplicación SNMP en la capa de aplicación. [3]

2.3 Protocolo Simple de Gestión de Red

SNMP es un protocolo de la capa de aplicación que permite el intercambio de mensajes informativos de gestión entre los dispositivos de una red. SNMP forma parte del protocolo TCP/IP. SNMP permite a los administradores de

red supervisar la productividad de la red, buscar y resolver sus problemas y planear el crecimiento de la red. [1]

Existen tres versiones de SNMP hasta ahora, que son: SNMPv1, SNMPv2 y SNMPv3. Cabe recalcar que SNMP no es una aplicación, por lo que es necesario de una aplicación para utilizar los mensajes de SNMP. Se puede utilizar cualquier aplicación de administración compatible con SNMPv3 para administrar su servidor. Cualquier equipo que ejecute un software o una aplicación de gestión SNMP es un sistema de gestión SNMP. Algunos sistemas de gestión de red SNMP son: HPOpenView, Net View, Boss, CiscoWorks, etc. [7],[12]

SNMP desempeña un papel importante en lo que es gestión de redes, ya que ayuda a proporcionar una interfaz uniforme para acceder y gestionar todos los dispositivos de red. [1]

2.3.1 Descripción

SNMP define una relación de cliente-servidor. El programa de cliente (llamado sistema administrador de red, o NMS) establece conexiones virtuales a un programa servidor (denominado el agente SNMP) que se ejecuta en un dispositivo de red remoto, y provee de información al NMS sobre el estado del dispositivo, enviando mensajes a través de IP utilizando paquetes UDP. La base de datos, controlados por el agente de SNMP, se refiere como la base de información de gestión SNMP (MIB) y es un conjunto estándar de valores estadísticos y de control. SNMP permite además la extensión de estos valores estándar con valores específicos para un agente en particular a través del uso de las MIB privadas.

Las directivas, emitidas por el cliente NMS a un agente SNMP, consisten en los identificadores de las variables SNMP (conocidos como identificadores de objetos MIB o variables MIB) junto con instrucciones para obtener ya sea el valor para el identificador o el identificador para establecer un nuevo valor. A través del uso de las variables del MIB privadas, los agentes SNMP se pueden adaptar para innumerables dispositivos específicos, tales como puentes, pasarelas de red y routers. Las definiciones de las variables del MIB soportados por un agente en particular se incorporan en los ficheros de

descripción, escritos en formato de notación de sintaxis abstracta (ASN.1), puesto a disposición de programas de la red de gestión de clientes para que puedan tomar conciencia de las variables del MIB y su uso. [1]

SNMP se dice es un protocolo que distribuye las operaciones de gestión. Un host puede ser un NMS y agente monitoreado a la vez, o solamente actuar como cualquiera de ellos; si desempeña ambas funciones, otro NMS es el encargado de consultar por la información que se almacena localmente a este host. [7]

2.3.2 Componentes básicos de SNMP

Una red administrada consiste en 4 componentes básicos de SNMP:

- **Dispositivos gestionados o Managed Devices (MD):** Es un dispositivo de red físico que forma parte de una red administrada y que contiene un agente SNMP. Los MD llevan una base de datos interna que almacena los objetos de administración poniéndolos a disposición del NMS usando SNMP. Ejemplos de MD son routers y

servidores de acceso, switches, hubs y puentes, hosts informáticos e impresoras.

- **Agente:** Es un módulo de software de gestión de red que reside en un dispositivo gestionado. Un agente tiene conocimiento local de gestión de la información y traduce esa información en una forma compatible con SNMP.
- **MIB:** Consiste en la gestión de la información que reside en el dispositivo gestionado. El agente proporciona un estándar de acceso a la MIB.
- **NMS o Sistema administrador de Red:** Es el que ejecuta aplicaciones de monitoreo y control que gestiona los dispositivos. El NMS proporciona la mayor parte de los recursos de procesamiento y memoria necesarios para la gestión de la red. Deben existir uno o más NMS´s en cualquier red gestionada. [1]

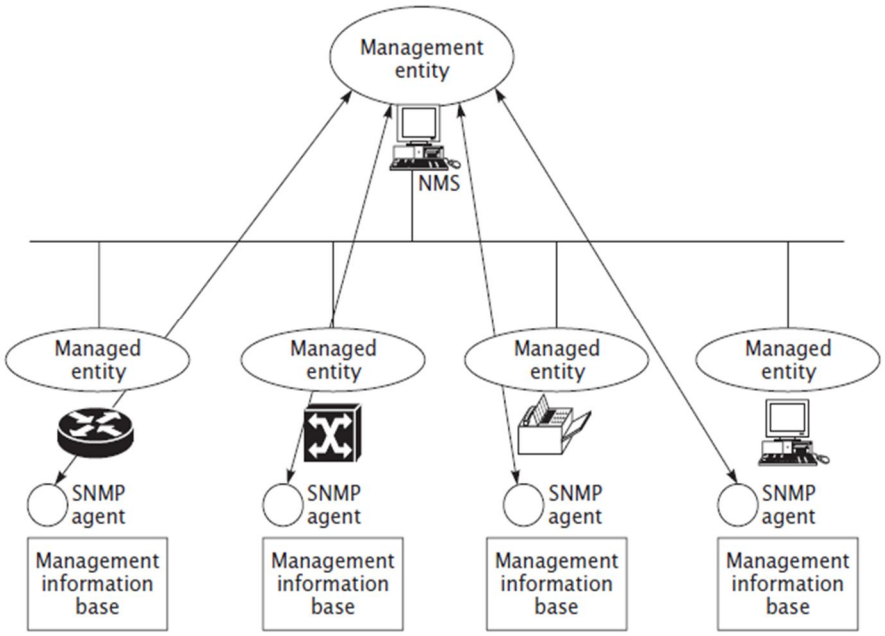


Figura 2.3 – Componentes básicos de SNMP. [1]

2.3.3 Comandos principales en SNMP

Los MD son supervisados y controlados utilizando 4 comandos básicos de SNMP:

- **Read:** Este comando es utilizado por un NMS para supervisar los MD. El NMS examina distintas variables que son mantenidas por los MD.
- **Write:** Este comando es utilizado por un NMS para controlar los MD. Cuando un NMS emite un comando de escritura cambia el valor de uno o varios objetos de la MIB del MD, los cuales permanecen invariables con los nuevos valores al menos que haya otra escritura futura sobre los mismos.

- **Trap:** Este comando es utilizado por los MD para reportar eventos de forma asíncrona a los NMS´s. Cuando ocurre algún tipo de evento fuera de lo normal, un MD envía un TRAP hacia el NMS.
- **Operaciones transversales o de recorrido:** Estas operaciones son utilizadas por los NMS´s para determinar las variables que pueden ser soportadas por los MD y obtener de manera secuencial la información en una tabla de variables, tal como una tabla de enrutamiento. [1]

2.3.4 MIB y OID

El componente básico de una implementación de SNMP es la base de información de administración o MIB (Management Information Base), esta base de información es equivalente a una base de datos, la cual está organizada jerárquicamente y es accesada utilizando el protocolo de administración de red SNMP; a través de la MIB se tiene acceso a la información para la gestión, contenida en la memoria interna del dispositivo en cuestión. MIB es una base de información completa y bien definida, está compuesta por objetos administrados e identificadores de objeto (OID).

Un objeto administrado (algunas veces llamado objeto MIB) es una entidad lógica que contiene cierto número de características específicas de un MD, donde cada dato vendría a ser una instancia de objeto que esencialmente es una variable. Los tipos de objetos administrados se dividen en dos: escalares y tabulares. Los objetos escalares definen una única instancia de objeto mientras que los tabulares definen múltiples instancias de objeto relacionadas entre sí, que están agrupadas conjuntamente en tablas MIB. [1]

Un identificador de objeto (OID) sirve para designar específicamente a un solo objeto administrado dentro de toda la jerarquía MIB. La jerarquía MIB se representa como un árbol con una raíz global (ISO) de la cual se derivan varios niveles, que son asignados por diferentes organizaciones y que se encuentran descritos en el documento de las MIB-I y MIB-II (RFC1066 y RFC1213 respectivamente). Los OID´s ubicados en la parte superior del árbol pertenecen a diferentes organizaciones estándares, mientras que los OID´s ubicados en las partes inferiores del árbol son colocados por las organizaciones asociadas, por ejemplo la rama private que pertenece al subárbol de internet y que agrupa objetos propietarios de varias empresas. [21]

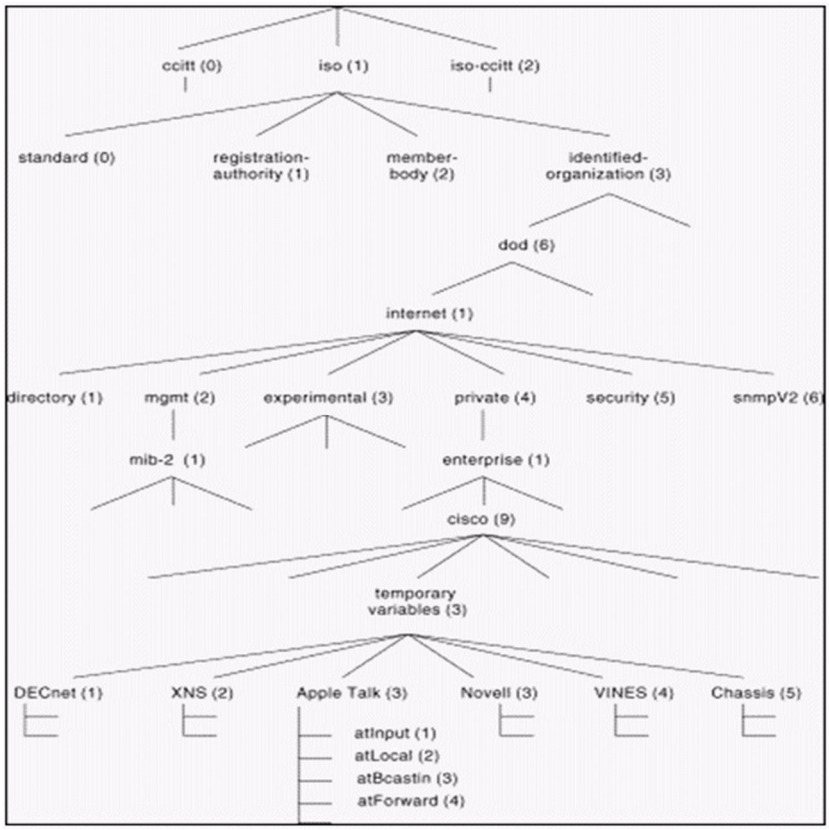


Figura 2.4 – Ejemplo de árbol MIB. [7]

El objeto administrado atForward (4) de la figura 2.4, podría ser identificado por un OID en 3 diferentes maneras:

- **Nombre:**
  iso.identified_organization.dod.internet.private.enterprise.cisco.temporary.AppleTalk.atForward
- **Descriptor de Objeto:** 1.3.6.1.4.1.9.3.3.4
- **Mixto:** 1.3.6.1.4.1.9.3.3.atForward

Toda rama principal, llamada nodo, del árbol MIB está compuesta por varios grupos de objetos, los cuales en su conjunto reciben un determinado nombre, un ejemplo es el árbol o nodo mib-2. Los grupos de objetos del mismo son los siguientes:

(1) System;

(2) Interfaces;

(3) AT;

(4) IP;

(5) ICMP;

(6) TCP;

(7) UDP;

(8) EGP;

(10) Transmission;

(11) SNMP.

Es importante destacar que la estructura de una MIB se describe mediante el estándar Notación Sintáctica Abstracta 1 (ASN.1). [7]

2.3.4.1 Tablas MIB SNMP

Los objetos tabulares, debido a sus múltiples variables, necesitan de una debida agrupación de sus instancias de objeto, de manera que permitan al

protocolo SNMP recuperar o alterar información de forma ordenada. Para esto SNMP tiene establecidas tablas estructuradas que están conformadas por varias filas (una tabla también puede tener 0 filas), donde una fila entera puede ser recuperada o modificada con un simple GET, GETNEXT o con el comando SET, ya que las filas están indexadas de manera que faciliten la obtención o modificación de datos. [1]

2.3.5 SMI

SMI significa Structure of Management Information y se encuentra definido en el documento RFC1155. Es un subconjunto adaptado de la notación ASN.1, y básicamente determina la forma de definir o nombrar los objetos administrados, su comportamiento y el tipo de variables que están asociadas a cada objeto) mientras que la MIB en cambio es la definición de los propios objetos.

El SMI utiliza 3 tipos de definiciones distintas:

- Definiciones de módulo.
- Definiciones de objeto.

- Definiciones de notificación. [24]

2.3.6 ASN.1

ASN.1 es un lenguaje formal estandarizado que fue definido por la CCITT (Comité Consultivo Internacional Telegráfico y Telefónico) y la ISO, que sirve para definir qué sintaxis tendrán los datos generados por las diferentes aplicaciones de capa 7; en el caso específico de SNMP, define la estructura del PDU.

Al ser un lenguaje estándar, permite tener muchas ventajas a la hora de desarrollar una aplicación: soporte interprotocolo, persistente a cambios de versiones por lo que no habrá problemas de compatibilidad entre las mismas, no depende de la compañía fabricante ni de plataforma o lenguaje de implementación; finalmente, es conocido en toda la industria.

Los siguientes términos son importantes en la notación ASN.1:

- Sintaxis abstracta: Describe la estructura general de los datos, definiéndoles tipos y valores posibles, sin importar qué técnica de codificación se usó para representarlos.
- Tipo de dato: Conjunto de nombres de valores, se describen más adelante.
- Codificación: Secuencia de octetos que representan los datos generados por las aplicaciones.
- Reglas de codificación: Define las formas de representar datos de una sintaxis específica en otra. Ver sección 2.3.7.
- Sintaxis de transferencia: Representación en bits de los datos cuando van de un host a otro. [23]

ASN.1 hace distinciones entre mayúsculas y minúsculas de la siguiente forma:

Elemento	Convención
Types	Inicial en mayúscula
Values	Inicial en minúscula
Macros	Todas las letras en mayúscula
Modules	Inicial en mayúscula
ASN.1 Keywords	Todas las letras en mayúscula

Tabla 2.1 – Tipos de Valor en ASN.1. [24]

Caracteres especiales en ASN.1:

Elemento	Nombre
-	Número con signo
--	Comentario
: :=	Asignación ("definido como...")
|	Alternativa (opciones de una lista)
{ }	Inicio y final de lista
[ ]	Inicio y final de una etiqueta (tag)
( )	inicio y final de una expresión de subtipo
. .	Indica un rango

Tabla 2.2- Caracteres especiales en ASN.1. [24]

ASN.1 define tres tipos de datos generales, los tipos que se mencionan a continuación son enfocados en la gestión de red de Internet:

a) Simples o primitivos: Almacenan un único valor, por lo que son llamados escalares. Los tipos primitivos más importantes se muestran en la tabla.

Tipo		Descripción
Integer		Numero entero positivo o negativo de hasta 32 bits.
Octet String	Display String	Cadena de caracteres ASCII imprimibles.
	OctetBitString	Usado para cadenas de bits mayores a 32 bits.
	PhyAddress	Representan direcciones de la capa de enlace de datos.
Object identifier		Identificador de objeto, que marca la posición del objeto dentro de la MIB.
Null		Representa la ausencia de valor.
Bolean		Representa un valor que puede ser verdadero o falso.

Tabla 2.3- Caracteres especiales simples en ASN.1. [24]

b) Estructurados o compuestos: Son registros vectoriales, puesto que sirve para definir filas y tablas. Son construidos a partir de otros tipos primitivos o compuestos. En la tabla se detalla los tipos de datos estructurados.

Tipo	Descripción
Sequence	Representa una fila, es decir una lista ordenada de tipos de datos diferentes. Son construidos a partir de tipos primitivos.
Sequence Of	Representa una tabla, es decir es una lista ordenada de varias filas iguales. Son construidos a partir de tipos compuestos.
Set	Es un tipo de datos similar a sequence, con la diferencia que la lista no está ordenada.
Set Of	Es un tipo de datos similar a sequence of, con la diferencia que la lista no está ordenada.
Choice	Es un tipo de dato en el que se debe escoger entre una lista previamente definida.

Tabla 2.4- Caracteres especiales estructurados en ASN.1. [24]

c) Definidos: Se construyen a partir de los tipos de datos anteriores (primitivos y compuestos) con la diferencia de que se les ha definido un nombre más descriptivo, se utiliza para distinguir los tipos dentro de una aplicación. Los tipos de datos definidos o etiquetados se detallan en la tabla. [24]

Tipo	Descripción
Network address	Representa una dirección de red de cualquier familia de protocolos.
IpAddress	Representa la dirección de red de internet (32 bits), definida en la pila de protocolos TCP/IP.
Counter	Representa un entero (integer) positivo, el cual se incrementa monótonamente hasta alcanzar $2^{12} - 1$. Se reinicia cuando alcanza el valor máximo.
Gauge	Representa un entero (integer) positivo, el cual se incrementa o decrementa monótonamente. Se reinicia cuando alcanza el valor máximo.
Time Ticks	Representa un entero (integer) positivo, el cual cuenta el tiempo transcurrido en centésimas de segundo.
Opaque	Representa un Octet String al que se le puede pasar cualquier valor ASN.1

Tabla 2.5- Caracteres especiales definidos en ASN.1. [24]

2.3.7 BER

Según la especificación ISO 8825, las Reglas Básicas de Codificación o BER, por sus siglas en inglés (Basic Encoding Rules), permiten traducir o codificar cualquier estructura de datos ASN.1 en cadenas de bytes para cuando se transmiten de nodo a nodo en una red, es decir, cuando viajan a través del cable. Este traspaso de los datos junto con su tipo, se da de forma eficiente, es decir, con la menor cantidad de bytes posibles.

SNMP utiliza un subconjunto de estas reglas. BER usa tres campos para codificar de forma recursiva los tipos de datos:

- *Tag:* Tipo de ASN.1.
- *Length:* Longitud del valor codificado que sigue.
- *Value:* La codificación del valor. [1],[2]

2.3.8 Operaciones del protocolo SNMP

SNMP es un protocolo simple de solicitud y respuesta. El NMS realiza una petición y los MD retornan una respuesta. Este comportamiento se implementa mediante el uso de las siguientes operaciones básicas:

- **Get-request.-** Utilizado por el NMS para consultar el valor de uno o varios objetos de la MIB del agente SNMP de la estación remota.
- **Get-next-request.-** Es similar a la anterior, con la diferencia que se obtiene el valor de una variable sin definir la que se envió en la solicitud, sino que se obtiene el valor de la variable siguiente al orden lexicográfico (jerárquico) dentro del árbol de la MIB.
- **Get-response.-** Es la respuesta del agente a una petición del NMS devolviendo el valor de una o más variables.
- **Set-request.-** Con este tipo de solicitud el NMS escribe o modifica uno o varios valores de las variables MIB de la estación remota.
- **Trap.-** Cuando en el agente se produce un evento inesperado que afecte a la MIB o a los recursos gestionados, envía un trap para notificarlo al NMS. Dado que el mensaje se envía de forma asíncrona (no se envía previa solicitud) y en cualquier momento, el NMS debe estar escuchando por un puerto específico (generalmente el 162) la llegada de estos traps, para almacenarlo o tomar una decisión dirigida a resolver el problema. [24]

2.4 Versiones de SNMP

2.4.1 SNMPv1

Para determinar qué NMS puede acceder a una parte o todos los objetos de la MIB, el esquema de autenticación de SNMPv1 es bastante simple; e incluye políticas de acceso relacionadas con comunidades, definición de modos de acceso y vistas MIB. [11]

Una comunidad agrupa varios hosts relacionados mediante un nombre de comunidad. Debido a que las comunidades básicamente son contraseñas que se ingresan en la solicitud del NMS permitiéndole el acceso a un agente SNMP, no es recomendable que sean configuradas usando palabras de diccionario o las comunidades por defecto public/private; sino, crear comunidades nuevas usando una combinación de letras minúsculas, mayúsculas, números o caracteres especiales. Adicionalmente, no hay que olvidar que las comunidades se transmiten en texto plano, por lo que es fácil interceptarlas y usarlas en perjuicio nuestro. Esta situación mejora considerablemente en la versión 3. [3]

El modo de acceso determina para alguna comunidad específica, qué tipo de acceso tiene permitido para realizar operaciones a los objetos de la MIB, ya sea ninguno (none), lectura y escritura (read-write), sólo lectura (read-only) o sólo escritura (write-only). En cambio, una vista de la MIB define sobre cuáles subárboles de la MIB una determinada comunidad puede acceder para realizar las operaciones de los modos de acceso.

Cuando el agente recibe una petición, se revisa en los campos del mensaje la IP de origen y el nombre de comunidad para determinar si el host que envió el mensaje realmente pertenece a esa comunidad. Luego, si esta verificación tiene éxito, se revisan las vistas y modos de acceso para la comunidad, para saber a cuáles variables tendrá acceso el agente y qué operaciones podrá realizar en ellas. Si existe una vista y el permiso de realizar operaciones en ella, el agente responderá a la petición de la forma adecuada; de lo contrario, devolverá el mensaje de error hacia el origen de la solicitud.

Las cuatro operaciones básicas para el acceso a la información de gestión contenida en la MIB son el Get, GetNext, Set y Trap, esta última no accede a información debido a peticiones, sino que envía notificaciones o avisos de problemas en el host. [11]

2.4.2 SNMPv2

SNMPv2 o mejor conocida como SNMPv2c (SNMPv2 basado en comunidades) es una extensión de SNMPv1 y usa los mismos elementos para su funcionamiento, esto es, las comunidades, vistas de la MIB y modos de acceso. Tampoco hay cambios significativos en la estructura del PDU, solamente cambia el número de la versión.

A las operaciones de acceso a objetos de SNMPv1 se suman el GetBulk e Inform. GetBulk se usa para obtener grandes bloques de información con una sola petición en vez de usar grandes cantidades de Get o GetNext, e Inform se usa para el envío de notificaciones que deben ser respondidas con un acuse de recibo por parte del receptor. Finalmente se definen nuevos tipos de objetos dentro de la MIB para mejorar la semántica. [11]

2.4.3 SNMPv3

SNMPv3 añade seguridad más robusta a las versiones previas y para ellos de vale de dos modelos, el modelo de seguridad basado en usuarios o USM (User-based Security Model), que añade autenticación y cifrado a los mensajes, y el modelo de control de acceso basado en vistas o VACM (View-based Access Control Model) para el control de acceso a las variables de la MIB. SNMPv3 también introduce la posibilidad de configurar remotamente al agente dando distintos valores a los objetos que representan su configuración. [11]

2.4.3.1 Entidad SNMPv3

Una entidad SNMPv3 está compuesta por el motor SNMP y las aplicaciones.

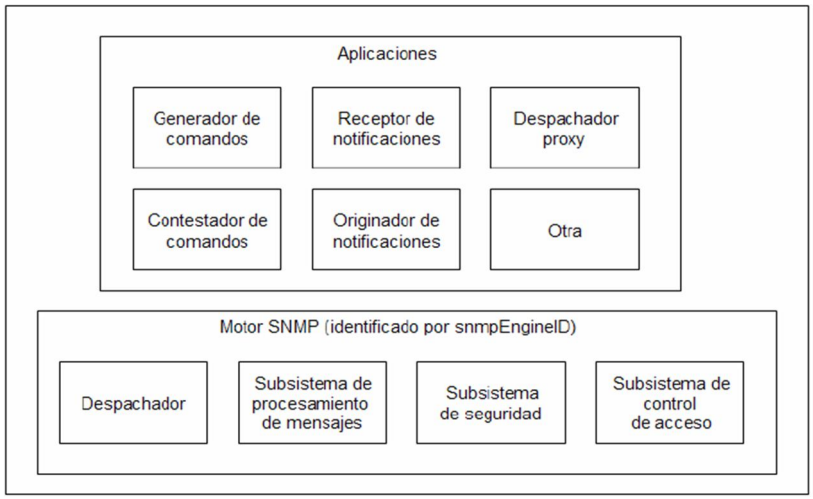


Figura 2.5 - Diagrama de una entidad SNMPv3. [23]

2.4.3.1.1 Motor SNMP

El motor está compuesto por 4 módulos: despachador, subsistema de procesamiento de mensajes, subsistema de seguridad y subsistema de control de acceso. El despachador verifica la versión del mensaje SNMP recibido antes de enviarlo al subsistema de procesamiento de mensajes; y además envía mensajes hacia motores externos. El subsistema de procesamiento contiene algunos sub módulos, por ejemplo, para procesar solicitudes de mensajes exclusivas para versión 1, 2 ó la 3; extrae la información de los mensajes recibidos y termina de preparar los que serán enviados. El subsistema de seguridad provee servicios de autenticación basados en comunidades o en usuarios, la de usuarios hace uso de los algoritmos MD5 (Algoritmo de resumen de mensaje v5) o SHA (Algoritmo de hash seguro) para autenticar los mensajes sin transmisión de contraseñas en texto plano. Este subsistema también provee servicios de privacidad para cifrar el contenido de los mensajes SNMP enviados haciendo uso de algoritmos como DES (Estándar de Cifrado de Datos) o AES (Estándar Avanzado de Cifrado). Finalmente, el subsistema de control de acceso permite definir a qué objetos un usuario puede acceder dentro de la MIB y qué operaciones puede realizar en ellos.

2.4.3.1.2 Aplicaciones SNMP

- Generador de comandos: Implementada por un NMS, genera solicitudes de lectura y escritura para las entidades monitoreadas y procesa sus respuestas.
- Respondedor de comandos: Implementada por un MD, responde a las diferentes solicitudes que recibe un NMS.
- Originador de notificaciones: Genera traps para ser enviadas al NMS.
- Receptor de notificaciones: Reside en el NMS, procesa las traps recibidas.
- Enviador proxy: Facilita el paso de mensajes entre entidades. [3]

2.4.3.2 Problemas de seguridad en un entorno de monitoreo

Las primeras versiones de SNMP no proveían mayores capacidades de seguridad que las solas comunidades, que actúan como claves, pero como ya sabemos, cualquiera puede adivinarlas si no se cambia las que vienen por defecto (public o private) para lectura o escritura respectivamente; también pudiendo obtenerlas por algún programa externo ya que se transmiten en texto plano con el resto de la información. Estas limitantes en la seguridad dejan expuesta a la red monitoreada a una serie de amenazas como son:

- **Alteración de la secuencia de los mensajes:** Es cuando una entidad no autorizada de manera maliciosa retrasa, fragmenta y reordena, copia y vuelve a enviar los mensajes que envía una entidad autorizada.
- **Modificación de la información:** Es cambiar el tráfico en tránsito generado por una entidad administradora autorizada, por parte de una no autorizada; para poder ejecutar tareas de administración de configuración o contabilidad no deseadas.
- **Enmascaramiento:** Es suplantar la identidad de una entidad administradora autorizada, para acceder a información restringida de la red mediante el uso de sus credenciales.
- **Revelación de datos confidenciales:** Cuando una entidad no autorizada captura tráfico entre los NMS´s y agentes de una red monitoreada, así como notificaciones o traps, teniendo así a mano información delicada de la red y la cual algunas veces es publicada de forma ilegal.

Frente a estos problemas, la última versión de SNMP implementa los algoritmos de firmas digitales y cifrado del tráfico de monitoreo para lograr sobre todo la integridad de los datos y que solamente los administradores de

la red sean los que puedan visualizar y modificar variables concernientes a su gestión. [1]

2.4.3.3 Modelo de Seguridad Basado en Usuarios (USM)

Utiliza el concepto del motor SNMP autoritativo (AuthoritativeEngine). En todo proceso de transmisión de mensajes, una de las dos entidades, ya sea, transmisora o receptora, será designada como el motor SNMP autoritativo, dependiendo de dos reglas:

- Cuando una entidad realiza una solicitud esperando una respuesta, la entidad receptora será autoritativa.
- Cuando una entidad recibe un mensaje sin solicitarlo (trap, response, report), la entidad transmisora será autoritativa.

Este motor autoritativo es el que sirve como referencia para el control de la sincronización entre agente y NMS.

Tiene tres módulos con diferentes especificaciones:

- *Módulo de autenticación:* Es el que se encargar de proporcionar la autenticación e integridad de un mensaje SNMP. Es decir, este

modelo se asegura de que el mensaje provenga de la fuente de donde dice ser generado y que en el proceso del envío no haya sido modificado. Para la comunicación entre entidades es necesario compartir una llave de autenticación.

- *Módulo timeliness:* Es el que proporciona la protección contra el retraso o retransmisión de los mensajes. El NMS copia unos parámetros del agente para la sincronización (descrito en la sección 2.6), por medio de esto el NMS puede hacer una estimación del tiempo en que los mensajes van a llegar al agente, con la cual se decide si se descarta o no un mensaje dependiendo si existe una diferencia en más o en menos de 150 segundos, esta estimación de tiempo se guarda en una memoria no volátil del agente.
- *Módulo de privacidad:* Es el que proporciona el cifrado y descifrado del contenido (datos) de un mensaje. Para realizar el cifrado del mensaje se necesitan los datos, una llave de privacidad y un vector de inicialización formado por un valor aleatorio de la entidad que envía el mensaje.

Según lo establecido por el protocolo SNMP no es posible realizar el cifrado de los datos sin realizar la autenticación, tampoco es posible lograr confiabilidad sin la autenticación y el cifrado de los datos. [19],[20],[22]

2.4.3.4 Modelo de Seguridad basado en Vistas (VACM)

Para realizar procesos de solicitudes y modificaciones de datos, se necesitan permisos de acceso a dichos datos para llevar un control. VACM tiene 5 elementos: grupos, niveles de seguridad, contexto, vistas de MIB y políticas de acceso.

Los grupos determinan los permisos de acceso de los NMS´s. Están formados por pares de modelo de seguridad y nombre de seguridad <securityModel , securityName>. Por medio de estos grupos se puede acceder a los objetos administrados. A cada grupo se le asigna un identificador único llamado groupName.

El nivel de seguridad del mensaje en una solicitud determina las diferentes posibilidades de acceso para un grupo. Pueden asignarse diferentes niveles de seguridad para distintos tipos de mensajes y una misma entidad. Estos niveles de seguridad son: noAuthNoPriv (llamado también noauth) que no ofrece seguridades para el mensaje ya que no lo autentica ni lo cifra; authNoPriv (llamado también auth) que solamente realiza la autenticación del mensaje y authPriv (llamado también priv) que autentica y cifra el mensaje.

Context es un subconjunto de instancias de objetos en el ámbito local de una MIB, pudiendo un objeto pertenecer a más de un solo contexto. Una entidad SNMP puede tener acceso a más de un contexto para lo cual debe tener definida una tabla llamada vacmContextTable para establecer a con qué contextos puede establecer comunicación solamente.

Las vistas MIB sirven como un filtro de acceso para las peticiones hechas por un grupo particular a un subgrupo de objetos de administración en el agente; cada objeto administrado en la MIB puede encontrarse incluido en una vista. Una vista MIB se puede formar por varios objetos pertenecientes a una sola

rama o de combinaciones de varios subárboles. También pueden definirse familias de vistas de subárboles, a las que deberá asignarse un nombre.

Las políticas de acceso definen los privilegios de acceso del NMS principal o uno o varios intermedios, para los datos informativos del agente y se configuran sobre los grupos pueden configurarse para los grupos. Para crear las políticas de acceso se necesita definir usuarios, grupos, nivel de seguridad, modelo de seguridad, contexto y tipo de acceso solicitado. [22]

2.4.3.5 Algoritmos de seguridad en la autenticación de usuario

Los protocolos usados por SNMPv 3 para la autenticación del usuario están basados en algoritmos de hash y son los siguientes:

- MD5, Message Digest Algorithm v5.
- SHA, Secure Hash Algorithm.

Un algoritmo hash es una función computacional, llamada también de resumen, por medio de la cual, a una entrada determinada (por ejemplo una contraseña, un documento de texto, archivo, etc) de cualquier longitud, se producirá una salida de tamaño fijo, totalmente diferente para cada entrada. Otra característica de una función hash es que es imposible reconstruir el texto origen a partir del valor resumen o resultado, además de que idealmente no habrá colisiones, es decir que a valores diferentes de entradas no se puede tener un mismo hash de salida. Obviamente, en la vida real, los algoritmos de hash sí pueden generar colisiones debido a que el conjunto de valores de entrada a la función es infinito, es decir, puedes ingresar cualquier contraseña de cualquier extensión para obtener siempre una salida de tamaño fijo.

Por ejemplo, el MD5 genera una salida de 128 bits en la forma de 32 dígitos hexadecimales. Para que no haya colisiones, el número de posibles entradas debería estar limitado a 2^{128} palabras, pero como ya se mencionó las posibilidades de ingreso son infinitas. Lo que se trata con los algoritmos de hash modernos, es que sean lo suficientemente complejos, que las colisiones sean las mínimas posibles y muy difíciles de hallar. [30]

2.4.3.5.1 MD5

El proceso de autenticación de un mensaje saliente comienza cuando dos llaves se originan de la llave secreta authKey de 16 octetos. Se comienza por extender a 64 octetos la authKey al aumentarle 48 octetos de ceros a cuyo resultado se le llama extendedAuthKey; se origina la cadena de octetos de relleno interior IPAD (inner padding) al replicar el octeto 0x36 64 veces y a la operación XOR entre el extendedAuthKey e IPAD se le denomina llave K1.
Luego se origina la cadena de octetos de relleno exterior OPAD (outer padding) al replicar el octeto 0x5c 64 veces y se obtiene la llave K2 haciendo la operación XOR entre el extendedAuthKey y el OPAD.

Se antepone la llave K1 al mensaje y a todo esto se le calcula el resumen MD5. Al resultado de la operación se le antepone la llave K2 y se vuelve a computar el resumen MD5 de 128 bits (16 octetos), del cual los primeros 96 bits (12 octetos) son el código de autenticación del mensaje (MAC) e irán en el campo msgAuthenticarionParameters del mensaje saliente.

Cuando una entidad recibe un mensaje que requiere autenticación, extrae el valor que viene en el parámetro msgAuthenticarionParameters y lo reserva; hace el mismo procedimiento para hallar las llaves K1 y K2 y con ellas calcula el resumen MD5 como se describió. De la misma forma, los primeros 12 octetos del resumen se colocan en el campo msgAuthenticarionParameters además de compararlos con el valor que se tenía en reserva. Si ambos coinciden, el mensaje es autenticado correctamente y se procede entonces a verificar la validez temporal. [19]

2.4.3.5.2 SHA

El proceso de autenticación de un mensaje saliente comienza cuando dos llaves se originan de la llave secreta authKey de 20 octetos. Se comienza por extender a 64 octetos la authKey al aumentarle 44 octetos de ceros a cuyo resultado se le llama extendedAuthKey; se origina la cadena de octetos de relleno interior IPAD al replicar el octeto 0x36 64 veces y a la operación XOR entre el extendedAuthKey e IPAD se le denomina llave K1. Luego se origina la cadena de octetos de relleno exterior OPAD al replicar el octeto 0x5c 64 veces y se obtiene la llave K2 haciendo la operación XOR entre el extendedAuthKey y el OPAD.

Se antepone la llave K1 al mensaje y a todo esto se le calcula el resumen SHA. Al resultado de la operación se le antepone la llave K2 y se vuelve a computar el resumen SHA de 160 bits (20 octetos), del cual los primeros 96 bits (12 octetos) son el código de autenticación del mensaje (MAC) e irán en el campo msgAuthenticarionParameters del mensaje saliente.

Cuando una entidad recibe un mensaje que requiere autenticación, extrae el valor que viene en el parámetro msgAuthenticarionParameters y lo reserva; hace el mismo procedimiento para hallar las llaves K1 y K2 y con ellas calcula el resumen SHA como se describió. De la misma forma, los primeros 12 octetos del resumen se colocan en el campo msgAuthenticarionParameters además de compararlos con el valor que se tenía en reserva. Si ambos coinciden, el mensaje es autenticado correctamente y se procede entonces a verificar la validez temporal. [19]

2.4.3.6 Algoritmos de seguridad en el cifrado del mensaje

Los protocolos usados por SNMPv3 para la autenticación del usuario son DES o AES, el cual usa llaves de 128 bits (también hay llaves de 192 y 256 bits).

El algoritmo DES, al poseer una clave relativamente corta no es muy fuerte para el cifrado por lo que ha sido reemplazado por AES ya que hasta ahora, AES se mantiene como un algoritmo seguro y que ha resistido a los ataques.

2.4.3.6.1 DES

- **Llave DES y vector de inicialización**

  Los primeros 8 octetos de la llave privKey de 16 octetos se usan como la llave DES de función hash para cifrar el mensaje; pero los bits menos significativos de cada octeto se descartan antes, ya que el algoritmo trabaja con llaves de 56 bits. Los 8 últimos octetos de la llave privKey se usan como pre vector de inicialización (pre-IV). El vector de inicialización (IV) se forma como se describe a continuación.

Una forma de asegurar que el IV no se repita para dos paquetes diferentes cifrados a partir de la misma llave, se necesita "condimentar" al pre-IV con algo único por paquete. Se llamará "condimento" a un valor de 8 octetos formado por la concatenación de 4 octetos correspondientes a snmpEngineBoots seguidos por 4 octetos que el motor local mantiene, generados de forma arbitraria al momento de booteo. Luego se hace la operación XOR entre el condimento y el pre-IV para formar el IV. El "condimento" se coloca en el campo privParameters del mensaje para habilitar a la entidad receptora calcular el IV correcto y así poder descifrar el mensaje. [19]

- **Cifrado del mensaje**

  El cifrado se realiza en modo Cipher Block Chaining - CBC por sus siglas en inglés (Encadenamiento de Bloques de Cifrado). Los datos a cifrar, específicamente el scopedPDU del mensaje (contextEngineID, ContexName y datos), se tratan como cadena de octetos y su longitud debe ser múltiplo de 8, si no lo es, se hace un relleno al final para cumplirlo.

El texto plano se divide en bloques de 64 bits (8 octetos). A cada bloque de texto plano (plaintext) se le hace un XOR con el texto cifrado (ciphertext) previo, el resultado se cifra y la salida de este proceso será el texto cifrado para el presente bloque. El procedimiento sigue hasta que ya no queden más bloques de texto plano. Para el primer bloque de texto plano, el vector de inicialización es usado en vez de un texto cifrado de algún bloque previo, ya que no lo hay.

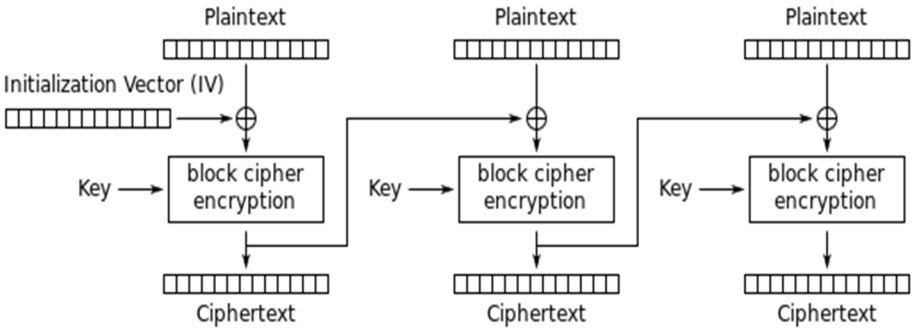


Figura 2.6 - Cifrado DES en modo Encadenamiento de Bloques de Cifrado. [28]

- **Descifrado del mensaje.**

  El primer bloque de texto cifrado es sometido al algoritmo de descifrado, a la salida del descifrado se le hace un XOR con el vector

de inicialización y el resultado es el primer bloque de texto plano. Para cada bloque siguiente, el bloque de texto cifrado se descifra, a la salida del descifrado se le hace un XOR con el bloque de texto cifrado previo y el resultado es el bloque de texto plano.

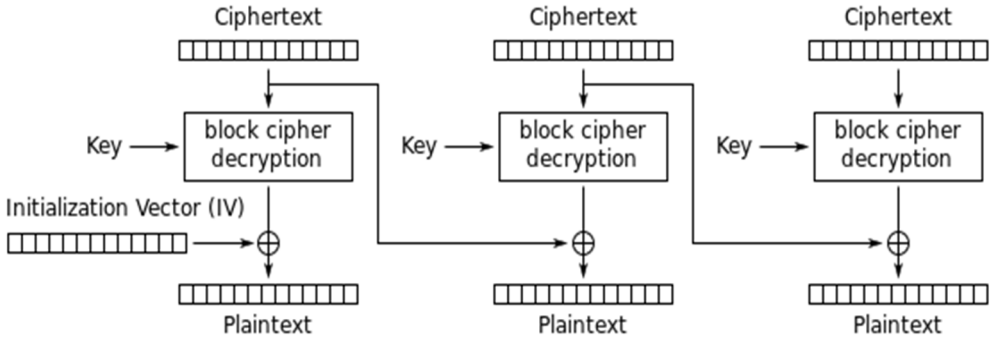


Figura 2.7 - Descifrado DES en modo Encadenamiento de Bloques de Cifrado. [28]

2.4.3.6.2 AES

- **Llave localizada, Llave de Cifrado AES y Vector de Inicialización.**

  Este protocolo de cifrado debe usarse en conjunto con protocolos de autenticación que generen llaves localizadas de 128 bits como mínimo. Más información en la sección 3.7.1. Los 128 bits (16 octetos) de la llave localizada se usan como la llave AES de la función hash

para cifrar el mensaje. El vector de inicialización (IV) es la concatenación de los siguientes valores: el valor de 32 bits correspondiente a snmpEngineBoots del motor autoritativo, 32 bits correspondientes a snmpEngineTime y 64 bits correspondientes a un valor generado de forma pseudo-aleatoria al momento de booteo por el motor local. Estos 64 bits son el “condimento” de los mensajes y son colocados dentro del campo msgPrivacyParameters como cadena de octetos para permitir que la entidad receptora descifre el mensaje (ya que al estar ambas entidades sincronizadas, comparten los valores del módulo timeliness y solo les falta este valor para generar el IV). [20]

- **Cifrado del mensaje**

  El cifrado se realiza en modo Cipher Feedback - CFB (Cifrado por Retroalimentación). Los datos a cifrar, específicamente el scopedPDU del mensaje (contextEngineID, ContexName y datos), se dividen en bloques de 128 bits (16 octetos), el último bloque podría tener menos de 128 bits pero no se requiere relleno.

Se aplica la operación de cifrado ($CIPH_K$) al IV para producir el primer bloque de salida (output block), al cual se le hace un XOR con el primer bloque de texto plano (plaintext) produciendo así el primer bloque de texto cifrado (ciphertext).

El bloque de texto cifrado se usa como bloque de entrada a la siguiente operación de cifrado. El último bloque de texto cifrado se obtiene haciendo un XOR entre el último bloque de texto plano de r bits (r puede ser menor a 128 bits) y los r bits más significativos del último bloque de salida.

- **Descifrado del mensaje.**

  El IV es el primer bloque de entrada, a la función de cifrado inversa ($CIPH^{-1}_K$). El primer bloque de texto cifrado se usa para el segundo bloque de entrada, el segundo texto cifrado se usa para el tercer bloque de entrada y así sucesivamente. La función de cifrado se aplica a cada bloque de entrada para producir los bloques de salida a los cuales se les aplica un XOR con los correspondientes bloques de

texto cifrado para recuperar los bloques de texto plano. Al último bloque de texto cifrado de r bits (r puede ser menor a 128 bits) se le hace un XOR con el segmento de los r bits más significativos del último bloque de salida para obtener el último bloque de texto plano.

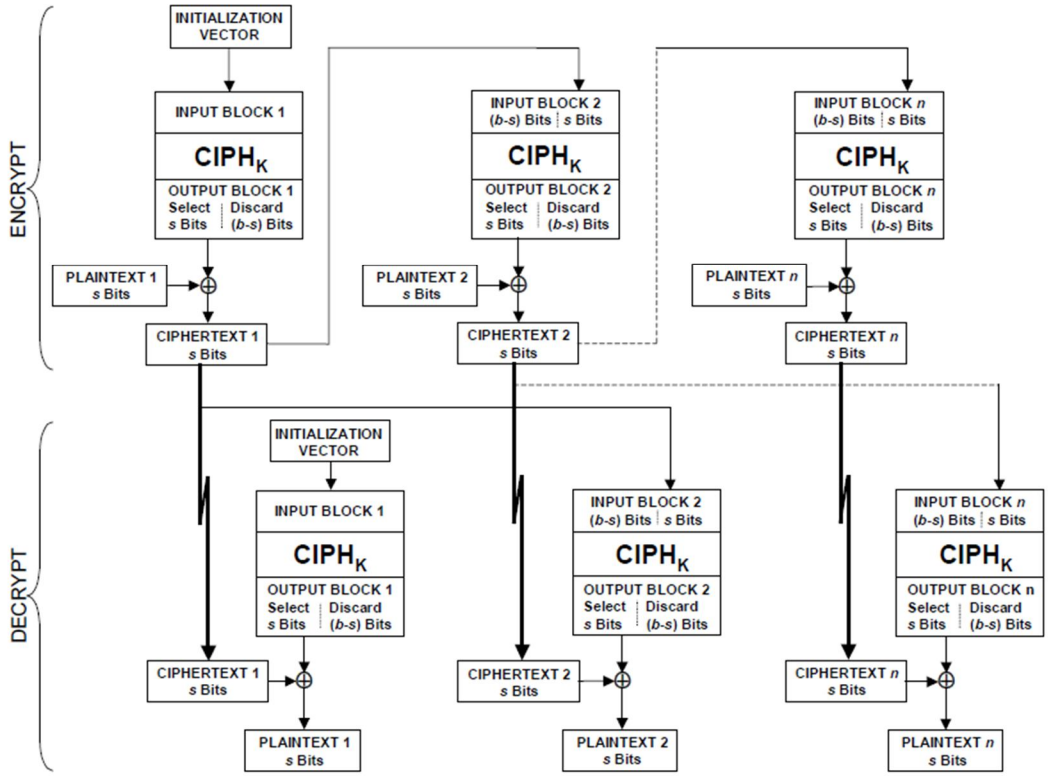


Figura 2.8 - Proceso de cifrado y descifrado AES en modo Cifrado por Retroalimentación. [13]

2.5 Formato de mensajes SNMP v1 y v2

Los Mensajes SNMP v1 y v2 contienen 2 partes, encabezado del mensaje y unidades de datos de protocolo o PDU.

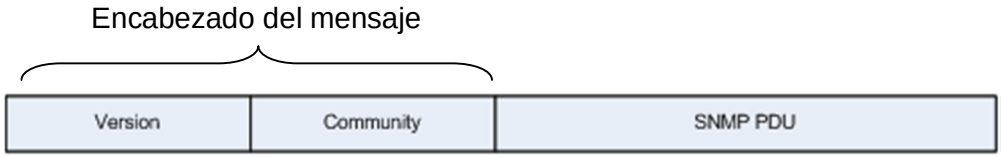


Figura 2.9 – Formato de mensajes SNMP v1 y v2. [5]

2.5.1 Mensaje de encabezado

El encabezado del mensaje para SNMP v1 y v2 contiene 2 campos:

1. El número de versión de SNMP (Version)
2. Nombre de la comunidad (Community)

El concepto de comunidad se describe con detalle en la sección 2.4.1.

2.5.2 PDU (Protocol Data Unit) v1 y v2

El PDU de SNMP contiene un comando especifico (Get, Set, etc.) y operandos que indican las instancias del objeto involucradas en la transacción. Los campos PDU de SNMP son de longitud variable, según lo especificado por ASN.1

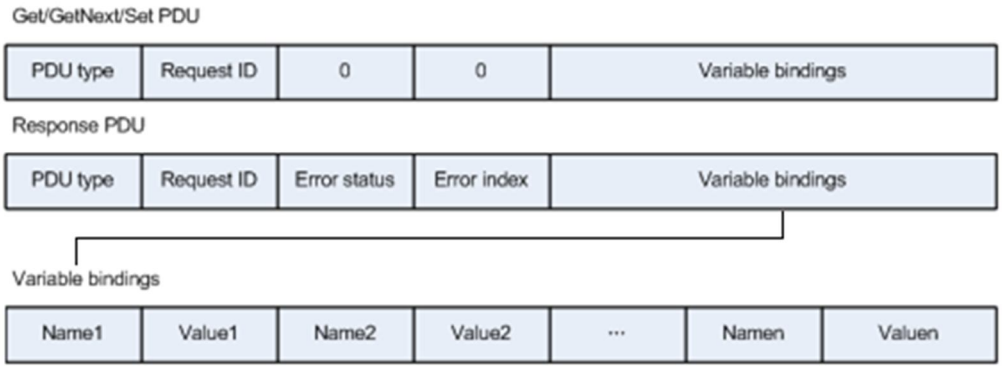


Figura 2.10 – Formato de la PDU v1 y v2. [5]

Descripción de los campos de la figura 2.10 para Get, GetNext, Set y Response:

- *PDU Type:* Especifica el tipo de mensaje. Los valores posibles son:

Tipo PDU	Nombre
0	Get-request
1	Get-next-request
2	Set-request
3	Response
4	Trap

Tabla 2.6- Tipos de PDU. [14]

- *Request ID:* Relaciona las solicitudes y respuestas SNMP; el emisor del mensaje coloca en un mensaje de solicitud de salida, un número

que de manera única va a identificarlo, así mismo como a la respuesta pendiente que debe llegar; de tal forma que cualquier intento de un atacante de enviar un valor ficticio para una consulta determinada será infructuoso ya que el mensaje será descartado al no coincidir los valores. Esto se debe sobre todo al esquema de transporte no seguro (UDP) que usa SNMP para enviar los mensajes.

- *Error Status:* La consulta no se ha podido llevar a cabo debido a algún inconveniente, los tipos de errores son los siguientes:

Estado de error	Nombre	Significado
0	NoError	No hay error.
1	tooBig	La PDU es demasiado grande para ser procesada.
2	noSuchName	No existe tal nombre o variable.
3	badValue	Valor incorrecto, no se ajusta a la definición de la variable
4	readOnly	Solo lectura, la variable no puede ser modificada.
5	genErr	Error general.

Tabla 2.7- Estado de error. [14]

- *Error Index:* Asocia un error con una instancia de objeto en particular. Cuando el estado de error contiene un valor diferente de cero, el índice de error indica qué instancia de objeto de la MIB generó el error. El agente la utiliza sólo para los errores "noSuchname","badValue" y "Readonly".
- *Variable Bindings:* Es una lista de nombres de variables con sus valores correspondientes (codificados en ASN.1). El campo *valor* existe tanto en las preguntas como en las respuestas, pero en las preguntas su contenido es nulo. [7],[14]

2.5.3 Formato de la PDU TRAP v1 y v2

La figura 2.11 muestra los campos del PDU TRAP.

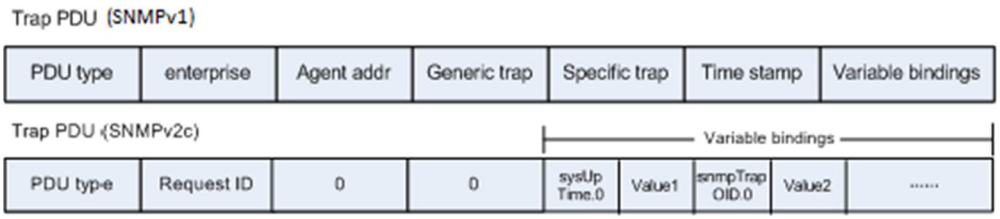


Figura 2.11 – Campos de la PDU Trap.

La descripción de los campos del PDU TRAP ilustrados en la figura 2.11 son los siguientes:

- *Enterprise:* Identifica el tipo de subsistema de gestión u objeto administrado que ha emitido el trap, a través del nombre del mismo, fabricante y versión.
- *Agent Address:* Proporciona la dirección IP del objeto administrado que genera el trap. [7]
- *Generic Trap Type:* Indica uno de una serie de tipos de trap genéricos así como las causas que originaron su envío. Entre las posibles causas se encuentran:

Tipo de Trap	Nombre	Significado
0	Cold Start	Indica que el agente ha sido reiniciado. Todas las variables de administración son reseteadas (como Counters o Gauges). Generalmente se debe a un crash.
1	Warm start	Indica que el agente se ha reinicializado, ninguna de las variables administrativas cambiarán. Generalmente es un reinicio controlado.
2	Link down	Indica que una interfaz de comunicación se encuentra fuera de servicio (inactiva). Dentro del trap se especifica primero el nombre y luego el valor del índice de la interfaz que tuvo el problema.
3	Link up	Indica que una interfaz de comunicación se encuentra en servicio (activa). Dentro del trap se especifica primero el nombre y luego el valor del índice de la interfaz que volvió a estar operativa.
4	Authenticantion failure	El agente ha recibido un mensaje que indica que un mensaje no ha pasado la

		autentificación en el agente (sea local o remoto).
5	EGP neighborLoss	Indica que en sistemas en que los routers están utilizando el protocolo EGP, un equipo vecino se encuentra fuera de servicio.
6	Enterprise specific	En esta categoría se encuentran todos los nuevos traps incluidos por los vendedores. Cualquier empresa puede incluir sus propios traps bajo la rama *private-enterprise* del árbol MIB. El NMS que recepte esta clase de trap debe decodificar el número de trap específico al interior del mensaje SNMP.

Tabla 2.8- Tipos de traps. [3]

- *Specific Trap Code:* Es usado para traps privados (de fabricantes), así como para precisar la información de un determinado trap genérico.
- *Time stamp:* Indica la cantidad de tiempo que ha transcurrido entre la última reinicialización de la red o agente y la consecuente generación del TRAP.

- *Variables binding:* Actúa como el campo de datos de la PDU SNMP. Es una serie de nombres de variables con sus valores correspondientes (codificados en ASN.1) indicando las causas específicas por las cuales se generó la alerta. [7]

2.6 Formato de mensajes SNMPv3

El formato de un mensaje SNMPv3 es muy distinto al de las dos versiones anteriores, ya que viene integrado de diferentes componentes de seguridad para la autenticación y cifrado de mensajes.

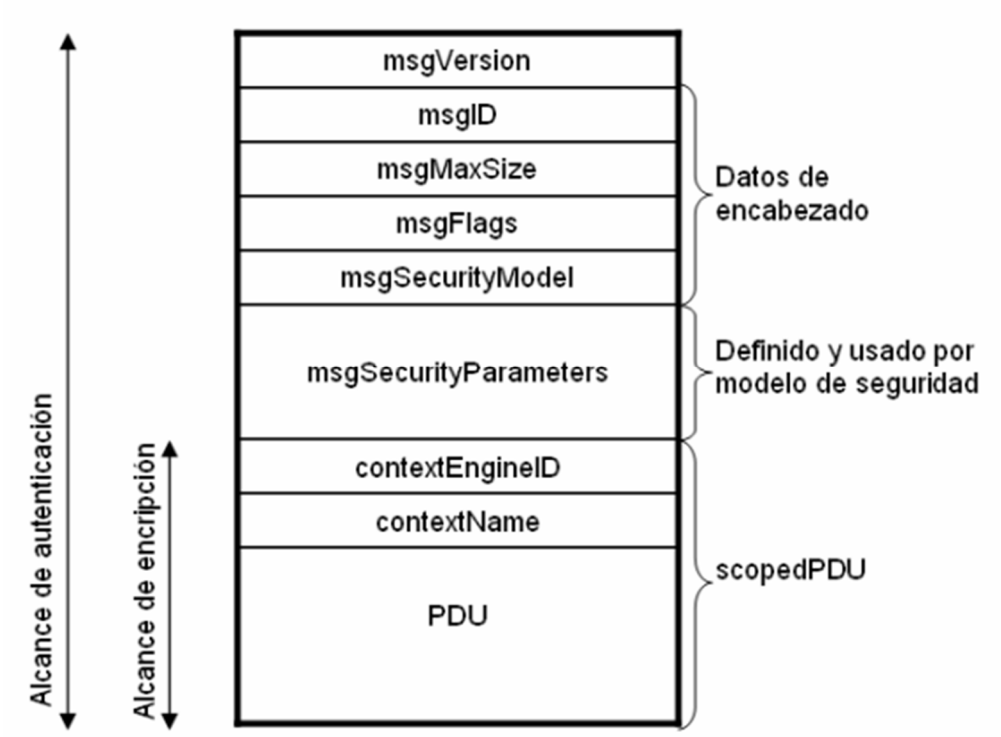


Figura 2.12 – FORMATO DE MENSAJE SNMPv3. [23]

- *msgVersion:* Indica la versión de SNMP, aunque su valor está por default en la v3.
- *msgID:* Número usado como identificador único de 32 bits que sirve para relacionar mensajes de solicitud y de respuesta.
- *msgMaxSize:* Número de 32 bits que indica el tamaño máximo en bytes que puede aceptar el emisor del mensaje.
- *msgFlags :* Cadena de 8 bits, que indica el nivel de seguridad usando solo 3 bits (menos significativos). Esta cadena contiene 3 banderas:
  - reportableFlag: El valor 1 en este subcampo indica que el receptor del mensaje debe enviar de vuelta un acuse de recibo.
  - privFlag: El valor 1 indica que se debe cifrar el mensaje.
  - authFlag: El valor 1 indica que se debe aplicar autenticación al mensaje.
- *msgSecurityModel:* Indica el modelo de seguridad empleado por el emisor del mensaje. Esto le permite saber al receptor que modelo debe usar: SNMPv1 (1), SNMPv2 (2) y USM de SNMPv3 (3).
- *msgSecurityParameters:* Cadena que contiene parámetros generados por el subsistema de seguridad en la entidad remitente y procesados por la entidad receptora.
- *contextEngineID:* Identifica de manera única una entidad SNMP. Para mensajes provenientes, determina a que aplicación el *scopedPDU* va a ser enviado.

- *contextName:* Identifica de manera única un contexto particular dentro del contexto del motor SNMP.
- *Datos:* un PDU que debe ser de tipo SNMPv2.[24]

El RFC2274 define una estructura llamada u*smSecurityParameters*, que especifica el formato interno del campo *msgSecurityParameters* en un mensaje SNMPv3, como se muestra en la figura 2.5.

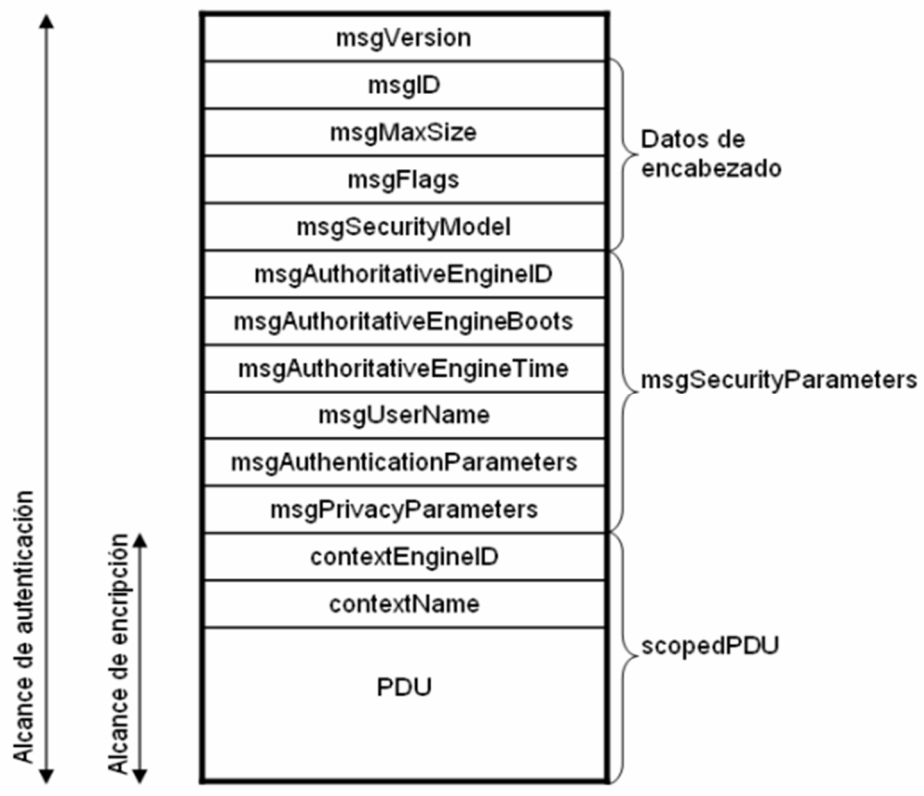


Figura 2.13 – Formato del mensaje SNMPv3 con el campo *msgSecurityParameters*. [23]

A continuación mostramos sus parámetros:

- *msgAuthoritativeEngineID*: Es el identificador *SNMPEngineID* de un motor SNMP autoritativo involucrado en el intercambio de mensajes.
- *msgAuthoritativeEngineBoots:* Indica el número de ocasiones que un motor SNMP reinició desde su configuración original.
- *msgAutoritativeEngineTime:* Indica el tiempo en segundos desde que un motor SNMP incrementó *snmpEngineBoots* por última vez.
- *msgUserName:* Indica el usuario a quien se le está intercambiando el mensaje.
- *msgAuthenticationParameters:* es nulo si no se usa privacidad, de lo contrario, es un código de autenticación de un mensaje HMAC.
- *msgPrivacyParameters:* es nulo si no se usa privacidad, de lo contrario, es un valor usado para formar el vector de inicialización en los algoritmos de cifrado. [23],[24]

2.7 Funcionamiento de los traps

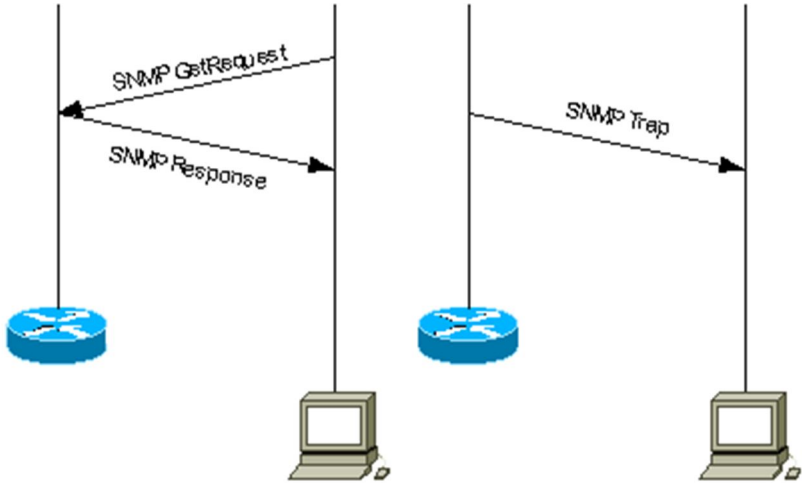


Figura 2.14 – Ejemplo de los traps de SNMP

En la figura 2.14, la configuración de la izquierda muestra un NMS solicitando información (get-request) y obteniendo su respectiva respuesta (get-response). La configuración de la derecha muestra a un agente enviando un trap asíncrono no solicitado hacia el NMS. [9]

2.7.1 Utilizando los Traps

En una empresa, se necesitan monitorear muchos dispositivos, donde cada dispositivo tiene una gran cantidad de objetos, y para un administrador de red sería muy complicado obtener o solicitar información desde cada objeto en cada dispositivo. La solución es que los agentes de cada MD notifiquen a los

NMS’s enviando alertas traps sobre eventos importantes ocurridos en cada uno de sus respectivos dispositivos y objetos. Después de que el administrador reciba las alertas, puede desplegarlas y guardarlas directamente desde los dispositivos y tomar las medidas respectivas dependiendo del tipo de evento efectuado.

El envío de traps dirigidos puede produce una marcada mejora en la economía de la red incluyendo los recursos del agente por la eliminación de solicitudes innecesarias del protocolo SNMP. Sin embargo, no es posible eliminar del todo las solicitudes SNMP ya que éstas permiten descubrir la red al inicio de las operaciones de monitoreo y detectar cambios en la topología ya que un agente MD no puede enviar un trap si el dispositivo ha tenido una interrupción catastrófica, por ejemplo, un corte de energía repentino. [9]

Los traps genéricos están descritos en la sección 2.5.3 en el campo Generic Trap Type.

2.7.2 Traps en la versión 3

La versión de los PDU es determinada de una manera dependiente de la implementación del protocolo SNMP en la red; para SNMPv3 la versión del PDU sería la misma que una de la versión 2 [15]. Además, según la RFC3584, el texto en su contenido sobre los tipos de PDU y operaciones del protocolo SNMPv2, aplican también para SNMPv3.

Por lo tanto, SNMPv3 se basa en las versiones anteriores, añadiendo autenticación y seguridad en el envío de los objetos de los dispositivos administrados; además de permitir el envío de grandes bloques de parámetros y traps para la mayoría de ellos. [1]

Internamente, un PDU trap de la versión 3, tiene el mismo esquema de un PDU notification o trap v2 simplemente que en su encabezado se han agregado todos los campos que permiten la autenticación y cifrado del mensaje. Como se vio en secciones anteriores, los PDU SNMPv2 con funcionamiento análogo al trap, como lo son el inform y report, están

presentes en la versión 3 con los beneficios extra de seguridad que ofrece esta última versión.

De forma general, para que una entidad pueda recibir o enviar traps en la versión 3 de SNMP, debe tener creados los usuarios con sus respectivos permisos de lectura o escritura en sus archivos de configuración, recordar que SNMPv3 funciona con el esquema de usuarios en vez de comunidades. Además es recomendable que la entidad que recibirá los traps conozca el identificador de motor SNMP de la aplicación de la entidad remota que enviará el trap, si no lo conoce, lo descubre para posteriormente enviarlo. Estas configuraciones se detallarán en un capítulo posterior.

CAPÍTULO 3

3 IMPLEMENTACIÓN DE SNMPv3 EN LA RED Y PRUEBAS

Este capítulo en su primera mitad explica cómo el servicio SNMPv3 se instala y configura en los principales agentes que se encuentran en una red corporativa. En la segunda mitad, ofrece pruebas e imágenes que ayudan a ver en forma sencilla cómo el protocolo funciona en la práctica además que son de gran apoyo para asimilar los conceptos del capítulo previos o nuevos conceptos que se explican en el desarrollo del presente capítulo.

3.1 Descripción general de SNMP en redes LAN

Toda red corporativa maneja su esquema de direccionamiento interno con direcciones privadas de cualquiera de las clases existentes, usando además subnetting para poder hacer una eficiente distribución de las IP´s. Al borde de la red se encuentra un dispositivo de enrutamiento que permite que cada

host interno pueda salir también hacia internet, usando NAT (Traducción de Dirección de Red) para poder tener una IP pública. Finalmente en el medio de los usuarios de la LAN y el router, se encuentra algún firewall, que proteja la red de intromisiones externas.

Hay que considerar también que en cualquier compañía la red de datos se encuentra segmentada en subredes físicas y lógicas (VLAN´s) de acuerdo a la división geográfica de los departamentos, tipo y cantidad de tráfico que generan cada uno de ellos, niveles de privilegio de los diferentes usuarios para poder hacer uso de la red y tener funcionalidades como VoIP (Voz sobre IP), acceso no limitado a internet, etc. Pero además, otra razón para dividir la red lógicamente en varias VLAN, es que se destina una determinada VLAN para permitir el flujo de tráfico de administración y en ellas se sitúan los equipos que administran o son administrados.

Por lo tanto es importante que la computadora que actúe como servidor SNMP haciendo solicitudes, valiéndose en nuestro caso de los aplicativos de monitoreo como WhatsUp Gold o SNMP JManager, esté dentro de la subred

y VLAN de administración de tal forma que el tráfico SNMP fluya normalmente y no se vea obstruido por firewalls, proxys o cuestión de permisos.

En nuestro caso implementamos una pequeña red física, compuesta de dos computadoras, una con sistema operativo Windows 7 que actúa como NMS con el programa WhatsUp Gold, y otra con el sistema operativo Ubuntu 13.10 que es la computadora monitoreada; ambas conectadas a un switch Cisco. Este switch tiene conexión a un router el cual también será monitoreado. Cabe destacar que se configuraron los mismos usuarios en el host Ubuntu y en el router, usando el mejor protocolo de autenticación y de cifrado, SHA y AES respectivamente. El esquema de nuestra red se puede apreciar en la figura 3.64.

3.2 Instalación y configuración del servicio agente SNMP en equipos monitoreados.

3.2.1 Hosts Windows

3.2.1.1 Windows 7

Nos dirigimos a Equipo y seleccionamos la opción *Desinstalar o cambiar un programa*.

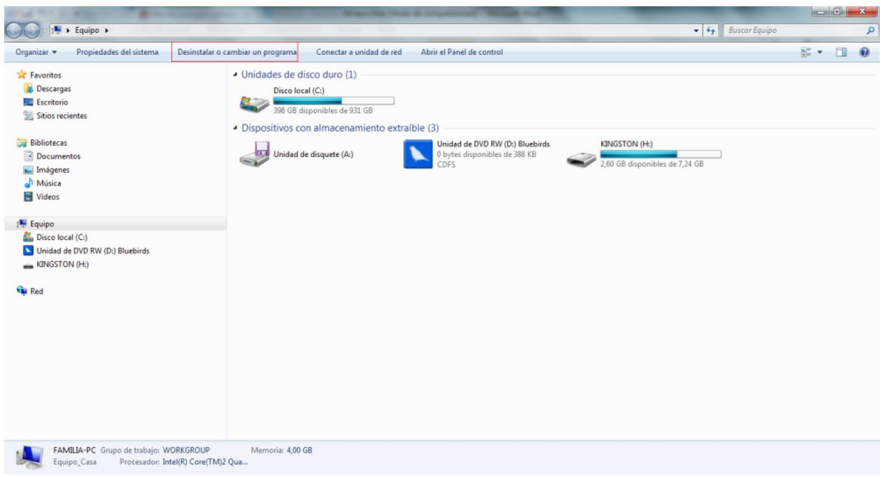


Figura 3.1- Opción para desinstalar o cambiar un programa

En la barra lateral izquierda seleccionamos la opción *Activar o desactivar las características de Windows* y esperamos que se abra una ventana que contiene diferentes tipos de servicios para ser instalados.

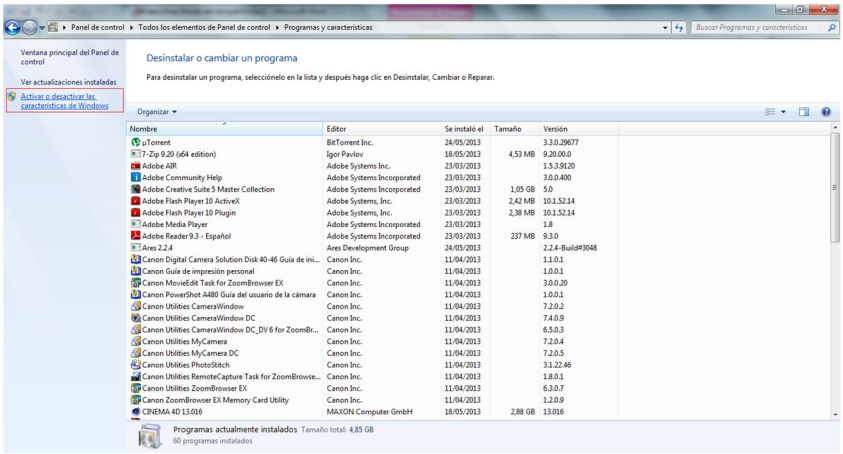


Figura 3.2- Activar o desactivar las características de Windows

Buscamos la casilla *Protocolo Simple de Administración de Red*, la activamos y presionamos aceptar. Una vez hecho esto el proceso de instalación del servicio SNMP empezará y solo hay que esperar que terminar para empezar a configurar el agente.

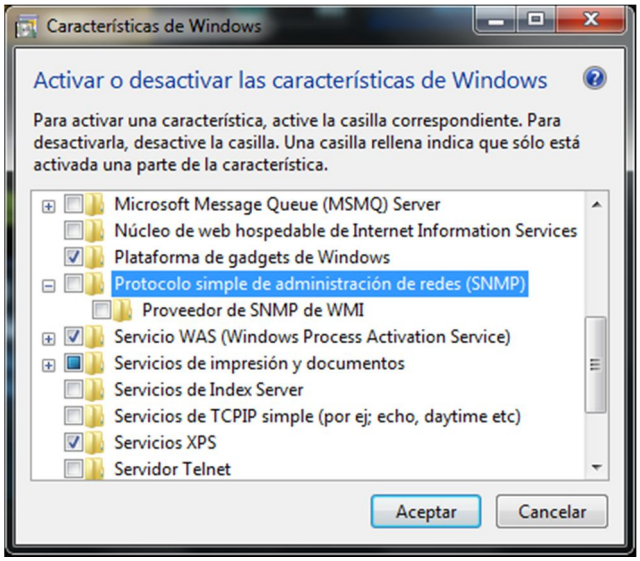


Figura 3.3- Casilla protocolo simple de administración de red

Verificamos que el servicio SNMP esté activo en la estación de trabajo en la ventana de servicios de Windows, que se encuentra en la ruta Inicio > Panel de Control > Sistema y Seguridad > Herramientas administrativas > Servicios.

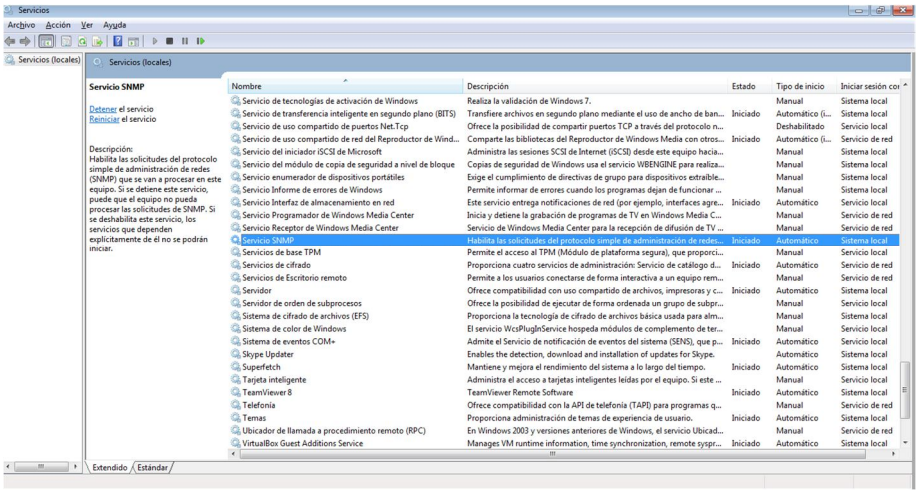


Figura 3.4- Servicio SNMP activo

De la misma forma verificamos si el servicio de traps SNMP se está ejecutando. Como se puede apreciar en la figura 3.5, no está ejecutándose ya que su inicio es de forma manual, y es preferible dejarlo así ya que la aplicación WhatsUp Gold es la que se encargará de recibir los traps.

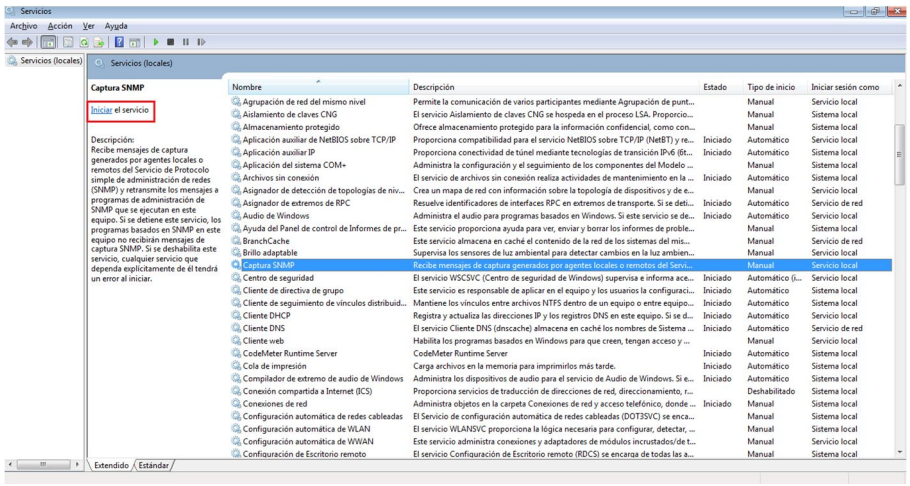


Figura 3.5- Servicio de traps SNMP inactivo

Necesitamos configurar el agente SNMP que reside en el host, para esto hacemos clic derecho sobre *Servicio SNMP* y seleccionamos *Propiedades*.

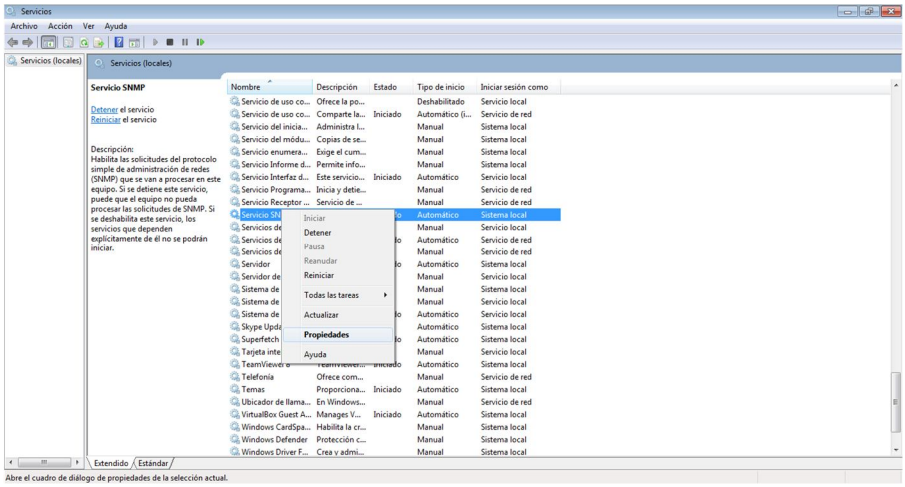


Figura 3.6- Configuración de agente SNMP

A continuación se listan las configuraciones principales para que el agente funcione correctamente y el equipo pueda ser consultado y notificar cualquier problema a las estaciones de monitoreo.

En la pestaña *General* aparece lo siguiente:

Descripción del servicio.- Breve reseña de las funciones del protocolo.

Tipo de inicio del servicio.- Es recomendable que esté configurado automático, debido a que los NMS´s continuamente hacen consultas a los equipos monitoreados y si se da el caso de equipos que no pasen prendidos continuamente, o que se llegaran enfrentar a un apagado fortuito, al iniciarlos se debe dar inicio de forma automática al servicio SNMP para que puedan ser consultados y aparecer en la topología de la red.

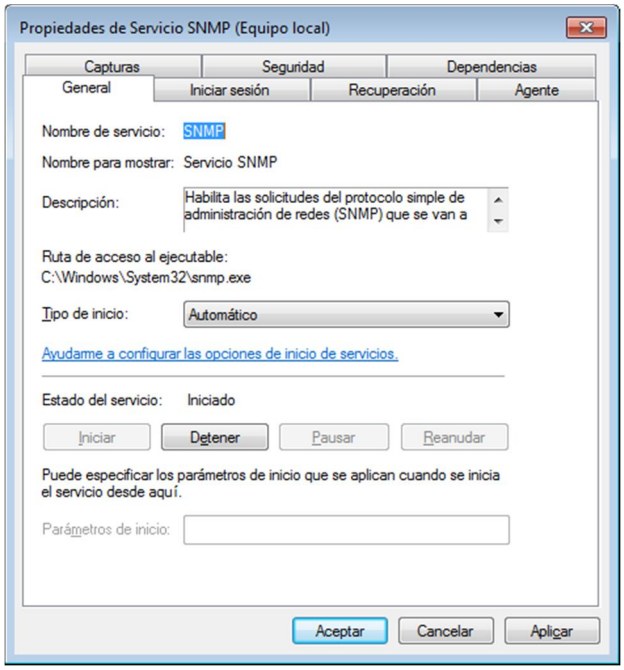


Figura 3.7- Propiedades de servicio SNMP - General

En la pestaña *Agente* aparece lo siguiente:

Contacto.- Se debe escribir datos del encargado directo del equipo.

Ubicación.- Lugar donde se encuentra el equipo, el cual puede ser tan general como la ciudad hasta lo más específico como edificio, piso, oficina, rack, etc.

Servicio.- Físico, Aplicaciones, Vínculo de datos y subred, Internet, De un extremo a otro. Recomendable seleccionar todas las casillas para que las consultas desde y hacia el equipo Windows tengan la mayor funcionalidad y alcance posible.

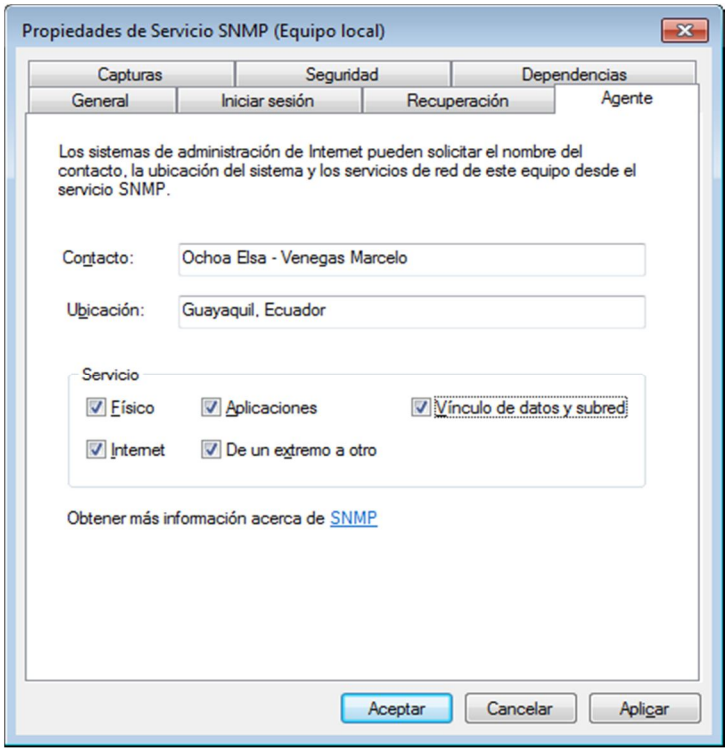


Figura 3.8- Propiedades de servicio SNMP - Agente

En la pestaña *Capturas* aparece lo siguiente:

Nombre de la comunidad.- Se debe ingresar el nombre de la comunidad en la cual se encuentra el equipo al que se le enviarán los traps. Luego se presiona el botón *Agregar a la lista*.

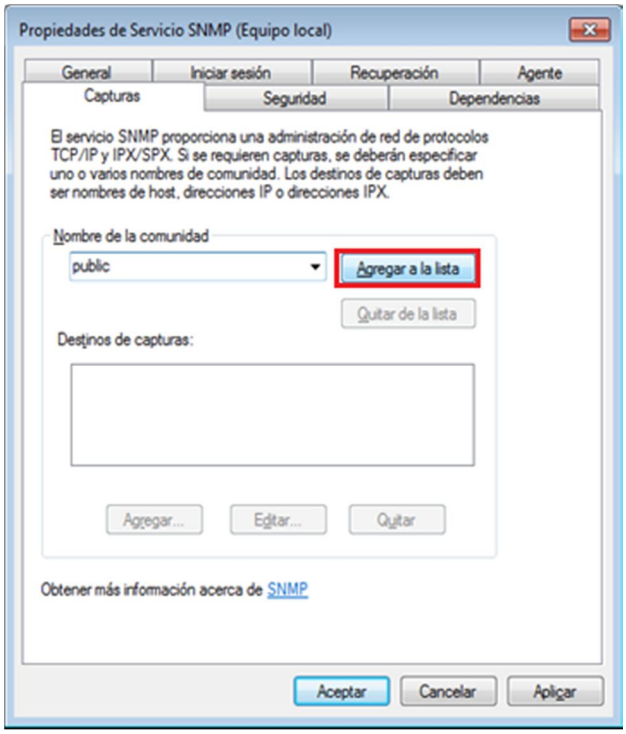


Figura 3.9- Propiedades de servicio SNMP - Capturas

Es recomendable que en una red LAN las comunidades por defecto public y private se deshabiliten y se usen otras alternas para evitar ataques valiéndose de ellas; al menos si en la red se usan las versiones 1 ó 2c de SNMP.

Destino de capturas.- Dirección IP del equipo al que se le enviarán los traps. Puede ser la dirección de loopback, con lo cual se enviarán a la misma máquina; u otra IP dentro de la misma red, la del equipo que actúa como NMS dentro de la red LAN.

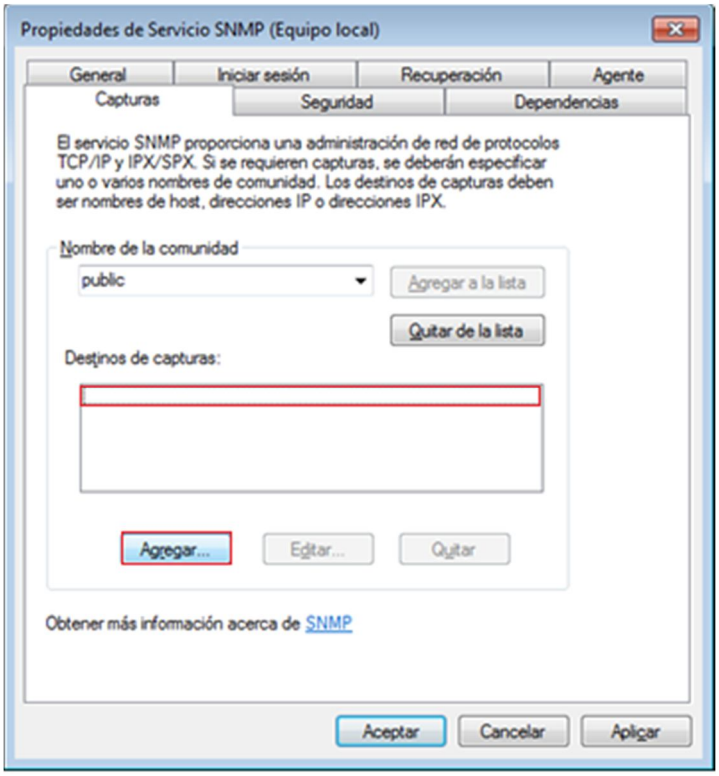


Figura 3.10- Propiedades de servicio SNMP - Destino de capturas

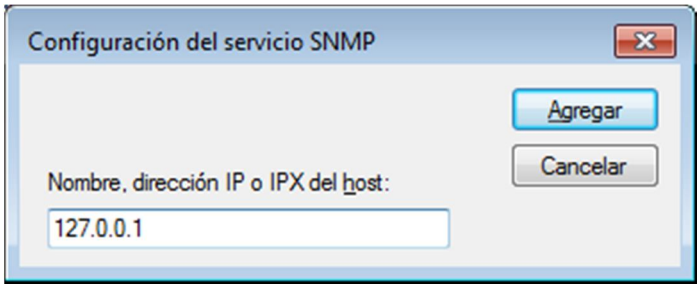


Figura 3.11- Dirección IP del NMS

Configuración final de las capturas o traps.

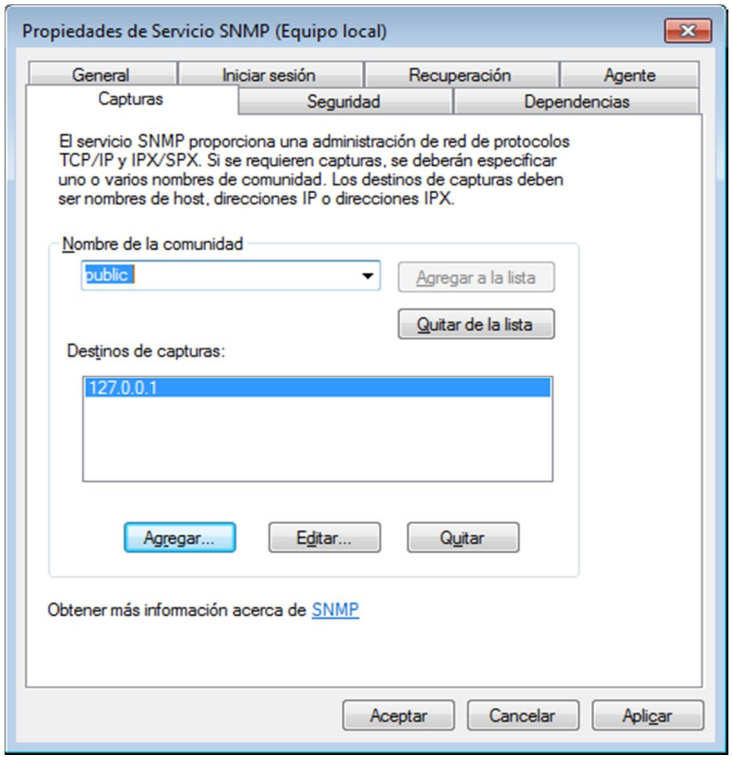


Figura 3.12- Dirección de destino de captura

En la pestaña *Seguridad* aparece lo siguiente:

Enviar captura de autenticación.- Cuando un agente SNMP externo logra comunicarse con el agente local, se envía un trap al agente remoto informándolo.

Nombres de comunidad aceptadas.- Aquí se indican qué comunidades y qué nivel de seguridad tienen, si sólo lectura o lectura/escritura; podrán comunicarse con el host Windows vía SNMP.

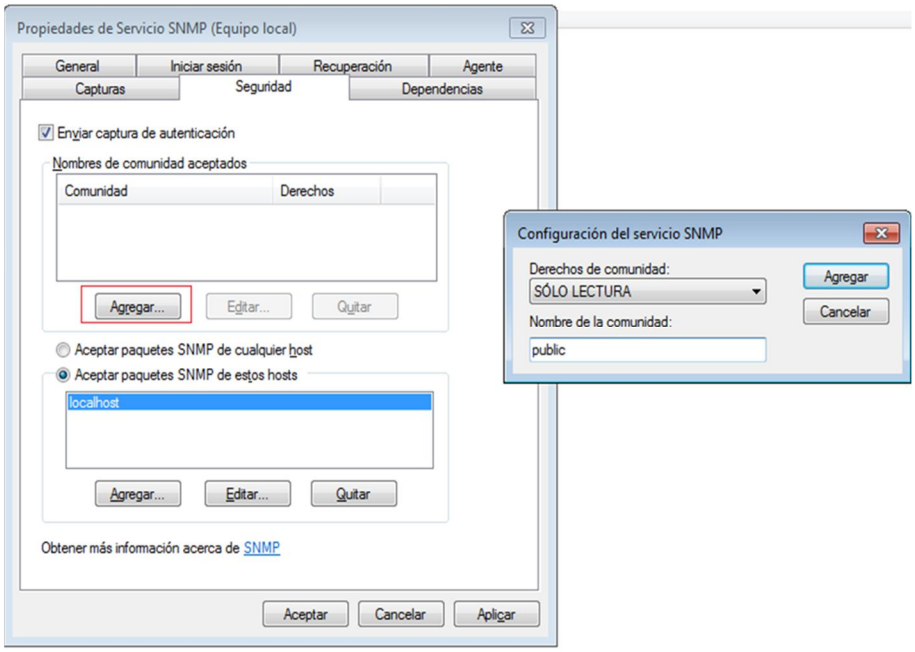


Figura 3.13- Comunidad para envío de capturas

Aceptar paquetes SNMP de cualquier o de algunos host.- Si se quiere restringir que solamente el NMS envíe consultas o traps al host Windows, entonces hay que agregar su IP en la opción de aceptación de paquetes de algunos host; de lo contrario, se puede elegir la opción de aceptar paquetes de cualquier host.

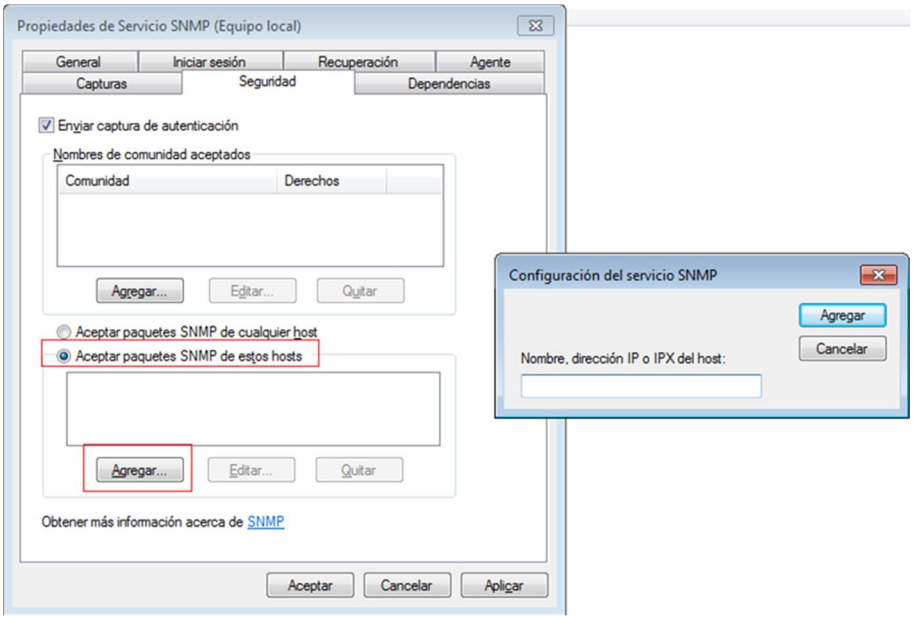


Figura 3.14- Aceptar paquetes SNMP de un host específico

Esa fue la configuración básica para un agente SNMP Windows, que puede funcionar para las versiones 1 y 2c; pero no para la versión 3, debido a que no ha sido incorporada en el sistema operativo como tal esta funcionalidad. Hay programas externos que pueden ser instalados en Windows para que ejecuten un agente SNMP en sus tres versiones, como por ejemplo, Net-SNMP. Más información sobre esta herramienta e instalación en http://www.net-snmp.org/

3.2.2 Hosts linux

3.2.2.1 Sistema operativo Ubuntu

Se recomienda actualizar los paquetes de la distribución actual del sistema Ubuntu o inclusive actualizar la distribución, para aquello se abre la terminal, y ejecutamos el comando *sudo apt-get upgrade*. Se presiona enter y se espera hasta que el proceso termine. La versión de Ubuntu en la que se instaló SNMP es la última hasta el momento, Saucy Salamander 13.10.

Se instala snmp y snmpd. Se digitan los comandos con sudo para ejecutarlos como administrador ya que la mayoría no funcionan si no se los ejecuta en ese modo.

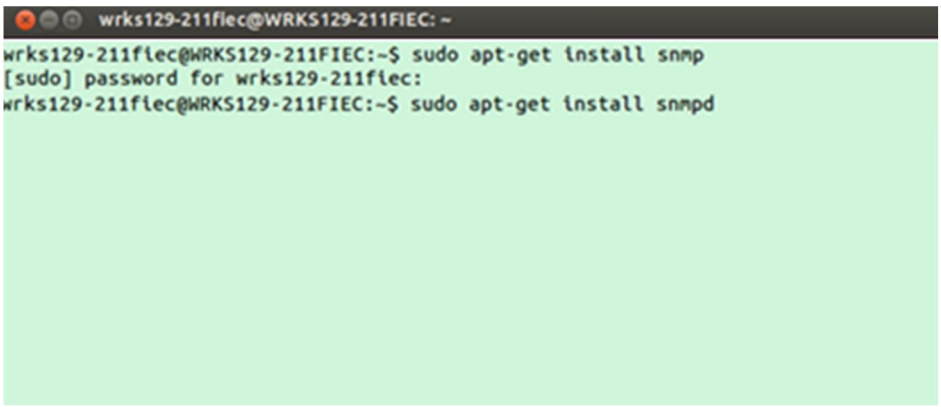


Figura 3.15- Comandos de instalación

Cuando ejecutamos los comandos anteriores se instala la aplicación Net-SNMP, la última versión hasta el momento es la 5.7.2. Hay dos *daemons* (procesos permanentes del sistema y transparentes al usuario) que permiten el correcto funcionamiento de las operaciones de administración en el sistema. El primero es *snmpd*, que en general, recibe, analiza y responde a las solicitudes SNMP entrantes. Luego, está el proceso *snmptrapd* que recibe y registra los traps, notificaciones e informes. Para este proyecto, la PC Ubuntu (192.168.56.101) es un agente monitoreado y la PC Windows 7 (192.168.56.3) es el NMS de la red, en donde se encuentra instalado WhatsUp Gold; de tal forma que desde esta última se hicieron consultas hacia los demás dispositivos y se recibieron las traps que generó la red. Por lo tanto solamente se hicieron configuraciones en el archivo de sistema que controla al servicio snmpd.

Hay dos formas de configurar el servicio de SNMP, por línea de comandos o configurando los archivos de sistema. Realizaremos la configuración por el segundo método, ya que es más directo y sencillo, obteniéndose los mismos resultados que si se hicieran por línea de comandos.

Antes de editar cualquier archivo de sistema relacionado a snmp se debe desactivar momentáneamente el servicio snmp, al final de todos los cambios, se lo inicia de nuevo. Se hace de la siguiente forma:

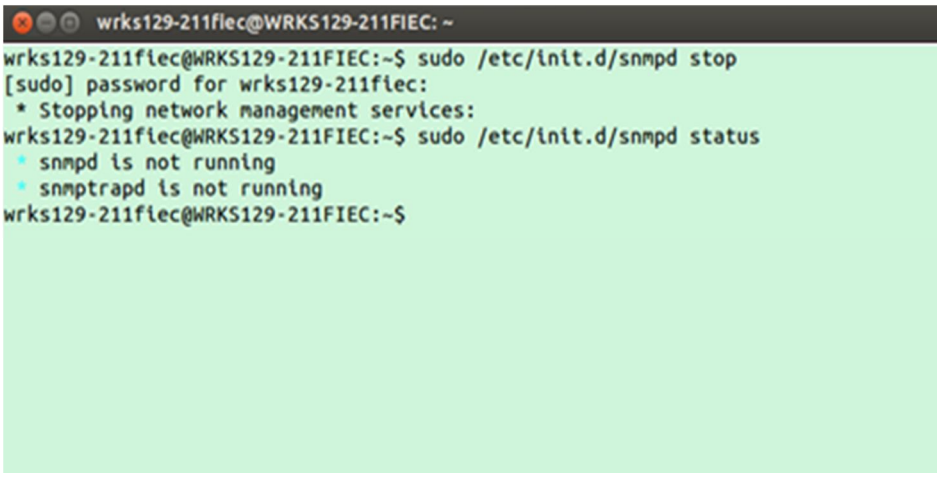


Figura 3.16- Desactivación del agente SNMP

Como se pudo apreciar en la figura 3.16, el primer comando sirve para detener los servicios de administración relacionados a snmp; y el segundo para ver el estado de los servicios, tanto el servicio del agente snmp, como el del generador y receptor de traps, que ya se encuentran detenidos.

En el terminal digitamos el comando sudo nautilus (Fig. 3.17) para poder explorar los directorios de la PC con permisos de administrador y así al abrir los archivos que queramos modificar, luego de editarlos podremos guardar

los cambios; evitando abrir por terminal cada uno de los archivos con permisos de administrador usando el editor de texto (gedit). Cuando se digita el comando se abre por defecto el directorio Carpeta Personal (Fig. 3.18).

Figura 3.17- Comando nautilus como administrador

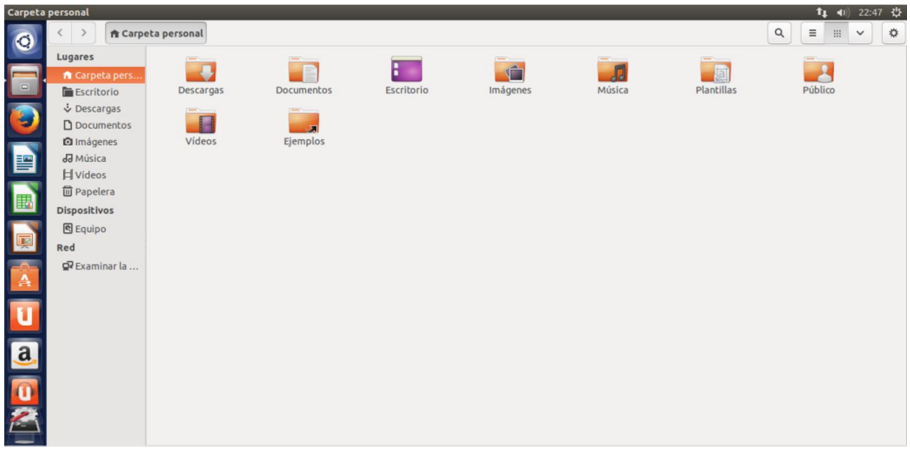


Figura 3.18- Carpeta personal

El primer archivo que se debe configurar es el que se encuentra en la ruta /etc/default/snmpd. Del lado derecho de la ventana Carpeta personal que se abrió al ejecutar nautilus, en la sección Dispositivos se hace clic en Equipo

para dirigirnos al directorio raíz. Una vez dentro del directorio raíz, ingresamos a la carpeta etc.

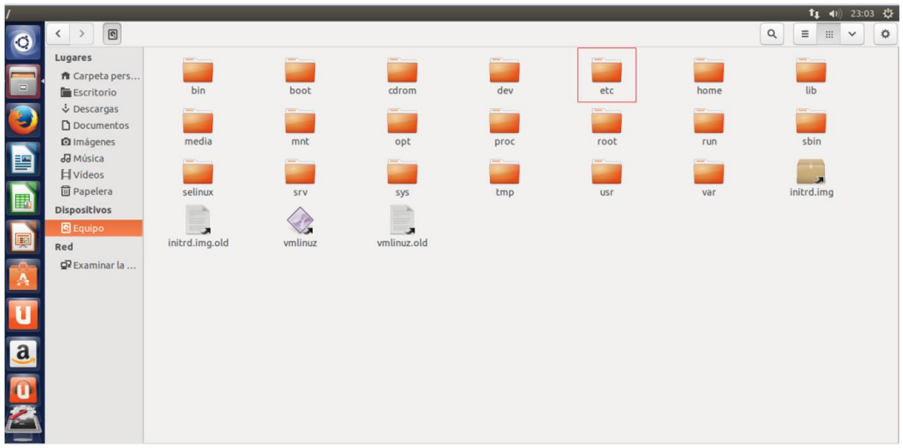


Figura 3.19- Carpetas de Sistema - etc

Dentro de etc nos dirigimos al directorio default y buscamos el archivo snmpd. Se lo abre para edición dando doble clic.

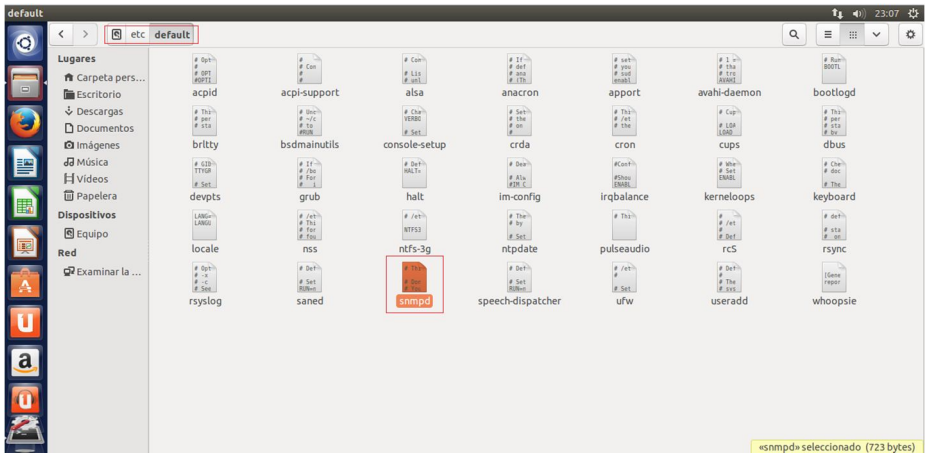


Figura 3.20- Carpetas del Sistema - etc/default

En este archivo se verifica que los servicios de snmp y snmptrap se encuentren operativos y si las operaciones que realizan tienen un alcance local a la PC Ubuntu o con cualquier otra PC. La configuración por defecto del archivo se muestra en la figura 3.21 y se analizan los aspectos más importantes para el buen funcionamiento de las aplicaciones. El servicio SNMP se encuentra habilitado ya que la línea SNMPDRUN está configurada con la opción sí, en cambio la recepción de traps está deshabilitada en la línea TRAPRUN con la opción no. Las líneas de opciones de snmp (SNMPOPTS) y traps (TRAPOPTS) indican qué alcance tienen las operaciones de administración sobre este agente, en versiones previas de Net-SNMP, al lado de la palabra pid se encontraba la IP local 127.0.0.1, lo que quería decir que cualquier operación de consulta podía hacerse al propio dispositivo pero nadie externo podía consultar por variables internas. Si se quiere que cierta PC o una red pueda hacer consultas a la PC Ubuntu, debemos escribir la dirección respectiva luego del pid y antes de la comilla simple; o si por el contrario queremos que cualquier host consulte al agente, se debe dejar sin IP como vemos que se encuentra en la configuración.

Si se deseara poder recibir traps en este agente, se cambia al valor ‘yes’ en TRAPRUN, pero como se mencionó previamente la PC NMS es la que

ejecuta WhatsUp Gold y a ella se enviaron todos los traps. De cualquier manera, si se realizaran cambios a esta configuración, no olvidar guardarlos.

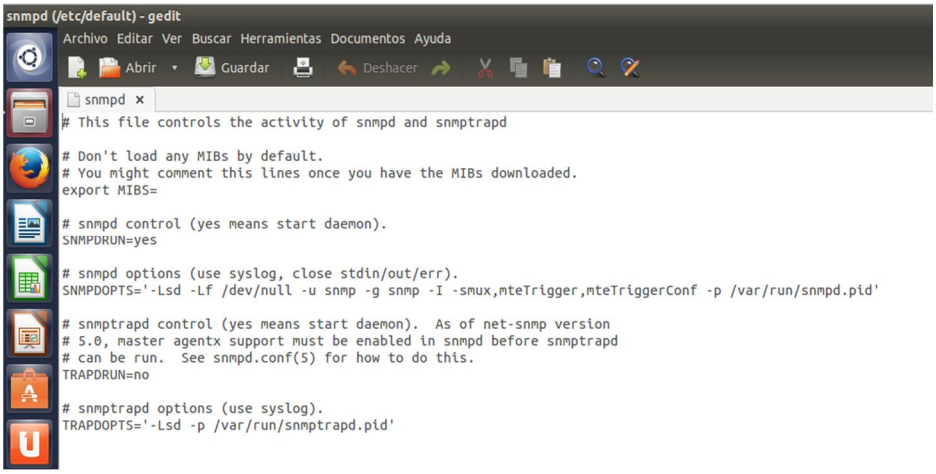


Figura 3.21- Carpeta del Sistema - etc/default/snmpd

Luego configuramos el archivo del directorio /etc/snmp/snmpd.conf, el cual contiene la información más importante para el funcionamiento del agente, incluyendo comunidades, usuarios, control de acceso, información del sistema, etc. Exploramos los directorios de la misma forma como se hizo para abrir el primer archivo de sistema, una vez localizado el archivo snmpd.conf se hace doble clic para abrirlo y modificarlo. La configuración se puede ver en el anexo 1, en los siguientes párrafos se la explica con detalle y qué efecto tienen en el funcionamiento del agente.

En el archivo se agregaron en la sección Access Control las directivas rouser y rwuser para definir usuarios SNMPv3 con diferentes niveles de seguridad, pero estrictamente hablando, a este punto todavía dichos usuarios no se encuentran activos ni se han originados sus entradas respectivas en las tablas usm o vacm, por lo que consultas hacia este agente con estos usuarios serán infructuosas. Para que los usuarios efectivamente sean reconocidos por el agente es necesario crearlos en el archivo persistente de snmpd, en el directorio /var/lib/snmp/snmpd.conf. El archivo persistente se usa para modificaciones de la información durante la ejecución del agente SNMP, las cuales necesitan ser grabadas entre una ejecución del agente y otra. Sólo se tiene que modificar para agregar las directivas de usuarios createUser. [27]

Es importante asegurarse que el daemon snmpd esté detenido antes de configurar el archivo persistente y guardar los cambios. Una vez dentro del directorio y abierto el archivo, se ingresan tantas directivas createUser como usuarios se hayan definido en las configuraciones anteriores, junto con los protocolos y contraseñas de autenticación y/o cifrado dependiendo del nivel de seguridad de cada usuario definido.

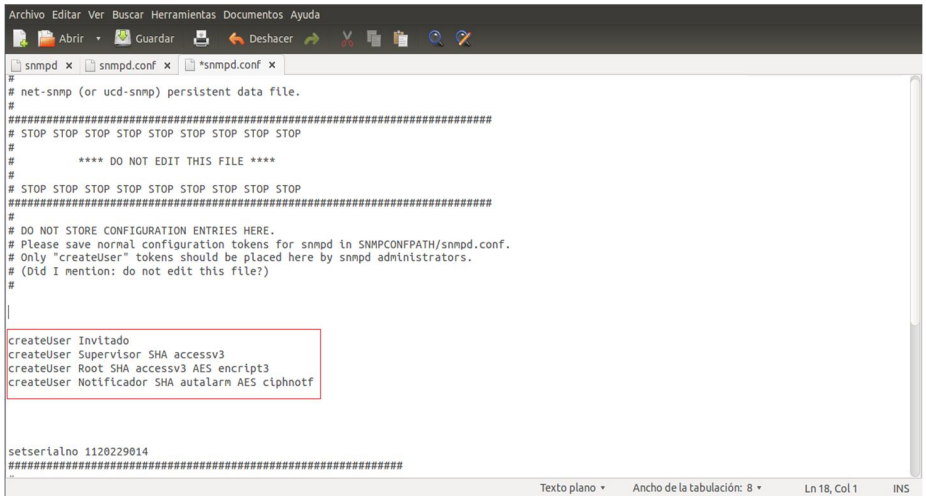


Figura 3.22- Archivo persistente snmpd.conf - directivas createuser

Se guardan los cambios y se reanudan los procesos snmp. El archivo persistente graba los cambios en el motor SNMP durante el reinicio de los servicios y remueve las directivas ‘createUser’ legibles al usuario y las reemplaza con unas entradas equivalentes ‘usmUser’. Toda la información que fue ingresada continúa, pero de una forma que es entendida solamente por el sistema, de hecho las contraseñas de autenticación y cifrado se transformaron en unas llaves (authKey y privKey) que son necesarias para los procesos de autenticación y cifrado de los mensajes enviados en representación de los usuarios de este motor SNMP. Si alguien intentara robar este archivo de configuración no podría usar la información de la entrada ‘usmUser’ para acceder a cualquiera de los otros agentes de la red (inclusive si tuvieran los mismos usuarios y contraseñas), ya que son llaves

localizadas a este motor SNMP en particular, como se ampliará en la sección 3.7.1.

Figura 3.23- Archivo persistente snmpd.conf - directivas usmUser

Hay 4 entradas usmUser correspondientes a cada usuario configurado en el archivo. A manera de ejemplo se explican los principales parámetros del usuario Root:

- 0x80001f88804fdfdc455ce79c52 corresponde al engineID del motor SNMP.
- “Root”. Nombre del usuario, es posible que en versiones anteriores de Net-SNMP aparezca en formato hexadecimal. Para este usuario aparecería 0x526f6f7400.
- Al no haber contextos definidos, aparece NULL en el siguiente parámetro.

- 1.3.6.1.6.3.10.1.1.3 corresponde al OID del protocolo de autenticación usmHMACSHAAuthProtocol.
- 0xdde0ae204bc4a522ebcd9c7d9085bb02acbc84ad es la llave localizada SHA.
- 1.3.6.1.6.3.10.1.2.4 corresponde al OID del protocolo de cifrado usmAesCfb128Protocol.
- 0x5fa7f25e3a17d8f96e98536ea0c3eee7 es la llave localizada AES.

Al final del archivo persistente se puede apreciar que el motor SNMP ha reiniciado 2 veces, indicado en el parámetro engineBoots; además del engineID para este motor específico.

Con las configuraciones que se hicieron, el servicio snmp debe funcionar sin problemas con cualquiera de sus versiones, se lo verificará haciendo solicitudes hacia el propio agente usando la comunidad y usuarios configurados. Para hacerlas, hay que reanudar el *daemon* snmpd (recordar que se lo desactivó a fin de configurar los archivos de sistema).

Figura 3.24- Activación del servicio SNMP

Para solicitar algún objeto al agente se digita el comando snmpget. Para hacerlo en las versiones 1 ó 2c, el formato es el siguiente:

comando snmp + versión + comunidad + host + OID a consultar

El formato para una solicitud usando la versión 3 es el siguiente:

comando snmp + versión + usuario + nivel de seguridad + protocolo y password autenticación + protocolo y password privacidad + host + OID a consultar

Para que las cadenas de caracteres sean visibles en formato ascii en vez de hexadecimal, es necesario especificar la bandera –Oa en cada comando.

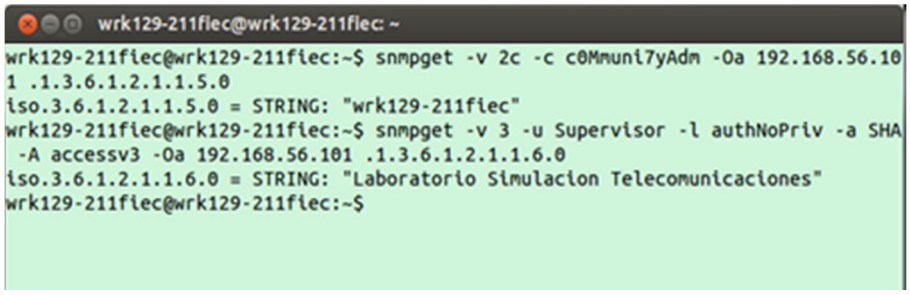


Figura 3.25- Consultas SNMPv2 y SNMPv3 al agente Ubuntu

Como se aprecia, ambos objetos fueron recuperados con éxito (sysName para la comunidad y sysLocation para el usuario). La solicitud en versión 3 se la hizo con el usuario Supervisor de nivel de seguridad authNoPriv, por lo que solamente fue necesario especificar el protocolo y password de autenticación y no de privacidad.

Aparentemente, no hubo diferencia alguna en las solicitudes independientemente si fueron hechas con comunidades o usuarios, pero en la sección 3.10 se analizará con detalle que las diferencias están en los mensajes generados y la posibilidad de ver o no los contenidos.

3.2.3 Equipos cisco

3.2.3.1 Configuraciones básicas del agente

Primero ingresamos al modo privilegiado para poder realizar configuraciones por medio de los comandos *enable* y luego *configure terminal*.

Para configurar el agente SNMPv3 en el router hay que crear usuarios, vistas y grupos. Una vista es un OID, cada OID tiene una rama de más OIDs (ver sección 2.3.4). El comando principal para configurar las vistas, grupos, usuarios, etc. es el comando snmp-server.

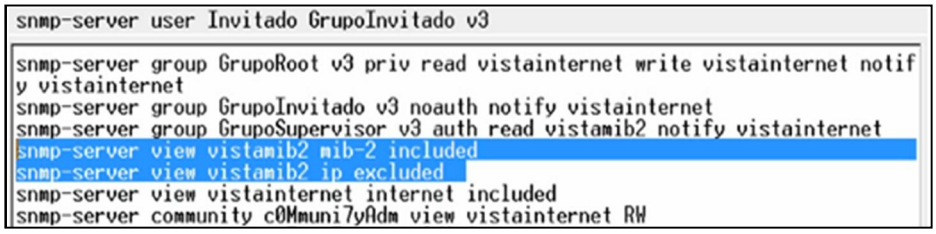


Figura 3.26- Configuración agente SNMPv3 en router

Para nuestro proyecto, hemos creado 2 vistas, de la siguiente manera:

```
snmp-server  view  vistamib2  mib-2  included

snmp-server  view  vistamib2  ip  excluded

snmp-server  view  vistainternet  internet  included
```

Después del comando *snmp-server*, está el comando *view* que sirve para crear una nueva vista, las palabras vistamib2 y vistainternet equivalen al nombre de las 2 vistas que creamos, es recomendable que los nombres estén relacionados con el OID del árbol de la vista que estamos creando. Como es de suponer mib-2, ip e internet, equivalen al OID de cada vista. En los equipos Cisco se escribe el nombre del OID en su notación numérica, pero también se puede escribir directamente el identificador de objeto. Las palabras *included* y *excluded* es para incluir o excluir un OID en la vista especificada (ver sección 3.8.3), por lo cual la primera y la segunda línea de comandos equivalen a una misma vista.

Para crear los grupos lo hicimos de la siguiente manera:

```
snmp-server group GrupoInvitado v3 noauth notify vistainternet
snmp-server group GrupoSupervisor v3 auth read vistamib2
notify vistainternet
snmp-server group GrupoRoot v3 priv read vistainternet write
vistainternet notify vistainternet
```

Donde *group* es el comando para crear un grupo. GrupoInvitado, GrupoSupervisor y GrupoRoot equivalen al nombre de cada grupo, luego se establece la versión en la que el grupo está siendo creado, en nuestro caso usamos la v3. A continuación escribimos el comando que especifica el nivel de seguridad al cual pertenece el grupo. Como son 3 tipos de niveles de seguridad, tenemos que crear 3 usuarios para cada nivel, ahora, cada usuario debe tener su propio grupo, al cual se le asignará una vista. Por esto hemos creado 3 grupos distintos, uno para cada usuario con su respectivo nivel de seguridad (noauth, auth y priv). A cada grupo se le debe de asignar una vista. La vista asignada puede ser la misma para cada grupo, o distinta, eso depende de los privilegios que queramos asignarle a cada usuario al cual va a pertenecer el grupo. Por lo tanto, cada uno de nuestros grupos tiene distintas restricciones. El GrupoInvitado como no tiene privilegios de autenticación y por consiguiente tampoco de privacidad, solamente se le ha asignado una vista de notificación por medio del comando notify. El GrupoSupervisor tiene privilegios de sólo autenticación, por lo cual, se le ha asignado una vista de lectura por medio del comando read, y una vista de notificación (notify). El GrupoRoot tiene privilegios de autenticación y de cifrado, por lo tanto, se le ha asignado 3 vistas, una para lectura (read), una para escritura (write) y otra para notificación (notify), estas 3 vistas asignadas corresponden a la misma vistainternet, pero no necesariamente debe ser la misma vista, se puede elegir una vista distinta para cada restricción, ya sea

de lectura, escritura o notificación. Las vistas asignadas corresponden a las 2 vistas creadas anteriormente.

Para crear los usuarios, lo hicimos de la siguiente manera:

```
snmp-server user Invitado GrupoInvitado v3
snmp-server user Supervisor GrupoSupervisor v3 sha accessv3
snmp-server user Root GrupoRoot v3 sha accessv3 aes128 encript3
```

Donde *user* es el comando para crear un usuario. La palabra siguiente del comando *user*, pertenece al nombre del usuario seguido del grupo creado para ese respectivo usuario. Después de haber asignado el grupo respectivo para ese usuario se debe especificar la versión del protocolo SNMP, que en este caso es la v3. Como mencionamos antes, hemos creado 3 usuarios con el objetivo de probar cada tipo de nivel de seguridad. Por lo tanto, cada usuario creado tiene diferentes restricciones. El usuario Invitado sólo tiene asignado su respectivo grupo y la versión del protocolo SNMP a usar, debido a que no tiene restricciones de autenticación, ni de privacidad. El usuario

Supervisor, tiene asignado el grupo, la versión del protocolo SNMP, pero se le ha agregado dos cosas más, que son el protocolo de autenticación y la clave de autenticación respectivamente, debido a que el usuario Supervisor tiene restricciones de autenticación. Para el último usuario que es Root, se le configura los mismos patrones que para el usuario de autenticación con la diferencia de que se le añade el protocolo de privacidad y a continuación su debida contraseña, para así, tener restricciones de autenticación y además privacidad (cifrado). Es recomendable que las contraseñas de autenticación y de cifrado sean distintas.

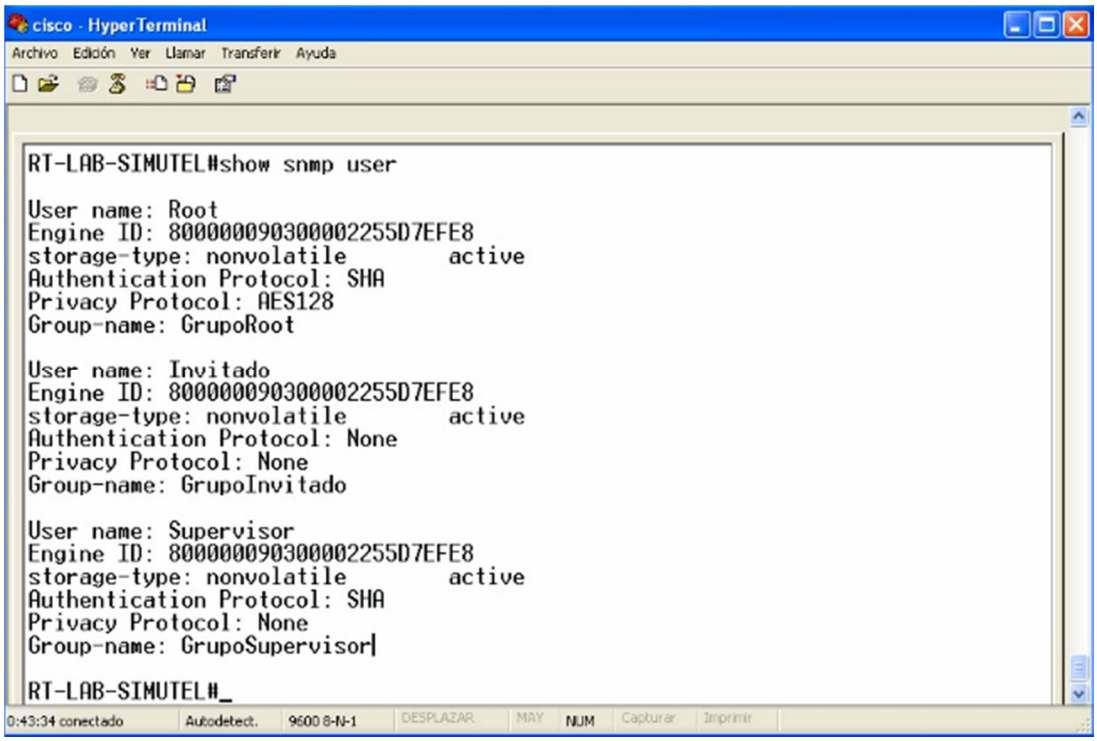


Figura 3.27- Usuarios configurados en router Cisco

Para crear una comunidad, no es necesario crear un grupo, ya que las comunidades pertenecen a la versión 2c del protocolo SNMP, donde no hay las restricciones de acceso como en la versión 3 de SNMP. Por lo tanto, se puede vincular directamente la comunidad creada, con la vista, sin necesidad de que exista un grupo de por medio. En nuestro proyecto hemos creado una comunidad de escritura con motivos de prueba, por medio de la siguiente línea de comando:

```
snmp-server community c0Mmuni7yAdm view vistainternet RW
```

Donde *community* es el comando para crear una comunidad. Luego se escribe el nombre de la comunidad, en nuestro caso es *c0Mmuni7yAdm*. A continuación escribimos el comando *view* que sirve para enlazar la comunidad con una vista. Luego del comando *view*, escribimos el nombre de la vista (vistainternet) que queremos asignarle a esa comunidad. Por último, debemos especificar si nuestra comunidad será de lectura (RO) o de escritura (RW), en nuestro caso, creamos una comunidad de escritura (RW).

Para la activación de Traps, se necesitan especificar 3 cosas:

1) La fuente: Interfaz de donde proviene la trap.
2) Tipo de trap: Definición de los traps específicos que se requiere en el router, hay muchos tipos de traps, pero no todos son necesarios.
3) Destino: Host donde los traps van dirigidos.

Líneas de comando de activación de traps:

```
snmp-server trap-source FastEthernet0/1.1
snmp-server enable traps snmp authentication linkdown linkup
coldstart warmstart
snmp-server traps envmon
snmp-server traps config-copy
snmp-server traps config
snmp-server traps entity
snmp-server traps cpu threshold
snmp-server host 192.168.56.3 version 3 priv Root
snmp-server host 192.168.56.3 version 2c c0Mmuni7yAdm
```

El comando *trap-source* es para la asignación de la fuente de la trap, en nuestro caso, la fuente de donde van a salir los traps es la interfaz FastEthernet0/1.1

Los comandos *enable trap* son para habilitar la función de los traps. Los comandos *snmp authentication* es una trap, que controla otras traps principales sobre los cambios de estado del dispositivo, y son las siguientes: linkdown, linkup, coldstart y warmstart.

Para seguir agregando traps solo se escribe el comando *traps* seguido del nombre del trap que se desea agregar.

- La trap *envmon* envía notificaciones de cambios de estado de shutdown, temperatura, ventilador, voltaje y fuente de poder.
- La *trap config-copy* envía notificaciones cuando se ha copiado algún tipo de configuración en el equipo.
- La trap *config* envía notificaciones cuando existe algún tipo de configuración en el equipo.

- La trap entity envía notificaciones cuando existe alguna modificación en la entidad MIB.
- La trap cpu *threshold* envía notificaciones cuando existe alguna violación de umbrales de CPU.

El comando *host* se escribe para especificar el destino de los traps y va seguido de la dirección IP del NMS. También se especifica la versión (versión 3) y el usuario (con su respectivo nivel de seguridad) o comunidad con que se van a recibir los traps.

```
snmp-server trap-source FastEthernet0/1.1
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart

snmp-server enable traps envmon
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps entity
snmp-server enable traps cpu threshold
snmp-server host 192.168.56.3 version 3 priv Root
snmp-server host 192.168.56.3 version 2c c0Mmuni7yAdm
```

Figura 3.28- Configuración de traps en router Cisco

3.2.3.2 Configuración NetFlow

NetFlow data es un tipo de configuración que sirve para transferir datos de tráfico de interfaces desde los dispositivos (router, switch) a la PC.

Ingresamos al modo privilegiado y allí escribimos los siguientes comandos:

```
ip flow-export  version 9
ip flow-export destination 192.168.56.3 9999
```

La primera línea de comando quiere decir que se está activando el flujo de datos con la version 9, hay 3 tipos de version, pero nosotros escogimos la última, que es la más actual. La segunda línea de comando quiere decir que el flujo de datos va a tener como destino la PC con dirección IP 192.168.56.3. El número 9999 corresponde al puerto de escucha por donde llegan los datos de Netflow. Este puerto es el que está establecido por defecto en WhatsUp Gold, pero si se desea, se lo puede modificar.

Luego ingresamos a la interfaz de donde queremos obtener el tráfico de datos, en este caso nuestra interfaz es la *fastEthernet 0/1.1,* y escribimos las siguientes líneas de comando:

```
ip flow ingress

ip flow egress

exit
```

Las palabras *ingress* y *egress* quieren decir que se va a obtener el flujo de tráfico de datos que va a entrar y salir por esa interfaz.

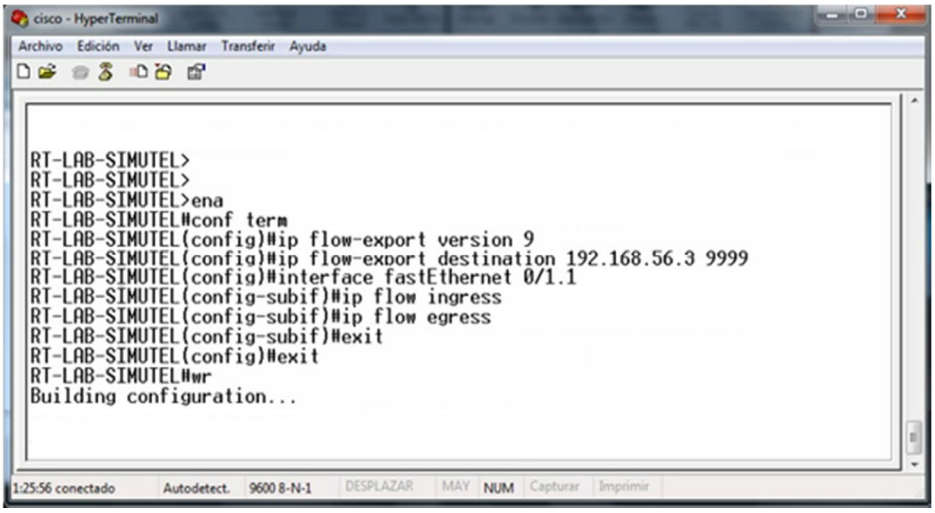


Figura 3.29- Configuración NetFlow en router Cisco. [8],[10]

3.3 Instalación de Wireshark

Wireshark es una herramienta de red que captura el tráfico de la máquina dónde se está ejecutando (analiza protocolos, direcciones IP, direcciones

MAC, etc) y nos muestra mediante su interfaz gráfica los paquetes capturados.

3.3.1 Sistemas operativos Windows.

1. Ingresamos a la página oficial de Wireshark: www.wireshark.org

2. Damos clic en la sección de descargas *download* y escogemos la opción según el sistema operativo Windows de nuestra PC. Nosotros escogimos Windows Installer de 32 bits.

3. Se guarda el ejecutable y se da doble clic, dándole permiso a la PC para que se ejecute.

4. Se abre una ventana de bienvenida y se da clic en *Next.*

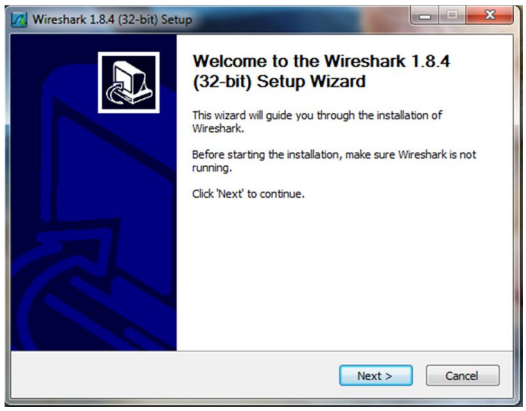


Figura 3.30- Ventana de bienvenida Wireshark

5. Se aceptan las condiciones de licencia y uso.

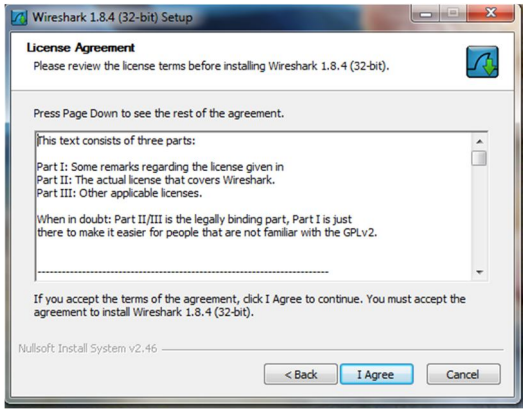


Figura 3.31- Ventana de licencia Wireshark

6. En la siguiente ventana se escogen todos los checkboxs, y se da clic en *Next*.

7. Se selecciona *Start Menu Item, Quick Launch Icon y File extensions*. Clic en *Next*.

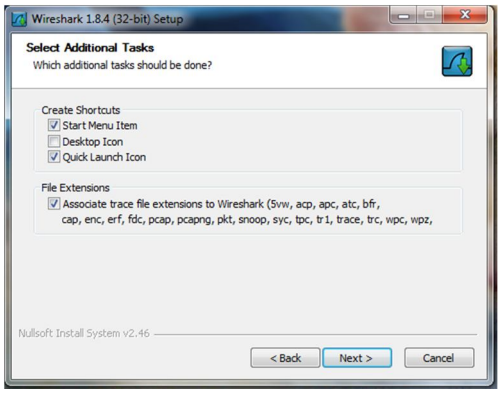


Figura 3.32- Ventana de checkboxs Wireshark

8. Clic en *Next*.

9. En esta nueva ventana nos presenta un checkbox donde nos pregunta si deseamos instalar WinPcap, este es un software de apoyo que Wireshark

utiliza para poder funcionar correctamente, así que si esta opción no está seleccionada hay que asegurarse de seleccionarla. Luego se da clic en *Install*.

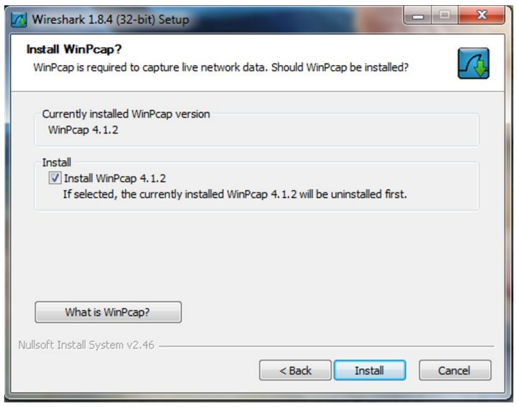


Figura 3.33- Instalación WinPcap

10. Aparece un cuadro de diálogo de WinPcap. Damos clic en *Next*.

11. Aparece ventana de bienvenida de WinPcap y damos clic en *Next* y aceptamos la licencia de uso.

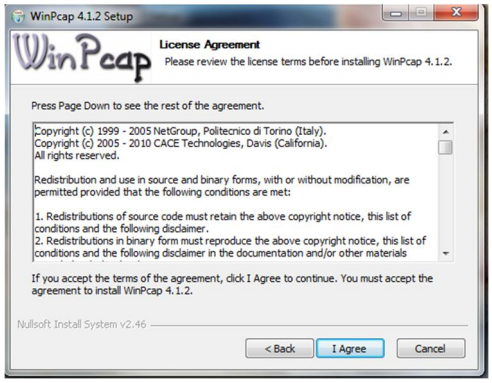


Figura 3.34- Ventana de licencia de WinPcap

12. Luego damos clic en *install*.

13. Esperamos que cargue y damos clic en *Finish*.

Figura 3.35- Finalización de instalación WinPcap

14. Luego de esto esperamos que Wireshark cargue todos sus complementos y damos clic en *next.*

15. En la última ventana aparecerán 2 checkbox, los cuales son opcionales. Podemos seleccionar el primero para ejecutar el programa al dar clic en *Finish*.

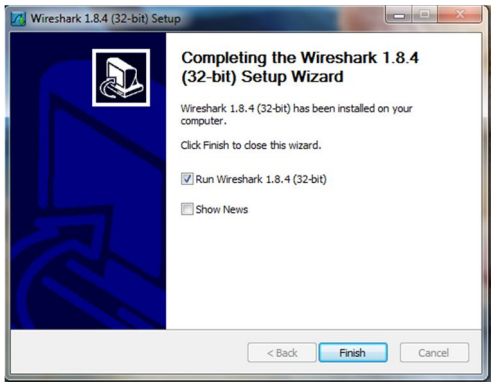


Figura 3.36- Finalización de instalación Wireshark

Cabe mencionar que la herramienta Wireshark es necesaria para poder capturar el tráfico que genera la estación de monitoreo en una red LAN, por lo que para los propósitos demostrativos de este proyecto es muy útil.

3.3.2 Sistemas operativos Linux.

Wireshark, como otros tantos otros programas, forma parte de los repositorios de las diferentes distribuciones de Linux así que podemos instalarlo directamente usando la terminal, escribiendo lo siguiente:

```
sudo apt-get install Wireshark
```

Una vez instalado Wireshark, podemos ejecutarlo como root, pero esta funcionalidad viene desactivada, por ser un riesgo de seguridad, ya que ejecuta muchos códigos con privilegio de administración. Como los usuarios no tienen permiso para manejar las interfaces de red directamente, se debe realizar una configuración para que un usuario regular en Ubuntu pueda usar Wireshark sin problemas [31]. Si Wireshark no está bien configurado no podremos ver las tarjetas de red ni tampoco capturar paquetes y veremos un mensaje de error diciendo lo siguiente:

No interface can be used for capturing in this system with the currence configuration.

(Couldn't run /usr/bin/dumpcap in child process: Permiso denegado)

See Capture Help below for details.

Lo primero que hicimos fue ejecutar los siguientes comandos en la terminal:

```
sudo addgroup –quiet –system wireshark
sudo chown root:wireshark /usr/bin/dumpcap
sudo setcap cap_net_raw,cap_net_admin-eip /usr/bin/dumpcap
```

Los últimos comandos permiten que para el nuevo grupo esté permitido usar dumpcap, que es el programa que usa Wireshark por defecto para poder capturar el tránsito de paquetes por las tarjetas de red; luego, solo falta añadir los usuarios que queramos al nuevo grupo.

```
sudo usermod –a –G wireshark nombreDeUsuario
```

En *nombreDeUsuario* se escribe el nombre del usuario de la PC. Lo siguiente es reconfigurar Wireshark para que los usuarios que no tienen permiso de administrador puedan capturar paquetes.

```
sudo dpkg-reconfigure wireshark-common
```

Luego se abre una ventana de configuración de Wireshark donde seleccionamos la opción *Sí,* y reiniciamos el computador por medio del siguiente comando:

```
sudo reboot
```

3.4 Instalación de la herramienta SNMP JManager

Una de las aplicaciones no licenciadas adecuadas para administrar de manera fácil y eficiente una red, es SNMP JManager, que permite monitorear sus principales aspectos de funcionamiento con un esquema de solicitud y respuesta simple. Tiene el soporte para operar con las tres versiones del protocolo SNMP tanto para direcciones de host IPV4 o IPv6, y entre los diferentes modos de desempeño destaca la funcionalidad de importar nuevas MIB para poder realizar consultas más específicas a los dispositivos.

Su interfaz gráfica facilita la configuración de conexiones con los dispositivos que queremos administrar, tales como dirección IP, número de puerto, comunidades en versiones 1 y 2c de SNMP, o usuarios con sus respectivas contraseñas de autenticación y cifrado en SNMPv3, entre otras; así mismo es sencillo elegir qué tipo de consulta se quiere realizar (lectura o escritura por ejemplo) y sobre qué objeto específico de la MIB hacerla, para lo cual podemos navegar entre los diferentes niveles de la MIB que hayamos elegido, ya sea la RFC1213 que aparece por defecto o alguna otra que se haya importado. Esta aplicación permite a cualquier NMS desempeñar un buen nivel de administración, por lo que es recomendable su descarga y probar su ejecución en entornos LAN.

1. Se ingresa a esta página http://snmp-jmanager.soft112.com/

2. Se da clic en *download*, y lo que se va a descargar es una carpeta donde tendrá todos los archivos de la aplicación que viene como un ejecutable.

3. Se ingresa a la carpeta y se da doble clic en el dibujo de computadores, el que es de tipo aplicación.

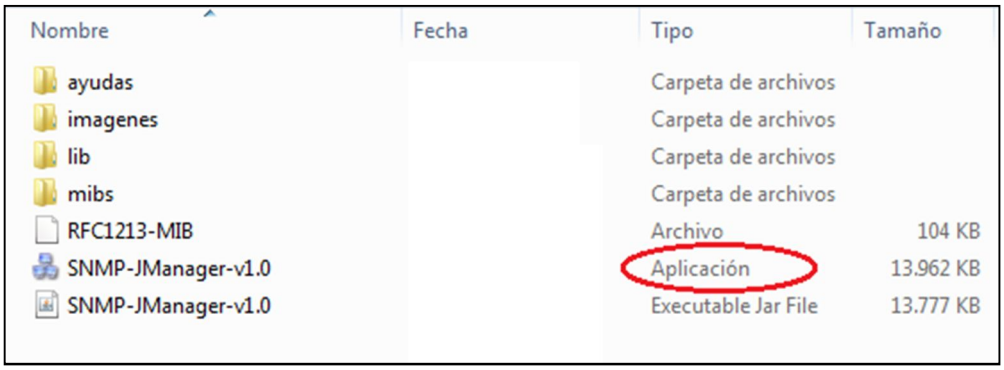


Figura 3.37- Ejecutando aplicación SNMP JManager

4. Finalmente se abre el programa SNMP JManager.

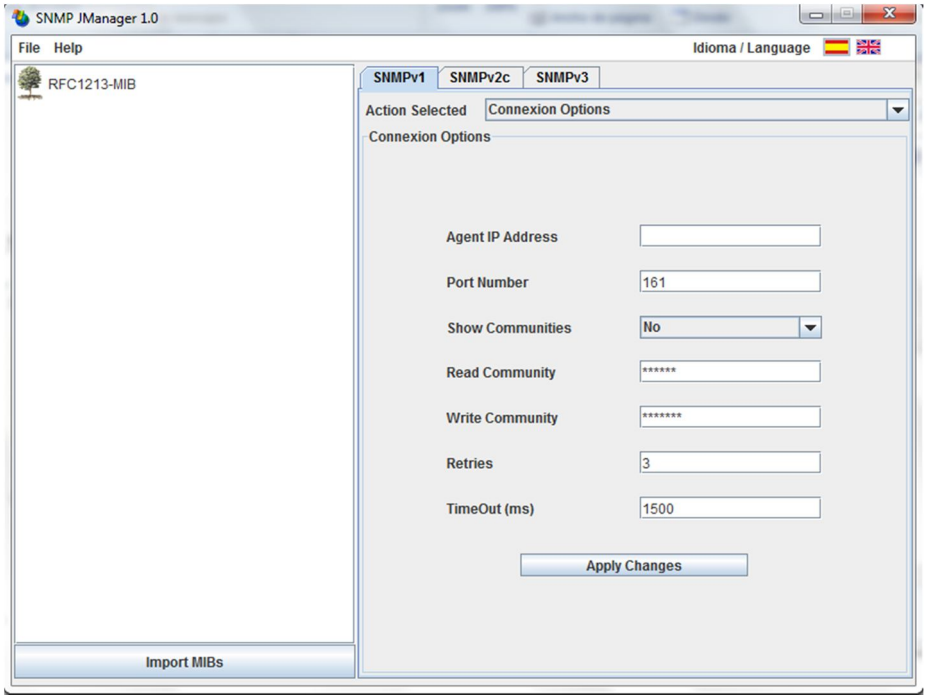


Figura 3.38- Aplicación SNMP JManager

El programa viene en inglés por defecto pero en la parte de arriba, del lado derecho se le puede cambiar el idioma a español. Para tener más

información acerca del manejo de esta aplicación, se puede ir al menú Help o Ayuda donde está un manual completo de esta aplicación.

3.5 Herramienta WhatsUp Gold V16.1.2.

3.5.1 Instalación

1. Ingresamos a la página oficial de WhatsUp Gold: http://www.whatsupgold.com/

2. Seleccionamos la opción *Downloads*, y nos saldrá algunos campos que debemos llenar con información personal, incluyendo un correo electrónico sin el cual no se podrá descargar el programa.

3. Luego de completar la información requerida, damos clic en *Descargar ahora*. Se descargará una versión de prueba de 1 mes del programa luego de lo cual se debe renovar la licencia pagando una suscripción.

4. Una vez descargado el instalador, se ejecuta dando doble clic.

5. Luego aparece una ventana de bienvenida, y se da clic en *Next*.

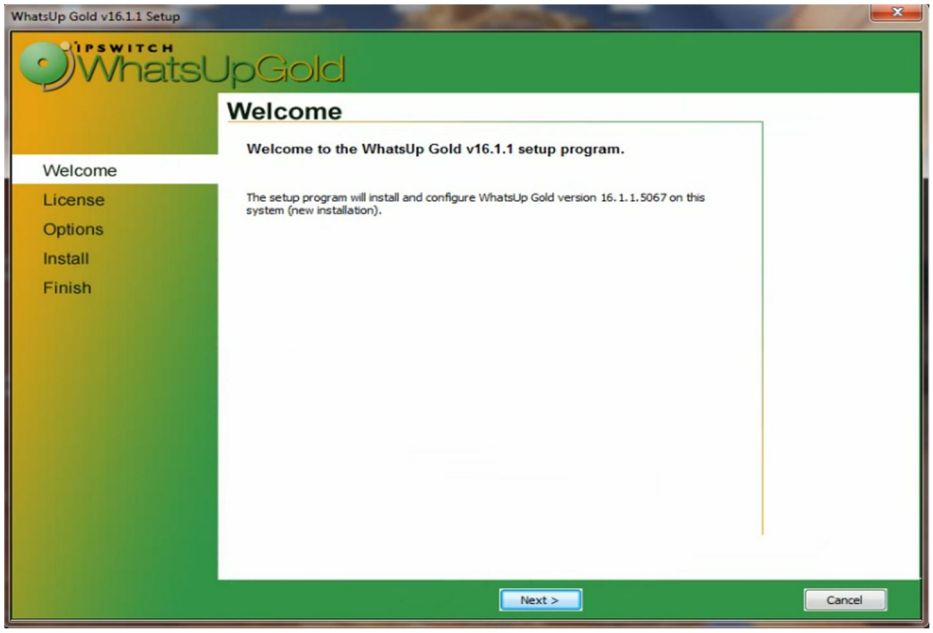


Figura 3.39- Ventana bienvenida WhatsUp Gold

6. El programa de instalación analiza el equipo donde se está instalando WhatsUp Gold para verificar qué programas o características adicionales se necesitan para su correcta instalación. Cualquier programa o característica de Windows faltantes, se instalarán y configurarán automáticamente luego de proceder con la instalación al hacer clic en la opción *Sí*.

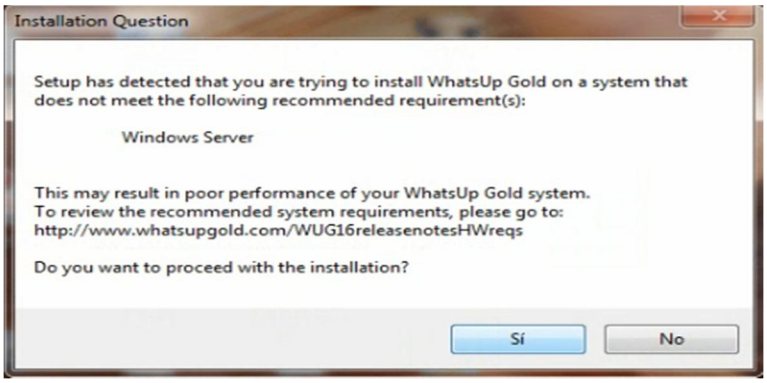


Figura 3.40- Permiso de instalación de WhatsUp Gold

En este caso, falta Windows SQL Server, el cual se instalará dentro de los siguientes pasos.

7. Seleccionamos la opción de aceptar licencia, y hacemos clic en *Next*.

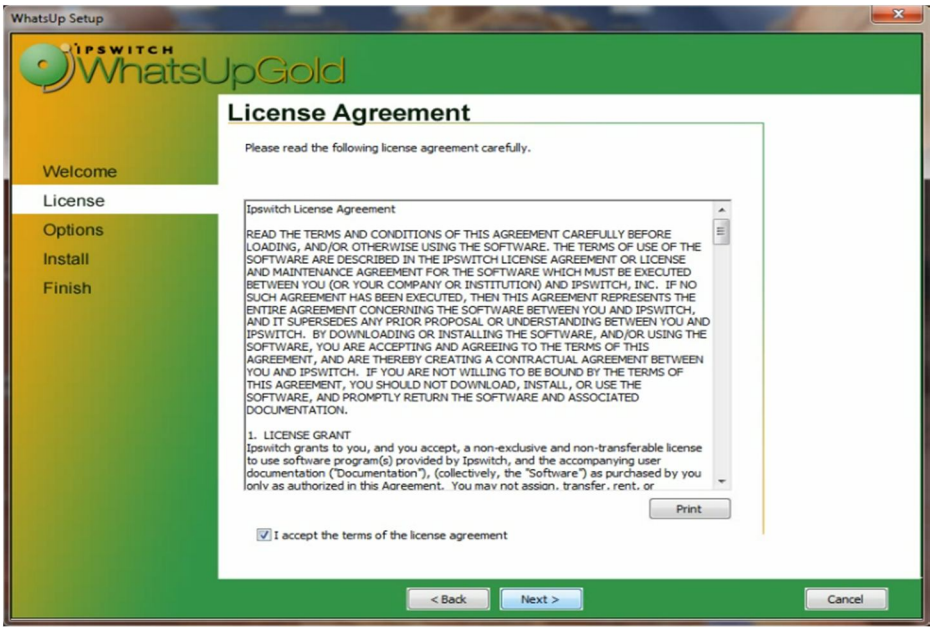


Figura 3.41- Aprobación de licencia de WhatsUp Gold

8. El programa de instalación verifica el serial de la versión de prueba, luego de lo cual nos aparece la ventana con el tiempo restante de la licencia, un mes.

9. Es recomendable tener activado el recuadro de *Ocultar opciones de instalaciones avanzadas*. Hacer clic en *Next*.

10. En la siguiente ventana, aparecen las rutas en las que se guardarán los archivos de aplicación y datos para la instalación de Microsoft SQL Server, el cual es complementario a WhatsUp Gold para poder almacenar los datos que

se recolectarán de los dispositivos. No se deben cambiar las rutas por defecto, presionamos *Next*.

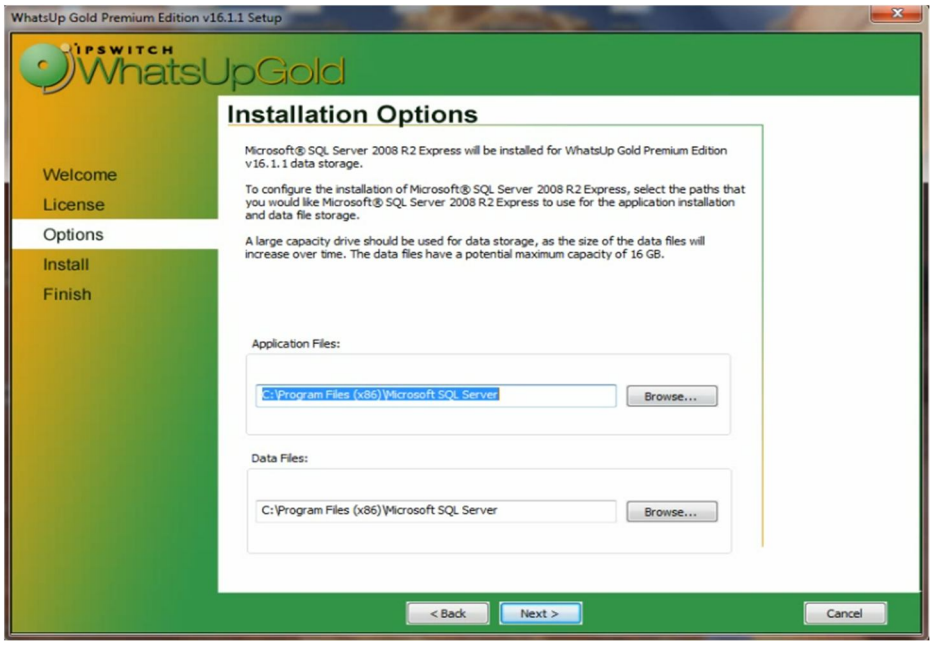


Figura 3.42- Opciones de Instalación de WhatsUp Gold

11. Le ponemos una contraseña al usuario administrador de SQL (sa) para la instancia de WhatsUp y damos clic en *Next*.

12. Escribimos otra contraseña para el SQL Server Login, el cual permite que WhatsUp Gold acceda a sus bases de datos. Damos clic en *Next*.

13. Se despliega una lista de direcciones IP (privadas y pública, si es que hay conexión hacia internet) de la máquina en la que estamos instalando WhatsUp Gold. Escogemos una dirección IP que sea estática, en este caso la 192.168.56.3, a donde llega toda la información de monitoreo, actuando nuestra máquina como un servidor de monitoreo. Damos clic en *Next*.

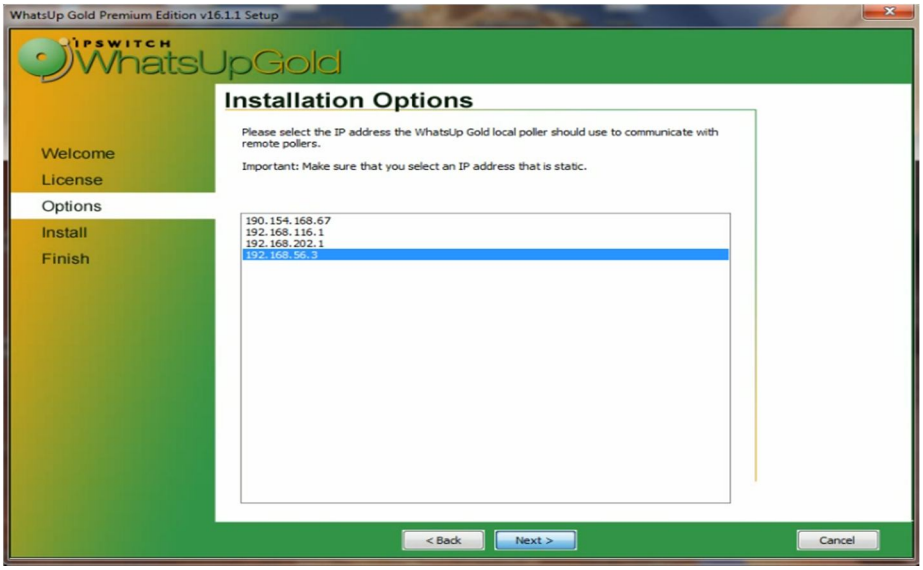


Figura 3.43- Direcciones IP de monitoreo

14. Aparece la carpeta por defecto en donde se instalará WhatsUp Gold, no es necesario cambiarla, a continuación, clic en *Next*.

15. Luego se abre una ventana preguntando si deseamos continuar, damos clic en *Sí*.

16. La siguiente ventana nos da la opción de escribir el puerto que el servicio IIS de Windows usará para la interfaz web de WhatsUp. Se recomienda dejarlo en el http 80, luego, clic en *Next*.

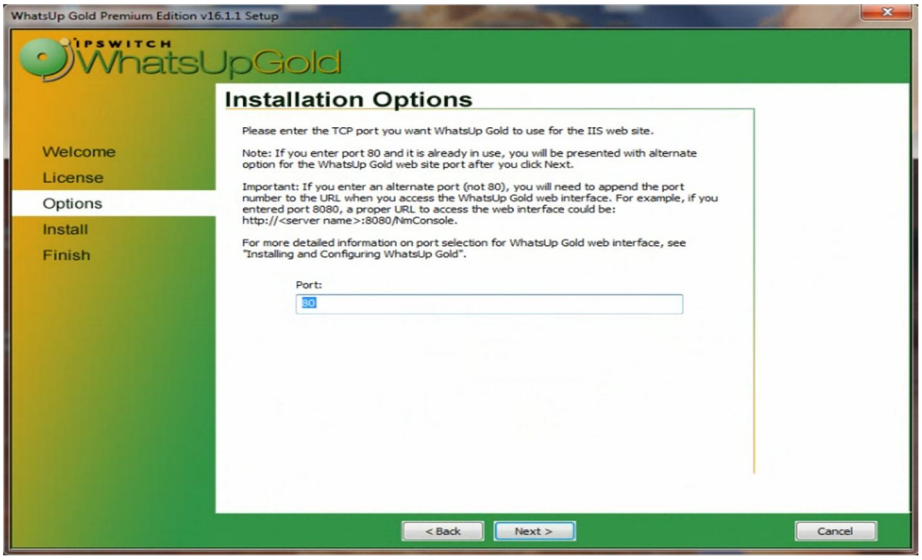


Figura 3.44- Puerto de interfaz web de WhatsUp Gold

17. En la siguiente ventana se da clic en *Proceed*.

18. Luego inicia el proceso de instalación de Microsoft SQL Server 2008 R2, el cual es automático. Se espera a que el proceso finalice, donde luego empieza la configuración automática de los Servicios de Información de Internet de Windows (IIS) y a continuación la parte final del proceso de instalación en donde se instala, configura, registra e inician los servicios de WhatsUp Gold.

19. Nos podemos dar cuenta de que tenemos instalado algunos otros programas en conjunto con WhatsUp:

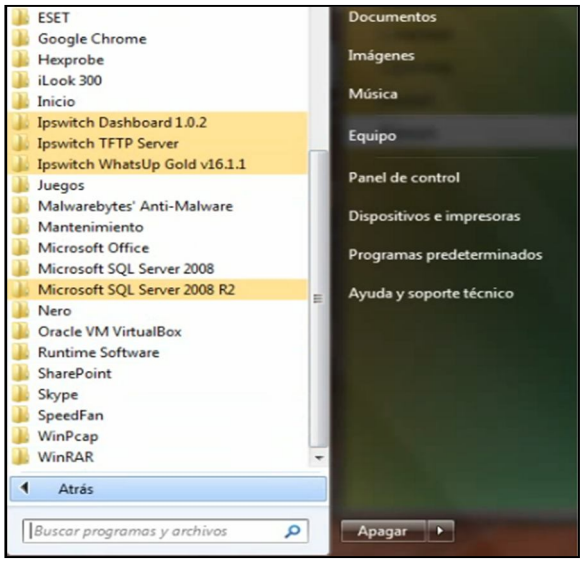


Figura 3.45- Programas instalados adicionales a WhatsUp Gold

20. Damos clic en *Finish* para terminar con la instalación.

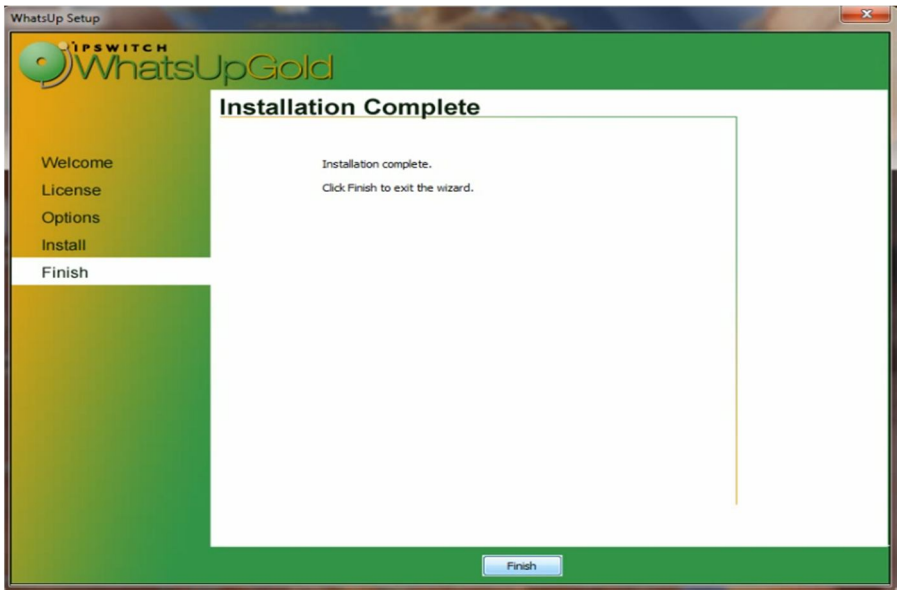


Figura 3.46- Finalización de instalación de WhatsUp Gold

Y ya podemos usar WhatsUp Gold desde nuestra PC. [6]

Las siguientes dos secciones detallan los procedimientos para descubrir los dispositivos de nuestra red, y luego modificar las propiedades que se configuran por defecto a los dispositivos, para que monitoreen las variables acorde a los requerimientos que sean necesarios. En la figura 3.47 se resume el mecanismo a seguir.

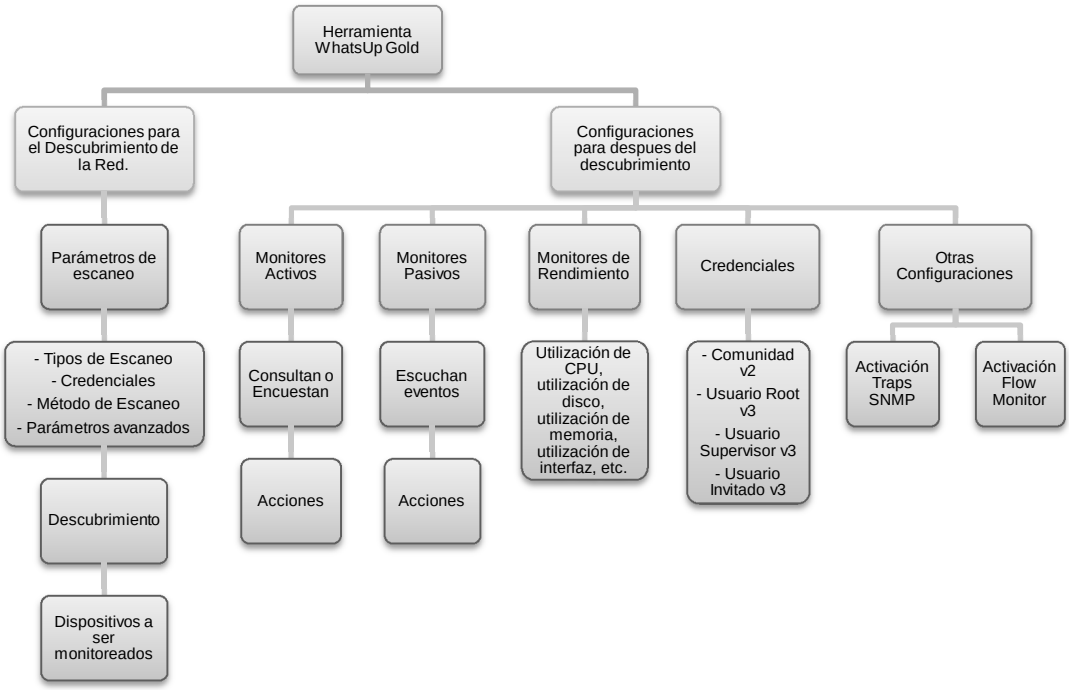


Figura 3.47- Esquema para descubrimiento de la red y configuraciones con WhatsUp Gold

3.5.2 Descubrimiento de la red.

Para el monitoreo de la red es necesario ejecutar el descubrimiento de todos los dispositivos que la conforman como computadores, servidores, impresoras, entre otros; de los cuales se guardan en WhatsUp Gold los más relevantes para su constante revisión. Dicho descubrimiento se puede hacer de dos formas:

1. Desde la consola de administración del programa, que se encuentra dentro del menú Inicio > Ipswitch WhatsUp Gold > WhatsUp Gold Admin Console.
2. Desde cualquier navegador web, digitando la IP de loopback o la propia IP privada, ya que al instalar WhatsUp Gold nuestra computadora se convierte en un servidor web que despliega en cualquier navegador la información más relevante de la red en formato comprensible al administrador.

Por motivos de mayores funcionalidades e interfaces manejables, escogeremos la segunda opción. Una vez ingresada la dirección IP en el navegador, ingresamos nuestro usuario y contraseña. Dentro de la interfaz web de administración, nos vamos a la pestaña *Devices* y dentro de la sección *Device Management*, escogemos la opción *Device Discovery*.

Del lado izquierdo de la pantalla aparecen algunas secciones que debemos configurar para que nuestro descubrimiento sea exitoso y obtenga la mayor cantidad de dispositivos útiles para nuestros fines, detalladas a continuación:

Parámetros de escaneo

Primero, se fija el tipo de escaneo. Hay tres clases:

- SNMP Smart scan: Este tipo de escaneo toma una dirección IP raíz, la cual tiene habilitado SNMP y en base a lo que se descubra en ésta, se descubre el resto de dispositivos de la red (y posibles subredes) a la que pertenece aquella IP.
- IP Range scan: Intervalo de direcciones IP que se incluirán en el descubrimiento.
- Hosts File scan: El descubrimiento se realiza en base a un archivo hosts de nuestro equipo, el cual contiene mapeos de nombres de hosts con direcciones IP.

El más recomendable es del rango de IP's, en donde definimos por medio de las direcciones inicial y final, el intervalo de IP's a las cuales se enviarán paquetes ICMP. Los elementos de red que sean capaces de responder a esas solicitudes de ping, aparecerán listados en la consola de descubrimiento. Dentro de la topología usada para este proyecto, la red 192.168.56.0/24 es en donde se encuentran las PC's a ser monitoreadas y el NMS, y la red 192.168.57.0/24 es la red en donde se encuentra el switch, al cual le hemos asignado una IP de esta red para poderlo administrar por WhatsUp Gold; si hubieran más switches en la red, tendrían IP's de esta red para que sea exclusiva de switches administrables.

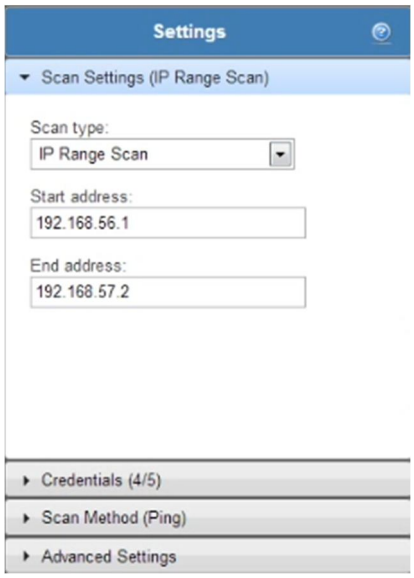


Figura 3.48- Tipos de escaneo

Credenciales

Este paso es muy importante porque tan pronto los dispositivos responden al ping y se ven activos, se les enviarán solicitudes SNMP para recolectar información importante de cada uno de ellos, alguna de ellas como nombre, contacto, ubicación, número y tipo de interfaces, servicios corriendo, etc.

Las credenciales no son más que las comunidades (versiones 1 y 2) y usuarios (versión 3). Como se vio en la sección 2.4, permiten acceder a las variables dentro de la MIB del dispositivo de red remoto, ya sea para consultar su valor o modificarlo; estos modos de autenticación deben estar configurados en ambos extremos de la comunicación de monitoreo: la máquina administradora o NMS y los MD.

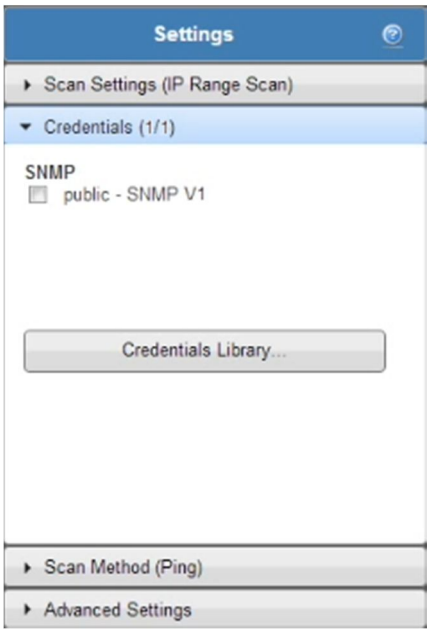


Figura 3.49- Parámetros de escaneo - Credenciales

Las credenciales se crean en la librería de credenciales, en donde al inicio aparece la credencial por defecto public de SNMPv1.

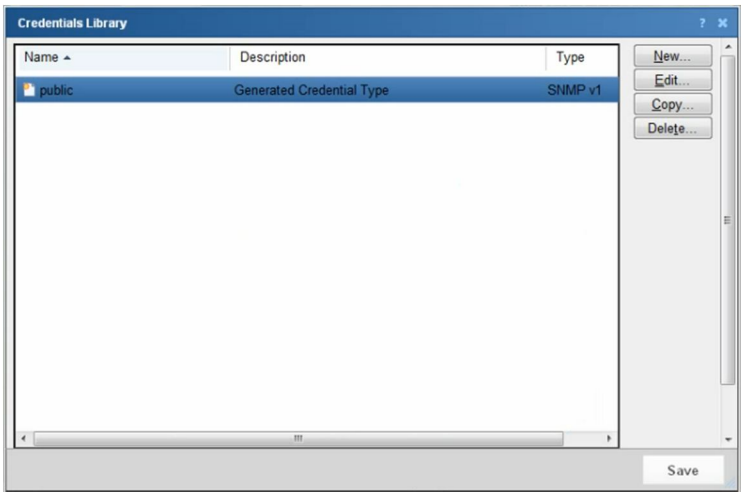


Figura 3.50- Librería de Credenciales

Haciendo un clic en *New*, seleccionamos el tipo de credencial que deseemos usar, en este caso primero definiremos una comunidad SNMPv2 para fines demostrativos solamente y damos clic en *OK* para crear la comunidad.

Escribimos el nombre de nuestra comunidad de lectura y escritura, en este caso “c0Mmuni7yAdm”, además de una descripción y nombres que servirán para guiarnos sobre el tipo y funciones de esta credencial.

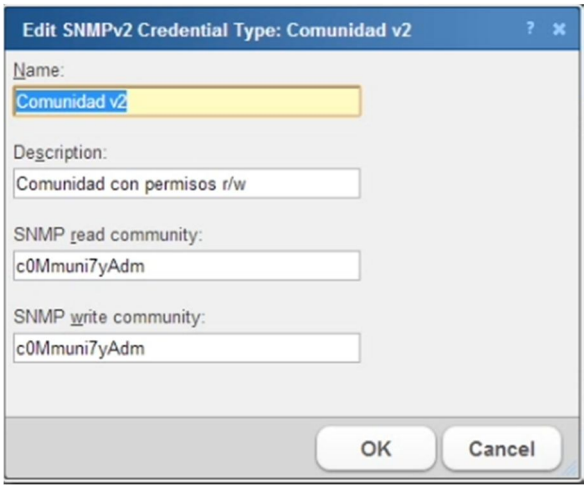


Figura 3.51- Tipo de credencial - Configuración de comunidad

Los campos que se han configurado para un usuario SNMPv3 son los siguientes:

- Name: Nombre de usuario, que se desplegará en la librería de credenciales.
- Description: Una ligera descripción del usuario para nuestra propia guía.
- Username: Es el usuario como tal, en este campo se tiene que escribir el nombre del usuario tal y como está configurado en los equipos remotos administrados.
- Authentication: Se definen los parámetros de autenticación, tales como el protocolo para autenticar los mensajes SNMP (MD5 o SHA) y la contraseña de autenticación, que debe tener como mínimo 8 caracteres.

- Encryption: Se definen los parámetros de cifrado, tales como el protocolo para cifrar el contenido de los mensajes SNMP (DES o AES128) y la contraseña de cifrado, que debe tener como mínimo 8 caracteres.

Hemos definido tres usuarios, cada uno con un nivel ascendente de seguridad: Invitado que no autentica ni cifra la mensajería SNMP; Supervisor que autentica los mensajes SNMP pero no los cifra; y finalmente, Root que es el usuario administrador que autentica y cifra el contenido de todos los mensajes SNMP. A continuación se muestran las imágenes de la definición de cada uno en la librería de credenciales (en el orden reciente en que fueron descritos):

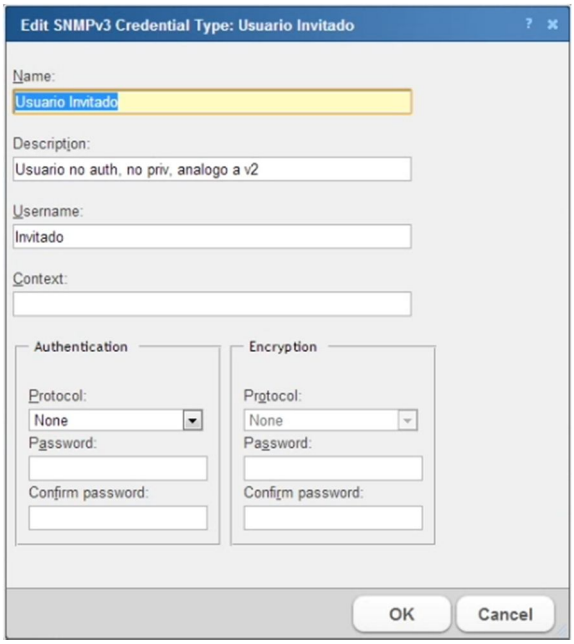


Figura 3.52- Tipo de credencial - Configuración usuario Invitado

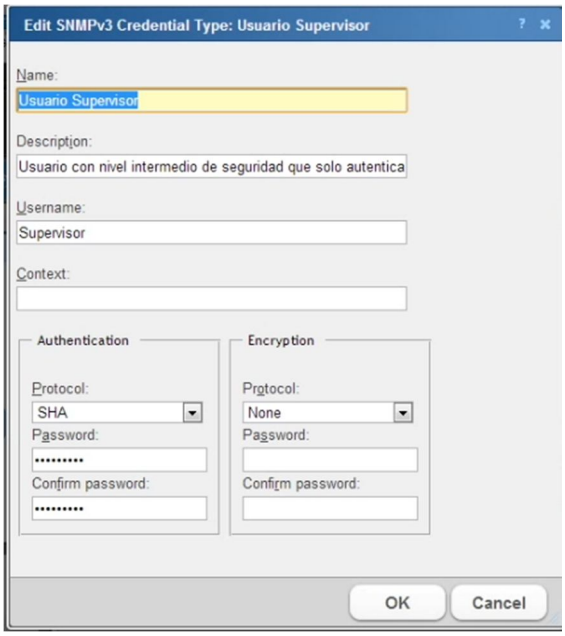


Figura 3.53- Tipo de credencial - Configuración usuario Supervisor

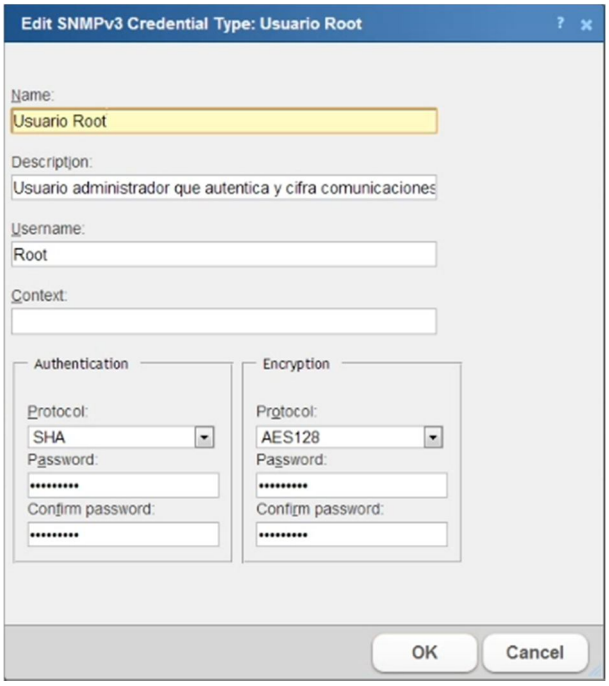


Figura 3.54- Tipo de credencial - Configuración usuario Root

Ya tenemos todas nuestras credenciales listas. Damos clic en *save* para guardarlas.

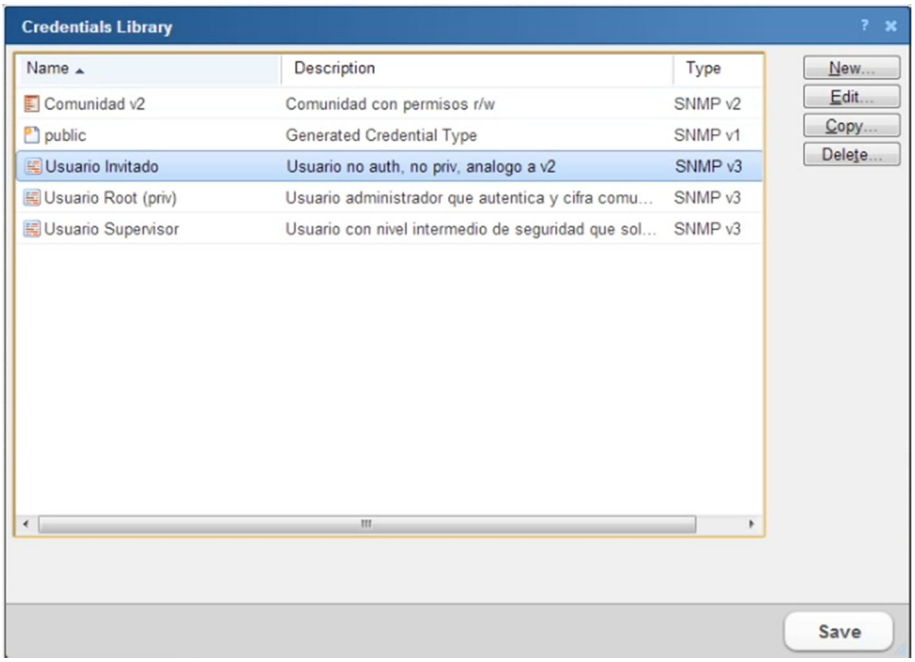


Figura 3.55- Librería de credenciales seleccionadas

Una vez guardadas, se las deja seleccionadas, para que se consulte a los equipos descubiertos con esas credenciales. Obviamente, esas mismas credenciales están configuradas en los diferentes equipos remotos que han sido descubiertos. Algo importante de recalcar es que la comunidad y los usuarios demostrados son los que están configurados tanto en el equipo router como la computadora, aunque en una organización existe total libertad de definir diferentes credenciales para los diversos equipos que conforman la red.

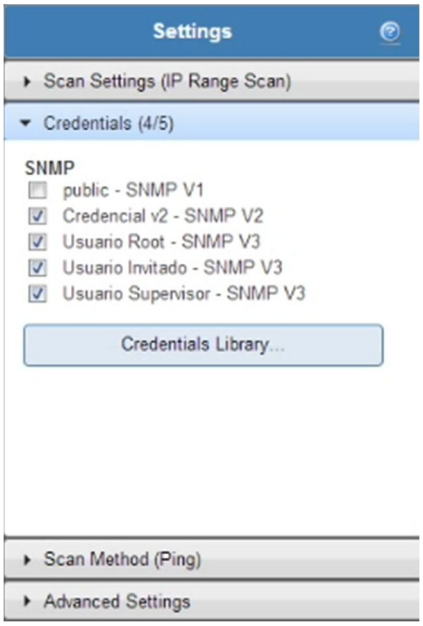


Figura 3.56- Selección de credenciales a usar en descubrimiento

Parámetros de escaneo.

Luego de configurar las credenciales, tenemos que ajustar los parámetros del ping en la sección *Scan Method* (Ping), tales como la duración y la cantidad de reintentos.

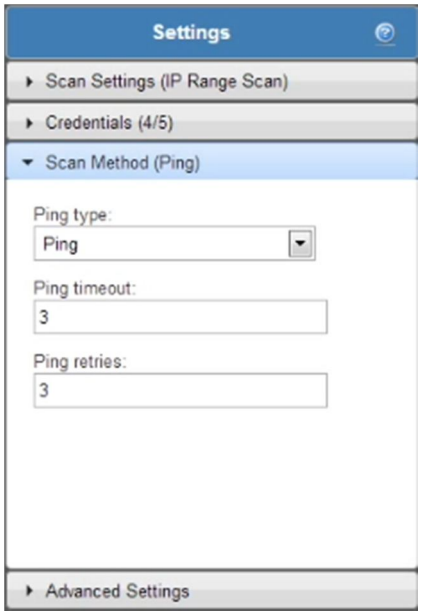


Figura 3.57- Método de escaneo

Parámetros avanzados.

Finalmente tenemos que definir los parámetros avanzados SNMP y del descubrimiento, en la sección *Advanced Settings*. Ahí definimos el tiempo de espera y reintentos de las solicitudes SNMP o WMI, la resolución de los nombres de los dispositivos descubiertos ya sea por DNS o el sysName (SNMP), la generación de un mapa de la topología descubierta y el escaneo de redes virtuales e inalámbricas.

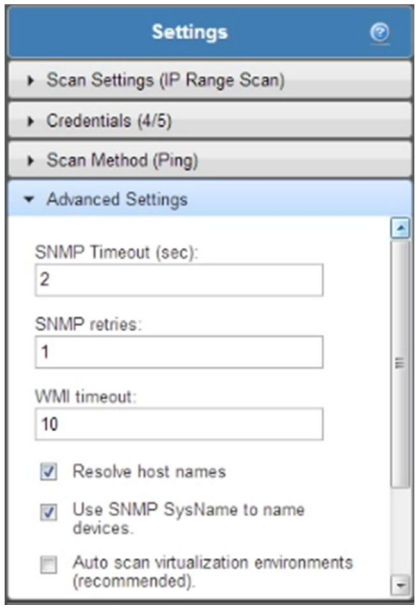


Figura 3.58- Parámetros avanzados

Una vez terminadas todas las configuraciones, hacemos clic en *Start a new scan session* para empezar el descubrimiento.

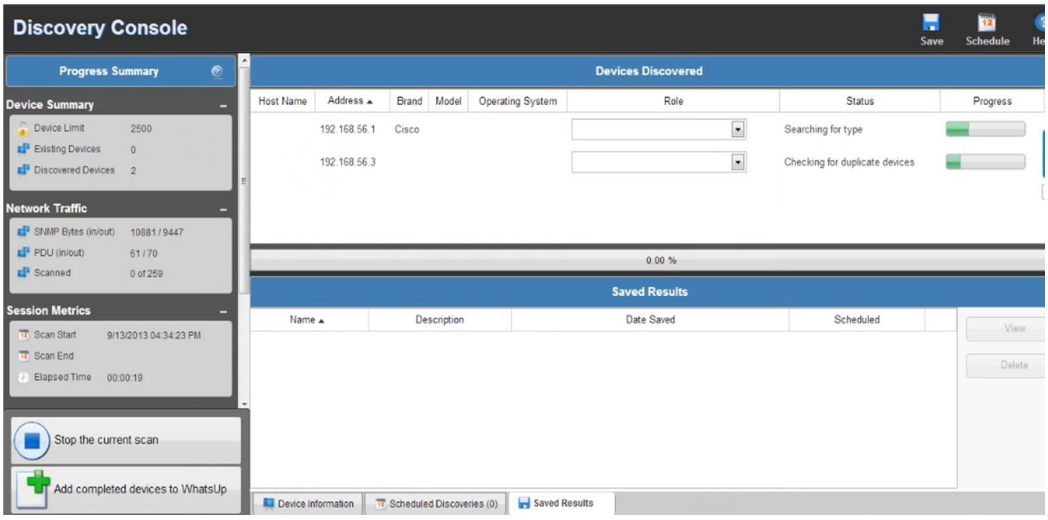


Figura 3.59- Inicio de escaneo o descubrimiento

Cada elemento de red despliega su IP y el tipo de dispositivo que es. Como nos podemos dar cuenta en la figura 3.60, el router cuya IP es 192.168.56.1 aparece como un switch, en el campo de rol (role). Los device roles son características asignadas a diferentes tipos de dispositivos predefinidos, durante el descubrimiento el host es consultado sobre diferentes parámetros y dependiendo de los que se descubran, se comparan con los diferentes roles que existen en WhatsUp, y el rol que se asemeje más a las características descubiertas en el host, será asignado. Claro está, se puede modificar luego de guardar los dispositivos los roles para una mejor búsqueda en el futuro y obviamente, modificar al tipo correcto de dispositivo si es que el rol inicial no lo describe con total exactitud, como es el caso del router.

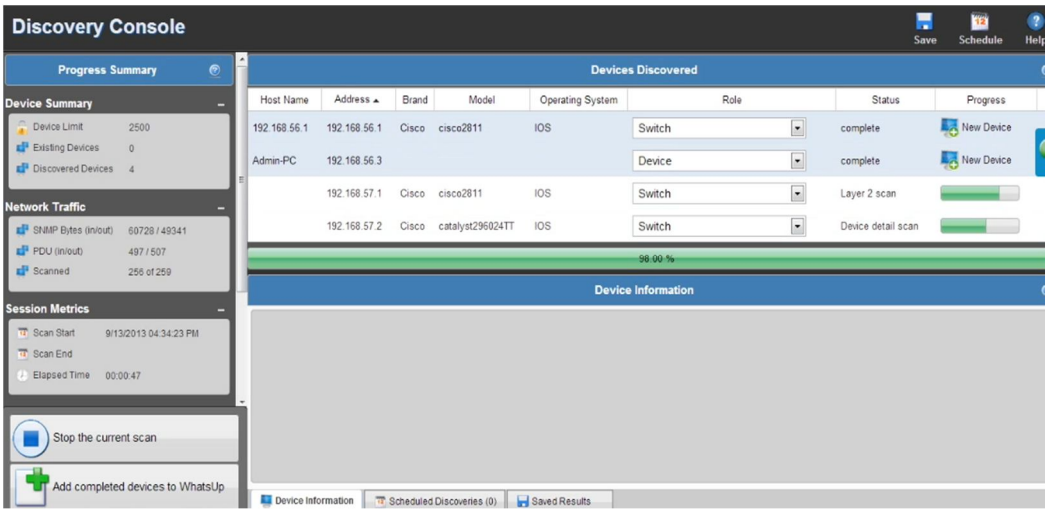


Figura 3.60- Detección de roles de dispositivos descubiertos

En las figuras 3.61 y 3.62, podemos observar que cada dispositivo (primero el router y luego el switch) muestra en la parte inferior de la pantalla de descubrimiento; información relevante como nombre, ubicación, contacto, descripción, interfaces de red y la credencial con la cual fue descubierto (un dispositivo puede tener varias credenciales, pero solo es descubierto con una). Esta información es consultada vía solicitudes SNMP, luego de que los pings son respondidos.

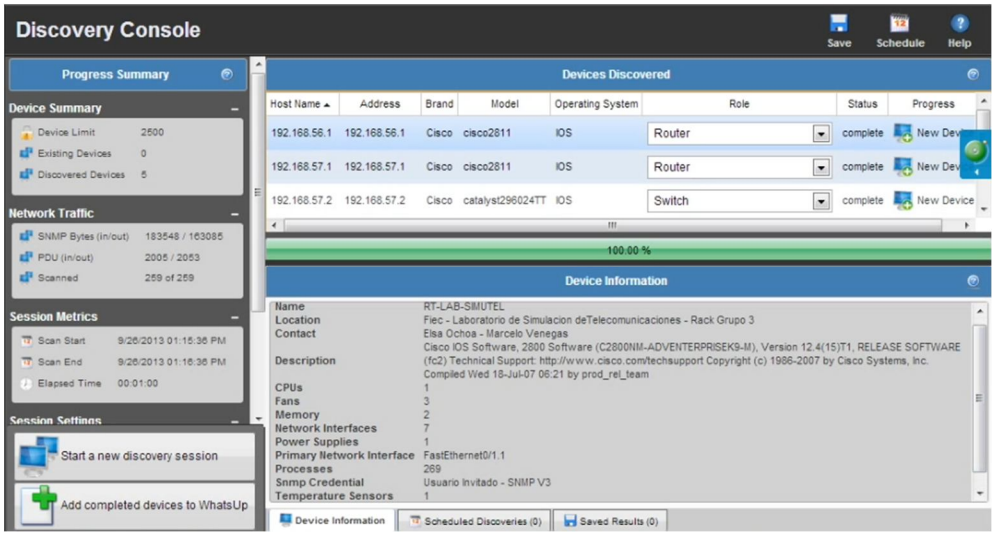


Figura 3.61- Router descubierto con credencial *Usuario Invitado*

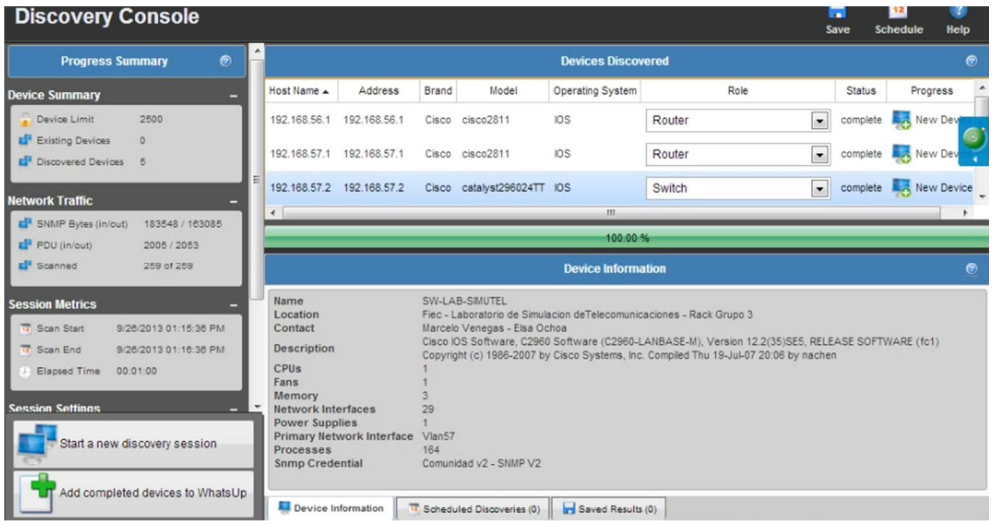


Figura 3.62- Switch descubierto con la comunidad

Una vez terminado el descubrimiento se hace clic en *Add completed devices to WhatsUp*, que se encuentra debajo del botón de inicio del descubrimiento, para guardar los dispositivos que nos interesen para su posterior monitoreo continuo.

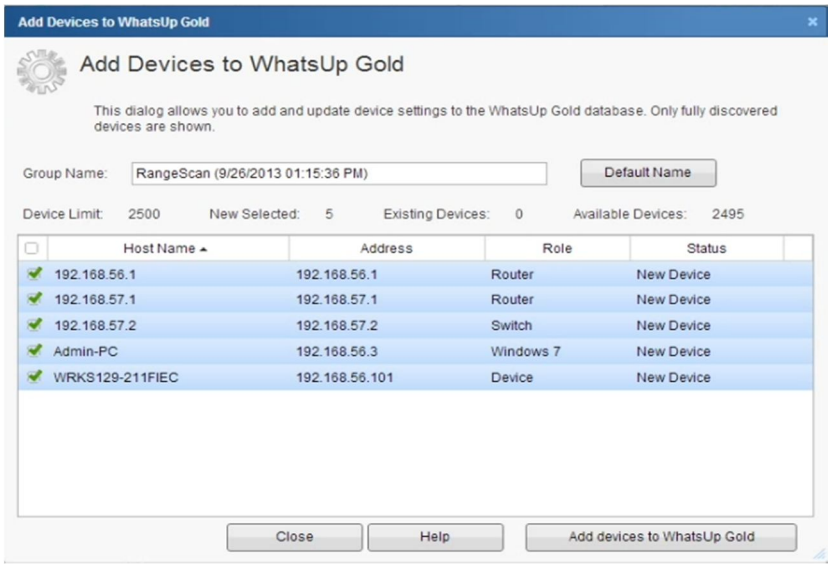


Figura 3.63- Dispositivos a guardar para monitoreo

Como se aprecia en la figura 3.63, todos los dispositivos salen seleccionados pero nosotros no seleccionamos el dispositivo con IP 192.168.57.1 ya que básicamente es el mismo que el 192.168.56.1; ambas se tratan solamente de subinterfaces del router. Finalmente, en la parte inferior derecha de la pantalla se hace clic en *Add devices to WhatsUp* para terminar de guardar los dispositivos. [6]

3.5.3 Configuraciones de propiedades de dispositivos

Dentro de *Devices* y específicamente en *Range Scan*, se encuentran todos los dispositivos añadidos del descubrimiento, para el monitoreo de WhatsUp Gold. Todos los dispositivos están de color verde, lo que quiere decir que están operativas (Up).

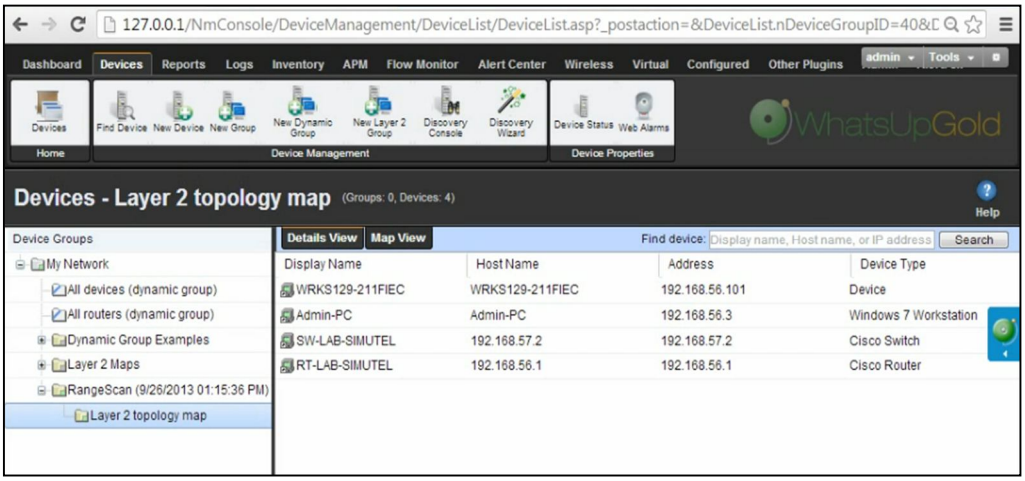


Figura 3.64- Dispositivos guardados y activos

Si damos clic en *Map View* podemos ver la topología de los dispositivos descubiertos.

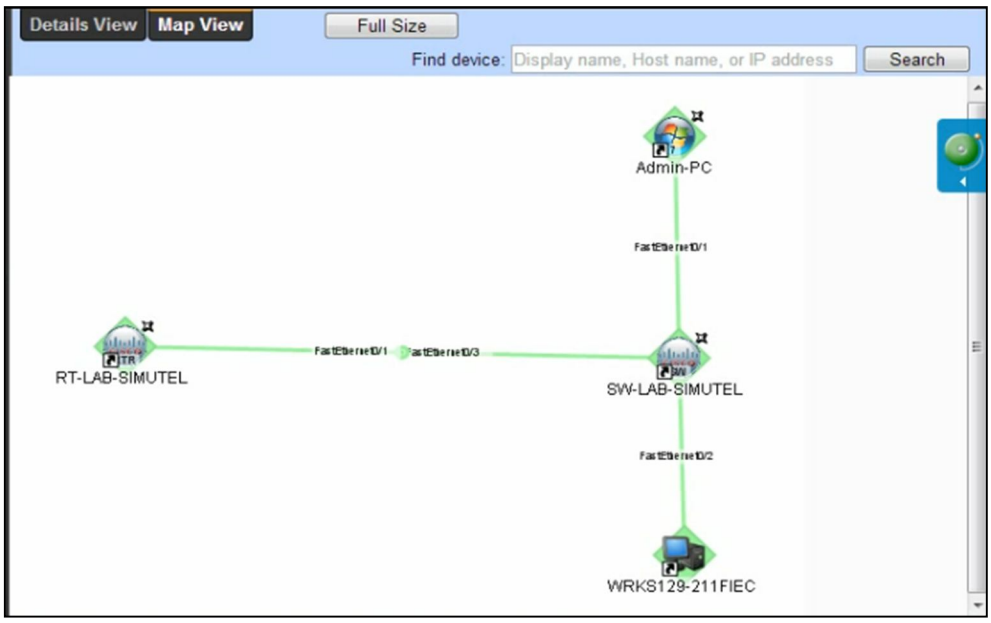


Figura 3.65- Topología de dispositivos descubiertos

Los dispositivos que han sido guardados o añadidos para el monitoreo respectivo, pueden ser configurados según las necesidades de la empresa/academia o los requerimientos que se necesiten para cada tipo de dispositivo según su propósito. A continuación veremos los tipos de configuraciones que se pueden realizar a un dispositivo, en este caso utilizaremos de ejemplo el router, pero cabe recalcar que todas las configuraciones a continuación explicadas las hemos realizado para todos los dispositivos guardados.

De la figura 3.64 le damos doble clic al router para configurarlo. Nos aparece una ventana donde están los datos estadísticos básicos del monitoreo pero salen vacíos ya que aún no hemos monitoreado nada. Damos clic en la parte superior derecha donde dice *properties*.

En la ventana *Properties* se pueden ver las características del equipo cuya mayoría son consultas de SNMP. Por defecto nos aparece la primera pestaña que es la de *Summary*, que contiene un resumen de las características del dispositivo seleccionado. Las características mostradas en *Summary* equivalen a la información adquirida en el descubrimiento del dispositivo entre los cuales están: categoría, contacto, descripción, versión, locación, etc.

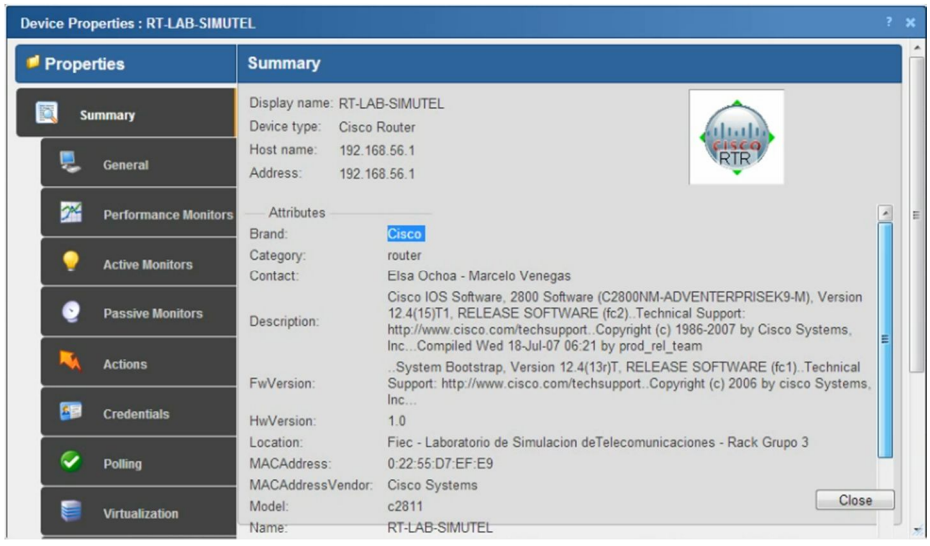


Figura 3.66- Propiedades de dispositivo - Summary

En la pestaña *General*, se muestra el nombre del dispositivo, el tipo de Polling, el host name, la dirección IP y el *device type*, el cual nos muestra que tipo de dispositivo es, en este caso identifica al router como un Cisco Router.

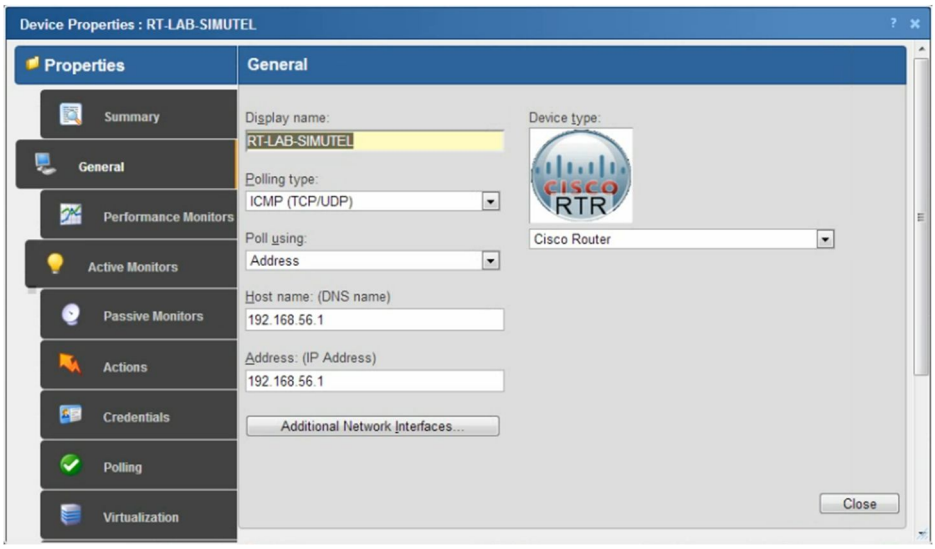


Figura 3.67- Propiedades de dispositivo - General

Los *Performance Monitors* o Monitores de rendimiento son los responsables de la recopilación de datos sobre los componentes de rendimiento de los dispositivos que se ejecutan en la red. Los datos luego son usados para crear reportes que tienden a utilizarse y disponerse para estos componentes de dispositivo. Los componentes de rendimiento de dispositivo son los siguientes: utilización de CPU, utilización de disco, utilización de memoria, utilización de interfaz y disponibilidad y estado latente del ping. Además, puede crear monitores de rendimiento personalizados para rastrear los monitores de rendimiento específicos para diferentes secuencias de

comandos. Para nuestro router, hemos seleccionado todos los tipos de monitores de rendimiento que vienen configurados por defecto. [6]

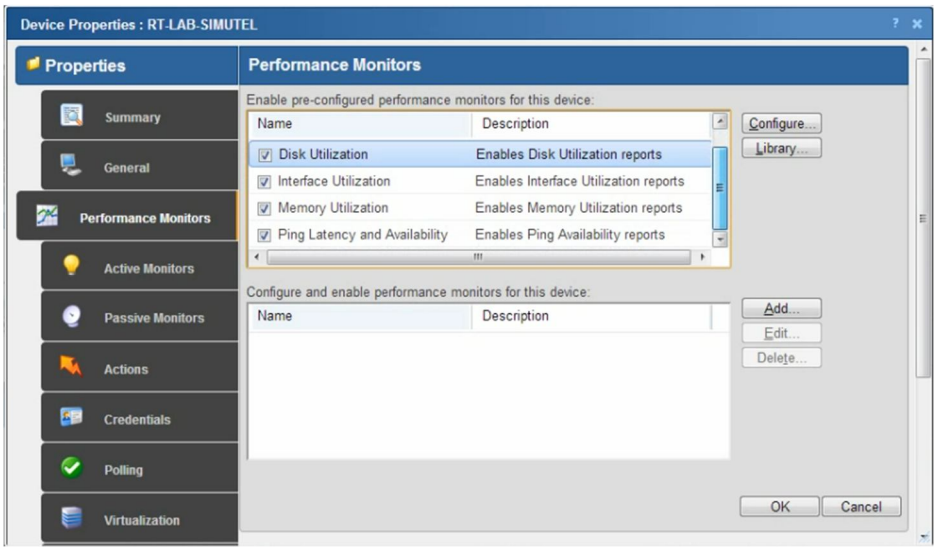


Figura 3.68- Propiedades de dispositivo - Performance Monitors

Luego damos clic en *OK*. En caso de querer agregar otro tipo de monitor que no sea parte de los críticos, se da clic en el botón *Add*, y se agrega el monitor que deseemos de la lista mostrada al usuario.

Los *Active Monitors* o Monitores Activos, encuestan los dispositivos de destino para obtener información como la accesibilidad ping, servicios de dispositivos, tales como servidores web o correo electrónico, y mucho más. Los Monitores Activos regularmente consultan o encuestan los servicios de

dispositivo para el que están configurados y esperan por respuestas. Si una consulta se devuelve con una respuesta esperada, el servicio consultado se considera activo o up. Si no se recibe una respuesta, o si la respuesta no es la esperada, el servicio consultado se considera inactivo o down; y un cambio de estado se emite en el dispositivo. [6]

Para nuestras encuestas, vamos agregar la del servicio SNMP, para saber por medio de las consultas, si el servicio SNMP de nuestro dispositivo está activo. Damos clic en el botón *Add* (Fig. 3.69).

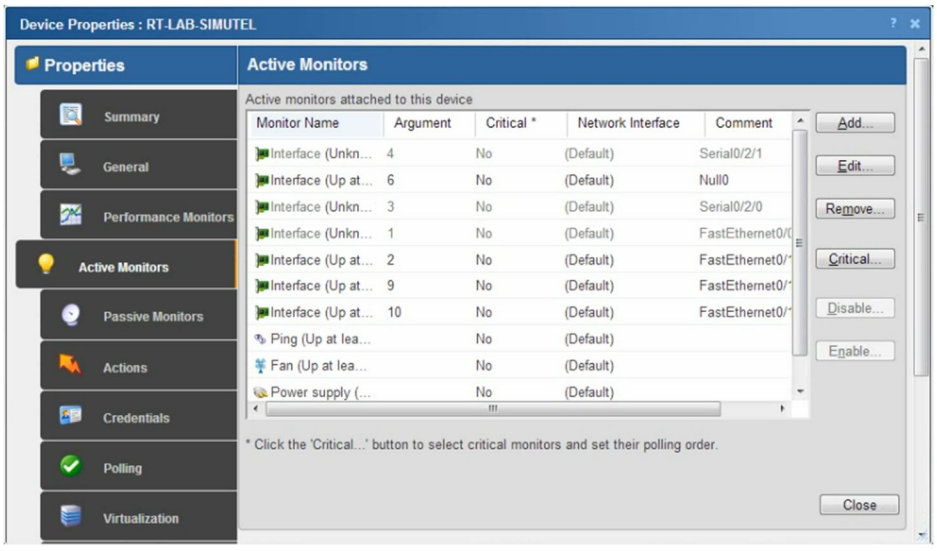


Figura 3.69- Adjunción de nuevo monitor activo

Damos clic en la flecha hacia abajo del combo box. De las opciones a escoger de la lista desplegable, elegimos *SNMP*. Una vez escogida la opción SNMP, damos clic en *Next*.

Figura 3.70- Selección del monitor *servicio SNMP*

En *Set Polling Properties*, seleccionamos el checkbox, una vez hecho esto, hemos activado el servicio de las consultas para el actual monitor activo, hay que tener en cuenta que si no activamos el checkbox, no vamos a poder realizar las consultas a nuestro dispositivo. Lo siguiente que hacemos es seleccionar la dirección IP del dispositivo y a continuación damos clic en *Next*.

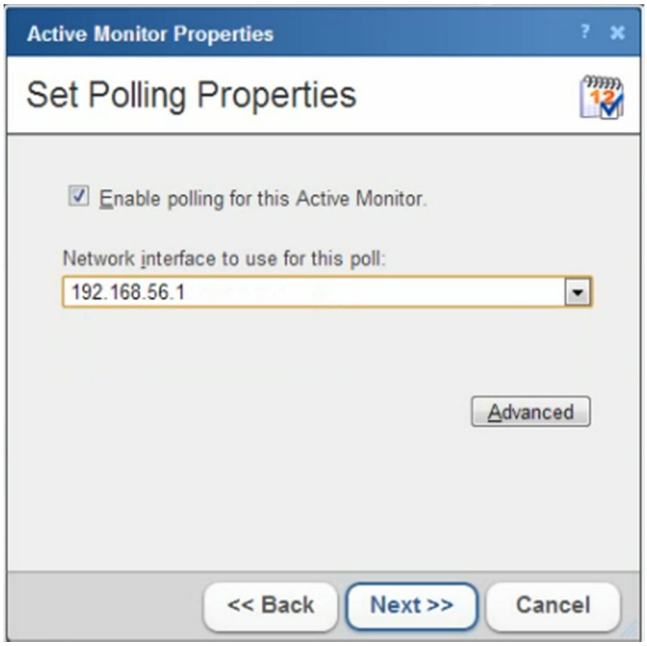


Figura 3.71- Habilitación de consultas y selección de IP a consultar

Setup Actions for Monitor State Changes es utilizado para cuando las solicitudes de WhatsUp Gold no encuentran respuesta del servicio SNMP (en nuestro caso, para el monitor activo que configuramos en adición a los preestablecidos) en el dispositivo monitoreado. Al no encontrar respuesta, el servicio SNMP adopta el estado down y al suceder esto se aplica una acción definida por una política. Si hay definidas políticas adicionales a las que aparecen por defecto, se despliegan las opciones del menú bajo *Apply this Action Policy* que se muestra en la figura 3.72 y se escoge la deseada, si deseáramos definir una acción personalizada, se hace clic en el botón de los puntos suspensivos a la derecha del combo box para definirla. Para nuestras

pruebas, no era esencial definir acciones, por lo que se continúa sin este paso, dando un clic en *Finish*.

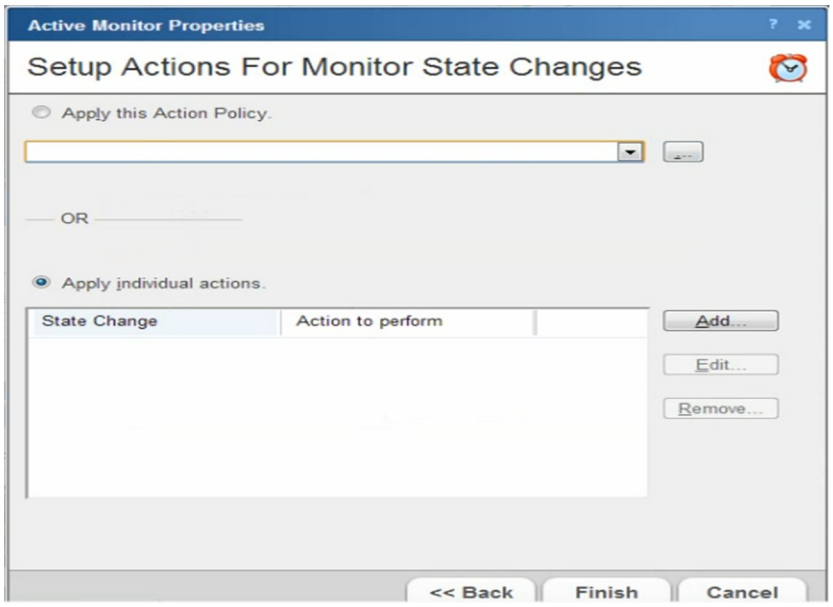


Figura 3.72- Ventana de aplicación de políticas

Como ya se tienen definidos los monitores activos, damos clic en *OK*.

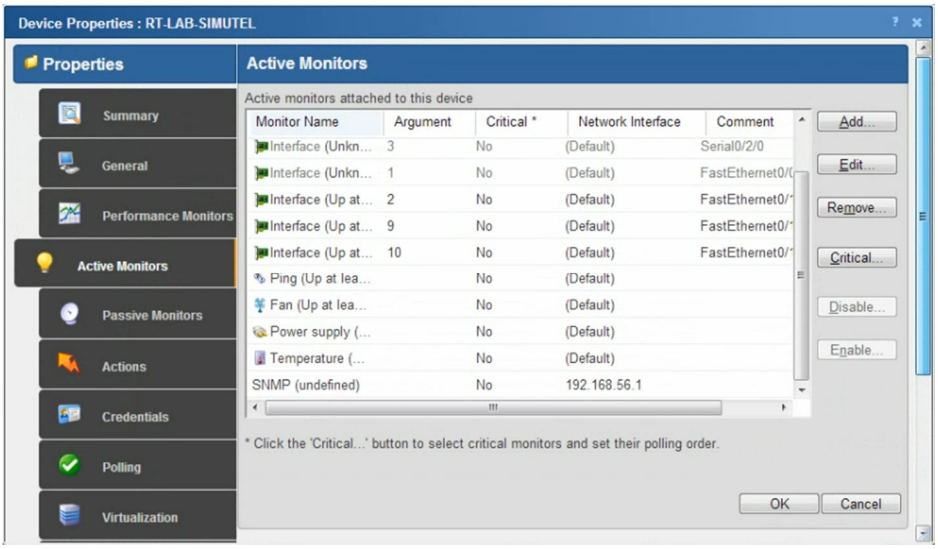


Figura 3.73- Finalización de configuración de monitores activos

Los *Passive Monitors* o Monitores Pasivos, son los que escuchan los eventos de un dispositivo. Así como los monitores activos activamente consultan o encuestan dispositivos por datos, los monitores pasivos escuchan pasivamente para eventos de dispositivo. Debido a que los monitores pasivos no hacen encuestas a dispositivos, utilizan menos ancho de banda que los monitores activos. Aunque los monitores pasivos son útiles, no se debe confiar en ellos exclusivamente para controlar un dispositivo o monitores de servicio pasivo, se debe utilizar en combinación con monitores activos. Vamos a la pestaña de *Passive Monitors*. Damos clic en *Add*.

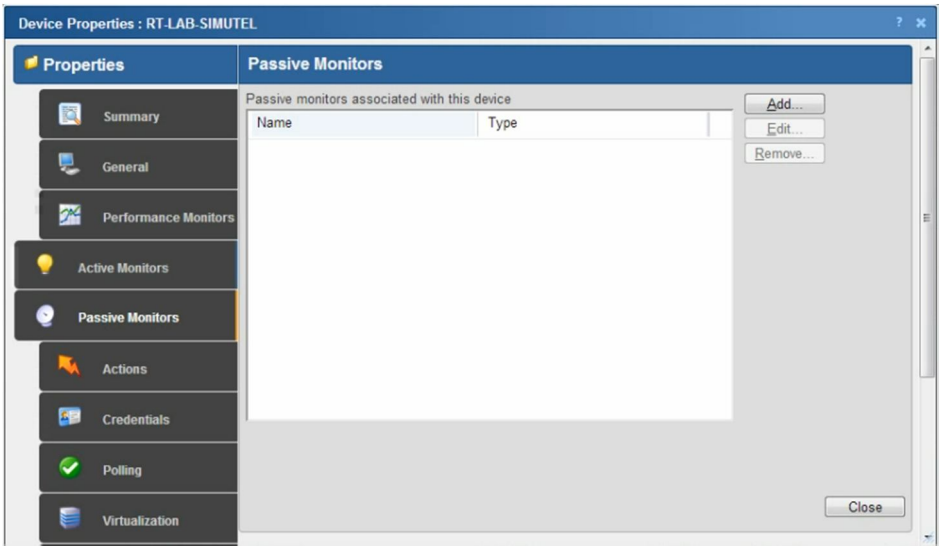


Figura 3.74- Passive Monitors

Escogemos el tipo de Monitor Pasivo que nos gustaría añadir. Seleccionamos *SNMP Trap*. También podemos seleccionar un tipo

específico de Trap. Nosotros escogimos *Any* (cualquiera). Damos clic en *Next*.

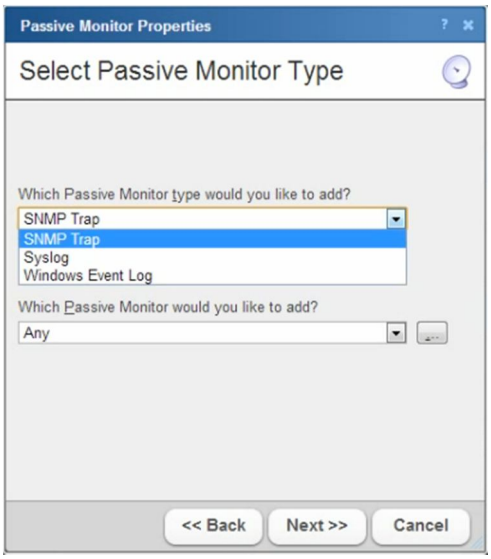


Figura 3.75- Tipos de monitores pasivos

De la misma manera que en *Active Monitor*, podemos obviar el paso de elegir una acción. Damos clic en *Finish*.

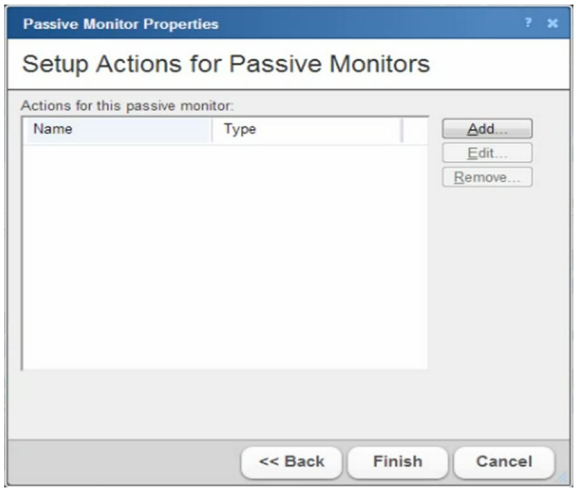


Figura 3.76- Ventana de aplicación de una acción

Ahora ya tenemos nuestro Monitor Pasivo. Damos clic en *OK*.

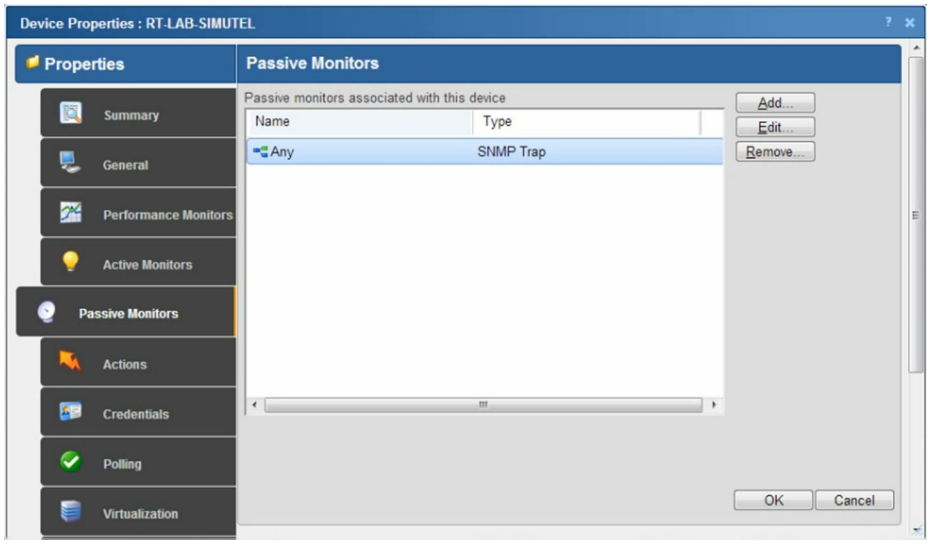


Figura 3.77- Monitor pasivo agregado – Traps

En *Credentials* definimos que tipo de credencial usaremos para las peticiones de nuestro dispositivo. Damos clic al botón *Edit*.

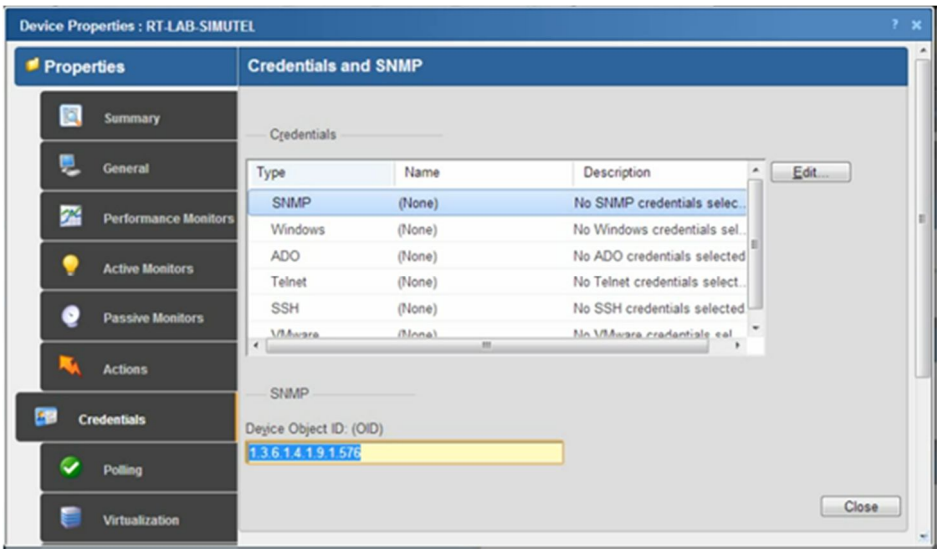


Figura 3.78- Configuración de credencial

A continuación escogemos el tipo de credencial a utilizar. Nosotros escogimos la credencial de *usuario Invitado v3*. Damos clic en *OK*.

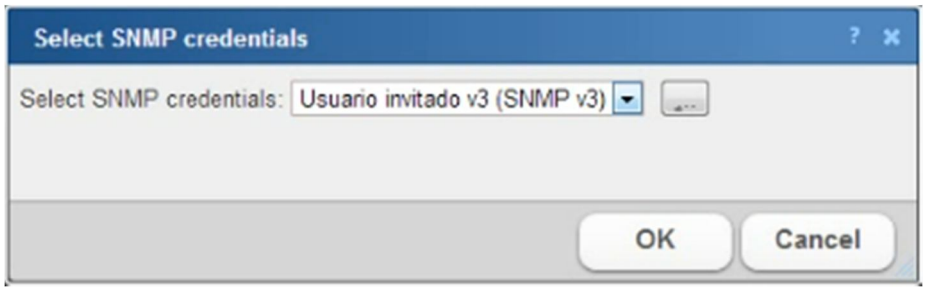


Figura 3.79- Selección de credencial - Usuario Invitado v3

Como podemos ver, al lado izquierdo de la palabra SNMP aparece una pequeña imagen, la cual significa que se ha definido la credencial a utilizar. Damos clic en ***OK***.

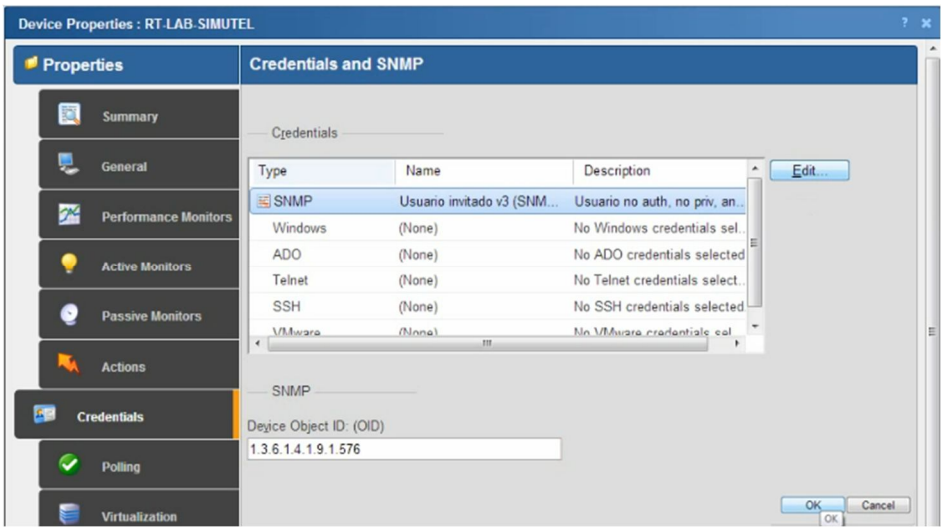


Figura 3.80- Credencial escogida

En la caja donde dice *Device Object ID*, se especifica la identidad del dispositivo por medio de un OID, que en este caso es 1.3.6.1.4.1.9.1.576, el cual corresponde al equipo cisco2811.

3.5.4 Otras Configuraciones

Otras de las configuraciones que se realizan y que no son parte de las propiedades, se las configura en las pestañas de WhatsUp Gold y en la consola de WhatsUp Gold, estas configuraciones son de activación del flujo de datos (Netflow) y Traps SNMP respectivamente.

3.5.4.1 Traps SNMP

Ingresamos a la consola de WhatsUp Gold, por medio del menú Inicio > Todos los programas > Ipswitch WhatsUp Gold v16.1.4 > WhatsUp Gold Admin Console.

Luego nos vamos al menú Configure > Program Options. Vamos a ver una ventana donde están seleccionados los checkbox que muestra la figura 3.81, y de ahí escogemos *Passive Monitor Listeners* en la parte lateral izquierda.

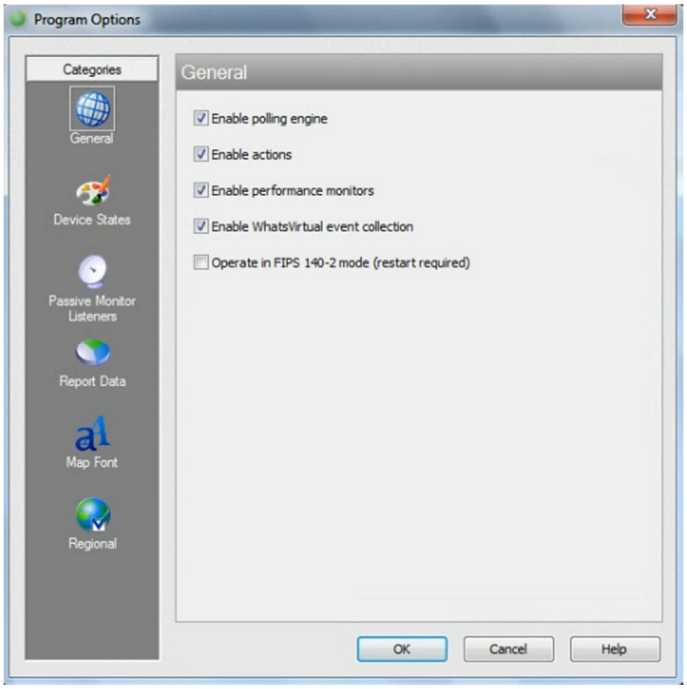


Figura 3.81- Program Options

Seleccionamos *SNMP trap* y damos Clic en *Configure*.

Luego seleccionamos los dos primeros checkbox y damos clic en *Ok*. Asegurarse de escribir el puerto 162 que es por donde se reciben los traps.

Figura 3.82- Configuración de escucha de los traps

A continuación aparece un mensaje donde se da clic en la opción *Sí* para aceptar traps no solicitados. Finalmente damos clic en *OK* para guardar los cambios.

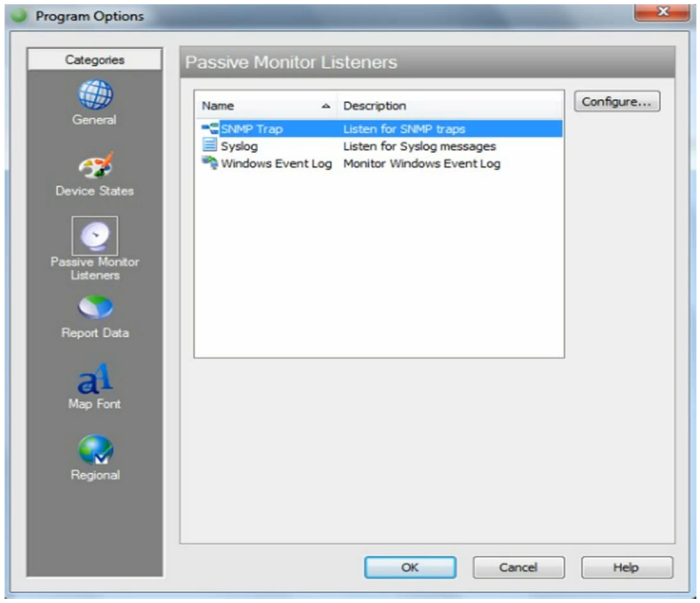


Figura 3.83- Finalización de Passive Monitors Listeners

3.5.4.2 Flow Monitor

Iniciamos el servidor web de WhatsUp Gold y vamos a la pestaña de Flow Monitor > Settings. El puerto de escucha para el Flow Monitor es por defecto el 9999 en WhatsUp. Log level debe ser Normal. Damos clic en *Ok*.

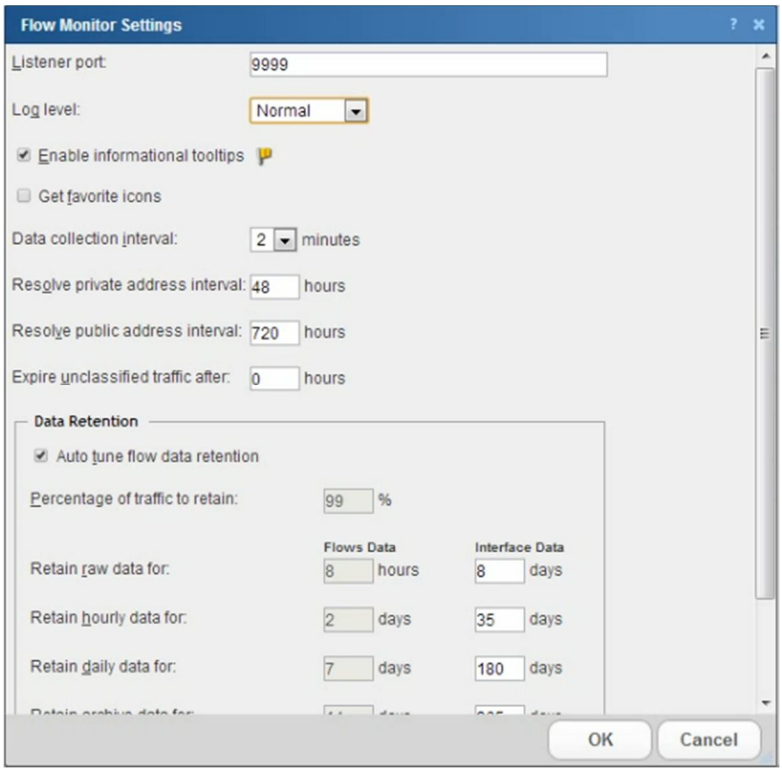


Figura 3.84- Configuración Flow Monitor

Vamos a la pestaña Flow Monitor > Sources. Vemos que tenemos una fuente (router) de donde provienen los datos estadísticos de Flow Monitor. Damos clic en *Edit*.

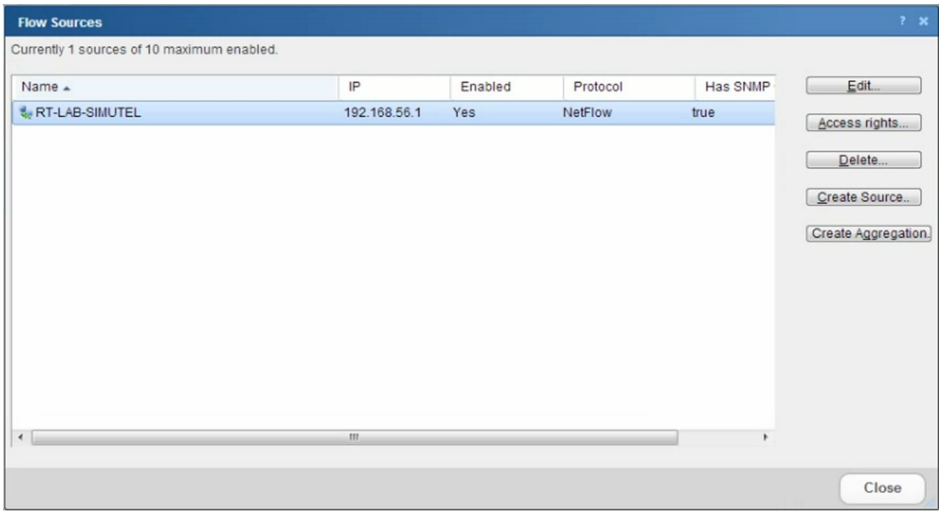


Figura 3.85- Fuente de datos estadísticos

Nos aseguramos que *Enable data collection from this source* esté seleccionado. Podemos ver las interfaces activas, la comunidad, nombre, etc. Damos clic en *OK*.

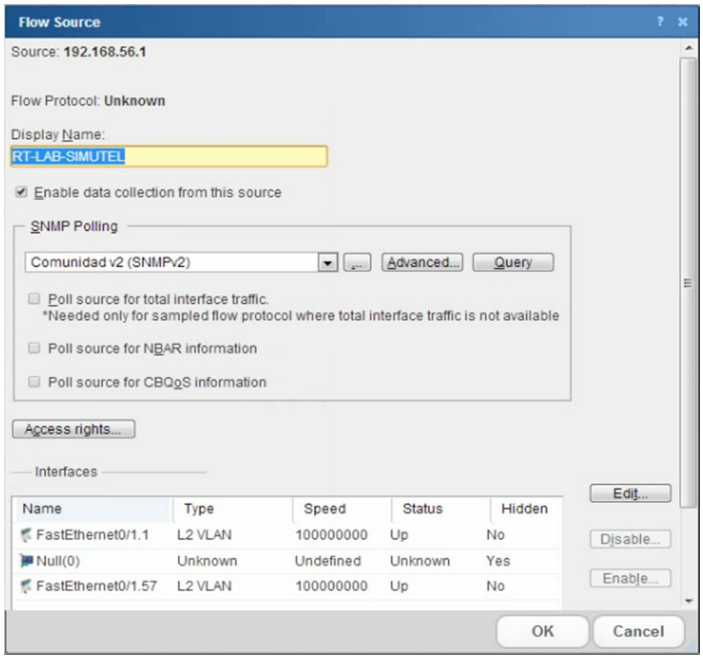


Figura 3.86- Configuración de flow source

3.6 Prueba de vulnerabilidades en una red usando SNMP v1 y v2

Esta prueba consiste en ingresar un código malicioso vía web por medio de un set a un dispositivo router, para poder cambiar la clave de acceso del mismo y poder ver o realizar cualquier tipo de configuración. Para realizar esta prueba necesitamos tener una máquina atacante (Windows) y un router victima (Cisco).

En el router configuramos la clave de administrador por medio de los comandos:

```
enable
configure terminal
enable secret adminaccess
```

Al mostrar las configuraciones con el comando *show run*, la contraseña se cifró automáticamente, y se mostró de la siguiente manera:

```
enable secret 5 $1$EBYg$4D0iRBwxs8z6GF2zAJsiL
```

Verificamos la conectividad del router vía web; ya que el router tenía activo el *http server*, pudimos ingresar por medio de su dirección IP.

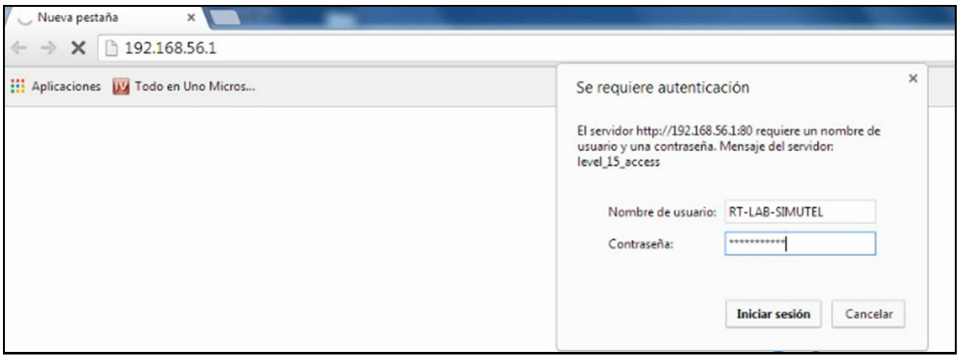


Figura 3.87- Ingreso vía web al router

Ingresamos con la contraseña de modo privilegiado que le asignamos al router, la cual es adminaccess.

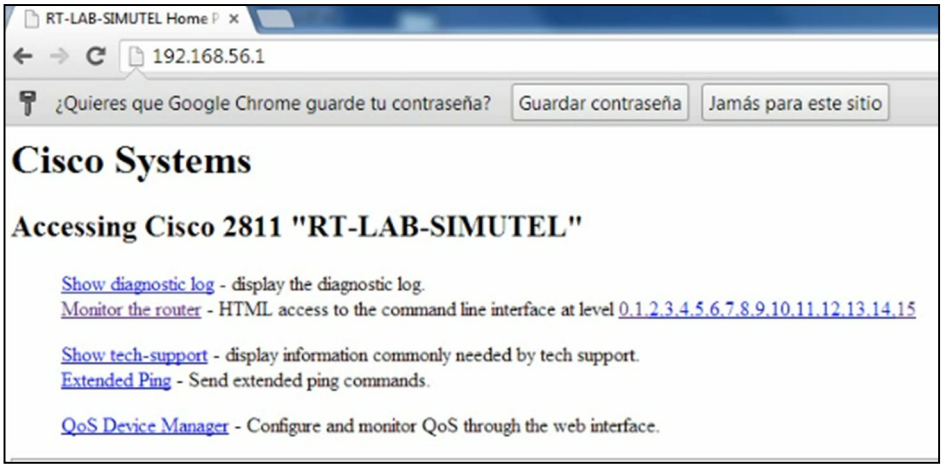


Figura 3.88- Página principal del router

Luego dimos clic derecho y seleccionamos la opción de “ver código fuente” para verificar que no está hecha una previa inserción del código malicioso.

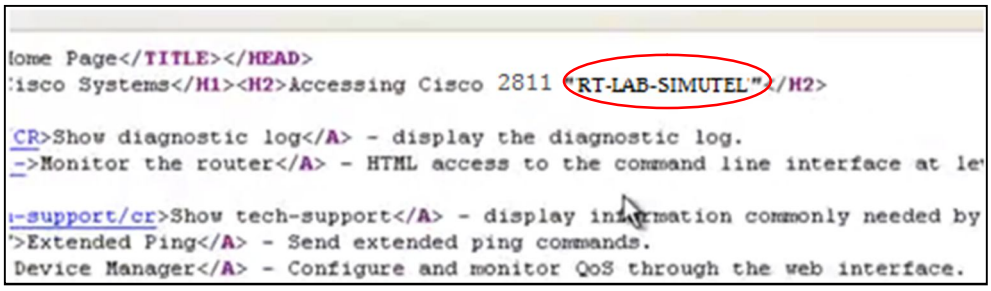


Figura 3.89- Código fuente de la página principal del router

Ingresamos al cmd (Inicio > barra de búsqueda > digitamos cmd y se da Enter) para verificar la conectividad desde la máquina atacante hacia el router con dirección IP 192.168.56.1 por medio de pings, los cuales según la figura 3.90 demuestran que hay conectividad entre ambos equipos.

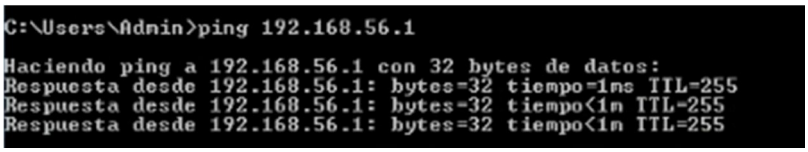


Figura 3.90- Conectividad de máquina Windows con router

Después de haber cerrado sesión en la vía web, tratamos de ingresar nuevamente pero con una contraseña distinta para verificar que no hay respuesta, ya que nos volvió a pedir la contraseña original.

Se inyecta una sentencia vía SNMP al router. Cabe destacar que cuando el router funciona como servidor HTTP, el método de autenticación por defecto es el método de password enable; si éste método se usa, el cliente se conecta al router con un nivel de privilegio de acceso 15. La línea de comando que se inserta al router modifica precisamente este nivel de acceso para establecer una nueva contraseña enable secret “nueva” para ingreso vía web. Se la ingresa a la línea de comandos como si fuera una imagen HTML. La sentencia a inyectar es la siguiente:

“Router<img/*.*/src=/level/15/configure/-/enable/secret/0/nueva>”

Ingresamos el código malicioso en la caja de texto con nombre Value, habiendo escogido el OID sysName como base para el set a realizar. Cabe mencionar que el nombre del router también fue modificado de “RT-LAB-SIMUTEL” a “ROUTER”, ya que en la línea de comandos escribimos “Router”. Al dar clic en el botón *Add*, se agrega el código que ingresamos, al OID de sysName y a continuación se da clic en el botón *Set*.

Figura 3.91- Seteo de ingreso de código malicioso

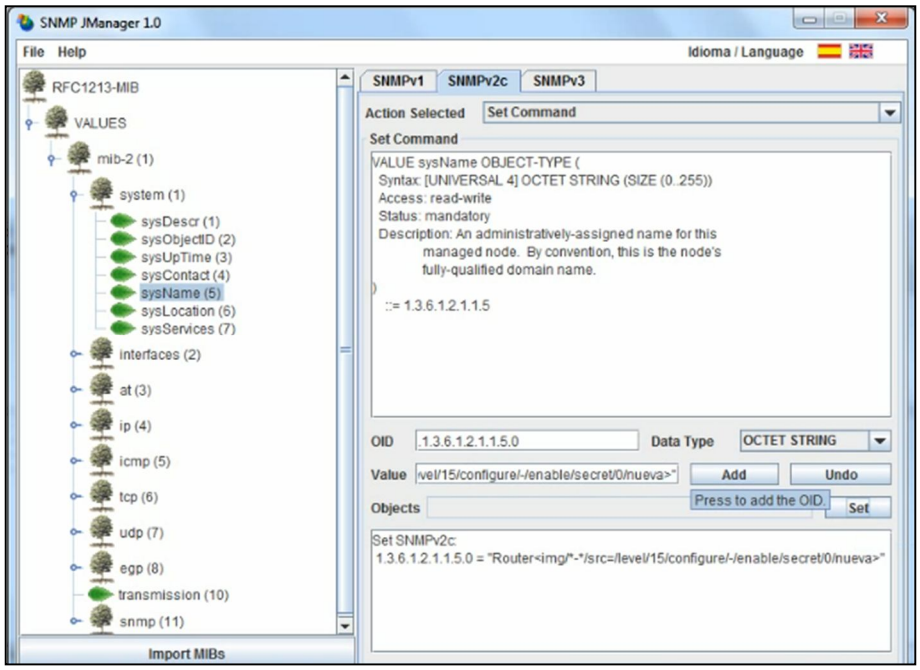


Figura 3.92- Modificación de nueva contraseña realizada

Una vez realizado el set, pudimos ver en Wireshark los paquetes SNMP enviados, en donde se encontraba el *set* realizado, pero también un trapv2 enviado desde el router hacia el NMS avisándole que ha sido configurado a través de un comando SNMP.

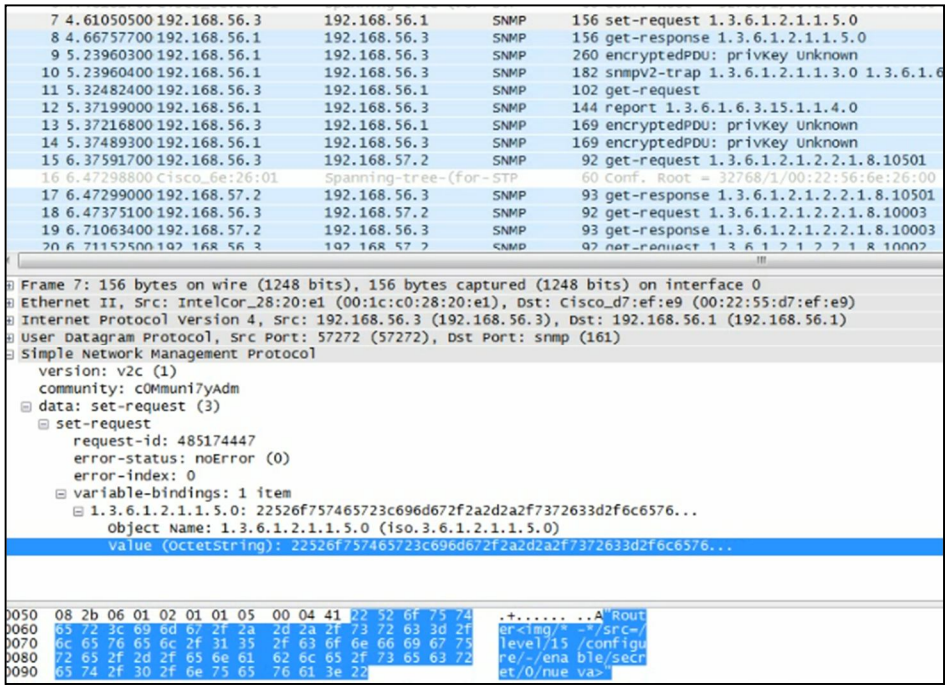


Figura 3.93- Captura en Wireshark del set de código malicioso

Verificamos la inserción de la contraseña intrusa que corresponde a “nueva”.

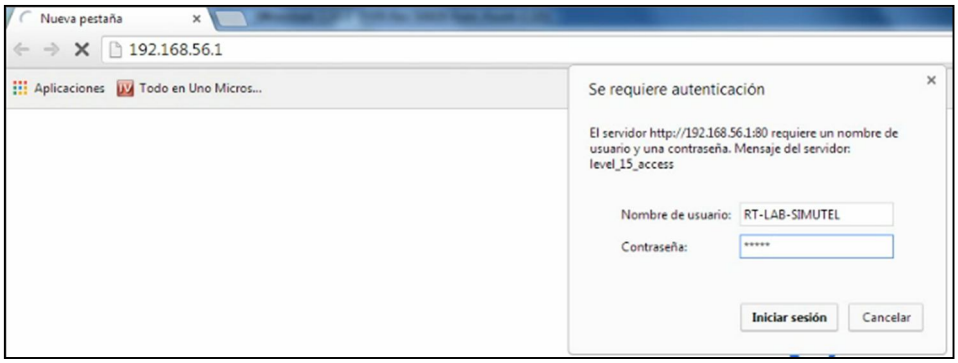


Figura 3.94- Ingreso al router vía web con contraseña intrusa

Como podemos observar en la figura 3.95, ingresamos al router sin ningún inconveniente. Al lado del nombre del router viene un pequeño recuadro que representa la “imagen” que es la que nosotros ingresamos con nuestro código malicioso.

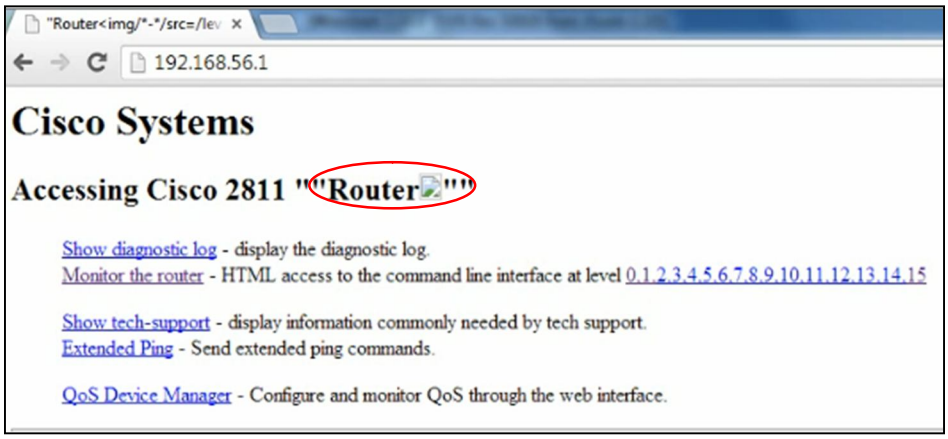


Figura 3.95- Ingreso realizado al router con contraseña intrusa

Para confirmar la inserción, se da clic derecho y se selecciona la opción de “ver código fuente” para observar que ya ha sido insertado el código malicioso.

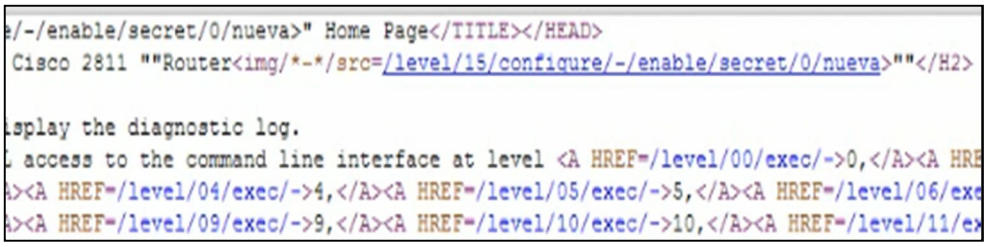


Figura 3.96- Código fuente del router con código malicioso

Como pudimos observar en la figura 3.92, para poder modificar una nueva contraseña con nuestro código malicioso, el comando *set-request* en SNMP JManager requiere del nombre de la comunidad, que en este caso, es c0Mmuni7tyAdm, nombre que fácilmente se puede obtener por medio de un programa que capture paquetes en proceso de envío, ya que los nombres de las comunidades se envían en texto plano, esto facilita a que cualquier persona modifique una nueva contraseña a nuestros dispositivos sin ningún problema, a pesar de cambiar el nombre de comunidad que viene por default y mantenerlo en secreto, por lo cual es recomendable utilizar la versión 3 de SNMP, donde la información es cifrada y autenticada, y se requiere de muchos parámetros para poder realizar un *set-request*, y así poder evitar el fácil acceso a nuestros equipos. [26]

3.7 Modelo de seguridad basado en usuarios USM SNMPv3

Como se vio en la sección 2.4.3.1.1 el motor SNMPv3 está conformado por diferentes módulos, entre ellos el subsistema de seguridad. Éste se encarga mediante mecanismos que se describirán en el desarrollo de la sección, de transformar las contraseñas ingresadas por el usuario en la aplicación del NMS, en llaves de autenticación y privacidad; y luego modificar aquellas llaves maestras en llaves localizadas acorde al engineID del motor SNMPv3. El subsistema se encarga también de almacenar en una tabla toda la información de los diferentes usuarios para que puedan hacer uso de la autenticación y cifrado de los mensajes que genera o que recibe el motor; y realiza operaciones tanto para crear nuevos usuarios a partir de otro preexistente de la tabla, como para cambiar las llaves que usará para asegurar los mensajes.

Cuando un mensaje llega a un MD, en el subsistema de seguridad se verifica que el usuario que generó el mensaje exista en la tabla de usuarios local y si usa autenticación o privacidad, someter el mensaje a los algoritmos respectivos. Una vez que se terminen estos procesos para el mensaje, éste se dirige al subsistema de control acceso.

El modelo USM es parte del subsistema de seguridad y en esta sección se analizarán los aspectos descritos en los anteriores párrafos, de la forma en que se muestra en la siguiente figura de síntesis.

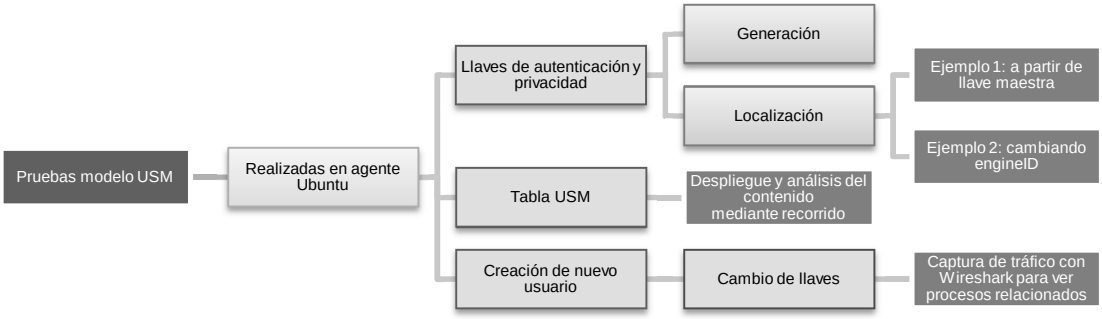


Figura 3.97- Esquema de pruebas modelo USM

3.7.1 Generación y localización de las llaves.

Cuando un usuario usa contraseñas de autenticación y/o privacidad (de mínimo 8 caracteres), el agente local las transforma en llaves de autenticación y/o privacidad mediante un algoritmo que hace uso de MD5 o SHA. Primero se forma una cadena de longitud 1’048.576 octetos al repetir el valor del password tanto como sea necesario y truncar si así se requiere al final. Esta cadena se ingresa como entrada al algoritmo hash del que hace uso el usuario para autenticar los mensajes y la salida (128 bits para MD5 o

160 bits para SHA) es la llave “maestra” o intermedia de autenticación authKey y/o privacidad privKey. [19]

Una manera de comprobar que efectivamente se generan primero las llaves maestras pero son las localizadas las que observamos en el archivo persistente de Ubuntu, es cambiar la forma en que creamos alguno de los usuarios. En la figura 3.98 se aprecia a manera de ejemplo, en el archivo persistente la definición del usuario Root pero en vez de configurarle una contraseña de autenticación, se le ingresa directamente una llave maestra obtenida de un password, el cual también se lo ingresa como password de privacidad. Adicionalmente, se cambia el engineID para comprobar si ambas llaves finales concuerdan de acuerdo al ejemplo del anexo A.3.2 del RFC3414, que se adjunta en el anexo 2 al final de este documento.

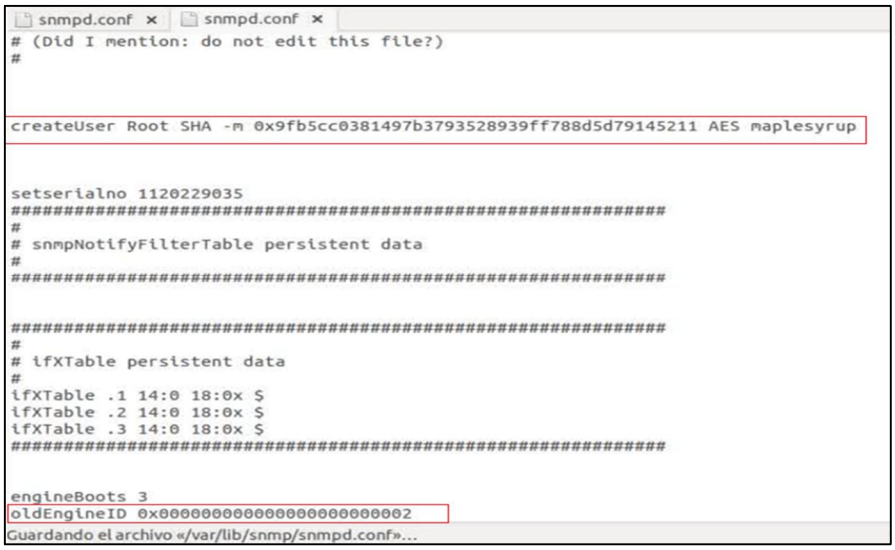


Figura 3.98- Archivo persistente - definición de llave maestra

Una vez reiniciado el servicio SNMP, vemos que efectivamente en el archivo persistente la entrada usmUser del usuario contiene las mismas llaves de autenticación y privacidad que se obtienen en el ejemplo, y que han sido localizadas por igual. Salvo que la llave de privacidad ha sido truncada a 16 octetos ya que es lo que requiere el algoritmo AES para sus procesos.

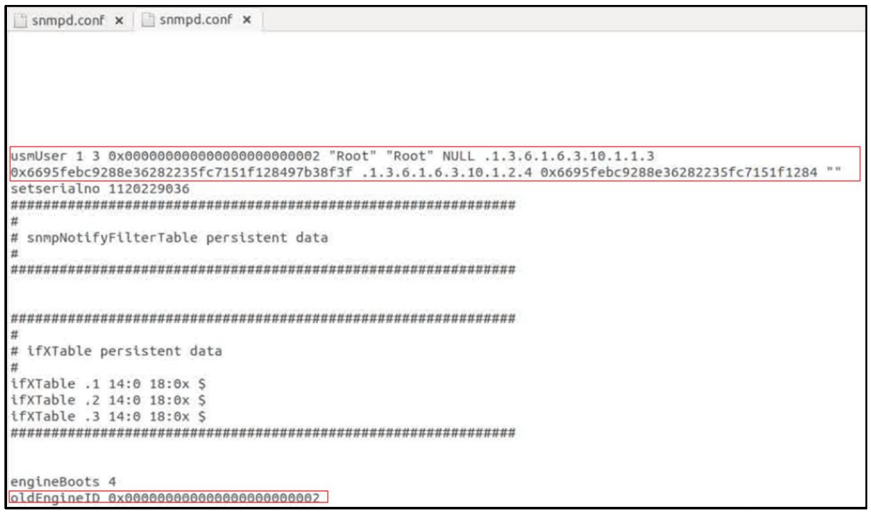


Figura 3.99- Archivo persistente - demostración de llaves iguales

Para convertir una llave maestra de usuario *Ku* a una llave localizada de usuario *Kul* en un motor autoritativo SNMP, se agrega el snmpEngineID del motor autoritativo a la llave Ku al inicio y al final de la misma para luego aplicar una función de hash que depende del algoritmo de autenticación del que hace uso ese usuario. La salida de la función de hash (128 bits para MD5 o 160 bits para SHA) es la llave localizada de usuario Kul de autenticación y/o privacidad en el motor autoritativo y es la que se puede observar en el archivo persistente; las llaves intermedias nunca se almacenan o son visibles en el nodo administrado, tampoco hay forma de recuperarlas mediante alguna operación de administración. [19]

Un motor no autoritativo, es decir el NMS que hace las consultas, posee la llave intermedia y mediante el proceso de descubrimiento, cuando por vez primera hace alguna solicitud a un motor SNMPv3, obtiene los engineID remotos. Con esos engineID calcula con el procedimiento previamente explicado, la llave localizada de cada uno de los dispositivos administrados con los que se tiene que comunicar.

A un cambio de engineID en un mismo motor SNMP o entre diferentes dispositivos (que obviamente tienen diferentes engineID), se observan diferentes llaves de autenticación y privacidad. A manera de ejemplo se crearán los mismos usuarios en el agente Ubuntu pero cambiando la forma en que el engineID es calculado en el agente. En la sección *AGENT BEHAVIOUR* de la configuración del agente Ubuntu (sección 3.2.2.1 y anexo 1), se agrega la directiva engineID Type 3, para que en el siguiente reinicio del agente el engineID se cambie en base a la MAC address de la tarjeta de red. Un engineID del tipo 1 está basado en la dirección IPv4 y tipo 2 está basado en la dirección IPv6. Luego se ingresa la directiva engineIDNic eth1 para que esta tarjeta de red sea de la que se tome la MAC address (80:00:1f:88:03:08:00:27:99:cf:8c).

Al haber hecho esto se vuelve a crear los usuarios en el archivo persistente y se reinicia el agente, para poder realizar consultas.

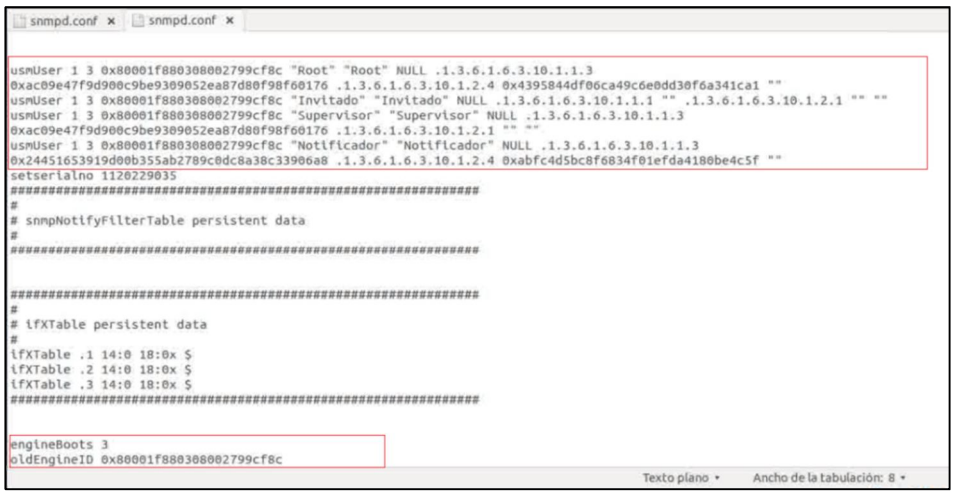


Figura 3.100- Creación de usuarios en archivo persistente

Como se ve en la figura 3.100, cada usuario configurado tiene diferentes llaves de autenticación y privacidad a las que se tuvieron en la configuración del agente de la sección 3.2.2.1.

3.7.2 Tabla de usuarios.

La tabla de usuarios usmUserTable (1.3.6.1.6.3.15.1.2.2) se encuentra dentro de la base de datos de configuración local (LCD) del motor SNMP. [19]

Cuando se genera un mensaje SNMPv3 saliente, basado en el nombre de usuario indicado en el campo msgUserName, se extrae información concerniente al mismo de la tabla USM, que sirve para los procesos de seguridad del mensaje ya sean de autenticación y/o privacidad.

Cuando un mensaje llega a un motor SNMP, se extraen sus parámetros de seguridad, entre otros, el nivel de seguridad; luego se extrae información del usuario indicado en el campo msgUserName de la tabla de usuarios. Si la información de seguridad del mensaje concuerda con la del usuario almacenado en la tabla, se procede a los procesos de autenticación y/o privacidad. [19]

Con la aplicación SNMP JManager se hará un recorrido por la tabla USM del agente Ubuntu para comprobar que la tabla se actualiza con la configuración de los usuarios que se hizo en la sección 3.2.2.1.

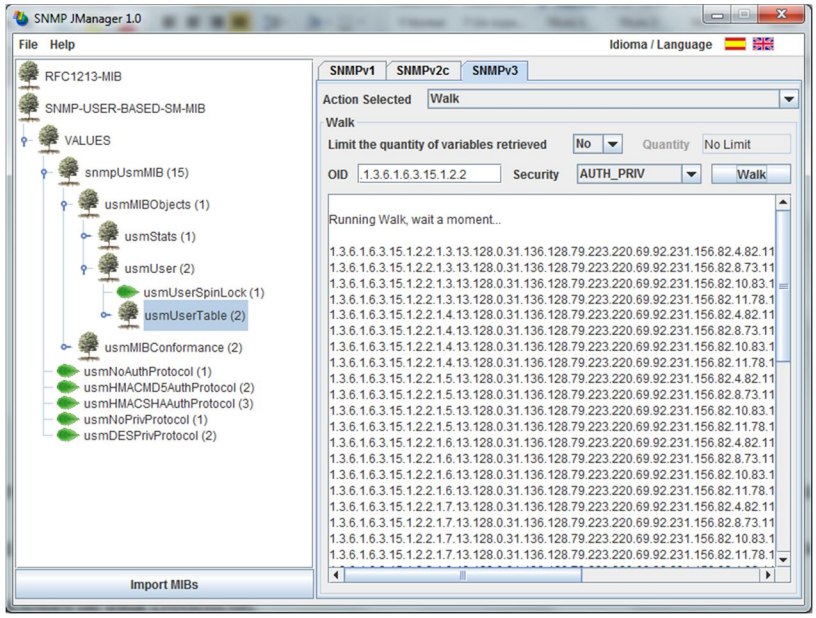


Figura 3.101- Recorrido de tabla USM

La tabla usmUserTable tiene diferentes entradas u objetos. Cuando se configura un usuario en el agente, se crea una fila para ese usuario en la tabla y tiene indexada el usmUserEngineID y el usmUserName; además dicha fila contiene varias columnas correspondientes a cada una de los objetos existentes en la tabla usmUserTable.

Cabe destacar que antes de las indexaciones del engineID y el userName en esta tabla; y antes de cualquier indexación de objetos en otras tablas (incluidas las de vacmMIBObjects de la sección 3.8.1), va indexado el

número de octetos que ocupa la siguiente indexación, lo que se cumple para toda indexación de objetos en cadena de octetos. [32]

Para todas las entradas de la tabla obtenidas, se observan las indexaciones:

- El engineID del motor SNMP que es el mismo para todos los usuarios en formato decimal. Como se vio en la configuración del agente Ubuntu, en el archivo persistente el engineID tenía el valor 0x80001f88804fdfdc455ce79c52 que en formato decimal aparece de la forma 128.0.31.136.128.79.223.220.69.92.231.156.82 (notación de OID, separado por puntos). Le antecede el número de octetos que ésta ocupa (13).
- El nombre de usuario en formato decimal (notación de OID, separado por puntos). ‘Root’ es 82.111.111.116, ‘Invitado’ es 73.110.118.105.116.97.100.111, ‘Supervisor’ es 83.117.112.101.114.118.105.115.111.114 y ‘Notificador’ equivale a 78.111.116.105.102.105.99.97.100.111.114. Antes de cada uno de ellos, se indexa el número de octetos que ocupan, para ‘Root’ 4, para ‘Invitado’ 8, para ‘Supervisor’ 10 y para ‘Notificador’ 11.

A continuación se detallan las entradas más importantes que se obtuvieron al hacer el walk de la tabla de usuarios para el agente.

- ***usmUserSecurityName 1.3.6.1.6.3.15.1.2.2.1.3***

  Objeto que describe en formato legible al usuario el nombre de seguridad. El nombre de seguridad junto con el engineID, identifica una fila en la tabla de usuarios que será usada para asegurar el mensaje. El securityName tiene un formato que es independiente del modelo de seguridad, pero cuando se usa el modelo USM, el securityName es el mismo que el userName. [19]

  Se observa que hay 4 instancias de este objeto debido a que se configuraron 4 usuarios distintos en el agente Ubuntu. Efectivamente, hay cuatro nombres de seguridad: ‘Root’, ‘Invitado’, ‘Supervisor’ y ‘Notificador’.

OID de la instancia	Valor
1.3.6.1.6.3.15.1.2.2.1.3.13.128.0.31.136.128.79.223.220.69.92.231.156.82.4.82.111.111.116	Root
1.3.6.1.6.3.15.1.2.2.1.3.13.128.0.31.136.128.79.223.220.69.92.231.156.82.8.73.110.118.105.116.97.100.111	Invitado
1.3.6.1.6.3.15.1.2.2.1.3.13.128.0.31.136.128.79.223.220.69.92.231.156.82.10.83.117.112.101.114.118.105.115.111.114	Supervisor
1.3.6.1.6.3.15.1.2.2.1.3.13.128.0.31.136.128.79.223.220.69.92.231.156.82.11.78.111.116.105.102.105.99.97.100.111.114	Notificador

Tabla 3.1- Instancias del objeto usmUserSecurityName

- ***usmUserAuthProtocol 1.3.6.1.6.3.15.1.2.2.1.5***

  Este objeto describe para determinado usuario si los mensajes que genere serán o no autenticados, y si lo son, qué protocolo se usa. Para los usuario ‘Root’, ‘Supervisor’ y ‘Notificador’ se autenticarán los mensajes usando el algoritmo SHA (usmHMACSHAAuthProtocol, 1.3.6.1.6.3.10.1.1.3) mientras que ‘Invitado’ no autentica sus mensajes (usmNoAuthProtocol, 1.3.6.1.6.3.10.1.1.1).

OID de la instancia	Valor
1.3.6.1.6.3.15.1.2.2.1.5.13.128.0.31.136.128.79.223.220.69.92.231.156.82.4.82.111.111.116	1.3.6.1.6.3.10.1.1.3
1.3.6.1.6.3.15.1.2.2.1.5.13.128.0.31.136.128.79.223.220.69.92.231.156.82.8.73.110.118.105.116.97.100.111	1.3.6.1.6.3.10.1.1.1
1.3.6.1.6.3.15.1.2.2.1.5.13.128.0.31.136.128.79.223.220.69.92.231.156.82.10.83.117.112.101.114.118.105.115.111.114	1.3.6.1.6.3.10.1.1.3
1.3.6.1.6.3.15.1.2.2.1.5.13.128.0.31.136.128.79.223.220.69.92.231.156.82.11.78.111.116.105.102.105.99.97.100.111.114	1.3.6.1.6.3.10.1.1.3

Tabla 3.2- Instancias del objeto usmUserAuthProtocol

- ***usmUserPrivProtocol 1.3.6.1.6.3.15.1.2.2.1.8***

Este objeto describe para determinado usuario si los mensajes que genere harán uso o no de privacidad, y si lo hacen, qué protocolo se usa. Para los usuario Root y Notificador se cifrarán los mensajes usando el algoritmo AES (usmAesCfb128Protocol, 1.3.6.1.6.3.10.1.2.4) mientras que Supervisor e Invitado no cifran sus mensajes (usmNoPrivProtocol, 1.3.6.1.6.3.10.1.2.1).

OID de la instancia	Valor
1.3.6.1.6.3.15.1.2.2.1.8.13.128.0.31.136.128.79.223.220.69.92.231.156.82.4.82.111.111.116	1.3.6.1.6.3.10.1.2.4
1.3.6.1.6.3.15.1.2.2.1.8.13.128.0.31.136.128.79.223.220.69.92.231.156.82.8.73.110.118.105.116.97.100.111	1.3.6.1.6.3.10.1.2.1
1.3.6.1.6.3.15.1.2.2.1.8.13.128.0.31.136.128.79.223.220.69.92.231.156.82.10.83.117.112.101.114.118.105.115.111.114	1.3.6.1.6.3.10.1.2.1
1.3.6.1.6.3.15.1.2.2.1.8.13.128.0.31.136.128.79.223.220.69.92.231.156.82.11.78.111.116.105.102.105.99.97.100.111.114	1.3.6.1.6.3.10.1.2.4

Tabla 3.3- Instancias del objeto usmUserPrivProtocol

- ***usmUserStorageType 1.3.6.1.6.3.15.1.2.2.1.12***

Describe cómo se maneja la fila en la memoria. Una fila que es volatile(2) se pierde cuando el agente reinicia. Una fila que es nonVolatile(3), permanent(4) o readOnly(5) está respaldada por un almacenamiento estable. Una fila permanente puede ser cambiada pero no borrada y una que es de sólo lectura no puede ser cambiada ni borrada [16]. Toda la información de los usuarios está almacenada en una memoria no volátil como se observa.

OID de la instancia	Valor
1.3.6.1.6.3.15.1.2.2.1.12.13.128.0.31.136.128.79.223.220.69.92.231.156.82.4.82.111.111.116	3
1.3.6.1.6.3.15.1.2.2.1.12.13.128.0.31.136.128.79.223.220.69.92.231.156.82.8.73.110.118.105.116.97.100.111	3
1.3.6.1.6.3.15.1.2.2.1.12.13.128.0.31.136.128.79.223.220.69.92.231.156.82.10.83.117.112.101.114.118.105.115.111.114	3
1.3.6.1.6.3.15.1.2.2.1.12.13.128.0.31.136.128.79.223.220.69.92.231.156.82.11.78.111.116.105.102.105.99.97.100.111.114	3

Tabla 3.4- Instancias del objeto usmUserStorageType

- ***usmUserStatus 1.3.6.1.6.3.15.1.2.2.1.13***

Este objeto maneja la creación y borrado de filas. Tiene 5 posibles valores: ‘active(1)’ que indica que la fila está lista para el uso del agente, ‘notInService(2)’ que indica la existencia de la fila en el agente pero que no está disponible para operaciones de administración, ‘notReady(3)’ que indica la existencia de la fila en el agente pero que falta definir información en alguna columna para poderla hacer activa, ‘createAndGo(4)’ es seteado por un NMS que desea crear una nueva fila de usuario y la hace activa para el inmediato uso del agente administrado, ‘createAndWait(5)’ es seteado por un agente administrativo que desea crear una nueva fila de usuario pero no la hace activa todavía y

‘destroy(6)’ es seteado por un NMS que desea borrar todas las instancias o columnas asociadas a una fila de usuario. [16]

Todos los usuarios se encuentran operativos en el agente para operaciones de administraciones locales o remotas.

OID de la instancia	Valor
1.3.6.1.6.3.15.1.2.2.1.12.13.128.0.31.136.128.79.223.220.69.92.231.156.82.4.82.111.111.116	1
1.3.6.1.6.3.15.1.2.2.1.12.13.128.0.31.136.128.79.223.220.69.92.231.156.82.8.73.110.118.105.116.97.100.111	1
1.3.6.1.6.3.15.1.2.2.1.12.13.128.0.31.136.128.79.223.220.69.92.231.156.82.10.83.117.112.101.114.118.105.115.111.114	1
1.3.6.1.6.3.15.1.2.2.1.12.13.128.0.31.136.128.79.223.220.69.92.231.156.82.11.78.111.116.105.102.105.99.97.100.111.114	1

Tabla 3.5- Instancias del objeto usmUserStatus

3.7.3 Clonación de usuarios y cambio de llaves

El modelo USM de SNMPv3 también permite la creación de usuarios mediante operaciones administrativas en el agente local o uno remoto, es decir emitiendo comandos. Generalmente a la creación de un usuario se la conoce también como clonación debido a que el nuevo usuario toma las

configuraciones de seguridad a partir de otro preestablecido que ya existe en el agente, en nuestro caso cualquiera de los que ya se encuentran activos y aparecen en el archivo persistente. [19]

La forma en que el programa Net-SNMP crea un usuario es mediante la siguiente instrucción:

snmpusm *+ [opciones comunes de snmpv3] + host +* ***create*** *+ nuevo usuario + usuario del que se clonan parámetros.*

Las opciones comunes de SNMPv3 abarcan al usuario que hace la solicitud, protocolo y contraseña de autenticación y/o protocolo y contraseña de privacidad.

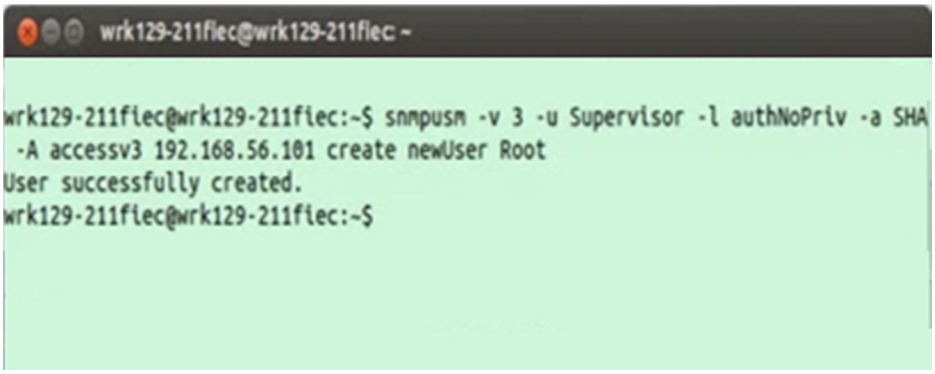


Figura 3.102- Clonación de usuario por medio del terminal

La figura 3.102 muestra la ejecución del comando en terminal. Es necesario reiniciar el servicio SNMP para que se persistan los datos del nuevo usuario,

como se puede apreciar en la figura 3.103, tiene las mismas llaves de autenticación y privacidad que el usuario Root del que fue clonado.

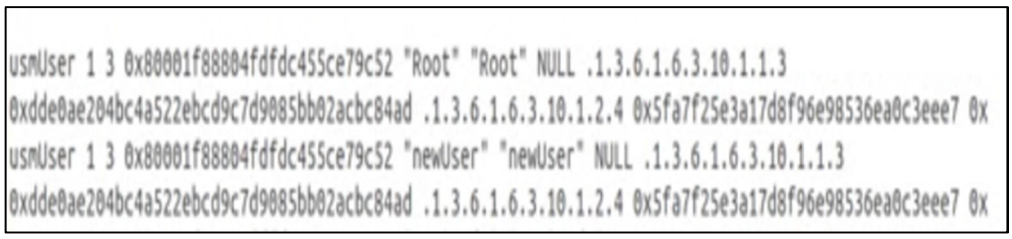


Figura 3.103- Llaves de usuario clonado

En la figura 3.104 se muestran los paquetes capturados en Wireshark de la operación de creación del usuario newUser (números 3 y 4).

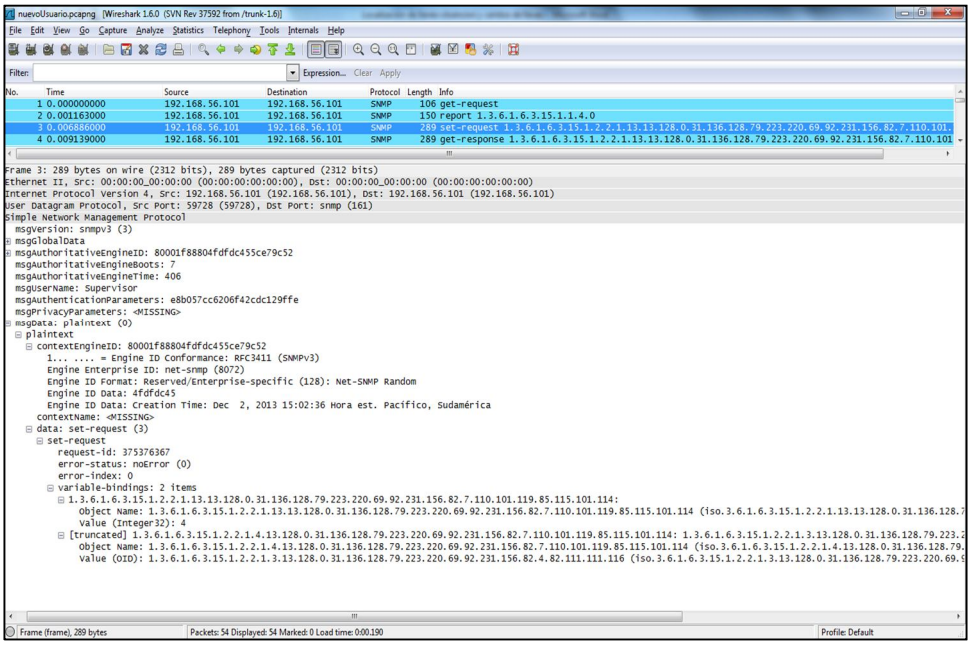


Figura 3.104- Captura de paquetes de clonación de usuario

Básicamente se hace un set-request que contiene las siguientes variables:

- usmUserStatus con el valor 4 para que se cree una nueva fila en la tabla de usuarios y que se encuentre operativa. Como se vio en la sección 3.7.2 toda entrada de la tabla, como esta nueva que se ingresa, tiene indexado el engineID (el mismo del agente Ubuntu de la sección 3.7.2) y el nombre de usuario newUser, que para este caso es el nuevo que se crea, en formato decimal (110.101.119.85.115.101.114).
- El objeto usmUserCloneFrom que es un puntero hacia otra fila de la tabla, indicando un usuario del cual se tienen que copiar las configuraciones de seguridad de protocolos y contraseñas.

  La variable vinculada a este objeto es usmUserSecurityName que tiene indexado al final el nombre de usuario “Root” (82.111.111.116), copiándose sus parámetros de seguridad al usuario “newUser”.

El get-response con el mismo contenido del set-request indica que la solicitud de escritura fue exitosa.

Vinculado a un proceso de creación de usuario generalmente va el proceso keyChange de cambio de contraseñas (y por ende, de clave). Los objetos authKeyChange y authOwnKeyChange de la tabla usmUserTable permiten cambiar local o remotamente la llave de autenticación localizada para algún usuario en particular, de una forma segura. De la misma forma los objetos privKeyChange y privOwnKeyChange de la tabla usmUserTable lo hacen con las llaves de privacidad localizadas. Todos estos objetos de cambio de llaves de autenticación y cifrado retornan el objeto 0.0 a solicitudes de lectura. [19]

Un proceso keyChange implica una serie de pasos, entre los que se encuentra la escritura en los objetos recientemente descritos de un valor generado por operaciones administrativas, causando que la llave especificada cambie. El valor se compone de dos partes concatenadas, una aleatoria y otra llamada delta, ambas de la misma longitud (en nuestro caso 16 octetos si la llave a cambiar es AES o 20 octetos si la llave a cambiar es SHA) [19]. La ventaja de este método es que no se transmite en texto plano el nuevo password ni la nueva llave, sino un valor totalmente diferente generado internamente al agente por el proceso keyChange que solamente la entidad receptora (local o remota) puede procesar para generar internamente la llave que se originó en el lado emisor del comando.

La forma en que el programa Net-SNMP realiza un cambio de llave es mediante la siguiente instrucción:

snmpusm *+ [opciones comunes de snmpv3] + protocolo de cifrado + host +* *Ca|Cx +* ***passwd*** *+ antiguo password + nuevo password + usuario destino del cambio*

Las opciones comunes de SNMPv3 abarcan al usuario que hace la solicitud, protocolo y contraseña de autenticación y/o protocolo y contraseña de privacidad. Ca especifica que se debe cambiar la llave de autenticación y Cx la llave de privacidad.

En la figura 3.105 se muestran los comandos para cambiar primero la llave de autenticación y luego la de privacidad, para esta última es necesario especificar el protocolo de privacidad en las opciones comunes de SNMPv3 para poder generarla, a pesar que la operación la hacemos con un usuario de nivel de seguridad authNoPriv (Supervisor). Al final de los cambios se reinicia el agente para poder ver los cambios en el archivo persistente (figura 3.106), en donde se verifica que el usuario newUser tiene llaves diferentes a las del usuario Root.

Figura 3.105- Cambio de llave de autenticación y privacidad

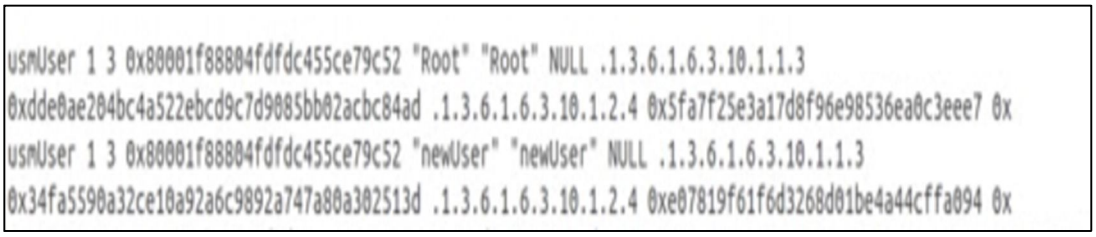


Figura 3.106- Verificación de llaves diferentes de usuarios

A continuación, en la figura 3.107 se muestran los paquetes capturados en Wireshark de la operación de cambio de llave de autenticación (números 17 y 18).

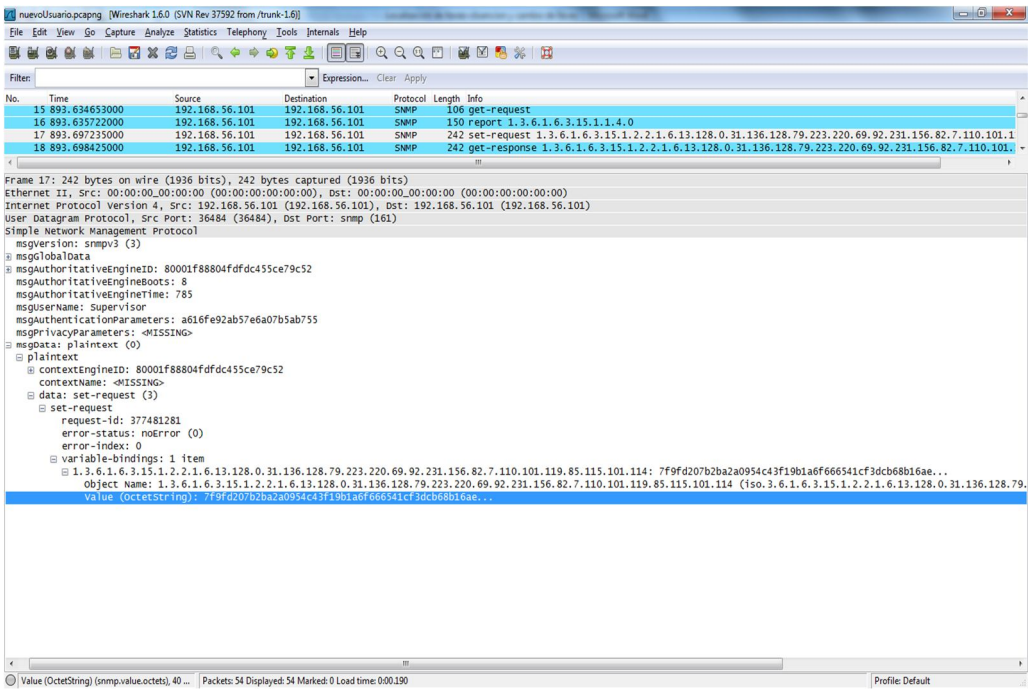


Figura 3.107- Captura de cambio de llave de autenticación

Básicamente se hace un set-request que contiene la siguiente variable:

- usmUserAuthKeyChange al que se le envía una cadena de 40 octetos obtenidos mediante el proceso de keyChange para que la entidad local genera la nueva llave de autenticación.

El get-response con el mismo contenido del set-request indica que la solicitud de escritura fue exitosa.

En la figura 3.108 se muestran los paquetes capturados en Wireshark de la operación de cambio de llave de privacidad (números 21 y 22).

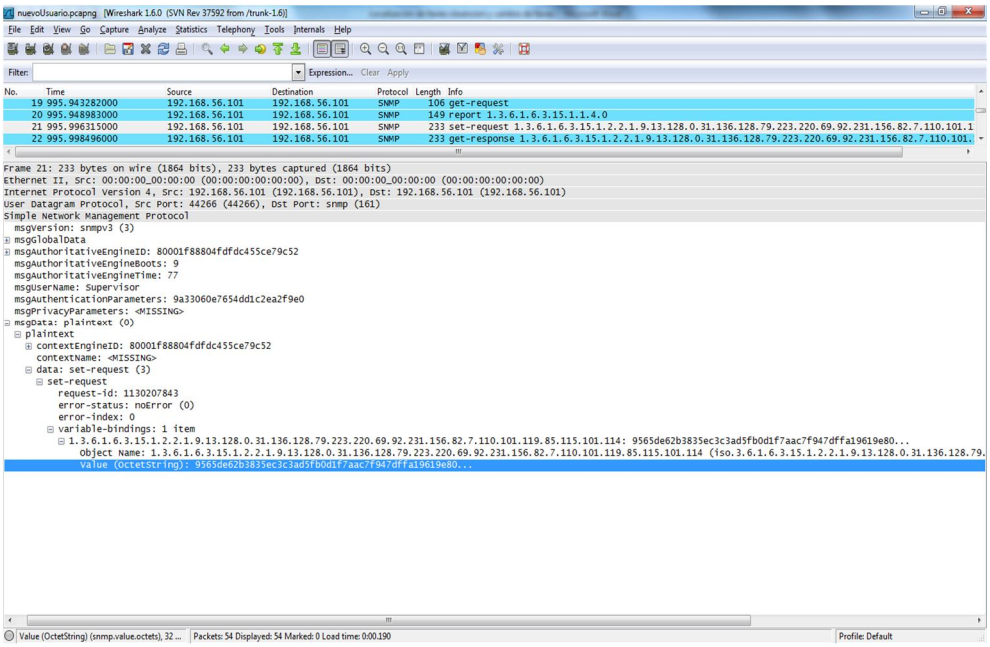


Figura 3.108- Captura de cambio de llave de privacidad

Básicamente se hace un set-request que contiene la siguiente variable:

- usmUserPrivKeyChange al que se le envía una cadena de 32 octetos obtenidos mediante el proceso de keyChange para que la entidad local genera la nueva llave de privacidad.

El get-response con el mismo contenido del set-request indica que la solicitud de escritura fue exitosa.

Es importante destacar que al crear un nuevo usuario a partir de otro, este en teoría está habilitado para operaciones de administración, sin embargo al hacer solicitudes para ese usuario y con las nuevas contraseñas (y llaves obviamente) obtuvimos respuestas nulas. Esto es porque el usuario no tiene definidos permisos de acceso a la MIB, los cuales se configuran como se muestra en la figura 3.109. Primero se creó un grupo mediante la directiva createSec2Group, luego se definió una vista con la directiva createView y finalmente se vincula al grupo con la vista usando el comando createAccess.

Figura 3.109- Configuración de permisos de acceso a MIB

Definidas las políticas de acceso se reinicia el agente y se observa que se agregaron al archivo persistente en la figura 3.110. A partir de ahí las solicitudes realizadas para el nuevo usuario son exitosas.

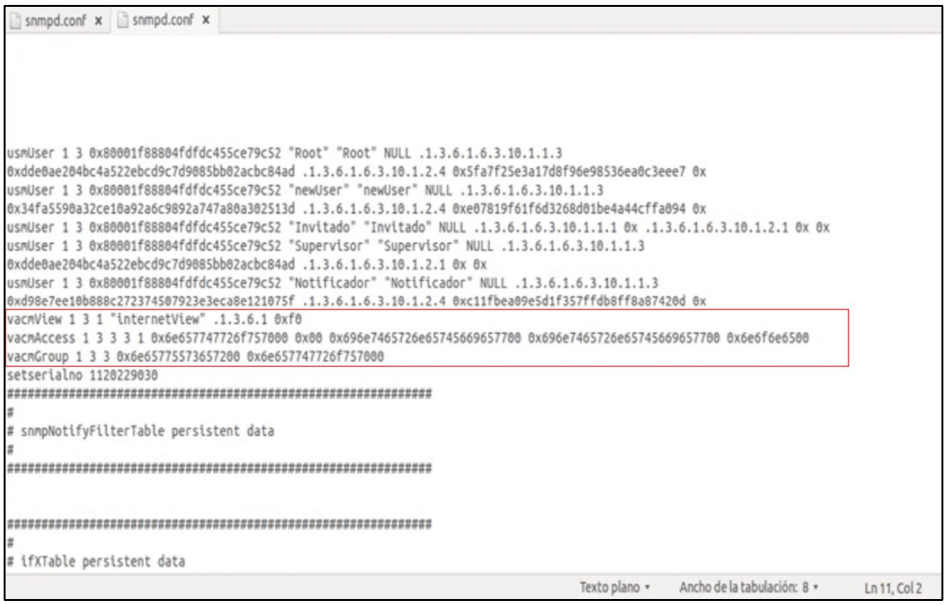


Figura 3.110- Archivo persistente con nuevas políticas de acceso

3.8 Modelo de seguridad basado en vistas VACM SNMPv3

Como se vio en la sección 2.4.3.1.1 el motor SNMPv3 se encuentra conformado por módulos entre ellos el subsistema de control de acceso, en el cual funciona el modelo VACM que maneja varias tablas que en conjunto, definen los permisos que tienen los usuarios para acceder a los objetos que se encuentran en la MIB. Una parte importante de VACM son los niveles de seguridad de los mensajes que son generados o recibidos por el motor y que definen si usan o no autenticación o privacidad; y las vistas, que agrupan varios objetos de la MIB, facilitando la tarea de permitir o no el acceso para todo el conjunto, a una solicitud entrante.

Cuando un mensaje llega al subsistema de control de acceso luego de haber sido sometido a la autenticación y/o cifrado, se analiza si el mensaje tiene los permisos para leer o escribir en una o varias ramas de la MIB y en base a eso, se genera el mensaje de respuesta que será enviado al NMS. En esta sección se analizarán los aspectos recientemente descritos, de la forma en que se muestra en la siguiente figura de síntesis

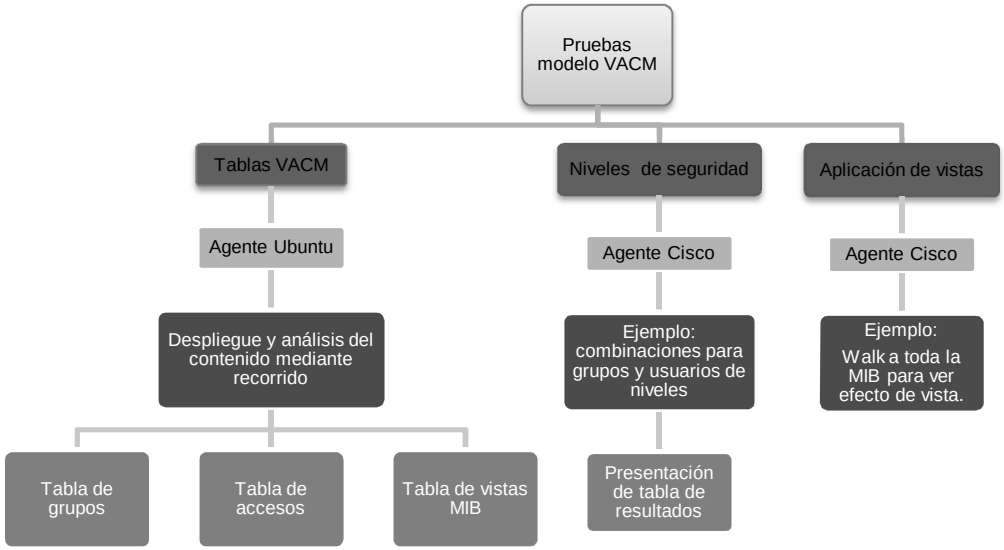


Figura 3.111- Esquema de pruebas modelo VACM

3.8.1 Tablas de control de acceso

Cuando se crea algún usuario dentro del agente, es necesario configurar el control de acceso a las variables de la MIB local definiendo qué ramas serán

visibles y qué operaciones se podrán realizar en ellas. Es importante señalar también que la información de la tabla usmUserTable debe ser solamente manipulada por usuarios administradores de red, ya que contiene información sensible de los usuarios como protocolos de autenticación y privacidad y además al setear valores dentro de cierto objeto de la tabla, es posible cambiar las llaves; de tal forma que hay que restringir el acceso a aquellos objetos mediante el uso de vistas y permisos de lectura/escritura dados a los usuarios.

Con la aplicación SNMP Jmanager se hará un recorrido por las diferentes tablas VACM del agente Ubuntu para comprobar que la tabla se actualiza con la configuración de las vistas conjuntas a los usuarios que se hizo en la sección 3.2.2.1.

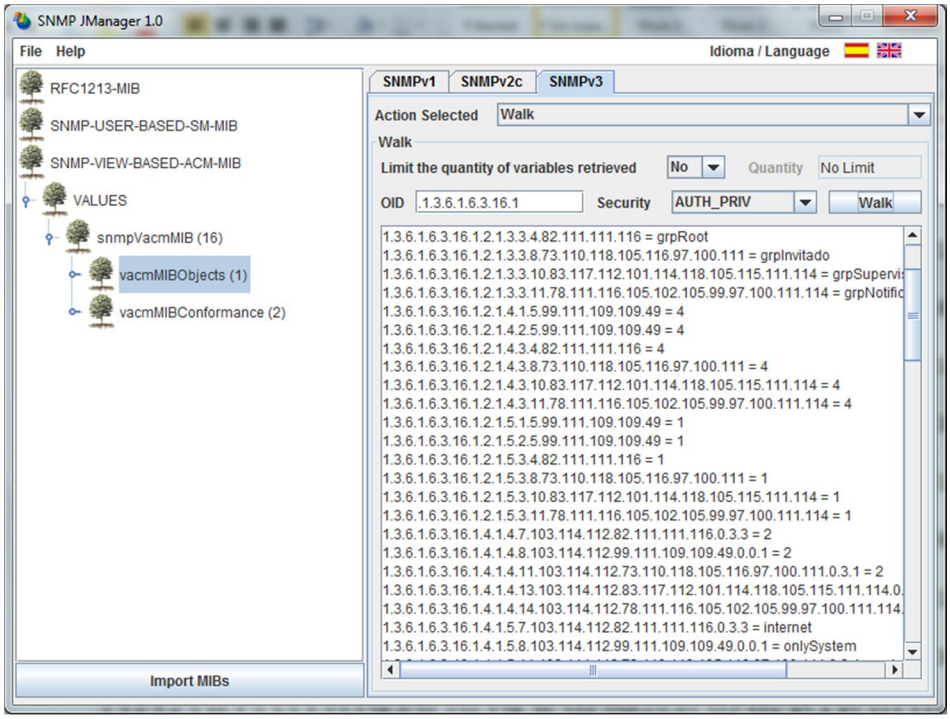


Figura 3.112- Recorrido tablas VACM

Hay cuatro tablas dentro de la MIB de VACM (vacmMIBObjects, 1.3.6.1.6.3.16.1): tabla de contextos vacmContextTable, tabla de grupos vacmSecurityToGroupTable, tabla de accesos vacmAccessTable y tabla de vistas vacmMIBViews.

- **Tabla vacmSecurityToGroupTable 1.3.6.1.6.3.16.1.2**

Esta tabla mapea una combinación de modelo de seguridad y nombre de seguridad en un nombre de grupo el cual es usado para definir una política de control de acceso para un grupo de usuarios [17].

Para todas las entradas de la tabla obtenidas, se observan las indexaciones:

- El securityModel que es el mismo para todos los usuarios configurados, cuyo valor es 3, debido a que todos usan el modelo USM de la versión 3. Esta indexación viene inmediatamente luego del identificador de objeto de la forma 1.3.6.1.6.3.16.1.2.1.x.
- El securityName en formato decimal (notación de OID, separado por puntos). Los valores son los mismos que se vieron en la sección 3.7.2. Esta indexación viene luego del índice del securityModel y el entero que indica la cantidad de octetos que tiene el securityName.

Dentro de esta tabla hay diferentes objetos, se detallan los más importantes a continuación:

- ***vacmGroupName 1.3.6.1.6.3.16.1.2.1.3***

Este objeto describe el nombre del grupo a la cual cada entrada pertenece. Hay 4 instancias diferentes para este objeto que describen 4 grupos diferentes: grpRoot, grpInvitado, grpSupervisor y grpNotificador, los cuales fueron creados automáticamente por el agente para cada usuario, debido a que las directivas de creación de usuarios *rouser* y *rwuser* solamente asocian a cada usuario vistas.

OID de la instancia	Valor
1.3.6.1.6.3.16.1.2.1.3.3.4.82.111.111.116	grpRoot
1.3.6.1.6.3.16.1.2.1.3.3.8.73.110.118.105.116.97.100.111	grpInvitado
1.3.6.1.6.3.16.1.2.1.3.3.10.83.117.112.101.114.118.105.115.111.114	grpSupervisor
1.3.6.1.6.3.16.1.2.1.3.3.11.78.111.116.105.102.105.99.97.100.111.114	grpNotificador

Tabla 3.6- Instancias del objeto vacmGroupName

Los objetos ***vacmSecurityToGroupStorageType 1.3.6.1.6.3.16.1.2.1.4*** y ***vacmSecurityToGroupStatus 1.3.6.1.6.3.16.1.2.1.5*** con sus diferentes instancias cada uno se definen de la misma forma que los objetos *usmUserStorageType* y *usmUserStatus* vistos en la sección 3.7.2 y tienen los mismos valores.

- **Tabla vacmAccessTable 1.3.6.1.6.3.16.1.4**

  Esta tabla define derechos de acceso para los grupos configurados. En esta tabla específicamente es donde se produce la “vinculación” de las vistas y permisos de lectura, escritura o notificaciones con los usuarios; ya que los usuarios están mapeados en grupos como se vio en la tabla vacmSecurityToGroupTable [17].

  Para todas las entradas de la tabla obtenidas, se observan las indexaciones:

  - El groupName, que define junto a las indexaciones restantes, una instancia específica para cualquiera de los objetos de esta tabla. Puede ser cualquiera de los grupos del objeto *vacmGroupName*, escrito en notación de OID decimal: grpRoot (103.114.112.82.111.111.116), grpInvitado (103.114.112.73.110.118.105.116.97.100.111), grpSupervisor (103.114.112.83.117.112.101.114.118.105.115.111.114) y grpNotificador (103.114.112.78.111.116.105.102.105.99.97.100.111.114).

    Le antecede un entero indicando su longitud en octetos.

- El context, define a cual contexto pertenecen las entradas de esta tabla. Como no fue definido contexto en la creación de usuarios, todas las entradas tienen un ‘0’.
- El securityModel es 3 para todas las entradas de la tabla, debido a que los usuarios configurados, usan el modelo USM de la versión 3.
- El securityLevel define, dependiendo del usuario, el nivel de seguridad relacionado a él. Si el usuario es ‘Root’ o ‘Notificador’ cuyo nivel de seguridad es authPriv, le corresponde el índice 3. Si es ‘Supervisor’ cuyo nivel de seguridad es authNoPriv, le corresponde el índice 2. Finalmente, si es un usuario de nivel se seguridad noAuthNoPriv le corresponde el índice 1.

Dentro de esta tabla hay diferentes objetos, se detallan los más importantes a continuación:

- ***vacmAccessReadViewName 1.3.6.1.6.3.16.1.4.1.5***

  Este objeto describe el nombre de la vista de lectura vinculada a un grupo (por ende, al usuario). El usuario ‘Root’ puede leer todo el contenido de la vista internet; ‘Invitado’ el contenido de la vista onlySystem, ‘Supervisor’ el contenido de la vista onlyMib2 y ‘Notificador’ el contenido de la vista internet.

OID de la instancia	Valor
1.3.6.1.6.3.16.1.4.1.5.7.103.114.112.82.111.111.116.0.3.3	internet
1.3.6.1.6.3.16.1.4.1.5.11.103.114.112.73.110.118.105.116.97.100.111.0.3.1	onlySystem
1.3.6.1.6.3.16.1.4.1.5.13.103.114.112.83.117.112.101.114.118.105.115.111.114.0.3.2	onlyMib2
1.3.6.1.6.3.16.1.4.1.5.14.103.114.112.78.111.116.105.102.105.99.97.100.111.114.0.3.3	internet

Tabla 3.7- Instancias del objeto vacmAccessReadViewName

- ***vacmAccessWriteViewName 1.3.6.1.6.3.16.1.4.1.6***

  Este objeto describe el nombre de la vista de escritura vinculada a un grupo (por ende, al usuario). Los usuarios ‘Root’ y ‘Notificador’ tienen permisos de escritura en todo el contenido de la vista internet, pero ‘Invitado’ y ‘Supervisor’ no tienen permisos de escritura en ninguna vista.

OID de la instancia	Valor
1.3.6.1.6.3.16.1.4.1.5.7.103.114.112.82.111.111.116.0.3.3	internet
1.3.6.1.6.3.16.1.4.1.5.11.103.114.112.73.110.118.105.116.97.100.111.0.3.1	none
1.3.6.1.6.3.16.1.4.1.5.13.103.114.112.83.117.112.101.114.118.105.115.111.114.0.3.2	none
1.3.6.1.6.3.16.1.4.1.5.14.103.114.112.78.111.116.105.102.105.99.97.100.111.114.0.3.3	internet

Tabla 3.8- Instancias del objeto vacmAccessWriteViewName

- ***vacmAccessNotifyViewName 1.3.6.1.6.3.16.1.4.1.7***

  Este objeto describe el nombre de la vista de notificación vinculada a un grupo (por ende, al usuario). Los usuarios ‘Root’ y ‘Notificador’ tienen permisos de enviar traps en todo el contenido de la vista internet, pero ‘Invitado’ y ‘Supervisor’ no tienen permisos de enviar traps en ninguna vista.

  Algo importante de aclarar es que con la configuraciones de usuarios del agente (sección 3.2.2.1) mediante el comando *rwuser* se definen vistas de lectura, escritura y de notificación, como se puede comprobar para los usuarios Root y Notificador en el primer y último objeto respectivamente; sin embargo, se generan los traps con el comando *trapsess* para todo el árbol de internet con el usuario Notificador, definido exclusivamente para envío de traps (ya que si un usuario esta

configurado en el comando *trapsess,* ya no está disponible para operaciones de lectura/escritura).

OID de la instancia	Valor
1.3.6.1.6.3.16.1.4.1.5.7.103.114.112.82.111.111.116.0.3.3	internet
1.3.6.1.6.3.16.1.4.1.5.11.103.114.112.73.110.118.105.116.97.100.111.0.3.1	none
1.3.6.1.6.3.16.1.4.1.5.13.103.114.112.83.117.112.101.114.118.105.115.111.114.0.3.2	none
1.3.6.1.6.3.16.1.4.1.5.14.103.114.112.78.111.116.105.102.105.99.97.100.111.114.0.3.3	internet

Tabla 3.9- Instancias del objeto vacmAccessNotifyViewName

Los objetos ***vacmAccessStorageType 1.3.6.1.6.3.16.1.4.1.8*** y ***vacmAccessStatus 1.3.6.1.6.3.16.1.4.1.9*** con sus diferentes instancias cada uno se definen de la misma forma que los objetos *usmUserStorageType* y *usmUserStatus* vistos en la sección 3.7.2 y tienen los mismos valores.

- **Tabla vacmViewTreeFamilyTable 1.3.6.1.6.3.16.1.5.2**

  Esta tabla especifica qué ramas de la MIB se encuentran incluidas o excluidas dentro de cada vista [17].

Para todas las entradas de la tabla obtenidas, se observan las indexaciones:

- ViewTreeFamilyViewName que es el nombre de la vista. Puede ser cualquiera de las vistas del objeto vacmAccessReadViewName escrito en notación de OID decimal: internet (105.110.116.101.114.110.101.116), onlyMib2 (111.110.108.121.77.105.98.50) y onlySystem (111.110.108.121.83.121.115.116.101.109).

  Le antecede un entero indicando su longitud en octetos.
- ViewTreeFamilySubtree que es el OID de la rama específica de la MIB que está definida por el nombre de vista previo. Le antecede un entero indicando su longitud en octetos. Por ejemplo, para la vista onlySystem el subárbol es 1.3.6.1.2.1.25.1.

Dentro de esta tabla hay diferentes objetos, se detallan los más importantes a continuación:

- ***vacmViewTreeFamilyType 1.3.6.1.6.3.16.1.5.2.1.4***

  Describe si el subárbol está incluido (1) o excluido (2) de la vista. Todos los subárboles en este caso se encuentran incluidos.

OID de la instancia	Valor
1.3.6.1.6.3.16.1.5.2.1.4.8.105.110.116.101.114.110.101.116.4.1.3.6.1	1
1.3.6.1.6.3.16.1.5.2.1.4.8.111.110.108.121.77.105.98.50.6.1.3.6.1.2.1	1
1.3.6.1.6.3.16.1.5.2.1.4.10.111.110.108.121.83.121.115.116.101.109.7.1.3.6.1.2.1.1	1
1.3.6.1.6.3.16.1.5.2.1.4.10.111.110.108.121.83.121.115.116.101.109.8.1.3.6.1.2.1.25.1	1

Tabla 3.10- Instancias del objeto vacmViewTreeFamilyType

Los objetos ***vacmViewTreeFamilyStorageType 1.3.6.1.6.3.16.1.5.2.1.5*** y ***vacmViewTreeFamilyStatus 1.3.6.1.6.3.16.1.5.2.1.6*** con sus diferentes instancias cada uno se definen de la misma forma que los objetos *usmUserStorageType* y *usmUserStatus* vistos en la sección 3.7.2 y tienen los mismos valores.

3.8.2 Relación niveles de seguridad de grupos con los de usuarios.

Cuando configuramos cualquier agente SNMP, tenemos que tener en cuenta la relación entre los niveles de seguridad que tienen los usuarios y grupos que configuramos y las interacciones de los usuarios con las solicitudes entrantes.

La interacción entre los niveles de seguridad de usuarios (agentes NMS y MD) tiene como regla general que el nivel de seguridad del usuario tiene precedencia sobre el nivel de seguridad de la solicitud proveniente de la consola de administración; ya que si el nivel de seguridad de la solicitud es mayor que el configurado para el usuario al que se le está preguntando, la solicitud no será exitosa y el OID correspondiente a “nivel de seguridad no soportado” (1.3.6.1.6.3.15.1.1.1) será enviado al NMS. Se muestra en las figuras desde la 3.113 hasta la 3.116 para este caso la configuración en el router (grupo y usuario noauth), la solicitud de nivel authNoPriv al router usando SNMP JManager y el contenido en Wireshark. El siguiente comando configura un grupo de nivel de seguridad noauth:

```
RT-LAB-SIMUTEL(config)#snmp-server group GrupoInvitado v3
noauth read vistainternet
```

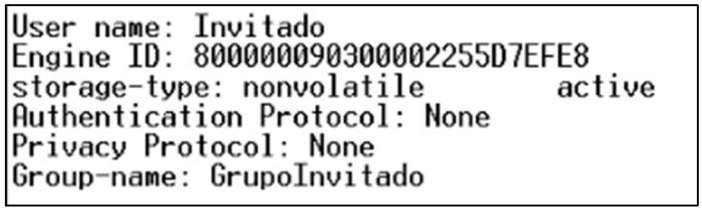


Figura 3.113- Usuario Invitado con grupo Invitado

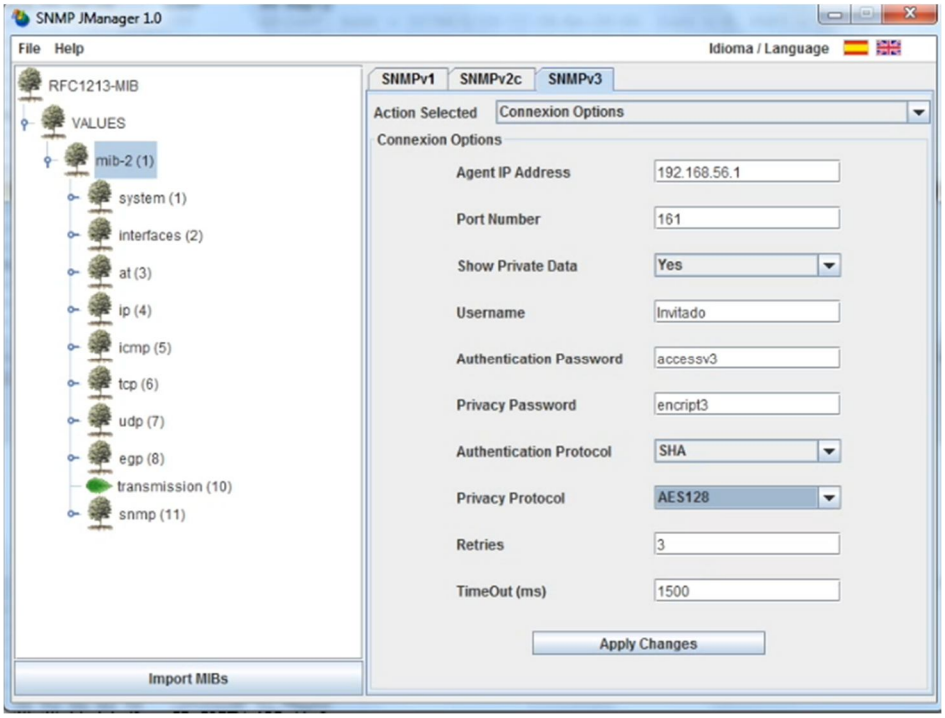


Figura 3.114- Configuración para solicitud con usuario Invitado

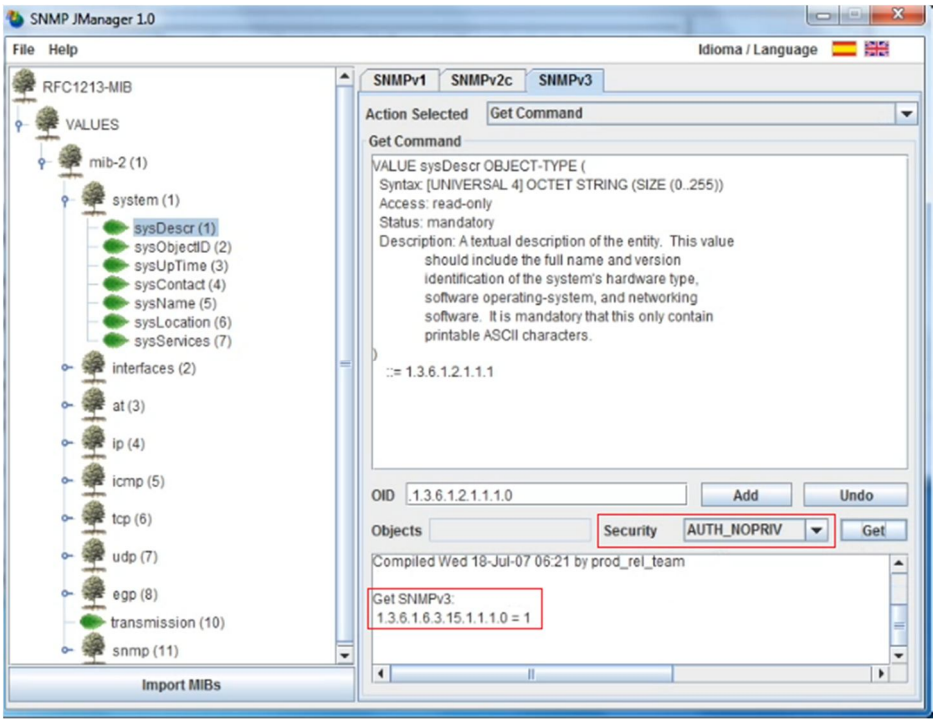


Figura 3.115- Solicitud de descripción con respuesta 1

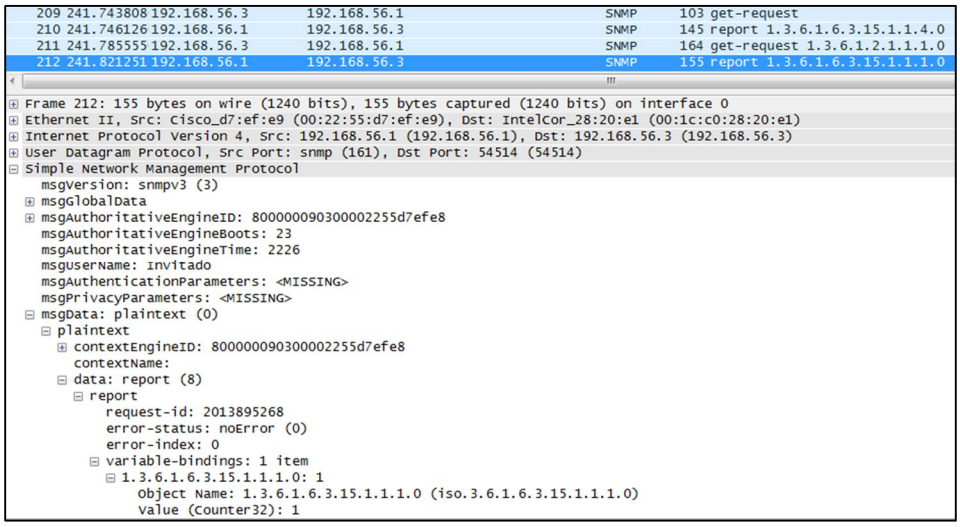


Figura 3.116- Captura en Wireshark de solicitud de descripción

Con respecto a los niveles de seguridad dentro del mismo agente monitoreado se tiene que el nivel de seguridad del grupo tiene precedencia sobre el del o los usuarios que pertenecen a ese grupo, si es que el nivel de seguridad del grupo es mayor que la configurada para los usuarios. Esto se explica mejor con las siguientes dos situaciones. En el primer escenario, si dentro del agente el grupo al que el usuario pertenece no tiene ningún tipo de seguridad (un grupo noauth) y el usuario dentro del grupo tiene un nivel de seguridad mayor, por ejemplo, authPriv; una solicitud externa que llegue a dicha entidad a ese usuario con un nivel de seguridad authNoPriv será capaz de recuperar la información que estaba buscando a pesar que el usuario de la entidad SNMP del router tenga configurada autenticación y cifrado de datos con sus respectivas claves. El siguiente comando configura un grupo de nivel de seguridad noauth:

```
RT-LAB-SIMUTEL(config)#snmp-server group GrupoInvitado v3
noauth read vistainternet
```

```
User name: Root
Engine ID: 800000090300002255D7EFE8
storage-type: nonvolatile        active
Authentication Protocol: SHA
Privacy Protocol: AES128
Group-name: GrupoInvitado
```

Figura 3.117- Usuario Root con grupo Invitado

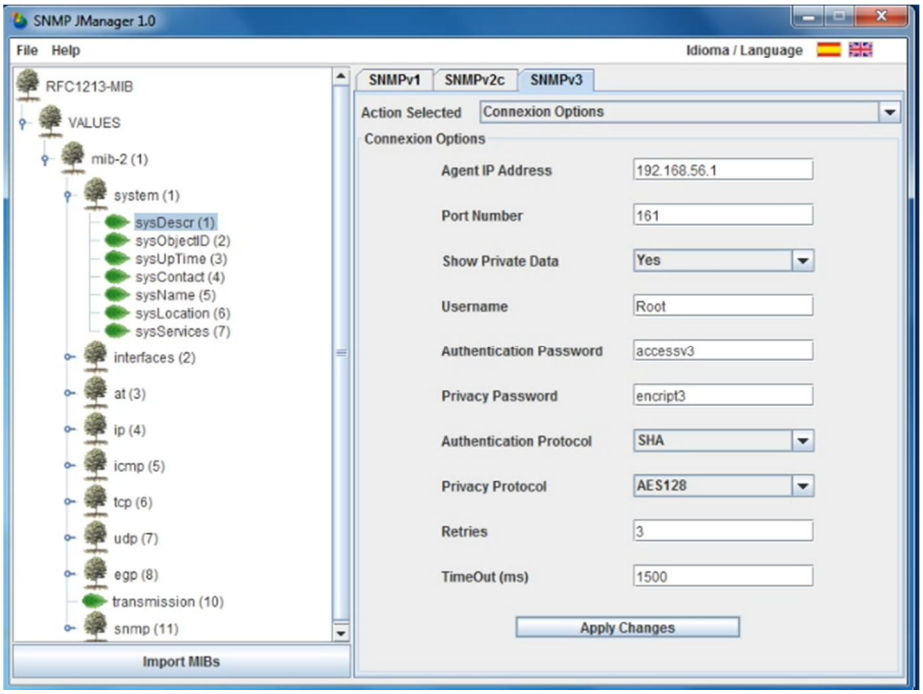


Figura 3.118- Configuración para solicitud con usuario Root

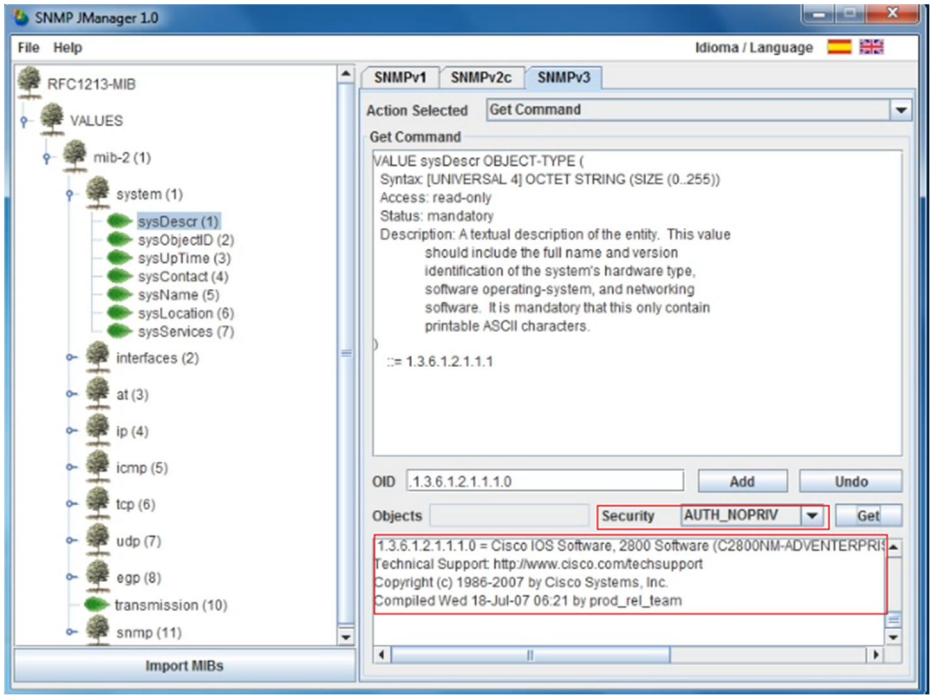


Figura 3.119- Solicitud de descripción con respuesta exitosa

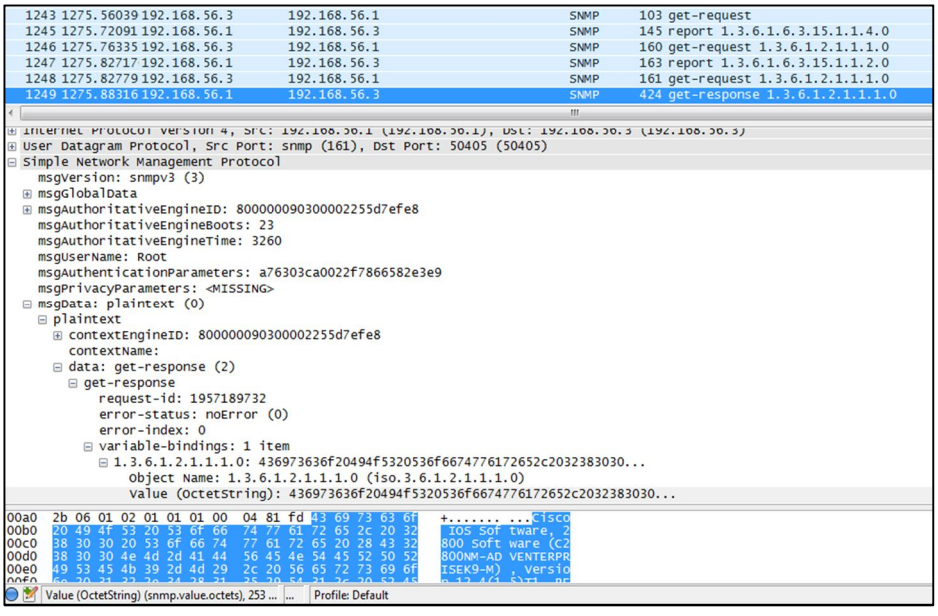


Figura 3.120- Captura en Wireshark de solicitud de descripción

En el segundo escenario, la situación opuesta. Cuando el nivel de seguridad del grupo es mayor, por ejemplo authPriv y el usuario que pertenece al grupo tiene nivel de seguridad menor (por ejemplo un usuario noAuthNoPriv o authNoPriv), una solicitud entrante externa al usuario del agente que tenga nivel de seguridad noAuthNoPriv o authNoPriv obtendrá un NULL como respuesta a la solicitud. El siguiente comando configura un grupo de nivel de seguridad priv:

```
RT-LAB-SIMUTEL(config)#snmp-server group GrupoRoot v3 priv
read vistainternet write vistainternet notify vistainternet
```

Figura 3.121- Usuario Supervisor con grupo Root

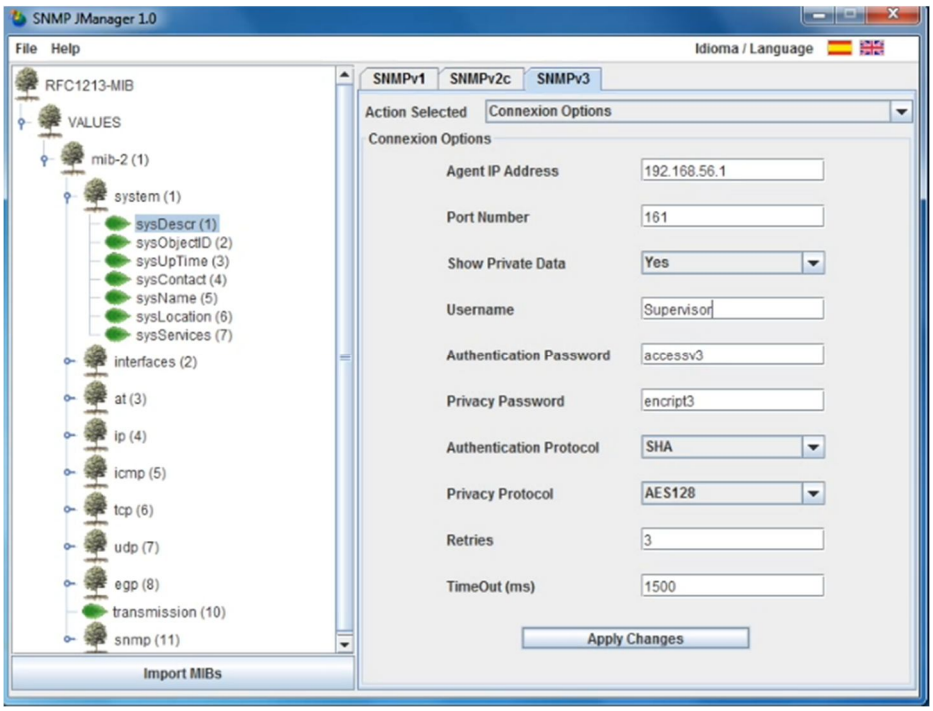


Figura 3.122- Configuración para solicitud con usuario Supervisor

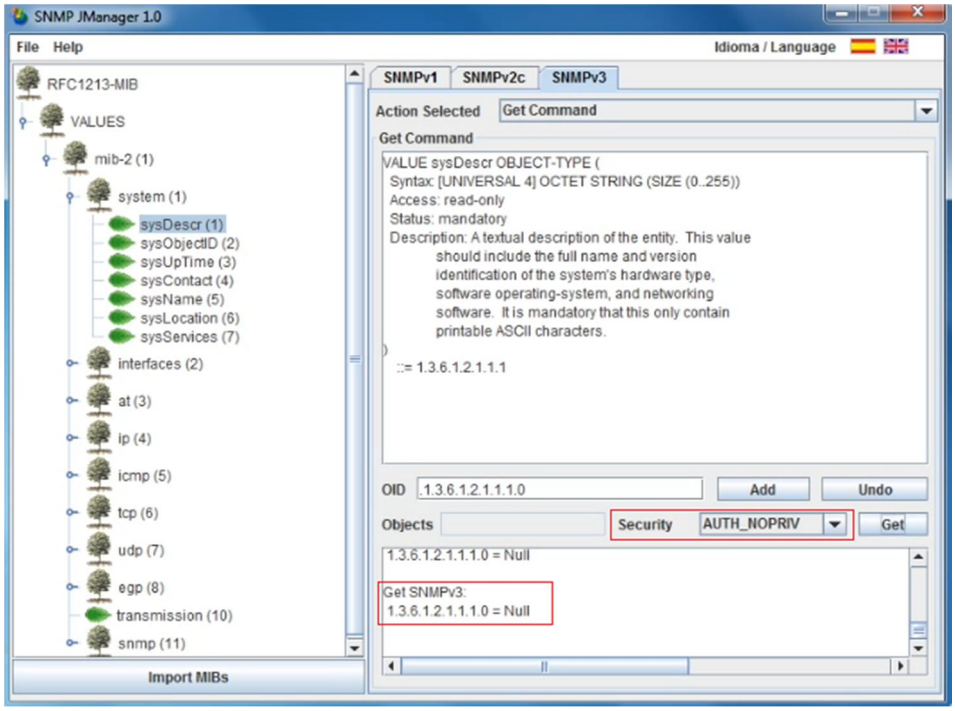


Figura 3.123- Solicitud de descripción con respuesta Null

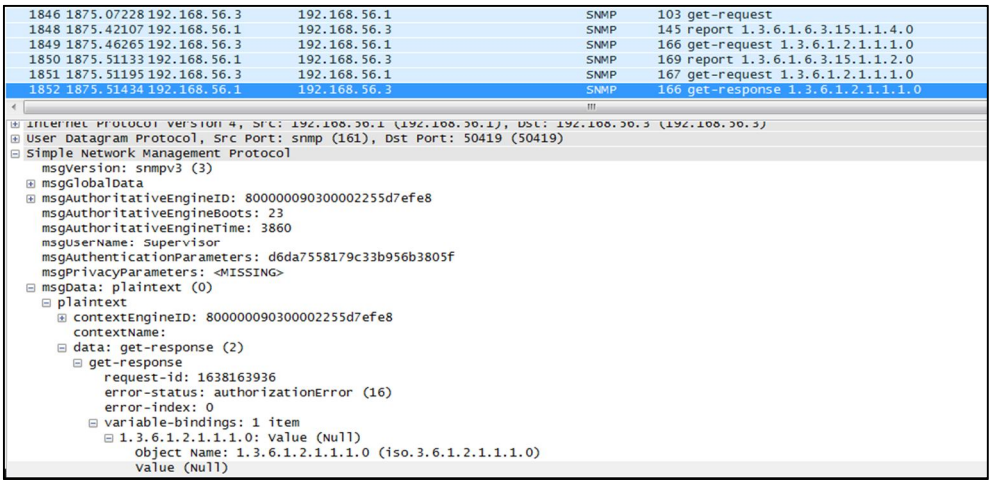


Figura 3.124- Captura en Wireshark de solicitud de descripción

Por eso es preferible que siempre exista concordancia entre los niveles de seguridad configurados para los grupos y los usuarios que pertenezcan a cada uno de ellos. La siguiente tabla muestra el resumen de resultados de las solicitudes realizadas al router.

Nivel de seguridad del motor		Nivel de seguridad consola NMS	
Grupo	Usuario del grupo	Solicitud de usuario	Resultados de solicitudes
noAuthNoPriv	noAuthNoPriv	noAuthNoPriv	OK
		authNoPriv	Nivel de seguridad no soportado
		authPriv	Nivel de seguridad no soportado
	authNoPriv	noAuthNoPriv	OK
		authNoPriv	OK
		authPriv	Nivel de seguridad no soportado
	authPriv	noAuthNoPriv	OK
		authNoPriv	OK
		authPriv	OK
authNoPriv	noAuthNoPriv	noAuthNoPriv	Null
		authNoPriv	Nivel de seguridad no soportado
		authPriv	Nivel de seguridad no soportado
	authNoPriv	noAuthNoPriv	Null
		authNoPriv	OK
		authPriv	Nivel de seguridad no soportado
	authPriv	noAuthNoPriv	Null
		authNoPriv	OK
		authPriv	OK
authPriv	noAuthNoPriv	noAuthNoPriv	Null
		authNoPriv	Nivel de seguridad no soportado
		authPriv	Nivel de seguridad no soportado
	authNoPriv	noAuthNoPriv	Null
		authNoPriv	Null
		authPriv	Nivel de seguridad no soportado
	authPriv	noAuthNoPriv	Null
		authNoPriv	Null
		authPriv	OK

Tabla 3.11- Niveles de seguridad de grupos y usuarios

3.8.3 Efecto de la aplicación de las vistas.

En la configuración del router, hemos establecido grupos, vistas y usuarios. Entre estas vistas, está la vista ***vistamib2,*** la cual excluye el árbol ***IP***, por medio del comando *excluded*.

```
snmp-server user Invitado GrupoInvitado v3

snmp-server group GrupoRoot v3 priv read vistainternet write vistainternet notif
y vistainternet
snmp-server group GrupoInvitado v3 noauth notify vistainternet
snmp-server group GrupoSupervisor v3 auth read vistamib2 notify vistainternet
snmp-server view vistamib2 mib-2 included
snmp-server view vistamib2 ip excluded
snmp-server view vistainternet internet included
snmp-server community c0Mmuni7yAdm view vistainternet RW
snmp-server trap-source FastEthernet0/1.1
snmp-server location Fiec - Laboratorio de Simulacion deTelecomunicaciones - Rac
k Grupo 3
```

Figura 3.125- Configuración del router de grupos, usuarios y vistas

Establecemos los datos en opciones de conexión. Para esta demostración utilizamos el usuario *Supervisor*.

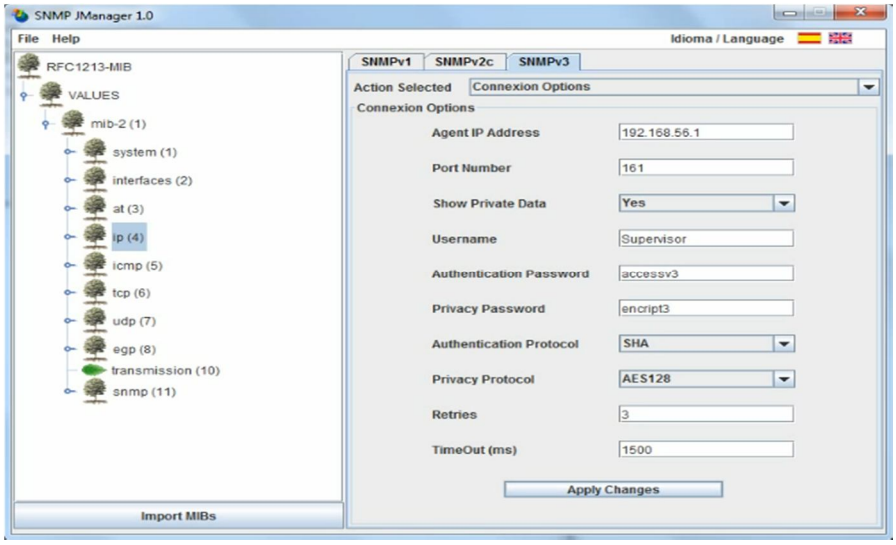


Figura 3.126- Configuración de solicitud con usuario Supervisor

La solicitud de *Walk,* es una orden que realiza una serie completa de get-next automáticamente y se detiene cuando devuelve resultados que no están en el rango del OID especificado originalmente. Es decir, al hacer la solicitud walk, se recorrió el árbol mib-2 excluyendo el árbol IP, hasta llegar al OID 1.3.6.1.2.1.92.1.2.2.0 que equivale a *endOfMibView*.

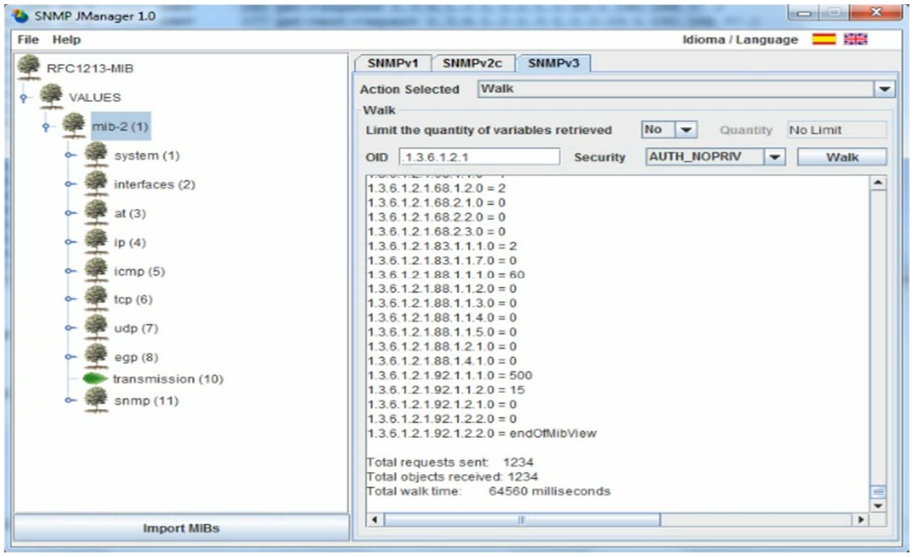


Figura 3.127- Recorrido de la mib-2

En la figura 3.128 podemos ver que la parte sombreada es exactamente donde se salta del árbol *at(3)*(1.3.6.1.2.**3**.1.1) al árbol *icmp(5)*(1.3.6.1.2.**5**.1.0), excluyendo el árbol *ip(4)*.

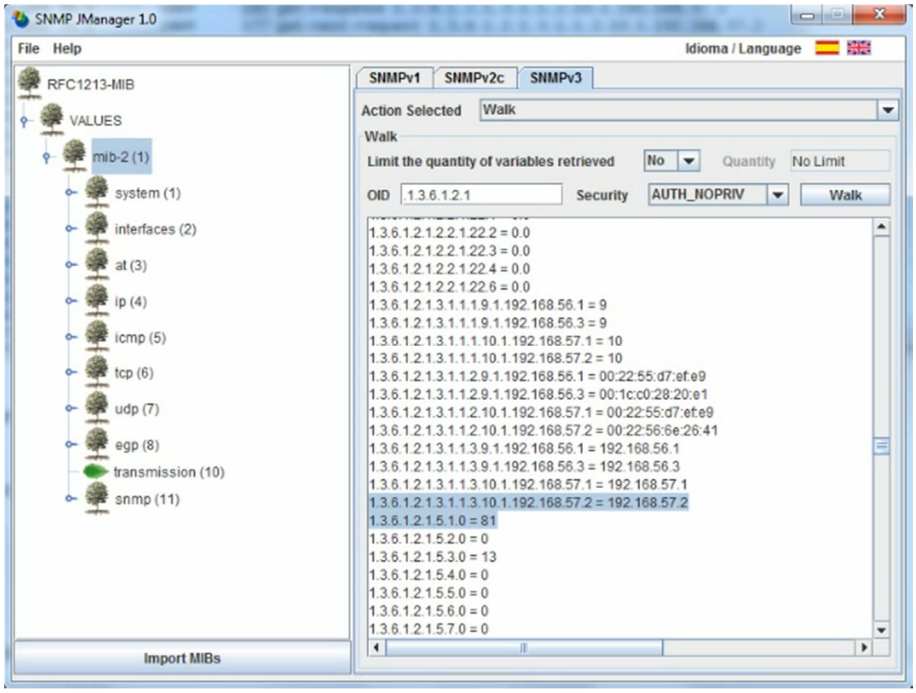


Figura 3.128- Intervalo de salto de un árbol a otro

En Wireshark hemos capturado los paquetes SNMP de las solicitudes de la orden *Walk*, donde también podemos ver el salto en donde se excluye el árbol IP.

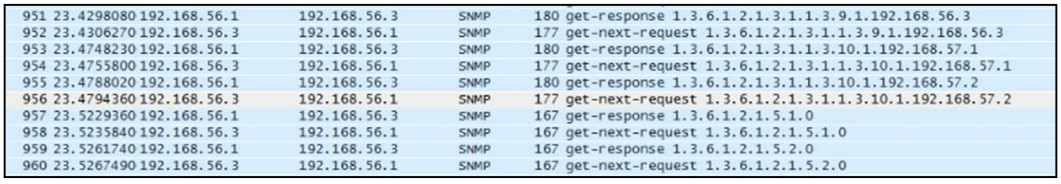


Figura 3.129- Objeto seleccionado del árbol 1.3.6.1.2.1.3

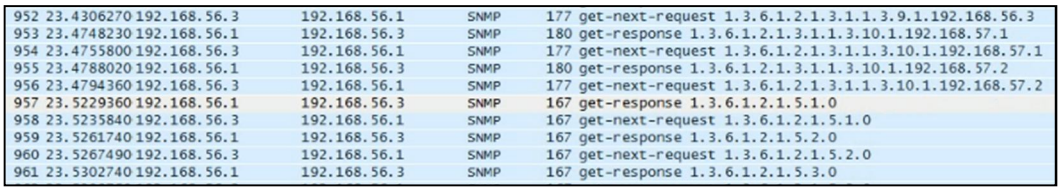


Figura 3.130- Objeto seleccionado del árbol 1.3.6.1.2.1.5

3.9 Funcionamiento de accesos a entidades para consultas

Para conocer las interacciones de solicitud y respuesta que se realizan entre las entidades involucradas, realizamos un get-request utilizando como NMS el programa SNMP JManager y como agente remoto una PC con sistema operativo Ubuntu.

3.9.1 Usuario y contraseñas correctas

Primero, establecimos los datos en las opciones de conexión escribiendo la dirección IP del host del agente, el usuario que utilizamos para realizar la solicitud, la contraseña de autenticación (no es necesario especificar la contraseña de cifrado ya que no usamos restricciones de privacidad) y protocolo de autenticación (tampoco es necesario especificar protocolo de privacidad).

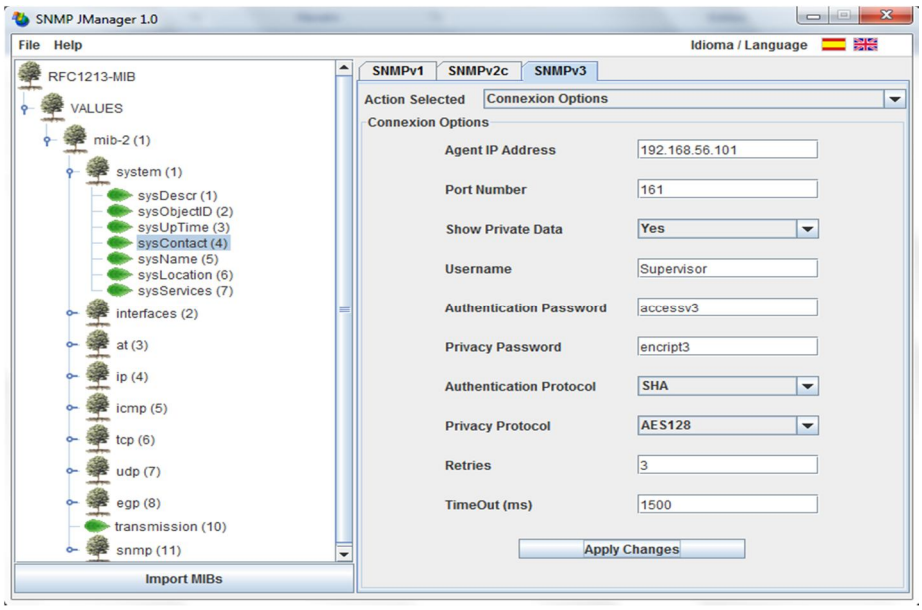


Figura 3.131- Configuración de solicitud con usuario Supervisor

Realizamos la solicitud del contacto, y obtuvimos la respuesta sin ningún error.

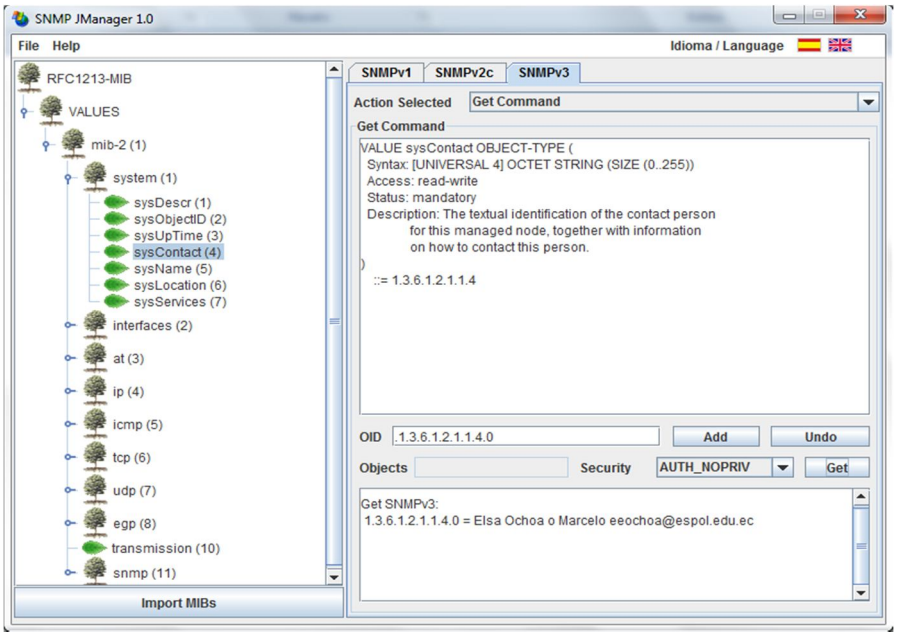


Figura 3.132- Solicitud de descripción con respuesta exitosa

Al capturar los paquetes en Wireshark, pudimos observar que existen seis interacciones con una sola petición. A estas interacciones se les llama descubrimiento, en el que el motor SNMP no autoritativo (agente NMS) descubre parámetros importantes del motor autoritativo (agente MD remoto) para poder realizarle consultas en SNMPv3.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.00309100	192.168.56.3	192.168.56.101	SNMP	103	get-request
4	0.02477800	192.168.56.101	192.168.56.3	SNMP	147	report 1.3.6.1.6.3.15.1.1.4.0
5	0.21484700	192.168.56.3	192.168.56.101	SNMP	168	get-request 1.3.6.1.2.1.1.4.0
6	0.23194300	192.168.56.101	192.168.56.3	SNMP	173	report 1.3.6.1.6.3.15.1.1.2.0
7	0.23307900	192.168.56.3	192.168.56.101	SNMP	169	get-request 1.3.6.1.2.1.1.4.0
8	0.23743600	192.168.56.101	192.168.56.3	SNMP	211	get-response 1.3.6.1.2.1.1.4.0

Figura 3.133- Captura de solicitud de descripción en Wireshark

Al analizar cada paquete, nos percatamos de las siguientes características:

1° Paquete.- Es un get-request dirigido desde la dirección IP 192.168.56.3 (NMS) hacia la dirección IP 192.168.56.101 (agente). Como hemos mencionado en la sección 2.6, los paquetes SNMPv3 tienen varios parámetros, pero para este primer paquete resulta importante enfatizar que los campos *msgAuthoritativeEngineID, msgAuthenticationParameters, msgPrivacyParameters* están vacíos, además de las *variable-bindings*, a pesar de que se hizo una solicitud con un usuario y OID's específicos; todo esto para dar inicio al proceso de descubrimiento.

Dentro de las banderas del mensaje (*msgFlags*) se observa que el mensaje es reportable, es decir, que espera una respuesta del motor SNMP con el que se está comunicando. Al ser un mensaje que no requiere autenticación en la entidad remota (es un mensaje de prueba que provoca un report), las banderas de autenticación y cifrado no están habilitadas ni tampoco sus parámetros (msgAuthenticationParameters y msgPrivacyParameters).

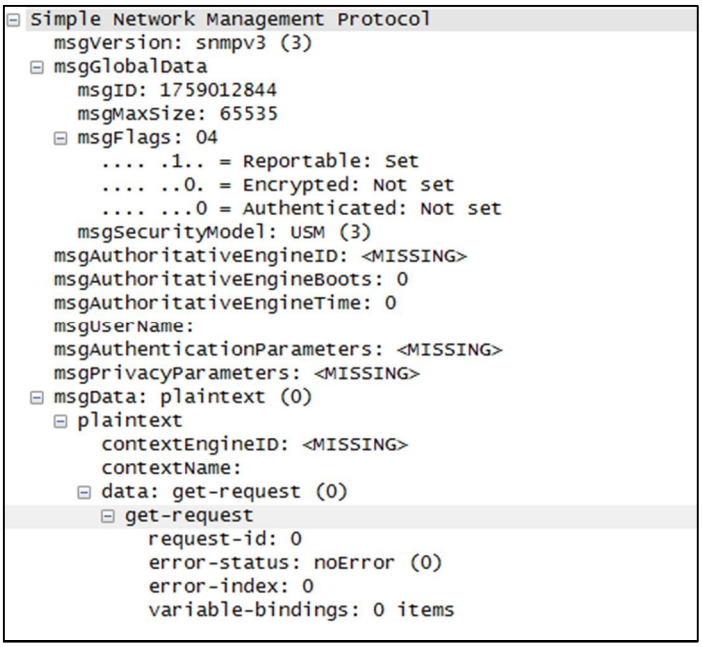


Figura 3.134- Contenido de la 1° interacción del mensaje

2° Paquete.- Este paquete es un report dirigido desde la dirección IP 192.168.56.101 (agente) hacia la dirección IP 192.168.56.3 (NMS), donde el parámetro *msgAuthoritativeEngineID* tiene el valor 80001f88804a9798550b502b52 y el campo *variable-bindings* contiene el OID 1.3.6.1.6.3.15.1.1.4.0 con el valor de 1 (paquetes descartados). El OID corresponde al contador *usmStatsUnknownEngineIDs*, el cual indica que el paquete enviado fue descartado ya que el snmpEngineID no era reconocido en el agente remoto, es decir, el NMS necesitaba conocer el snmpEngineID del agente al que le iba a realizar la solicitud previamente. En resumidas cuentas el MD le hace conocer al NMS su engineID.

No define banderas ni tampoco parámetros de autenticación o privacidad, ya que hasta este punto solamente es un mensaje informativo al motor no autoritativo.

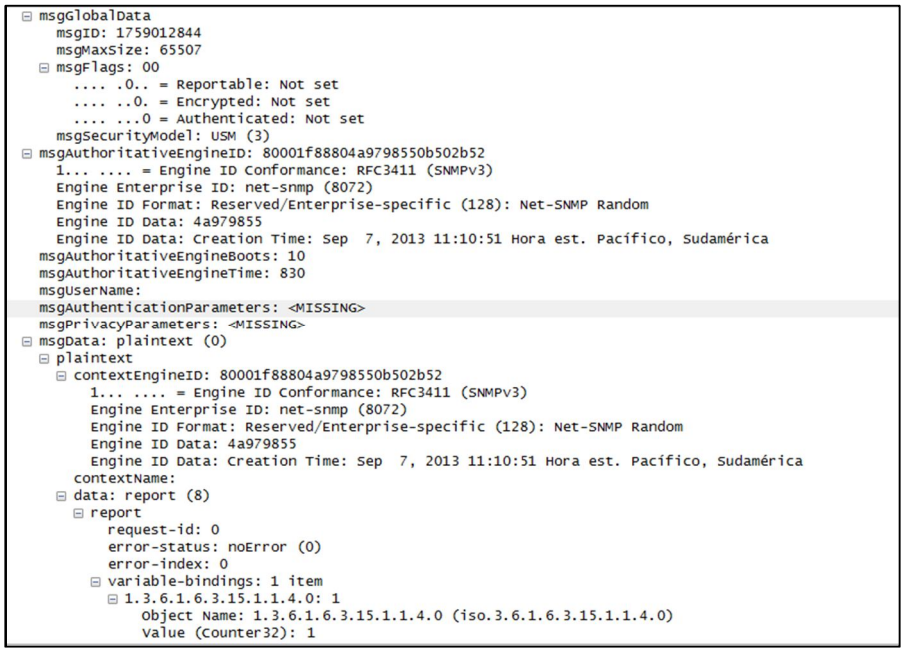


Figura 3.135- Contenido de la 2° interacción del mensaje

3° Paquete.- Es un nuevo get-request dirigido desde la dirección IP 192.168.56.3 (NMS) hacia la dirección IP 192.168.56.101 (agente), recordemos que el anterior se descartó como se explicó en el segundo paquete. Este paquete ya contiene el valor del snmpEngineID autoritativo en el parámetro *msgAuthoritativeEngineID,* y el campo *variable-bindings* ya contiene el OID (1.3.6.1.2.1.1.4.0, sysContac) de la solicitud que se desea realizar, con el valor de NULL; pero los parámetros de sincronización *msgAuthoritativeEngineBoots* y *msgAuthoritativeEngineTime* están en cero.

Los contenidos del campo *variable bindings* son visibles ya que la solicitud fue hecha con un nivel de seguridad que no requerían privacidad (cifrado).

El mensaje es reportable, y tiene habilitadas las banderas de autenticación y cifrado. Como se mencionó en la sección 2.4.3.5, un motor SNMP genera un valor de resumen del cual una parte se coloca en el campo de autenticación del mensaje para que la entidad remota (MD) lo compare con el valor que obtenga luego de someter al mensaje que reciba al algoritmo de autenticación y así verificar que el mensaje es auténtico y no ha sido alterado. Se envían los parámetros de autenticación pero no todavía los de privacidad (si el nivel se seguridad del mensaje también incluyera privacidad) debido a que falta sincronización todavía entre los motores SNMPv3.

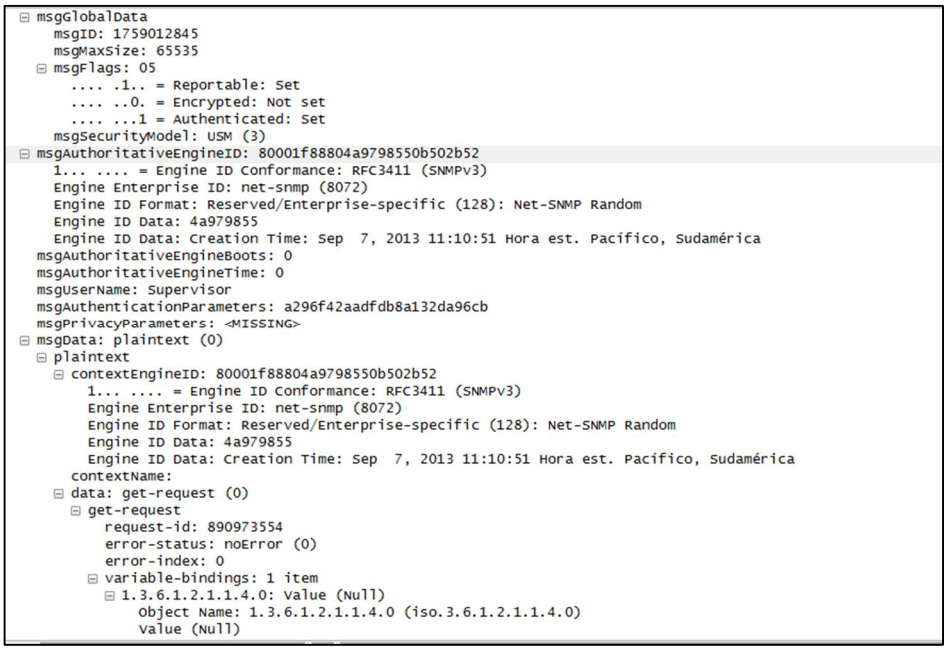


Figura 3.136- Contenido de la 3° interacción del mensaje

4° Paquete.- Es un report dirigido desde la dirección IP 192.168.56.101 (agente) hacia la dirección IP 192.168.56.3 (NMS), conteniendo en el campo *variable-bindings* el contador de OID 1.3.6.1.6.3.15.1.1.2.0 cuyo nombre es *usmStatsNotInTimeWindows*, con valor de 1, indicando que aunque el paquete anterior está autenticado, fue descartado ya que no se encontraba dentro de la ventana de tiempo del snmpEngineID autoritativo (agente). La ventana de tiempo es un intervalo de segundos determinado, con respecto al EngineTime existente en el motor autoritativo, en el que una solicitud es considerada válida. Los parámetros *msgAuthoritativeEngineBoots* y *msgAuthoritativeEngineTime* de este report tienen un valor específico, que sirve para que la entidad solicitante los adopte localmente y así se sincronice

con la entidad autoritativa. Para poder sincronizarse, el NMS necesita conocer el *msgAuthoritativeEngineBoots* y *msgAuthoritativeEngineTime* del agente al que le va a realizar la solicitud.

Con respecto a las banderas, tiene habilitada la autenticación ya que ya existe autenticación entre ambos agentes, y envía el mensaje de resumen calculado dentro de los parámetros de autenticación *msgAuthenticationParameters*, el cual se comparará con el valor que calcule el motor remoto (NMS) para este mensaje.

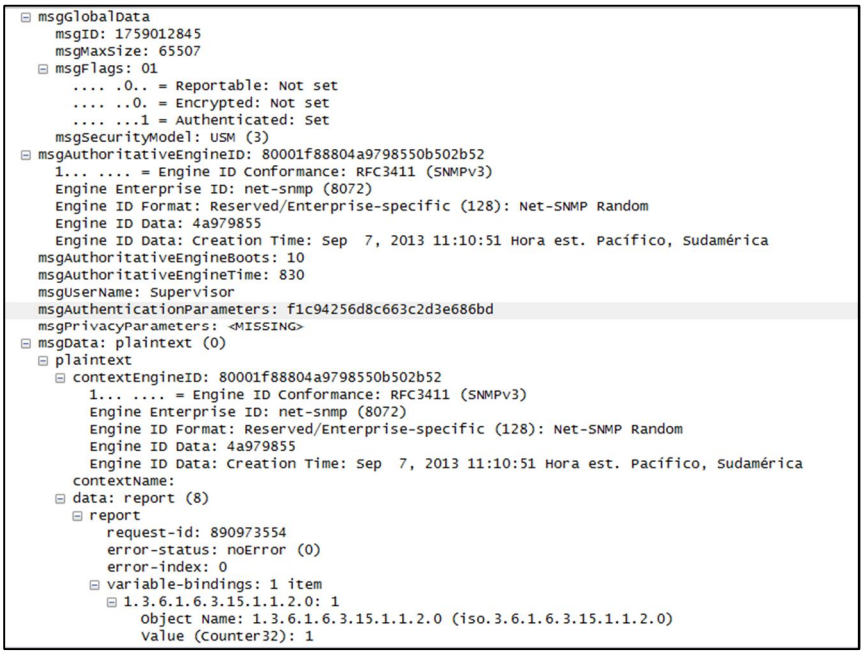


Figura 3.137- Contenido de la 4° interacción del mensaje

5° Paquete.- Es un get-request dirigido desde la dirección IP 192.168.56.3 (NMS) hacia la dirección IP 192.168.56.101 (agente). Como nos podemos dar cuenta, este paquete ya tiene todos los parámetros completos (*msgAuthoritativeEngineID, msgAuthoritativeEngineBoots* y *msgAuthoritativeEngineTime*) que necesitaba conocer del agente para poder realizar la solicitud. Ya hay sincronización entre ambos motores, por lo que las banderas indican que el mensaje es reportable y autenticable pero no cifrado ya que la solicitud fue hecha con el usuario Supervisor. Tiene definido el campo msgAuthenticationParameters que contiene el valor de resumen de esta solicitud, para que la entidad remota (MD) lo compare con el valor que obtenga luego someter al mensaje que reciba al algoritmo de autenticación.

El campo *variable-bindings* también viene con el OID (1.3.6.1.2.1.1.4.0, sysContac) correspondiente a la solicitud que realizamos por medio del programa SNMP JManager. Ahora el paquete ya está listo para poder realizar la solicitud sin ser descartado, ya que el agente remoto logra autenticar el mensaje y comprobar que está dentro del tiempo esperado para luego también descifrar la solicitud si es que hubiera sido realizada con nivel de seguridad priv.

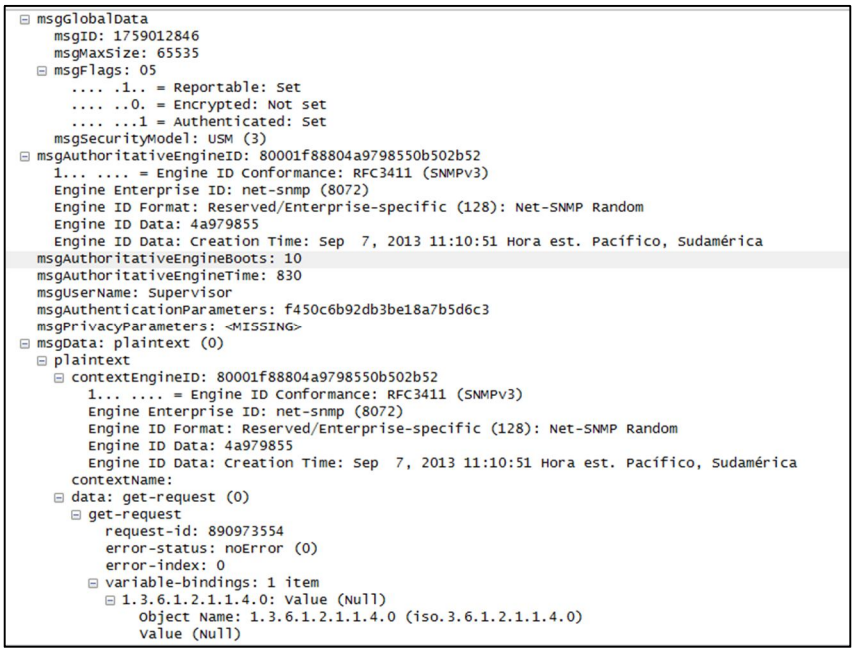


Figura 3.138- Contenido de la 5° interacción del mensaje

6° Paquete.- Finalmente este paquete es un get-response dirigido desde la dirección IP 192.168.56.101 (agente) hacia la dirección IP 192.168.56.3 (NMS), con el valor de la respuesta solicitada en el campo *variable-bindings*. Este valor es de tipo OctetString, por lo tanto sólo veremos una cadena de caracteres hexadecimales, pero en la sección bytes de paquete de la pantalla de Wireshark, se puede ver el texto que la cadena de hexadecimales simboliza.

La NMS ya no enviará otro mensaje o acuse de recibo, por lo que la bandera de report está en cero, el mensaje tiene habilitada la autenticación por lo que las bandera y el parámetro respectivo están definidos en el mensaje.

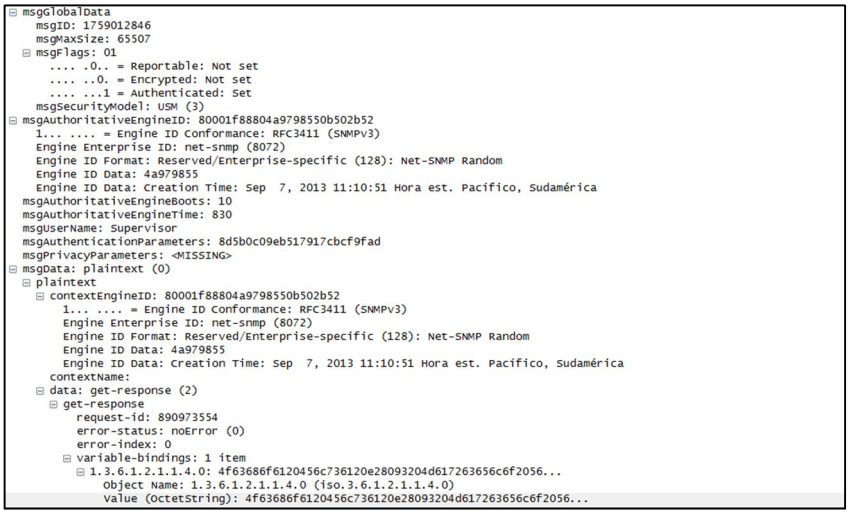


Figura 3.139- Contenido de la 6° interacción del mensaje

Luego para comprobar el sistema de seguridad que permite el ingreso y procesamiento de solicitudes entrantes con los niveles de seguridad authNoPriv y AuthPriv a un agente, en las subsiguientes secciones testeamos los posibles errores que un NMS puede tener al momento de tratar de recuperar información de algún dispositivo; estos errores son los de utilizar para alguna solicitud un usuario erróneo o contraseñas incorrectas ya sea de autenticación o de privacidad.

Antes de hacer las pruebas verificamos que el agente esté corriendo correctamente en nuestro dispositivo.

Figura 3.140- Verificación de activación del servicio SNMP

3.9.2 Usuario incorrecto

Escribimos mal intencionalmente el nombre de usuario mientras que las contraseñas de autenticación y los demás datos están escritos correctamente en las opciones de conexión de SNMP JManager.

En este caso utilizamos el nivel de seguridad authNopriv, ya que nuestro objetivo es ver el efecto cuando se ingresa una solicitud con usuario inexistente y no es necesario utilizar privacidad, ya que si lo hacemos no podríamos ver el contenido de las capturas de los paquetes en Wireshark. El usuario designado para este nivel de seguridad es *Supervisor*, pero como

podemos ver en la figura 3.141 hemos escrito *Superviso* para realizar la prueba.

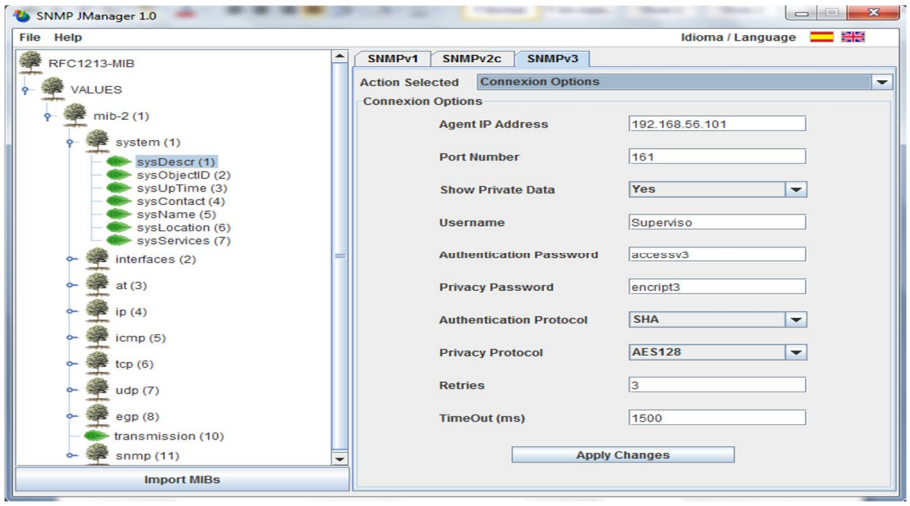


Figura 3.141- Configuración de solicitud con usuario erróneo

Es importante que el nivel de seguridad especificado en la ventana donde se ingresa el OID a consultar, sea el mismo que ha sido configurado en el agente remoto para el usuario al cual estamos consultando.

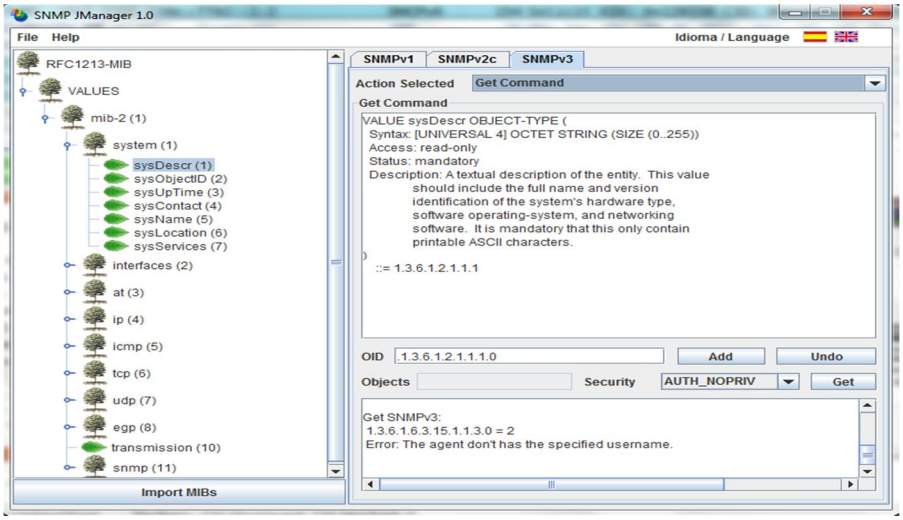


Figura 3.142- Solicitud de descripción con respuesta de error

Al realizar la petición de *sysDescr* que equivale a la descripción del equipo, hemos obtenido como resultado el error: The agent don't has the specified username. Esto quiere decir que el usuario ingresado no ha sido reconocido en el agente remoto, debido a que el usuario no está creado dentro del mismo.

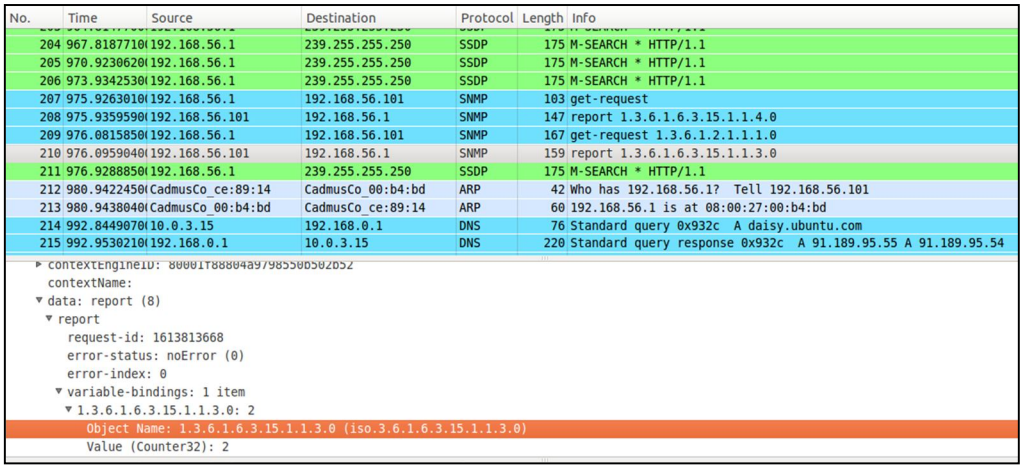


Figura 3.143- Captura de solicitud de descripción en Wireshark

Revisando las capturas SNMP en Wireshark de la solicitud realizada, podemos observar que en el report 210 enviado desde la PC Ubuntu a la PC NMS que ejecuta JManager, tiene como OID 1.3.6.1.6.3.15.1.1.3.0, que es el mismo OID que mostró el resultado de error consultado con el programa JManager, este OID tiene como respuesta en la variable *value* el número 2. Consultando este OID en un Repositorio de OIDs [33] encontramos el nombre y la descripción.

Su nombre es ***usmStatsUnknownUserNames***, describe el número total de paquetes recibidos por el motor SNMP que fueron descartados (y aumentan con los sucesivos errores de usuarios inexistentes que puedan haber) porque el usuario referenciado no fue reconocido dentro del motor del MD.

3.9.3 Contraseña de autenticación incorrecta

Configuramos los datos correctamente en las opciones de conexión exceptuando la contraseña de autenticación. La contraseña de autenticación designada para el nivel de seguridad que utilizaremos (authNoPriv) es accessv3, pero nosotros hemos escrito accessv378 (Fig. 3.144).

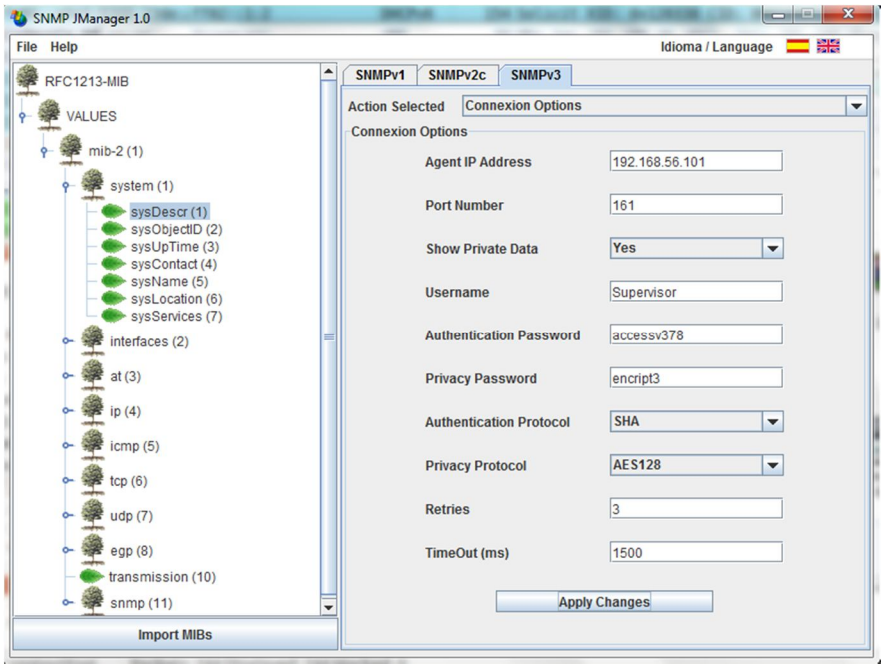


Figura 3.144- Configuración con contraseña de autenticación errónea

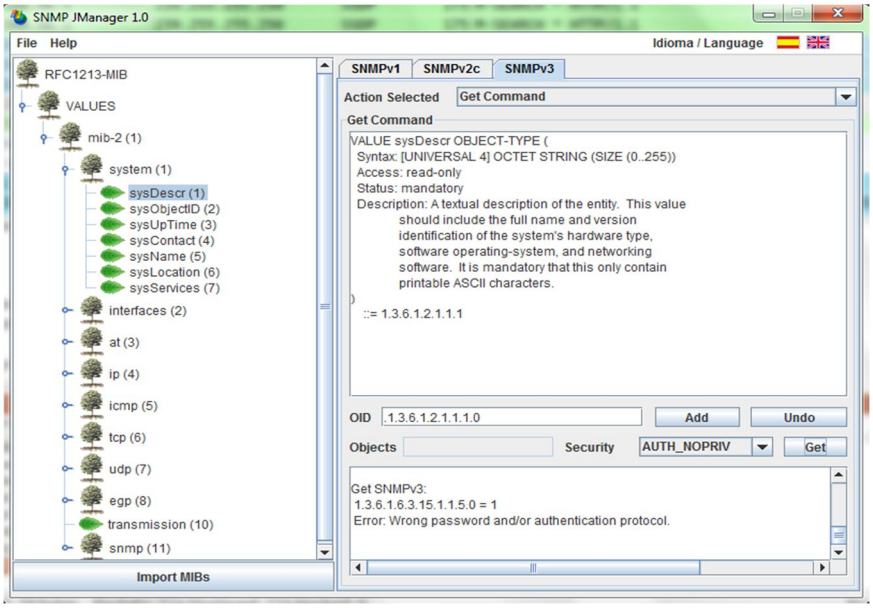


Figura 3.145- Solicitud de descripción con respuesta de error

Como se ve en la figura 3.145, el programa nos da el error: Wrong password and/or authentication protocol. Esto quiere decir que la contraseña y/o protocolo de autenticación que hemos ingresado es incorrecto.

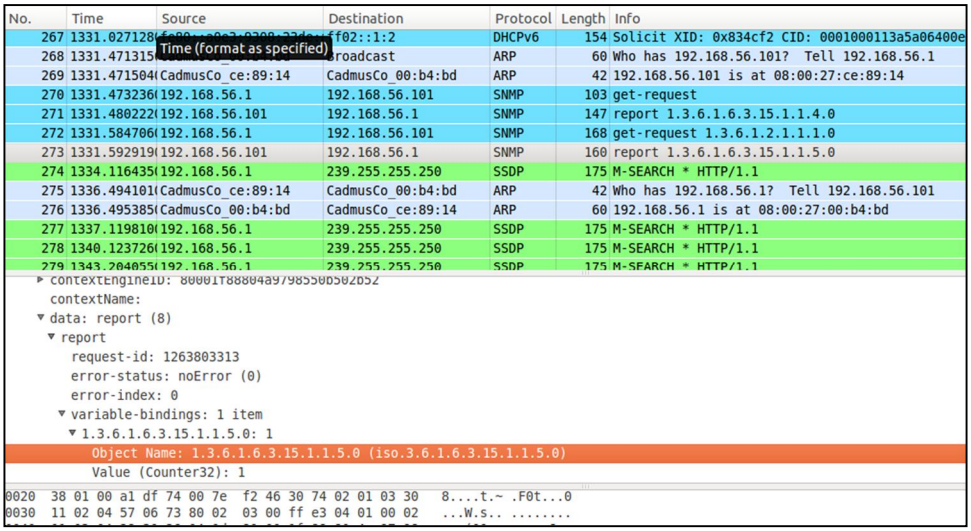


Figura 3.146- Captura de solicitud de descripción en Wireshark

Observando el report 273 que contiene el OID 1.3.6.1.6.3.15.1.1.5.0 de la captura de Wireshark y del SNMP JManager, vemos que no sólo es el mismo sino que tiene como respuesta el valor 1.

El OID, según [33] tiene como nombre ***usmStatsWrongDigests***, según su descripción el valor que contiene es el número total de paquetes recibidos por el motor SNMP, que fueron descartados porque no contienen el valor resumen esperado. Una solicitud que llega al MD debe tener un valor resumen de autenticación MAC determinado, resultado de someter al mensaje generado por un usuario existente, junto con la llave de autenticación correcta al algoritmo de autenticación especificado. Como la contraseña no era la correcta, la llave de autenticación del NMS difiere a la que conoce el MD y el valor resumen de la solicitud entrante difiere del valor resumen del MD y el motor SNMP descartó el paquete por esa razón. Este OID también es acumulativo, y si hay otro error de clave de autenticación futuro, la cuenta aumentará a 2.

3.9.4 Contraseña de cifrado incorrecta

En este caso utilizamos el nivel de seguridad AuthPriv, con usuario *Root* y contraseña de cifrado *encript3* (la contraseña de autenticación es la misma que en el nivel de seguridad authNopriv). Configuramos los datos correctamente en las opciones de conexión exceptuando la contraseña de cifrado. Para realizar la prueba hemos escrito como contraseña de cifrado *encript378*.

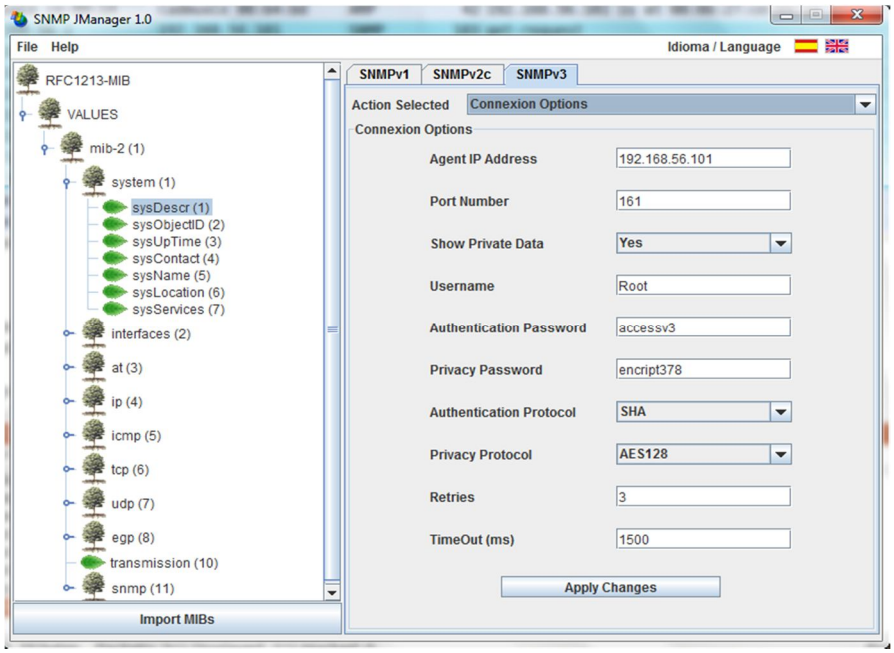


Figura 3.147- Configuración con contraseña de cifrado errónea

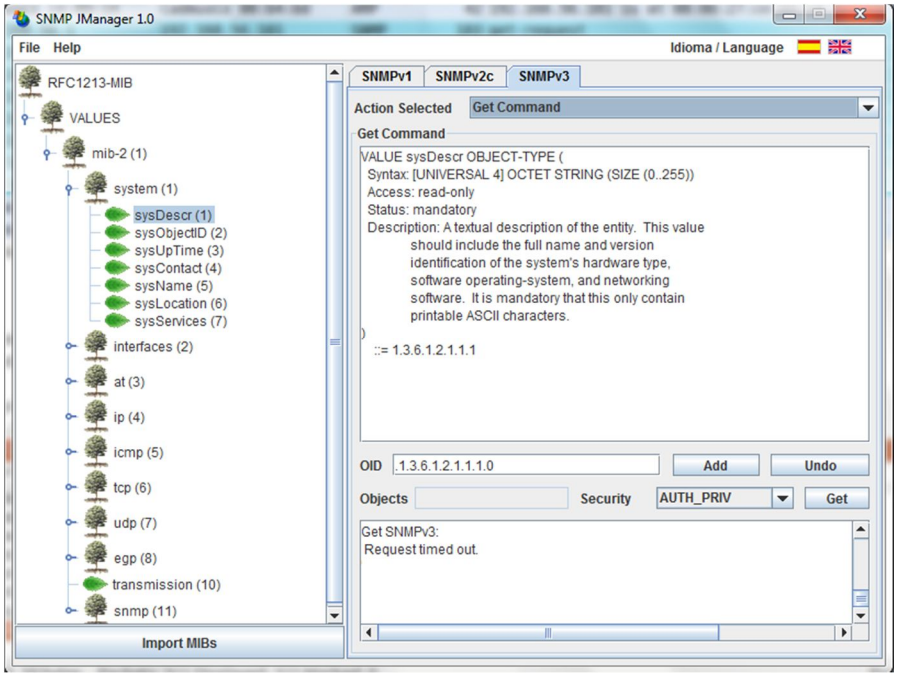


Figura 3.148- Solicitud de descripción con tiempo de espera agotado

Como podemos ver en la figura 3.148, el resultado obtenido es: Request timed out. Esto quiere decir que se agotó el tiempo de espera para una respuesta, lo cual indica que el receptor nunca pudo descifrar el paquete de la solicitud realizada, ya que la llave con la que el NMS envió el paquete no es la misma que el agente remoto tiene configurado en el nivel de seguridad authPriv, es decir, la solicitud nunca pudo ser procesada por el agente remoto y el tiempo de espera caducó.

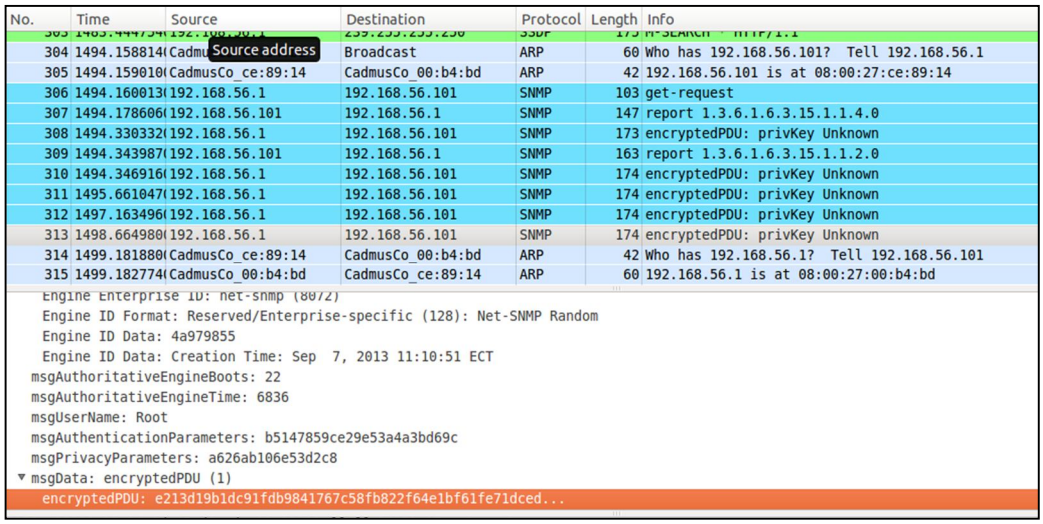


Figura 3.149- Captura de solicitud de descripción en Wireshark

Si observamos las ultimas capturas SNMP en Wireshark, el paquete 310 se repite 3 veces más (311, 312 y 313), esto es a causa del número de reintentos (3) que ingresamos en el casillero de reintentos (*Retries)* en las opciones de conexión para un mensaje SNMP.

3.10 Escenarios de tráfico SNMP en la red

En esta sección se analizan los diferentes tipos de escenarios de circulación de mensajes SNMPv2 y SNMPv3 en nuestra red, a saber, mensajes de solicitud de lectura de variables de la MIB, así como de escritura y recepción de las traps en el NMS. Durante el desarrollo de estas pruebas se capturó

con Wireshark el tráfico generado por las aplicaciones de monitoreo para comparar los paquetes en el siguiente capítulo.

En la figura 3.150 se detalla la forma en que se realizarán las pruebas en el desarrollo de la sección 3.10.

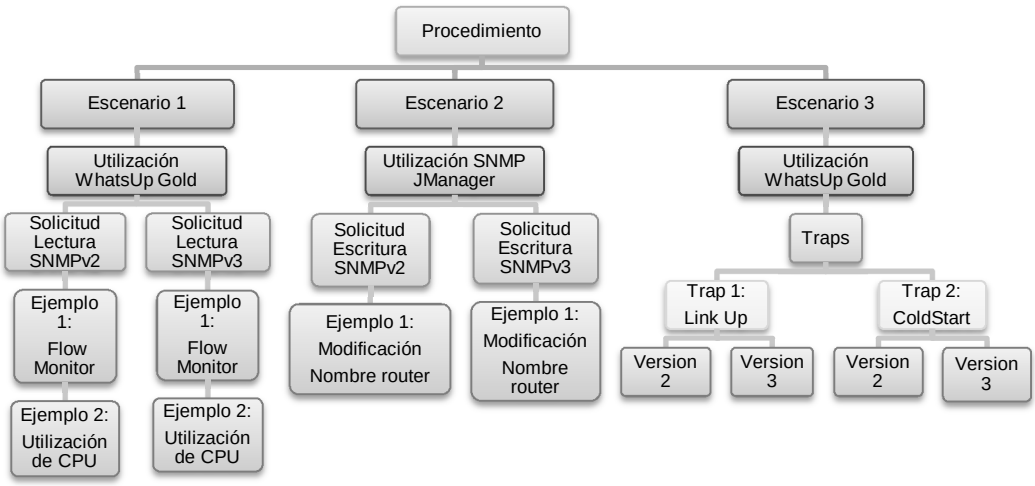


Figura 3.150- Desarrollo de escenarios de tráfico SNMPv2 y SNMPv3

3.10.1 Escenario 1: pruebas de solicitudes de lectura SNMP.

En esta sección analizaremos en detalle las primitivas que se envían a una agente SNMPv3 remoto cuando se lo consulta desde un NMS comparando los contenidos (y la capacidad de analizarlo) de los mensajes dependiendo a

cuál usuario/nivel de seguridad se hace la solicitud. Vamos a exponer dos ejemplos, el primero de solicitudes con versiones 2 y 3 de SNMP al router Cisco usando Flow Monitor de WhatsUp Gold y luego solicitudes de uso de CPU a una computadora usando WhatsUp Gold.

3.10.1.1 Solicitudes SNMP a equipos monitoreados usando v2.

3.10.1.1.1 Flow Monitor

Dentro de la consola web de WhatsUp Gold, elegimos la pestaña *Flow Monitor* para configurar y desplegar los diferentes reportes sobre tráfico de interfaces de red. En la sección *Configuration*, dentro de *Settings*, verificamos que la fuente de la información de tráfico es el router y vamos a consultarlo con la comunidad SNMPv2 c0Mmuni7yAdm. Como se sabe, los mensajes que genera, tienen un contenido de datos en texto plano.

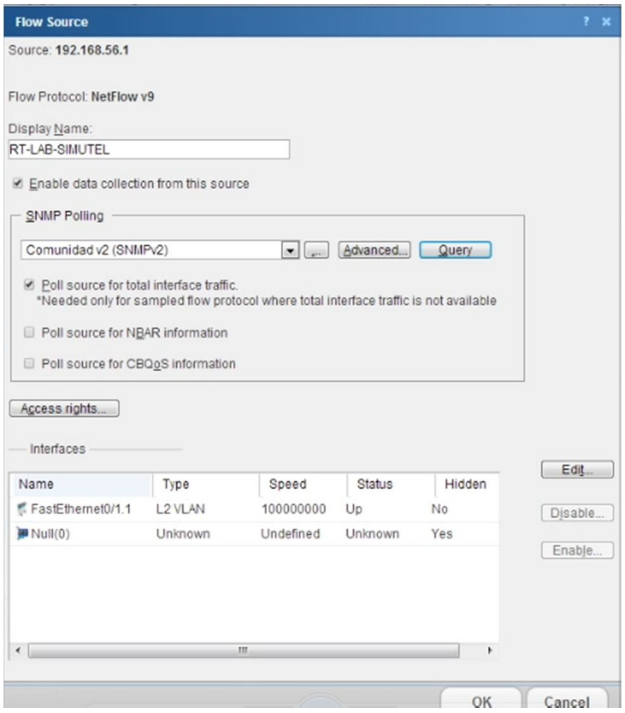


Figura 3.151- Configuración de captura de tráfico con comunidad v2

La recepción y consulta de todos los datos de tráfico se encuentra en la sección *Reports*. Las funcionalidades de control de tráfico en esta sección incluyen a: *Interface Details*, *Interface Overview, Flow Monitor Log, Bandwidth Usage, Interface Usage, NBAR & CBQoS* y *Host Traffic*; que entre otras cosas, permiten generar gráficos del tráfico entrante y saliente de una interfaz o una fuente completa (la sumatoria del tráfico de todas las interfaces), identificar las principales aplicaciones y protocolos que generan dicho tráfico, calcular el ancho de banda y porcentaje usado del mismo, entre varias otras funciones relacionadas.

Para la interfaz FastEthernet0/1.1 del router, tan pronto se hace clic en *Interface Details*, se envían una serie de mensajes de solicitud SNMPv2 con la comunidad c0Mmuni7yAdm hacia dicha interfaz del router, conteniendo diferentes OIDs en el campo de *variable bindings* relacionados al tráfico de la interfaz para poder dibujar los datos que aparecen en la consola web. Al mismo tiempo de ingresar a este tipo de consulta de información, se está ejecutando una captura de tráfico con Wireshark para ver la mensajería SNMP que hay entre el computador NMS y el router. A continuación se muestra la información de *Interface Details.*

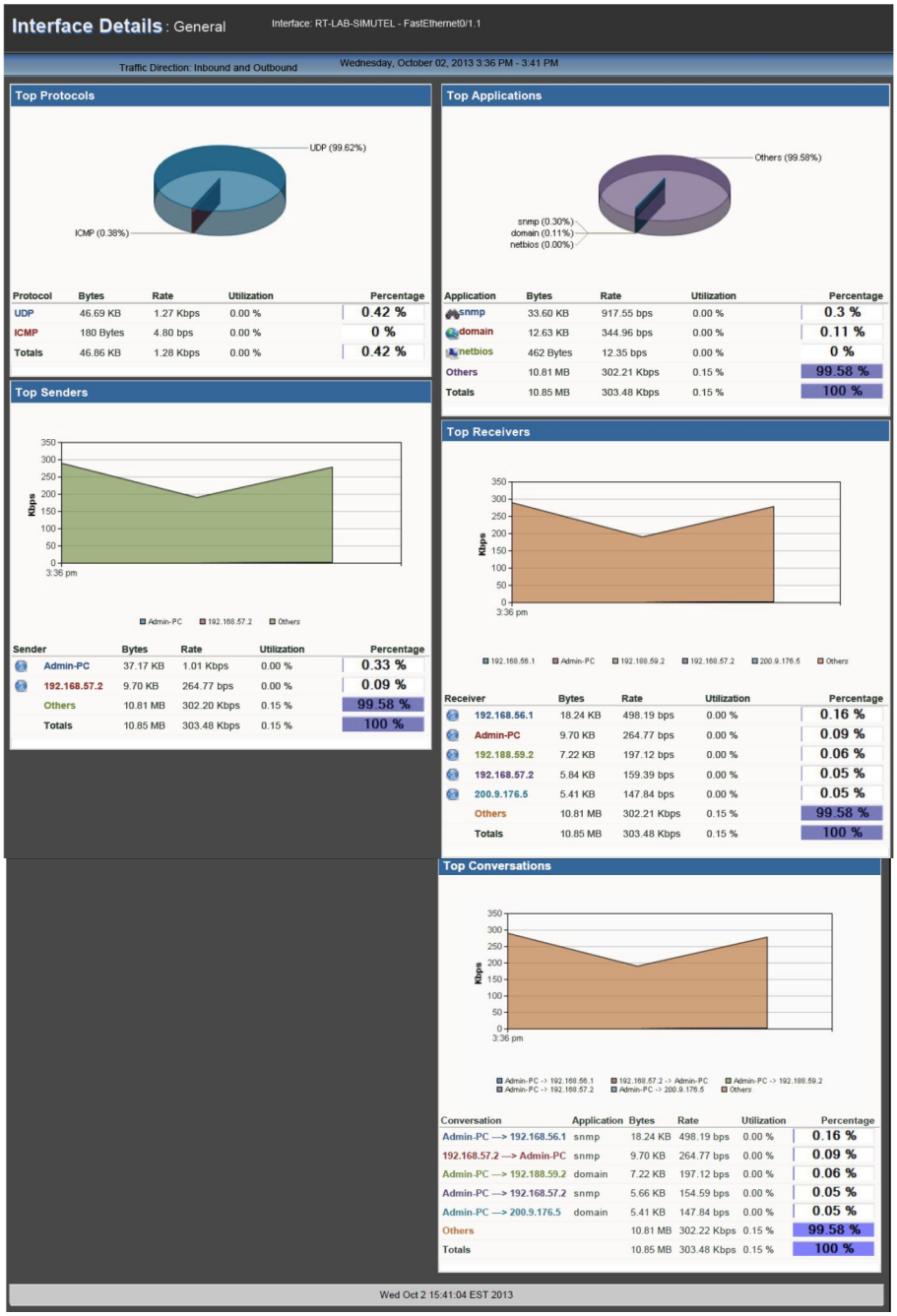


Figura 3.152- Información de *Interface Details* - v2

Como se puede apreciar en la figura 3.153, en Wireshark se han capturado todos los mensajes de solicitud y respuesta que se envían del NMS hacia el router.

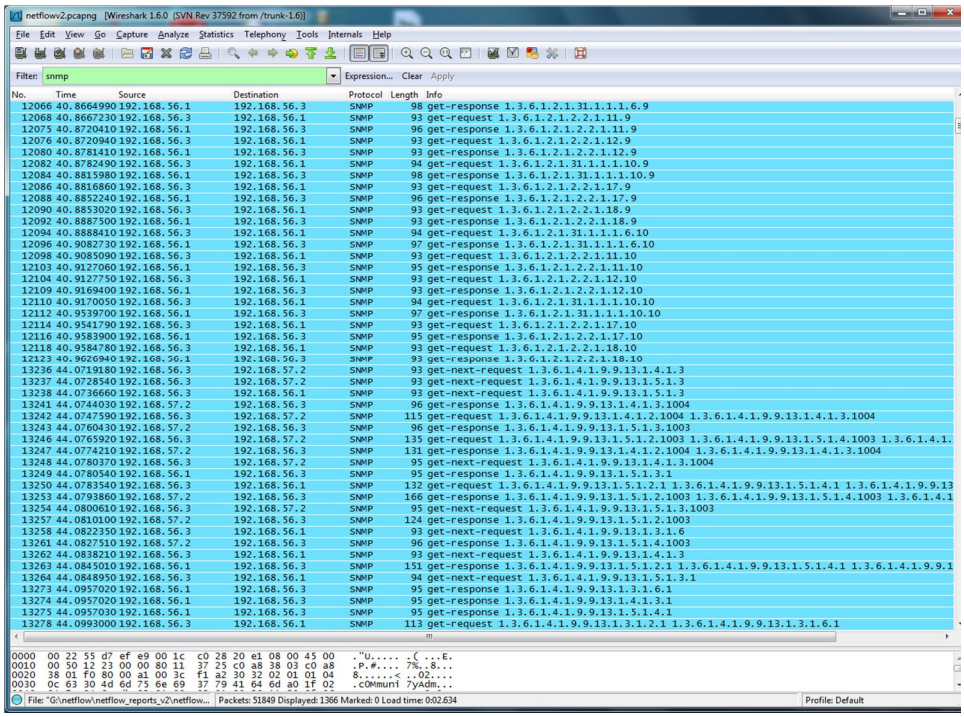


Figura 3.153- Captura de solicitudes y respuestas de la pc al router - v2

Las solicitudes pertenecen a las MIB´s *interface(2)* e *ifMIB(31)* de la mib-2 en su mayoría, que recopilan datos importantes sobre el tráfico que circula en alguna interfaz (bytes enviados, recibidos, errores, velocidad, etc); aunque hay unos mensajes que contienen OIDs propietarios de Cisco, específicamente de la familia *ciscoEnvMonMIB*, que preguntan en general el estado de la fuente de poder y ventilador del equipo. Todas las solicitudes y obviamente, las respuestas a cada una de ellas, vistas de manera individual,

son valores puntuales de enteros, contadores32, contadores64, por mencionar algunos tipos de datos. Entonces, el envío cíclico de las consultas por todos aquellos OIDs permite obtener nuevos valores cada vez y así, poder graficar en la consola de reportes los gráficos que observamos previamente.

3.10.1.1.2 Utilización de CPU

Se monitoreó la actividad del CPU de una PC de la red. Dentro de la consola web de WhatsUp Gold, elegimos la pestaña *Reports*, dentro de la cual nos ubicamos en la sección *Performance* y dimos clic en *CPU Utilization* en el cual se desplegó un gráfico del porcentaje de uso del CPU de nuestra PC Ubuntu en tiempo real.

Por defecto la funcionalidad no aparece vinculada a ningún dispositivo en especial, si queremos que se despliegue el reporte de uso de CPU para alguno específico, se hace clic en la carpeta *All Devices* y dentro se elige de la lista el equipo que se quiera monitorear. También es necesario asegurarse que la PC esté siendo consultada con la comunidad c0Mmuni7yAdm de

SNMPv2, en las propiedades del dispositivo, pestaña *Credentials* como se vio en la sección 3.5.3.

Una vez definida la PC de la que se quiere obtener estadísticas del uso del CPU, en la parte inferior de la interfaz de *CPU Utilization* observamos un gráfico que está siendo actualizado en tiempo real de acuerdo a las transacciones que realiza el CPU; para esto WhatsUp Gold envía varias solicitudes hacia la computadora pertenecientes a la MIB *hrDevice*, específicamente la tabla de dispositivos (*hrDeviceTable*) y procesadores (*hrProcessorTable*). Dichas solicitudes serán repetidas mientras permanezcamos en la interfaz de *CPU Utilization*, lo que permite actualizar los valores y así ir graficando el reporte.

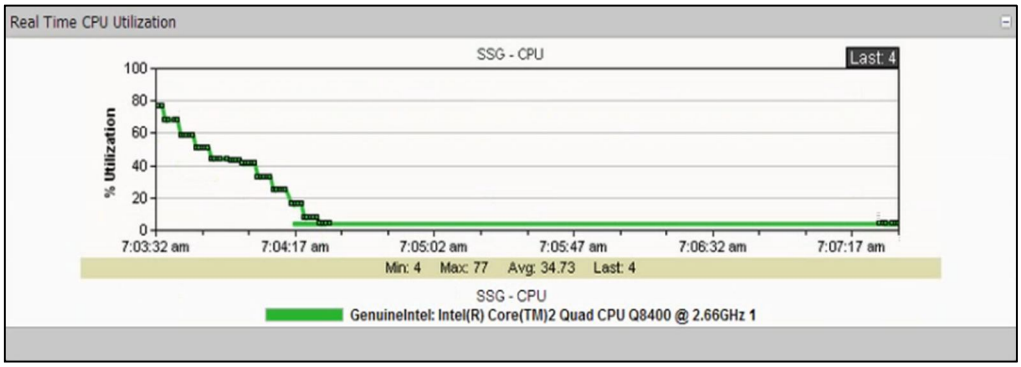


Figura 3.154- Gráfico de datos de utilización del CPU - v2

Mientras se realiza esta operación, se ejecutó Wireshark para capturar el tráfico de las solicitudes y respuestas SNMP que WhatsUp Gold está generando.

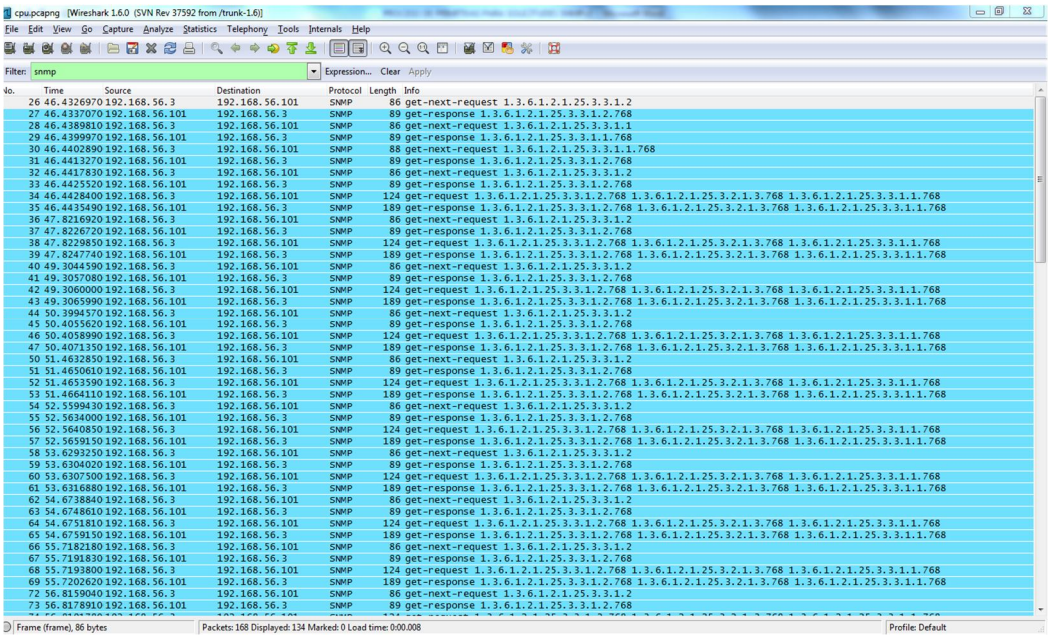


Figura 3.155- Captura de tráfico de solicitudes y respuestas del CPU - v2

3.10.1.2 Solicitudes SNMP a equipos monitoreados usando v3

3.10.1.2.1 Flow Monitor

Dentro de la consola web de WhatsUp Gold, elegimos la pestaña *Flow Monitor* para configurar y desplegar los diferentes reportes gráficos sobre tráfico de interfaces de red. En la sección *Configuration*, dentro de *Settings*,

verificamos que la fuente de la información de tráfico es el router y vamos a consultarlo, a la vez que nos enviará la información, con el usuario Root. Como se sabe, los mensajes que genera este tipo de usuario (authPriv), realizan una autenticación cada vez que arriban a una entidad SNMPv3 (de ida o vuelta) aparte de cifrar su contenido.

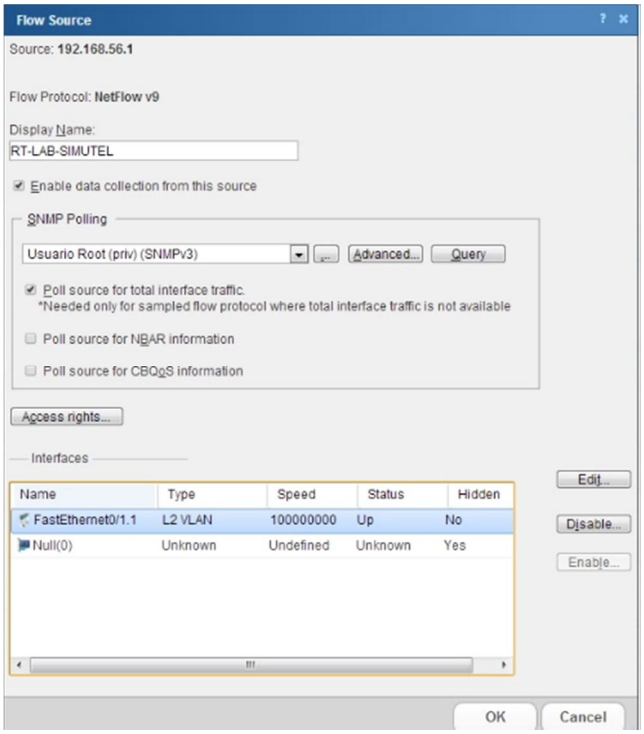


Figura 3.156- Configuración de captura de tráfico con usuario Root

La mecánica de la prueba no cambia mucho con respecto a la demostración efectuada para SNMPv2. Dentro de la pestaña *Flow Monitor* del programa, escogemos *Interface Details* de la sección *Reports*, para poder visualizar todas las estadísticas de la interfaz del router que estamos monitoreando, la FastEthernet0/1.1.

Tan pronto escogemos esta opción, se enviarán simultáneamente varias solicitudes SNMPv3 al router con los mismos OIDs de la *ifMIB* y de la *ciscoEnvMonMIB* que se enviaron en la prueba anterior; además de enviarse cada cierto tiempo para permitir un muestreo aceptable y así poder realizar las gráficas. El reporte generado se muestra en la figura 3.157.

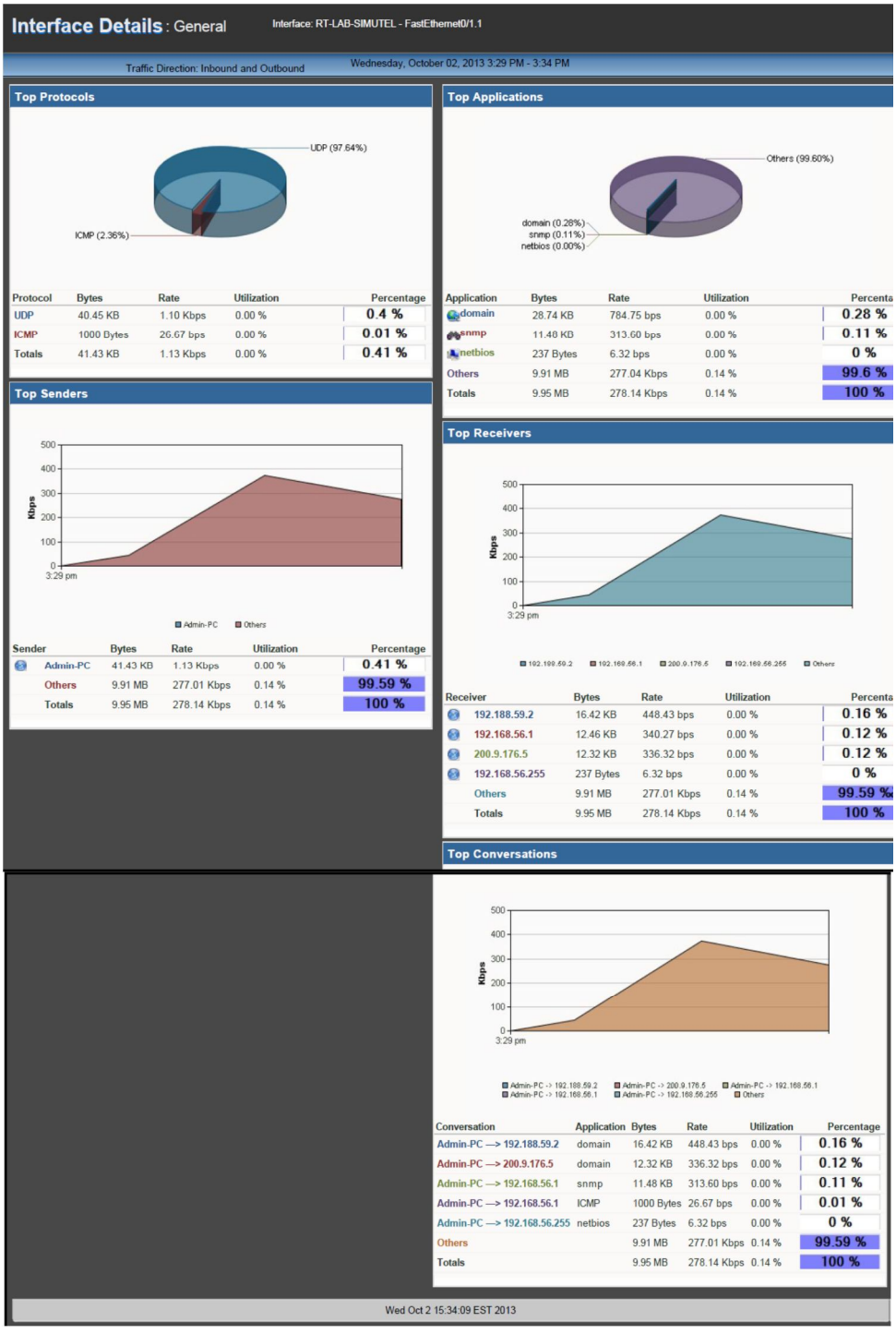


Figura 3.157- Información de *Interface Details* - v3

Prácticamente son los mismos gráficos y cantidades desplegadas, el que puedan variar significativamente depende de la cantidad de tráfico y conexiones simultáneas que tenga la interfaz que esté siendo consultada, lo que en nuestro caso es mínimo. De la misma forma, que la interfaz del dispositivo sea consultada por alguna comunidad o usuario SNMPv3 determinado, no influye en nada en cómo se muestran los resultados en la consola.

Como se observa en la figura 3.158, en Wireshark se ha capturado todo el tráfico mientras se desplegaba el reporte de *Interface Overview*. Como se analizó en la sección 3.9.1 la consulta hacia el router por una variable específica usando el programa SNMP JManager, implica un intercambio de mensajes consecutivos entre ambas entidades SNMP, debido a que primero el NMS debe conocer detalles del usuario remoto al que le solicita la información como su engineID y parámetros de tiempo; en la figura 3.158 se aprecia que el proceso es más directo en WhatsUp, un mensaje por solicitud y uno por respuesta, ya no existen los reports informativos; debido a que con el descubrimiento inicial de los dispositivos y posteriores consultas, el programa ya tiene almacenados los engineID de los dispositivos con los cuales se comunica y ya tiene sincronización con ellos.

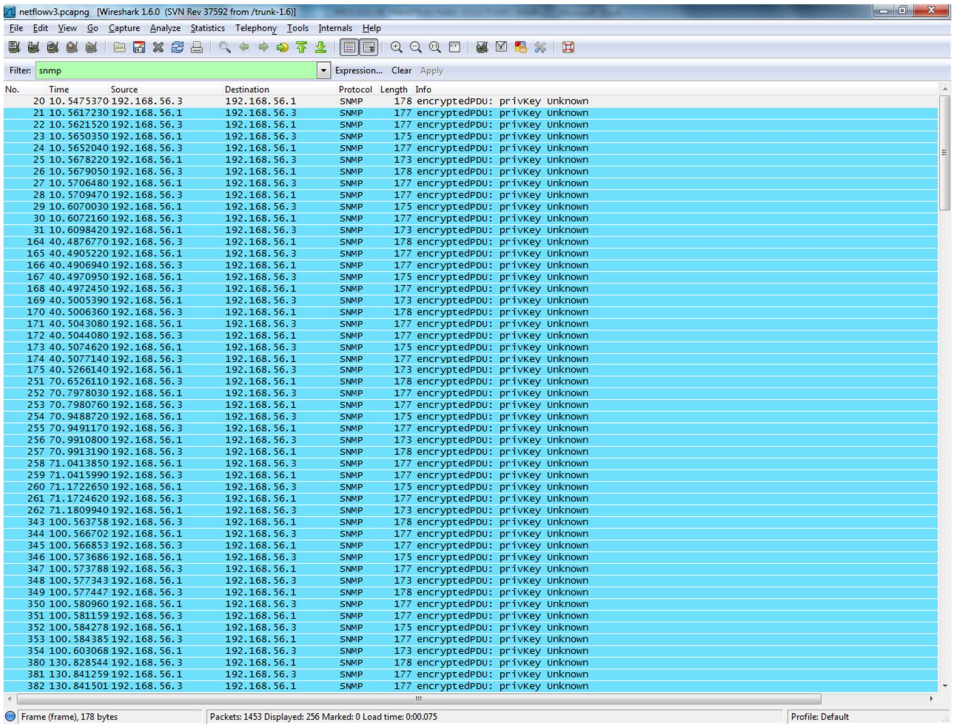


Figura 3.158- Captura de solicitudes y respuestas de la PC al router - v3

3.10.1.2.2 Utilización de CPU

En las propiedades de la PC Ubuntu, pestaña *Credentials*, verificamos que esté seleccionada la credencial del usuario Root. Se accedió a la misma funcionalidad de *CPU Utilization* dentro de la pestaña *Reports* en la consola web de WhatsUp Gold para la PC Ubuntu, y se obtuvo en la parte inferior de la pantalla el gráfico de uso de CPU en tiempo real para la computadora monitoreada.

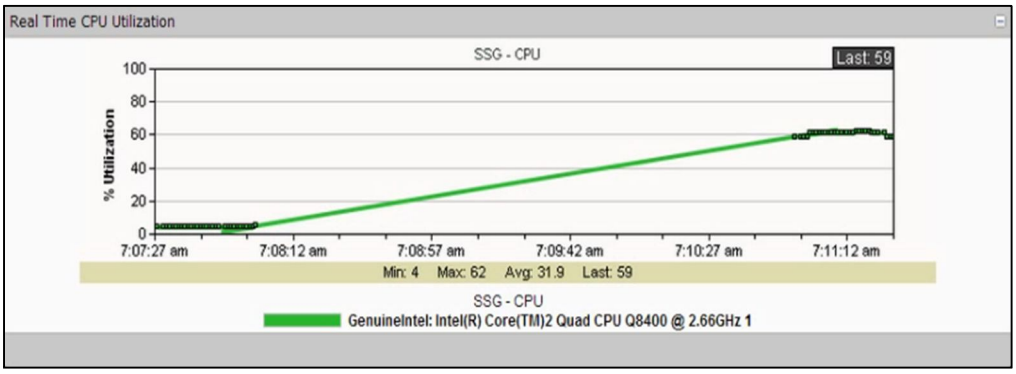


Figura 3.159- Gráfico de datos de utilización del CPU - v3

Los OIDs que son consultados son los mismos que vimos en el despliegue cuando se consultó a la PC usando la versión 2, es decir los pertenecientes a la MIB *hrDevice*. Además como se vio en el ejemplo anterior las consultas se repiten cada cierto tiempo para poder graficar con detalle el reporte de uso del CPU, lo que se puede apreciar en la captura de Wireshark mostrada en la figura 3.160.

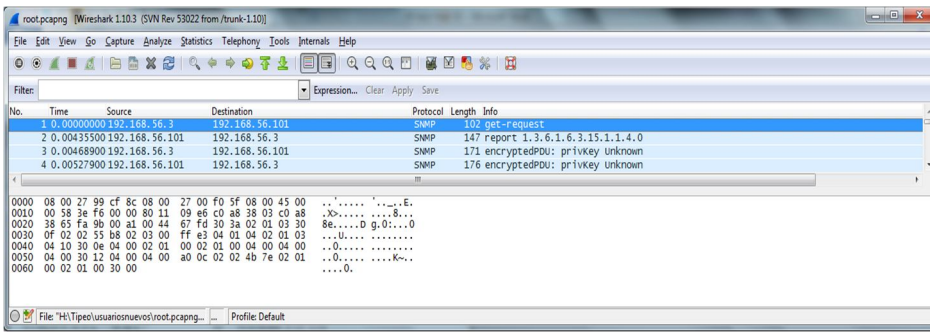


Figura 3.160- Captura de tráfico de solicitudes y respuestas del CPU - v3

3.10.2 Escenario 2: pruebas de solicitudes de escritura SNMP.

En esta sección se analizó en detalle las primitivas que se envían a una agente SNMPv3 remoto cuando por medio de alguna aplicación de monitoreo SNMP, se modifica el valor de alguna de sus variables. Se capturaron los paquetes usando Wireshark, y se observó la posibilidad que el contenido del PDU de datos esté expuesto o no, dependiendo si la petición de set se realiza con una comunidad o un usuario de determinado nivel de seguridad. Se realizaron dos ejemplos, el primero un set-request usando SNMPv2 y el segundo el mismo set-request con SNMPv3, ambos con la aplicación SNMP JManager, para poder cambiar el hostname del router y capturar los paquetes con Wireshark.

3.10.2.1 Solicitudes SNMP a equipos monitoreados usando v2.

Se puede apreciar que el router tiene definido el nombre RT-LAB-SIMUTEL.

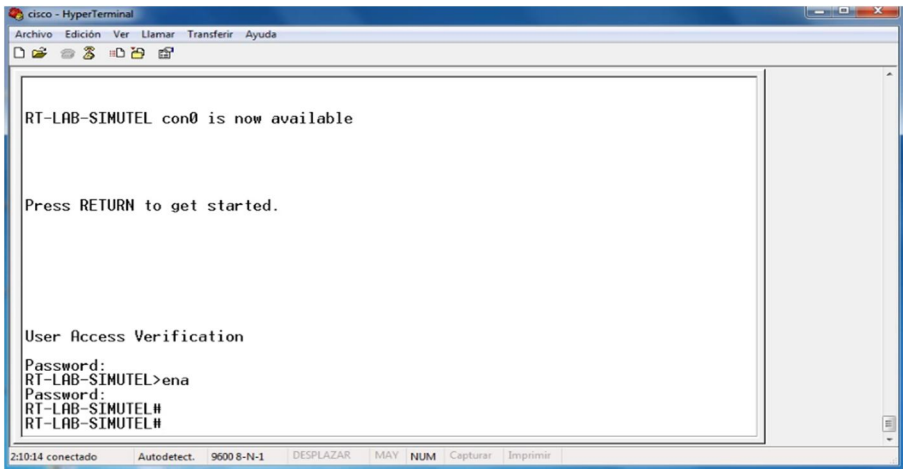


Figura 3.161- Nombre del router pre-definido

Para poder realizar cualquier cambio a alguna de las variables contenidas en la MIB de algún equipo de la red, es necesario utilizar la comunidad de escritura, en nuestro caso la comunidad read-write c0Mmuni7yAdm que configuramos en las opciones de conexión del programa SNMP JManager.

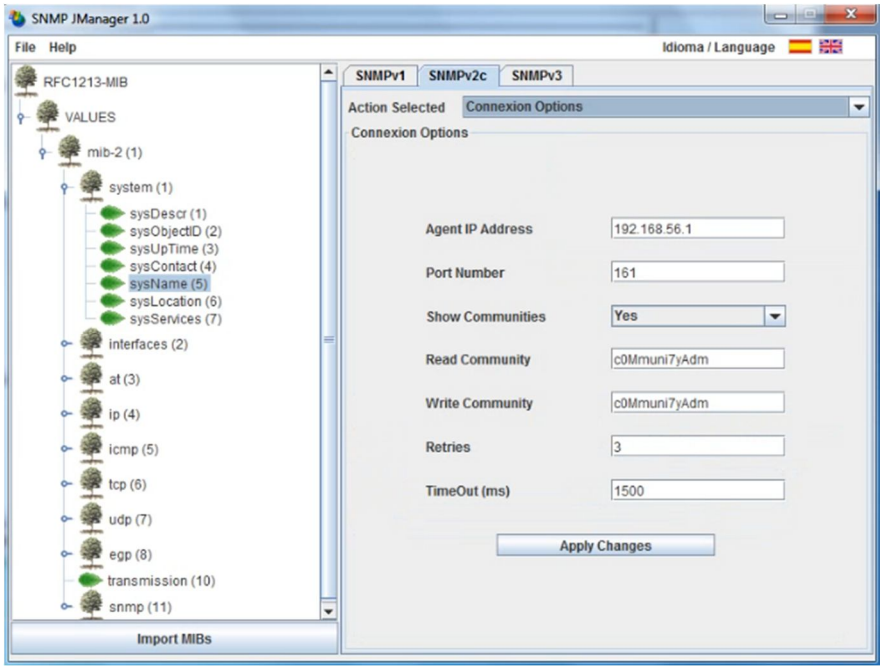


Figura 3.162- Configuración para realizar un set con comunidad

Para cambiar el hostname del router modificamos el objeto sysName cuyo OID es 1.3.6.1.2.1.1.5.0 , su valor es del tipo octect string (cadena de caracteres) y en el campo valor escribimos el nuevo nombre que deseamos que el router tenga. Al presionar Set, en el área Objects de la ventana, aparece la respuesta que el router nos envía con el nombre que seteamos para el router, lo que indica una operación exitosa, como aparece en la figura

3.163. La aplicación del set-request es inmediata, ya que en el router aparece un mensaje de aviso de una configuración hecha vía SNMP (Fig. 3.164) y luego aparece el nuevo nombre “CISCO2800”.

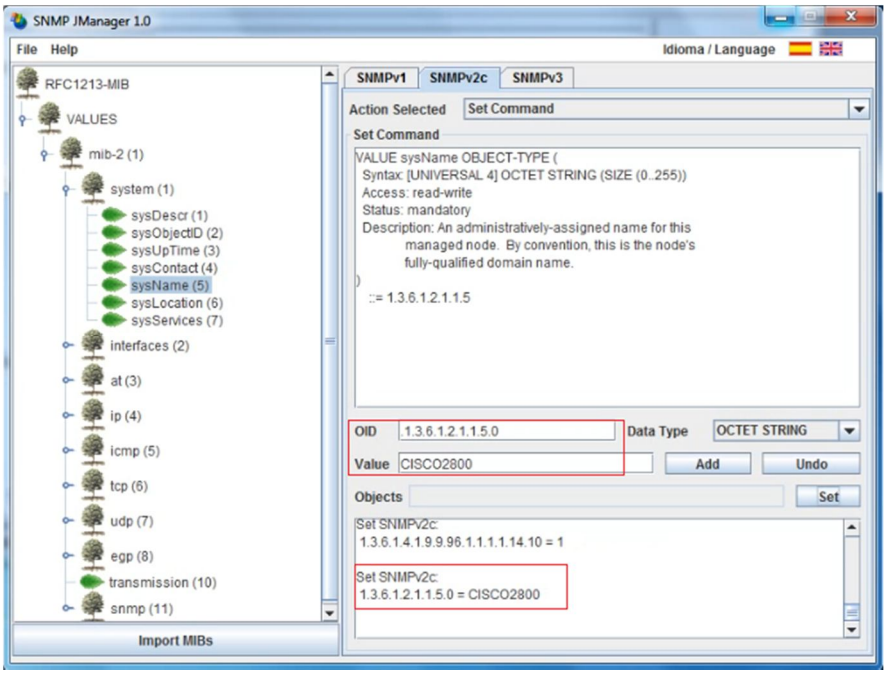


Figura 3.163- Modificación del nombre del router

```
RT-LAB-SIMUTEL#
*Oct  2 20:52:00.891: %SYS-5-CONFIG_I: Configured from 192.168.56.3 by snmp
CISCO2800#_
```

Figura 3.164- Mensaje de configuración en el router

3.10.2.2 Solicitudes SNMP a equipos monitoreados usando v3.

El router inicialmente tiene definido el nombre RT-LAB-SIMUTEL, como se muestra en la figura 3.161. En las opciones de conexión de SNMP JManager se configura el usuario Root de nivel de seguridad authPriv, además de los protocolos y contraseñas de autenticación y cifrado (que deben ser las mismas configuradas en las entidades remotas osea en el router).

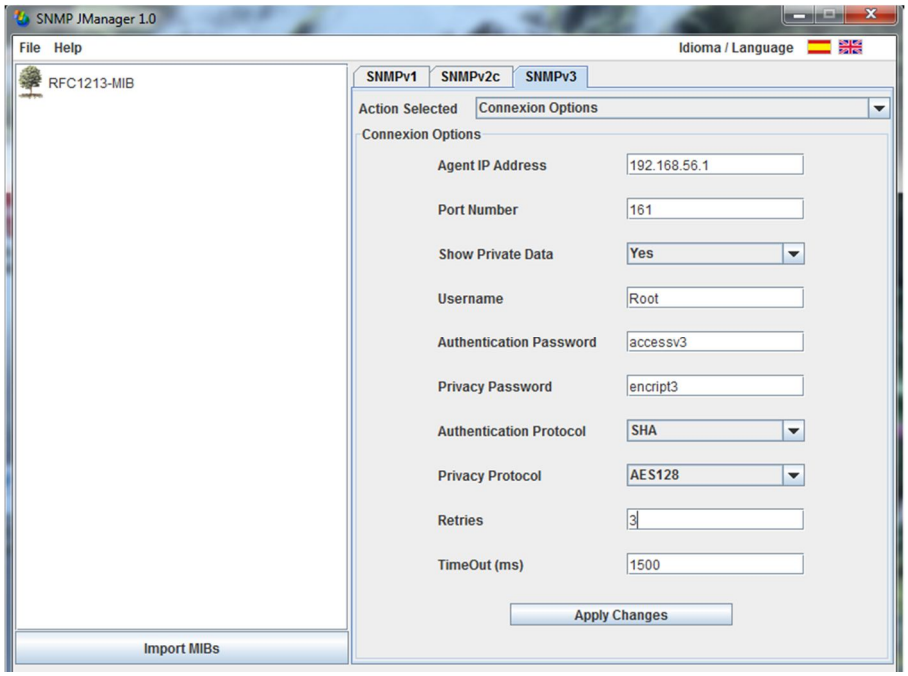


Figura 3.165- Configuración para realizar un set con usuario Root

Al hacer clic en Set, se observa que en área Object de la aplicación aparece el mensaje enviado por el router en respuesta a nuestra solicitud de escritura, lo que indica que el cambio de nombre fue exitoso (Fig. 3.166).

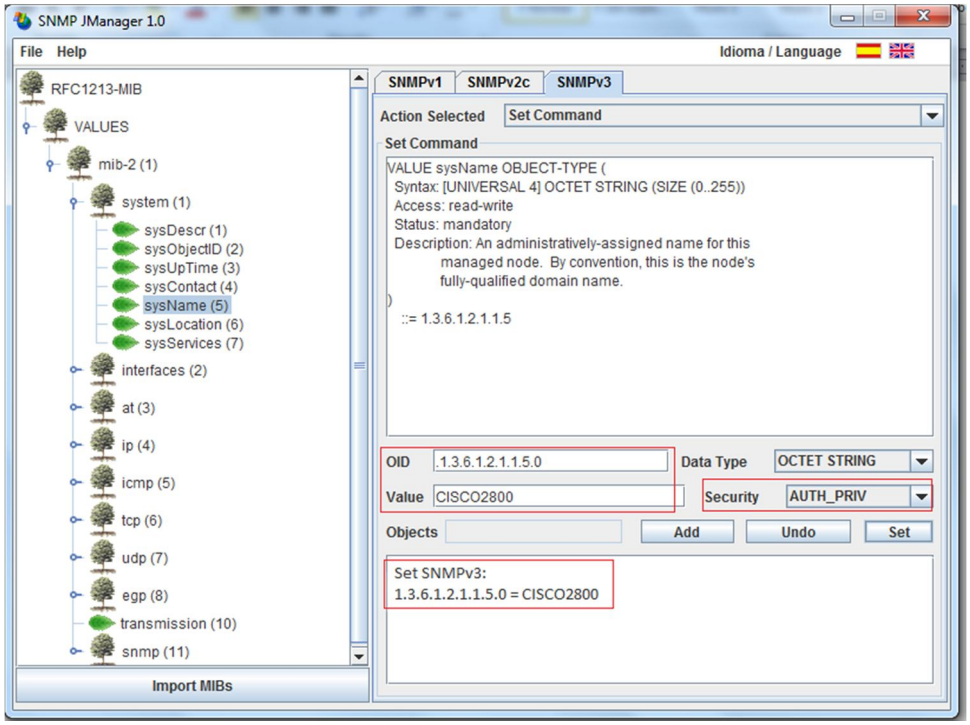


Figura 3.166- Modificación del nombre del router con usuario Root

Mientras que en el router se observó un mensaje de cambio de configuración realizado a través de SNMP, y el inmediato cambio de nombre.

```
RT-LAB-SIMUTEL#
*Oct  2 20:52:00.891: %SYS-5-CONFIG_I: Configured from 192.168.56.3 by snmp
CISCO2800#_
```

Figura 3.167- Mensaje de configuración en el router

3.10.3 Escenario 3: pruebas de traps v2 y v3 recibidas.

En las secciones 3.2.3.1 y 3.5.4.1 hemos visto cómo se activan y configuran los traps en los equipos Cisco y en WhatsUp Gold respectivamente. En esta sección veremos específicamente la ejecución de las traps.

Las traps que han sido configuradas son las siguientes:

```
snmp-server enable traps snmp authentication linkdown linkup
clodstart warmstart
snmp-server enable traps envmon
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps entity
snmp-server enable traps cpu threshold.
```

Intencionalmente deshabilitamos la conexión de red, y luego la volvimos a habilitar, para recibir traps de linkDown, linkUp, entre otras. Aparte de eso, hicimos algunas configuraciones en el router, como por ejemplo, enviamos el comando shutdown en una interfaz y luego la volvimos a habilitar. Luego nos

desplazamos al menú Logs > SNMP trap de WhatsUp Gold para poder ver las traps recibidas.

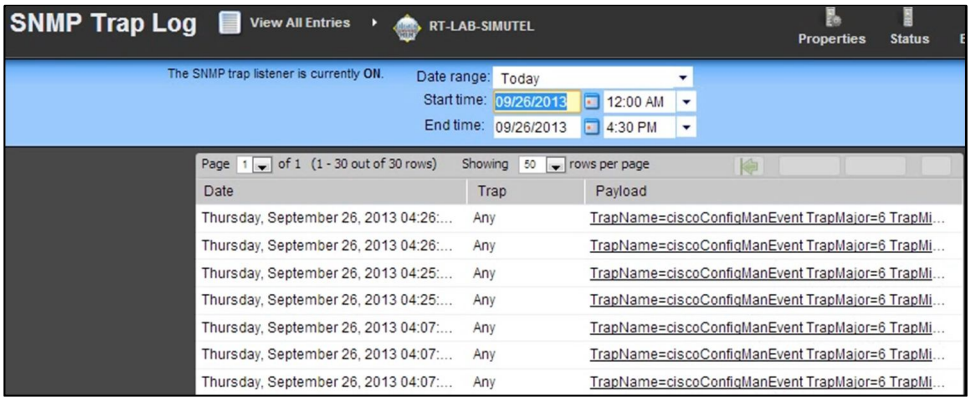


Figura 3.168- Traps recibidas por medio de WhatsUp Gold

Podemos observar en la figura 3.168 algunas traps de la gran cantidad que recibimos, unas de configuración de eventos, otras de linkUp y linkDown, etc.

CAPÍTULO 4

4 ANÁLISIS DE RESULTADOS SNMPv3

Este capítulo presenta los resultados de los escenarios del capítulo 3. Se explora en profundidad el contenido de los paquetes capturados por Wireshark y se hace la comparación y análisis de las principales diferencias de un mensaje SNMPv2 con respecto a uno SNMPv3. El esquema del desarrollo se muestra en la siguiente figura:

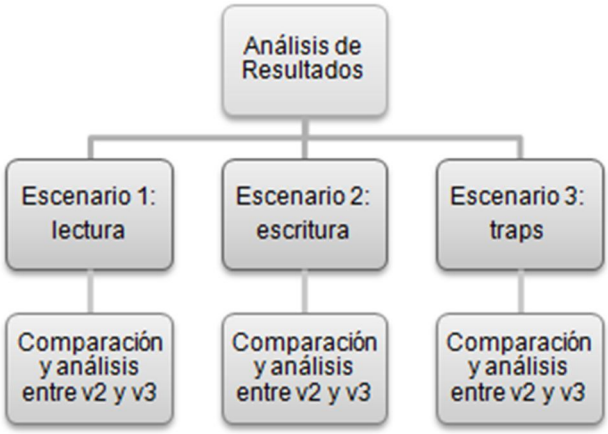


Figura 4.1- Análisis de resultados de escenarios de tráfico

4.1 Comparación de capturas obtenidas en solicitudes de lectura

4.1.1 Flow Monitor

El uso de una comunidad SNMPv2 para el análisis de tráfico permite que podamos analizar el contenido de cada uno de los mensajes capturados en Wireshark. De las múltiples peticiones realizadas al router, tomaremos un par solicitud/respuesta a manera de ejemplo. El mensaje número 11.756 solicita al router el estado operativo de la interfaz Fa0/1.1 y la respuesta del router se puede ver en el mensaje 11.760.

Como vimos en la sección 2.5.2 un mensaje SNMP contiene su respectiva versión, en este caso el mensaje de solicitud tiene la v2c (1), nuestra comunidad c0Mmuni7yAdm, el request-id 14.545 que permite emparejar la solicitud con el futuro mensaje de respuesta (sobre todo si son varias solicitudes simultáneas como en este caso), indicadores de error (en este caso, no hubo error de ningún tipo) y las variables (objetos con sus valores respectivos). Dentro del campo de variables (*variable bindings*) se encuentra el OID 1.3.6.1.2.1.2.2.1.8.1 con el valor *NULL* que luego será reemplazado en el agente del router por el valor que corresponda.

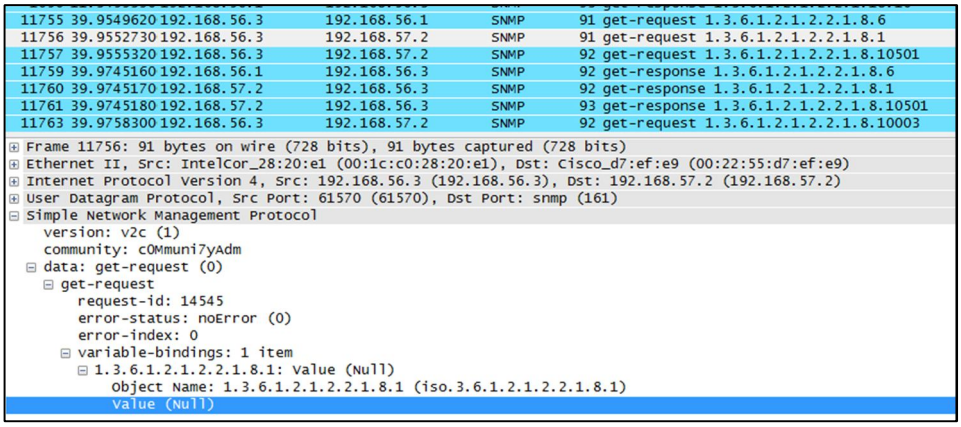


Figura 4.2- Paquete solicitud datos de tráfico interfaz - v2

Este OID corresponde al objeto *ifOperStatus*, tiene al final indexado el número que le corresponde a la interfaz consultada, en este caso la Fa0/1.1 le corresponde el índice 1, y así las demás (sub)interfaces del router tendrán cada una asignados sucesivos índices. El OID indica el estado operacional

actual de la interfaz Fa0/1.1. Los posibles estados son: *up (1), down (2), testing (3), unknown (4), dormant (5), notPresent (6) y lowerLayerDown (7).*

El mensaje de respuesta que el router envía a la PC tiene el mismo contenido, salvo que el campo *variable bindings* ya contiene el valor de respuesta asociado al OID 1.3.6.1.2.1.2.2.1.8.1. El valor 1 significa que la interfaz consultada está habilitada (*up*).

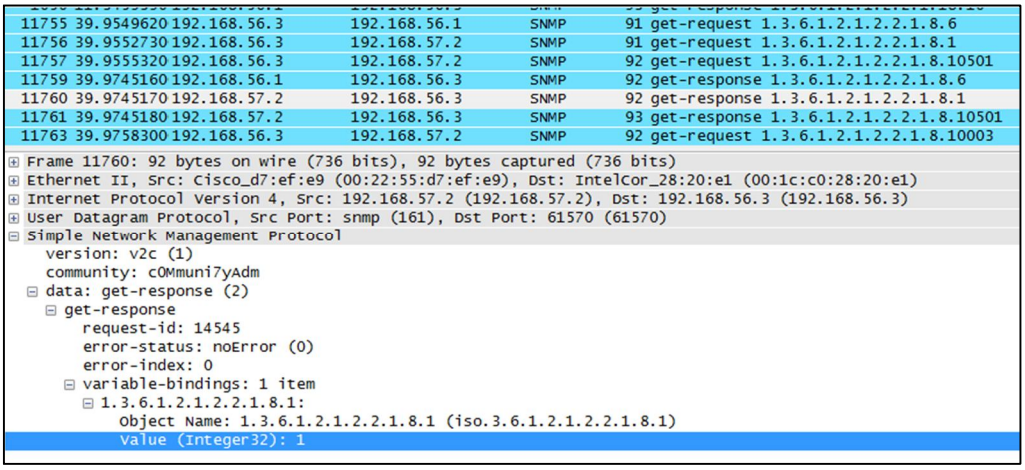


Figura 4.3- Paquete respuesta datos de tráfico interfaz - v2

Al configurar en Netflow un usuario SNMPv3 de nivel de seguridad authPriv, logramos que los mensajes de consulta y respuesta de Netflow estén vinculados a procesos de protocolo encargados de su autenticación en ambas entidades y del total cifrado de su contenido de datos. Es por eso que a diferencia de la captura de Wireshark realizada en el mismo despliegue

para el router en versión 2, acá observamos que la totalidad de los mensajes están cifrados.

El paquete número 20 (Fig. 4.4) corresponde a la solicitud y el número 21 a la respuesta recibida, ya que tienen el mismo msgID. En la sección 2.6 se analizó la teoría relacionada al encabezado de un mensaje SNMPv3, ahora con un ejemplo real analizando uno de los paquetes capturados (solicitud) se refuerzan aquellos conceptos.

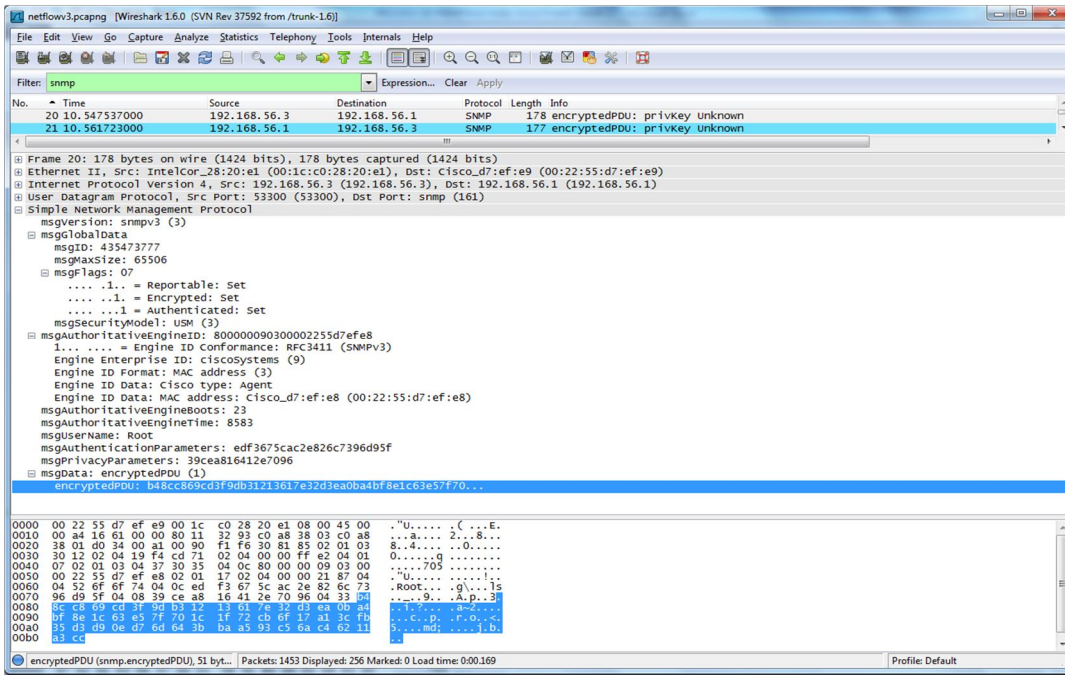


Figura 4.4- Paquete solicitud datos de tráfico interfaz - v3

1. El msgVersion es 3 ya que el paquete pertenece a la versión 3.
2. Datos globales, msgGlobalData que contiene:
    - Identificador de mensaje (msgID) igual a 435’473.777 que será el mismo de la respuesta que arribe.
    - Máximo tamaño de mensaje (msgMaxSize) hasta 65.506 bytes que la máquina administradora de red o NMS puede procesar sin problemas.
    - Banderas (msgFlags). En tres bits se indican tres banderas, como vemos tiene un valor de 111 (7 en decimal) lo que indica que este mensaje es reportable, espera una respuesta por parte del receptor; tiene habilitada la autenticación, el mensaje tendrá que someterse a un proceso de autenticación en la entidad remota y además, tiene habilitada el cifrado, el mensaje se cifra totalmente antes de partir hacia el destino.
    - Modelo de seguridad (msgSecurityModel). Modelo de seguridad basado en usuarios (USM) vinculado a SNMPv3.
3. Parámetros de seguridad: Incluyen los de autenticación y cifrado.
    - Identificador del motor SNMP autoritativo (msgAuthoritativeEngineID). Tiene el valor 800000090300002255d7efe8 que identifica únicamente al router que contiene al motor SNMP que responderá a las solicitudes (autoritativo).

Las siguientes líneas al identificador son un desglose de cómo se forma este engineID de 12 octetos de longitud, formato hexadecimal. [18]

La primera línea que le sigue al engineID indica el valor del primer bit del primer octeto, el cual es el binario 10000000 (80h) indicando que el identificador se formó según las reglas del RFC3411.

La segunda línea indica el número asignado por la IANA a Cisco, el 9, cuyo equivalente binario se escribe dentro de los 4 primeros octetos, rellenándose con bits aleatorios los espacios vacíos.

La tercera línea indica el contenido del quinto octeto, MAC Address (3) indica que del sexto al doceavo octeto se rellenará los bits con la dirección MAC del motor autoritativo, completándose los espacios vacíos con octetos aleatorios, ceros la mayoría de las veces. IANA (Autoridad de Asignación de Números en Internet) define números entre 6 a 127 en el quinto octeto para múltiples formas de completar el engineID, entre algunas otras direcciones IPv4, IPv6, texto, etc.

La cuarta y quinta línea indican que el motor autoritativo es un agente Cisco y definen la dirección MAC 00:22:55:d7:ef:e8 con la que se completó del octeto 6 al 12.

- Reinicios del motor autoritativo (msgAuthoritativeEngineBoots). El motor autoritativo ha reiniciado 23 veces.
- Tiempo del motor autoritativo (msgAuthoritativeTimeBoots). Han pasado 8583 segundos desde la última re-inicialización del motor SNMP.
- Usuario (msgUserName). Root es el usuario al cual se le consulta y el que nos responderá la petición. Cabe recalcar que el poder ver al usuario no constituye problema porque hay muchos otros factores que intervienen en la seguridad de la comunicación entre motores SNMP y no solo los usuarios, como sí sucede en cambio con las comunidades de SNMPv1 y v2c.
- Parámetros de autenticación y privacidad (msgAuthenticationParameters, msgPrivacyParameters), contienen los valores que este motor usa para autenticar y cifrar sus mensajes cuando se comunica con otro motor SNMP, como se indicó en las secciones 2.4.3.5 y 2.4.3.6.

4. scopedPDU. Esta parte del mensaje es la que se cifra.
    - Identificador del motor SNMP de contexto (contextEngineID). Tiene la misma información del motor SNMP autoritativo, es decir, el mismo engineID.

- Nombre del contexto. No hay definidos contextos (la mayoría de implementaciones los omiten) por lo que no aparece un nombre definido.
- PDU de datos. Es la información como tal, contiene todos los *variable bindings* consultados en este caso.

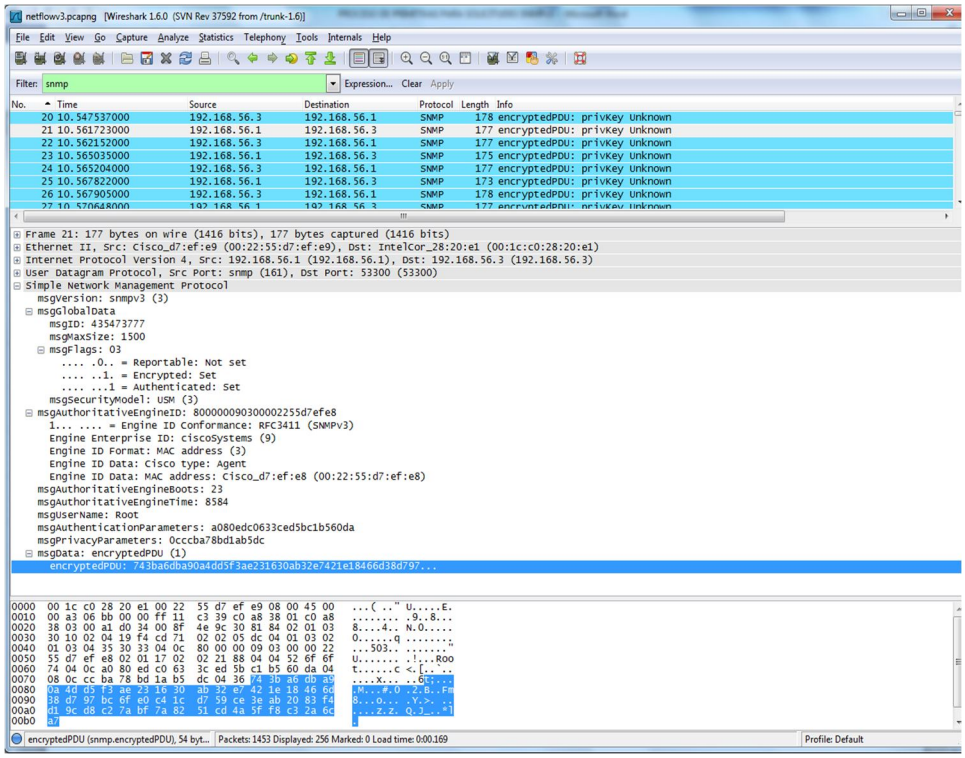


Figura 4.5- Paquete respuesta datos de tráfico interfaz - v3

En la imagen 4.5 correspondiente a la respuesta del router, observamos que coincidió el msgID con el de la solicitud y el campo de datos está cifrado. Es importante recalcar que en el caso de la solicitud lo que se codifica es el contenido del mensaje con los OIDs que serán consultados al router pero los

valores correspondientes a cada OID al inicio están en NULL y en la respuesta en cambio se cifra el contenido del mensaje constituido por los OIDs ya con su respectivo resultado; en ambos casos, ya sea en la solicitud o respuesta, esos datos son sensibles y deben ser protegidos.

4.1.2 Utilización de CPU

Se muestra la captura de Wireshark usando la comunidad de SNMPv2. Los paquetes 34 y 35 corresponden a una solicitud y respuesta respectivamente. En el get-request (Fig. 4.6), podemos apreciar que contiene en el campo *variable bindings*, tres pares de objetos con los valores NULL para que sean respondidos en el agente de la máquina Ubuntu.

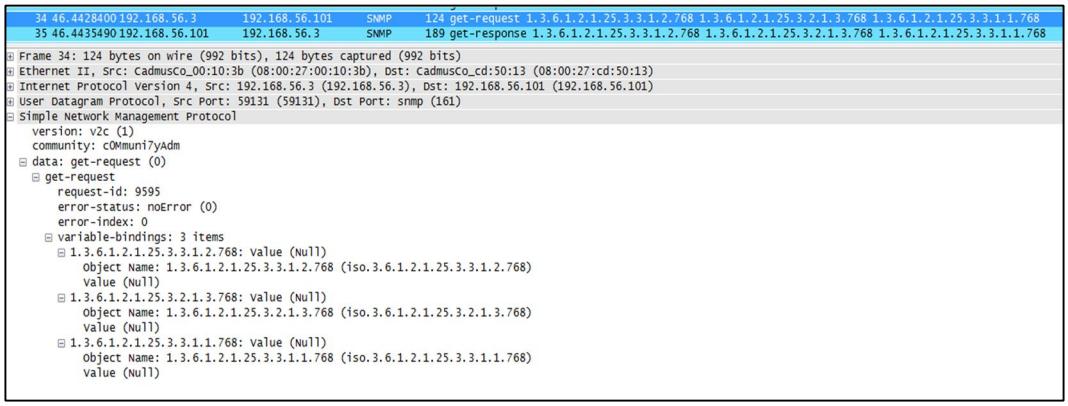


Figura 4.6- Paquete solicitud datos de uso CPU - v2

El primer objeto corresponde a *hrProcessorLoad*, que recupera el valor del porcentaje de uso del procesador en el último minuto; el segundo corresponde a *hrDeviceDescr* que describe textualmente al dispositivo, incluyendo al fabricante, revisión y posiblemente su serial; finalmente el tercer objeto es *hrProcessorFrwID* que es el ID de firmware asociado con el procesador. Todos estos OIDs tienen al final indexado un número, que es el correspondiente a hrDeviceIndex, tanto para los objetos pertenecientes a la tabla de dispositivos hrDeviceTable como para los objetos de la tabla de procesadores hrProcessorTable. Entonces, el número 768 al final de los tres OIDs dentro del campo *variable bindings* de la solicitud, indica una instancia particular para esos objetos y es un número aleatorio que el agente asigna a un procesador (si hay más de uno, se les asignarán otros números) y que permanece constante al menos, hasta el próximo reinicio del agente.

A continuación se muestra el detalle del mensaje de respuesta.

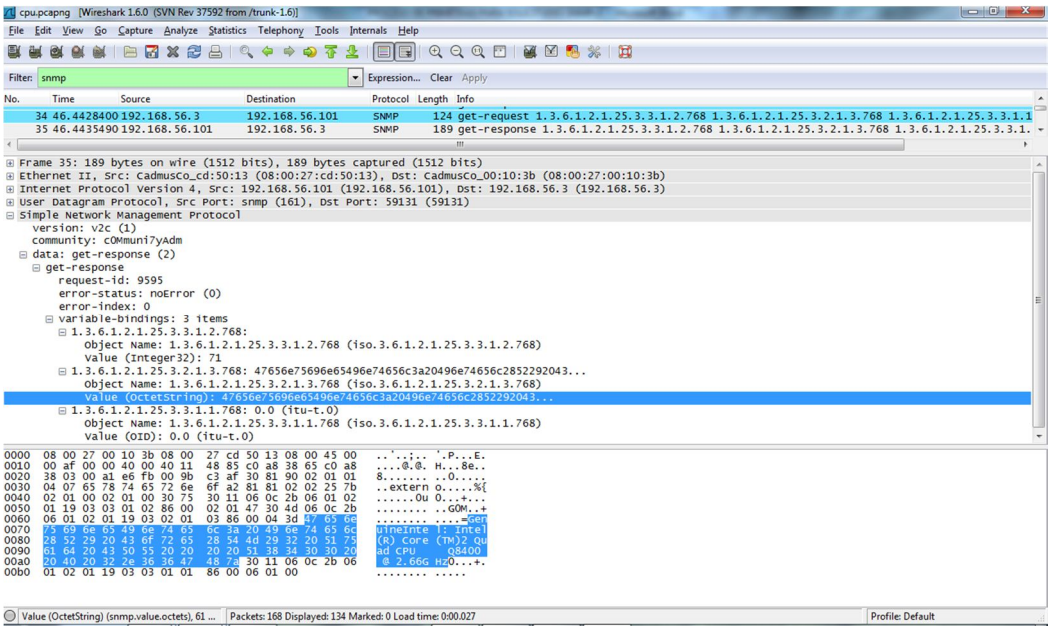


Figura 4.7- Paquete respuesta datos de uso CPU - v2

Vemos que la carga del procesador para el último minuto ha sido del 71%, que la descripción del procesador es una cadena de octetos cuyo detalle se observa en la ventana de bytes de paquete en el área sombreada, y es el mismo que aparece en la parte inferior del gráfico del uso de CPU en la consola web (detalles del procesador Intel); y que no hay un ID de firmware, ya que se despliega zeroDotZero (0.0) que la IETF define como un valor usado para identificadores nulos. [34],[35]

La situación cambia al usar un usuario SNMPv3. En WhatsUp al estar la PC vinculada con una credencial de nivel de seguridad authPriv como es Root, se logra que sus mensajes de solicitud y respuesta realicen una debida autenticación y además el cifrado de la PDU de datos. Se pudo apreciar que los paquetes capturados en Wireshark están cifrados, igual que en el despliegue de datos de Flow Monitor usando la versión 3.

Tomamos los mensajes 3 y 4 a manera de ejemplo para su análisis, el detalle de cada uno de los campos que componen el encabezado y contenido de los mensajes SNMPv3 fue revisado en el ejemplo anterior con Netflow, ahora se mencionará algunos aspectos importantes solamente. El mensaje 3 corresponde a la solicitud (Fig. 4.8):

- Su msgID es 21.945 que será el mismo en el mensaje respuesta que le corresponda.
- Identificador del motor SNMP autoritativo (msgAuthoritativeEngineID). Tiene el valor 80001f88804fdfdc455ce79c52 que identifica únicamente al host Ubuntu que contiene al motor SNMP que responderá a las solicitudes (autoritativo).

El número asignado por la IANA a Net-SNMP es el 8072, que se escribe en formato hexadecimal dentro de los 4 primeros octetos, lo que se observa específicamente en el tercer y cuarto octeto: 1f88. El contenido del quinto octeto, Net-SNMP Random (128) indica que del sexto al doceavo octeto se rellenará los bits con un valor aleatorio (4fdfdc45) generado por el motor SNMPv3 al momento de iniciar.

- El msgUserName es Root y el msgData, que contiene las variables que son solicitadas con su valor cada una (NULL inicialmente), y que como vemos, es un PDU cifrado.

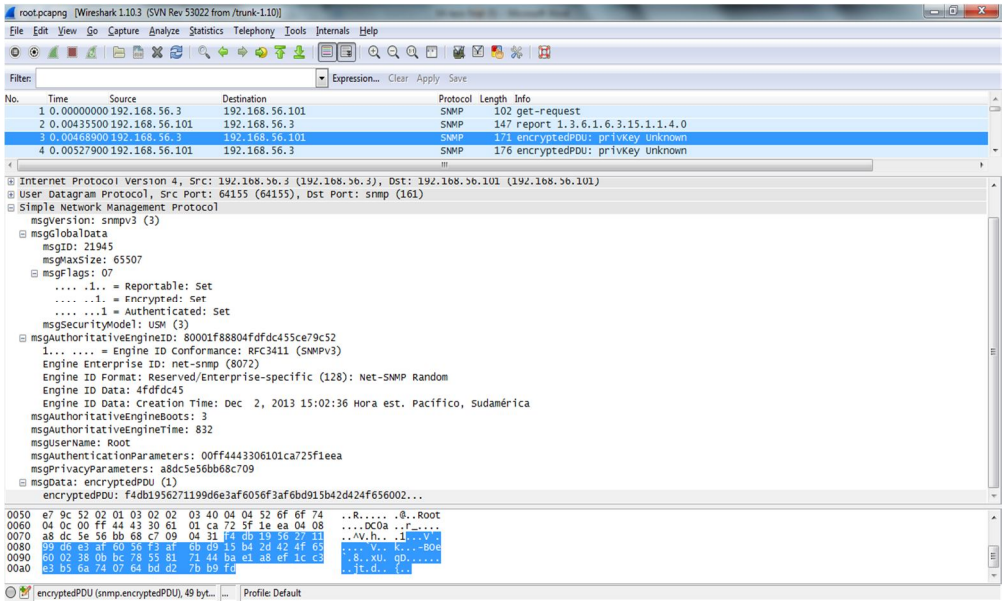


Figura 4.8- Paquete solicitud datos de uso CPU - v3

El paquete 4 (Fig. 4.9) se comprueba es el de respuesta a la solicitud debido a que su msgID coincide con el de petición y el campo de msgData que contiene todos los OIDs con los valores de respuesta está cifrado.

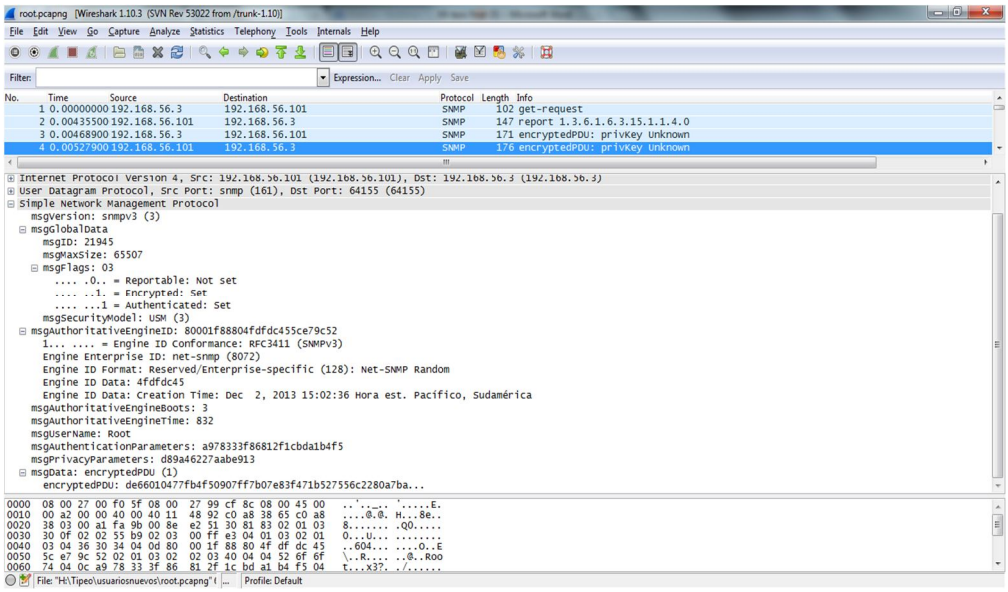


Figura 4.9- Paquete respuesta datos de uso CPU - v3

4.2 Comparación de capturas obtenidas en solicitudes de escritura

En Wireshark se puede apreciar los paquetes asociados a la transacción usando la comunidad c0Mmuni7yAdm. El paquete 521 (Fig. 4.10) corresponde al set-request, que contiene el request-id 1.517’591.479 para asociarlo con el paquete de respuesta; no hay información que indique error

en el error status y error index y en el campo *variable bindings* se observa el nombre de objeto, que es el OID correspondiente al sysName y su valor, la cadena de octetos en hexadecimal para el nombre CISCO2800.

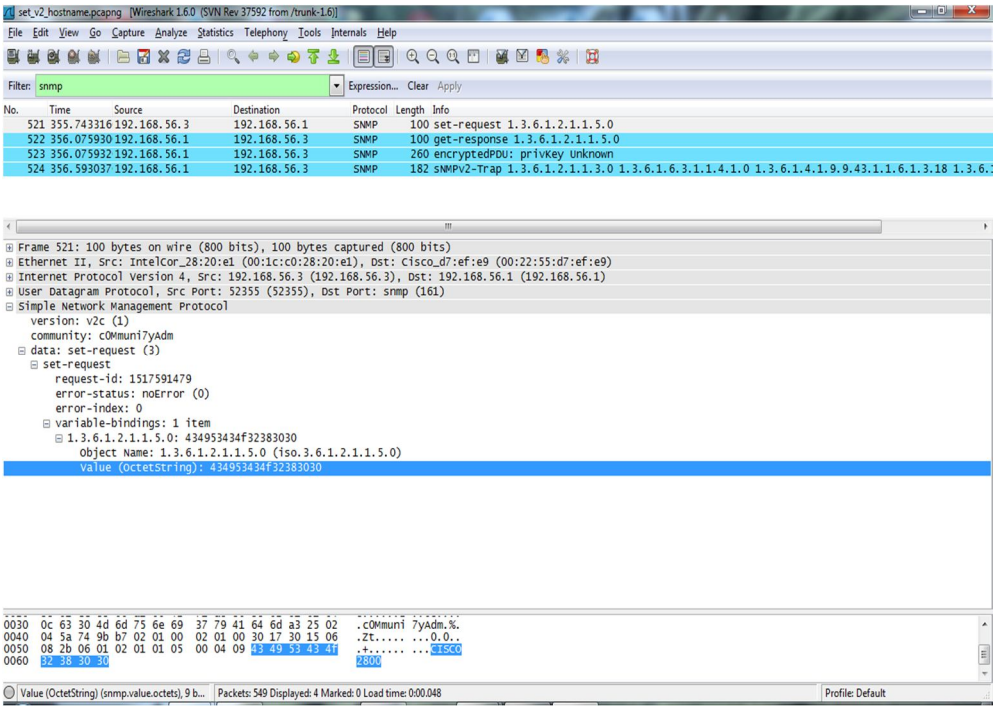


Figura 4.10- Paquete solicitud escritura al router - v2

El paquete 522 (Fig. 4.10) de respuesta contiene exactamente los mismos campos explicados para el de la solicitud, incluidos los del OID con su valor, excepto que es un get-response. Cuando se hace una solicitud set y es exitosa, arriba un mensaje de respuesta get-response con la misma PDU de datos que fue enviada en el set.

Los paquetes 523 y 524 (Fig. 4.10) corresponden a traps, uno en v3 y el otro en v2, enviados por el router hacia la NMS en respuesta a la modificación que recibió, esto debido a que en el router se configuró el envío de algunas traps (tanto en v2 y v3 simultáneamente), entre ellas las *config* las cuales notifican cualquier modificación a su configuración. Mayor detalle sobre traps v2 y v3 en la sección 3.12.

Los paquetes capturados en Wireshark durante la operación set con el usuario Root de SNMPv3 se muestran en la figura 4.11, desde la solicitud set-request hasta el get-request hay 6 mensajes (paquetes 27 al 32). Como se vio en la sección 3.9.1, una comunicación entre motores SNMPv3 tiene algunas interacciones para que la comunicación se pueda realizar con éxito. Los paquetes 33 y 34 corresponden a los traps enviados, en v3 y v2c, desde el router al NMS luego de la configuración.

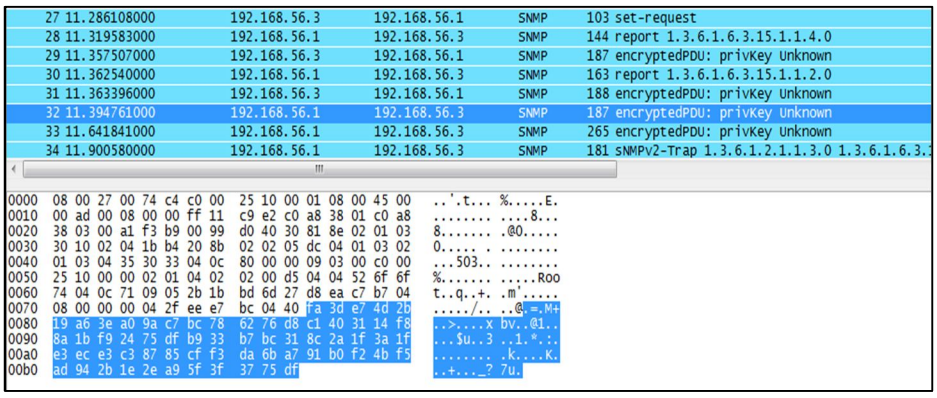


Figura 4.11- Paquete solicitud escritura al router - v3

En esta sección no se analizará en detalle cada apartado del encabezado de los mensajes, ya que se lo hizo con los ejemplos de las solicitudes de lectura; sin embargo el lector puede ver la similitud con lo que ya ha sido explicado en los paquetes 31 y 32. El paquete 31 (Fig. 4.12) es la solicitud set-request con ID 464'789.463. Sus datos están codificados por el motor SNMPv3 local, por lo que no es posible observar las variable bindings, que básicamente, tiene el mismo contenido que la misma solicitud de escritura para la versión 2. Como se vio en la sección 3.9.1, un mensaje que envía el agente es sometido a los algoritmos de seguridad y los valores relacionados se colocan en los campos correspondientes msgAuthenticationParameters y msgPrivacyParameters para que el MD en este caso, pueda descifrar la solicitud y autenticarla.

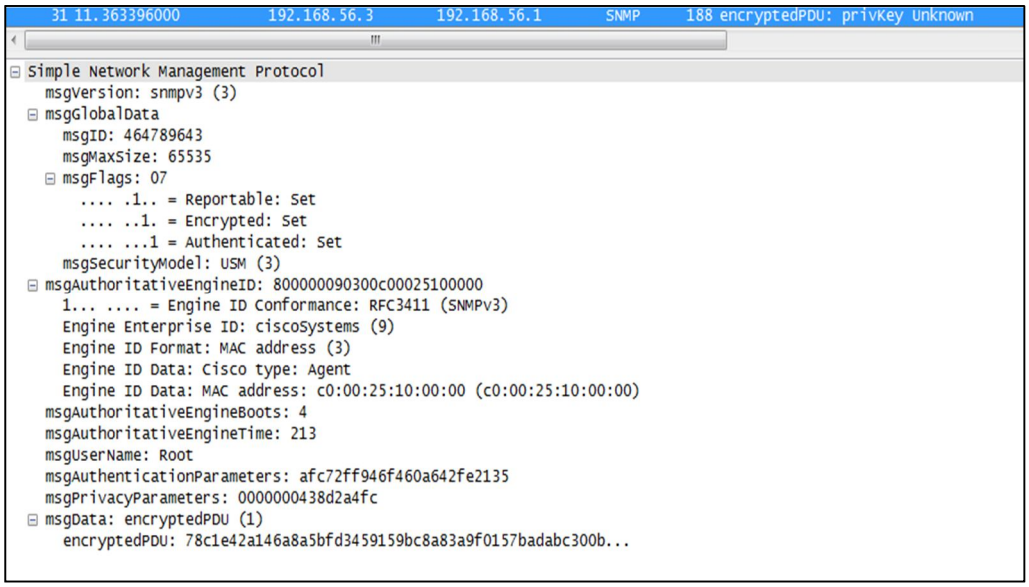


Figura 4.12- Set-request cifrado con parámetros de seguridad - v3

El paquete 32 (Fig. 4.13) es el get-response (de ID 464’789.463) que se sabe indica una operación set exitosa y cuyo contenido de datos fue cifrado por el motor SNMPv3 del router (autoritativo). De la misma forma se observan los parámetros de autenticación y cifrado definidos por el router para este mensaje.

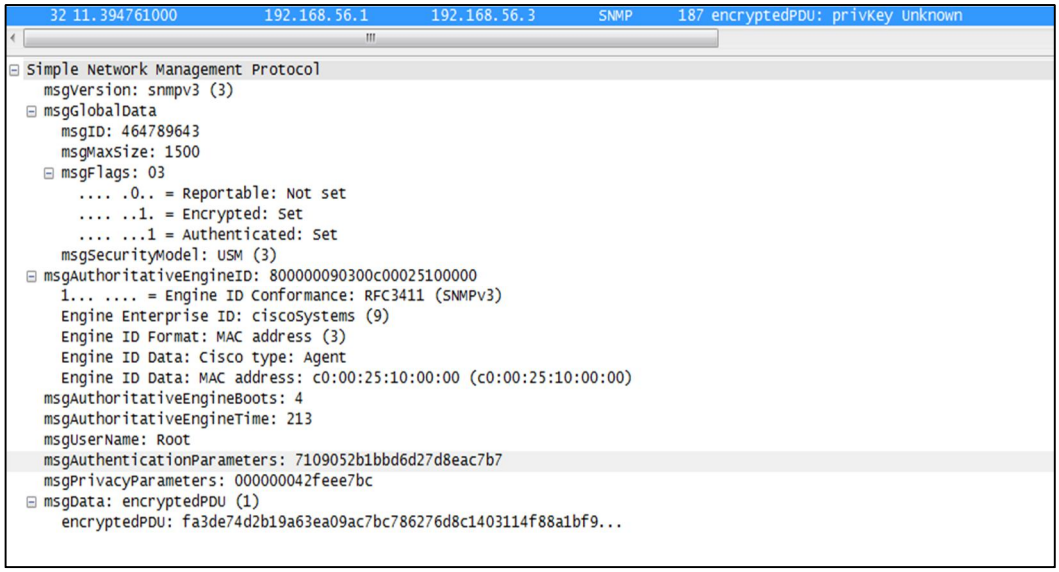


Figura 4.13- Get-response cifrado con parámetros de seguridad - v3

4.3 Comparación de capturas obtenidas en traps

Para realizar la comparación, necesitábamos ver las especificaciones de cada una de las traps recibidas, dimos doble click en cada una de las traps subrayadas de la fila derecha donde dice *payload* de la figura 3.168, donde sólo podíamos ver una trap a la vez, así que, para tener una mejor

visualización de todas las traps, las exportamos las traps a un archivo pdf dando click en la parte superior derecha donde dice *export*, al lado de *status*.

4.3.1 Trap LinkUp

Thursday, September 26, 2013 03:47:11 PM Any	TrapName=linkUp TrapMajor=3 TrapMinor=0 1.3.6.1.2.1.2.2.1.1.2=2 snmpTrapOID.0=1.3.6.1.6.3.1.1.5.4 (linkUp) ifType.2=6 ifIndex.2=2 1.3.6.1.2.1.1.3.0=0days 00:01:03.62 Object=1.3.6.1.2.1.11 (snmp) local.2.1.1.20.2=Link up 1.3.6.1.4.1.9.2.2.1.1.20.2=Link up Timetick=0days 00:01:03.62 ifDescr.2=FastEthernet0/1 Packet Type=SNMPv2 Trap 1.3.6.1.6.3.1.1.4.1.0=1.3.6.1.6.3.1.1.5.4 Protocol Version=SNMPv3 1.3.6.1.2.1.2.2.1.3.2=6 1.3.6.1.2.1.2.2.1.2.2=FastEthernet0/1 sysUpTimeInstance=0days 00:01:03.62
Thursday, September 26, 2013 03:47:11 PM Any	TrapName=linkUp TrapMajor=3 TrapMinor=0 1.3.6.1.2.1.2.2.1.1.2=2 snmpTrapOID.0=1.3.6.1.6.3.1.1.5.4 (linkUp) ifType.2=6 ifIndex.2=2 1.3.6.1.2.1.1.3.0=0days 00:01:03.62 Object=1.3.6.1.2.1.11 (snmp) local.2.1.1.20.2=Link up 1.3.6.1.4.1.9.2.2.1.1.20.2=Link up Timetick=0days 00:01:03.62 ifDescr.2=FastEthernet0/1 Packet Type=SNMPv2 Trap 1.3.6.1.6.3.1.1.4.1.0=1.3.6.1.6.3.1.1.5.4 Protocol Version=SNMPv2 CommunityName=c0Mmuni7yAdm 1.3.6.1.2.1.2.2.1.3.2=6 1.3.6.1.2.1.2.2.1.2.2=FastEthernet0/1 sysUpTimeInstance=0days 00:01:03.62

Figura 4.14- Traps linkUp recibidos por consola WhatsUp Gold

La figura 4.14 nos muestra dos traps de tipo *linkUp*; este tipo de trap se encarga de notificar al NMS que la interfaz respectiva ha cambiado de estado "*Down*" a "*Up*". La primera trap fue enviada con protocolo SNMP versión 3, y la segunda con protocolo SNMP versión 2. Las dos traps tienen los mismos datos, con la diferencia de que el trap v2 especifica el nombre de la comunidad a la cual pertenece, en cambio la trap v3 no especifica nombre de usuario alguno. Entre los datos que hay dentro del trap, existe un OID perteneciente a Cisco 1.3.6.1.4.1.9.2.2.1.1.20 (*locIfReason*), el cual imprime en pantalla una cadena de caracteres indicando la razón del *linkUp.* El resto de datos define todo lo relacionado con la interfaz que ha cambiado su

estado, como por ejemplo la descripción, el nombre de la interfaz, el índice, etc. Captura de la trap realizada en Wireshark:

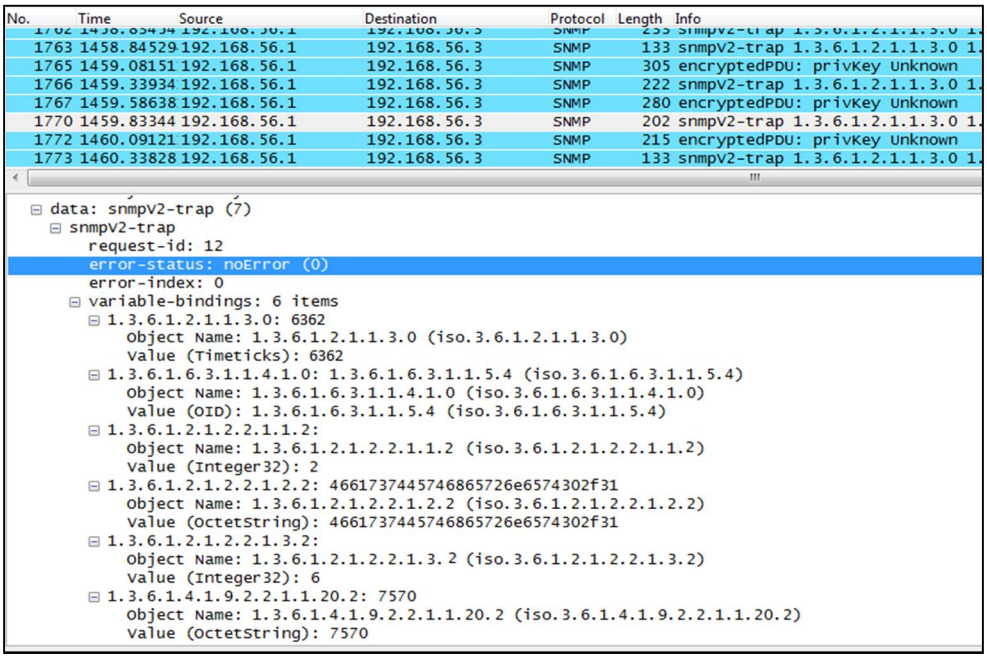


Figura 4.15- Traps linkUp capturados por Wireshark

Variables:

- TimeTicks
- snmpTrapOID
- ifIndex
- ifDescr
- ifType
- locIfReason

En la captura SNMP de la figura 4.15 podemos ver que el mensaje numero 1770 es una trap linkup v2, la cual contiene los OID's corespondientes a las datos ya explicados anteriormente. El mensaje numero 1.767 está cifrado ya que corresponde a un mensaje SNMPv3, y podemos asegurar que se trata de la misma trap linkup v2 con numero 1.770 pero en v3, ya que al enviarse los traps de v2 y v3, lo hacen de manera simultanea, es decir, siempre llegan juntas.

4.3.2 Trap ColdStart

Thursday, September 26, 2013 03:47:11 PM Any	TrapName=coldStart TrapMajor=0 TrapMinor=0 whyReload.0=power-on snmpTrapOID.0=1.3.6.1.6.3.1.1.5.1 (coldStart) 1.3.6.1.2.1.1.3.0=0days 00:01:04.11 1.3.6.1.4.1.9.2.1.2.0=power-on Object=1.3.6.1.2.1.11 (snmp) Timetick=0days 00:01:04.11 Packet Type=SNMPv2 Trap 1.3.6.1.6.3.1.1.4.1.0=1.3.6.1.6.3.1.1.5.1 Protocol Version=SNMPv2 CommunityName=c0Mmuni7yAdm sysUpTimeInstance=0days 00:01:04.11
Thursday, September 26, 2013 03:47:11 PM Any	TrapName=coldStart TrapMajor=0 TrapMinor=0 whyReload.0=power-on snmpTrapOID.0=1.3.6.1.6.3.1.1.5.1 (coldStart) 1.3.6.1.2.1.1.3.0=0days 00:01:04.11 1.3.6.1.4.1.9.2.1.2.0=power-on Object=1.3.6.1.2.1.11 (snmp) Timetick=0days 00:01:04.11 Packet Type=SNMPv2 Trap 1.3.6.1.6.3.1.1.4.1.0=1.3.6.1.6.3.1.1.5.1 Protocol Version=SNMPv3 sysUpTimeInstance=0days 00:01:04.11

Figura 4.16- Traps coldStart recibidos por consola WhatsUp Gold

La figura 4.16 nos muestra dos traps de tipo *coldStart*, este tipo de trap notifica que la entidad de protocolo está siendo reinicializada y que la configuración del agente o la entidad del protocolo pueden estar alteradas. La primera trap fue enviada con protocolo SNMP versión 2 y la segunda con

protocolo SNMP versión 3. Las dos traps tienen los mismos datos, con la diferencia de que el trap v2 especifica el nombre de la comunidad a la cual pertenece, en cambio la trap v3 no especifica ningún nombre de usuario. Entre los datos que hay dentro del trap, existe un OID perteneciente a Cisco 1.3.6.1.4.1.9.2.1.2.0 (*whyReload*), el cual imprime en pantalla una cadena de caracteres indicando la razón del *coldStart,* que en este caso es debido al encendido del equipo. El resto de datos corresponden a información de la trap. Captura de la trap realizada en Wireshark:

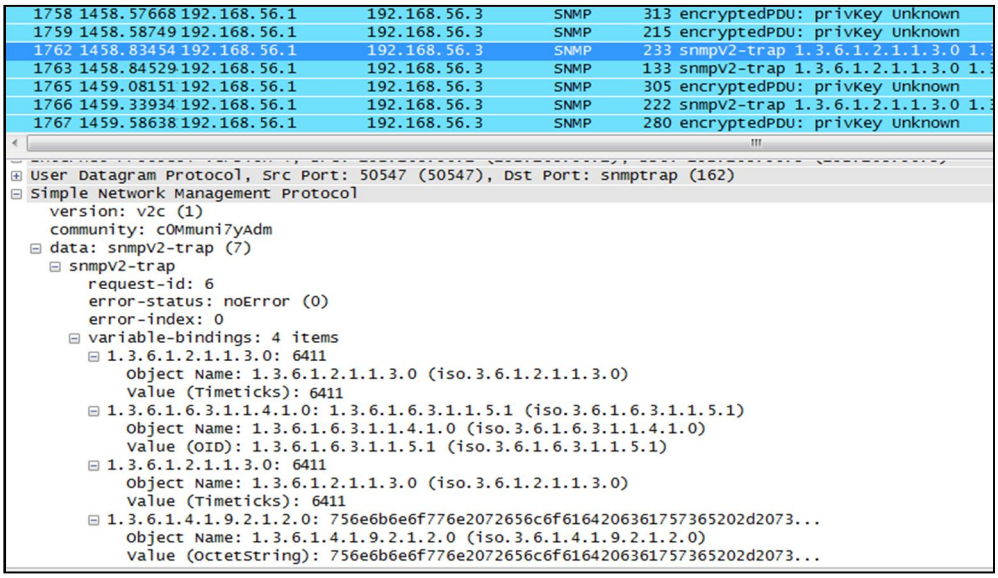


Figura 4.17- Traps coldStart capturados por Wireshark

Variables:

- TimeTicks
- snmpTrapOID
- sysUpTime

- whyReload

En la captura SNMP de la figura 4.17 podemos ver que el mensaje numero 1.762 es una trap coldStart v2, la cual contiene los OID's corespondientes a los datos ya explicados anteriormente. El mensaje numero 1.759 está cifrado ya que corresponde a la misma trap coldStart v2 con numero 1.762 pero en v3, ya que como explicamos en la trap linkup, los traps de v2 y v3, sin importar que traps son, siempre se envían de manera simultánea.

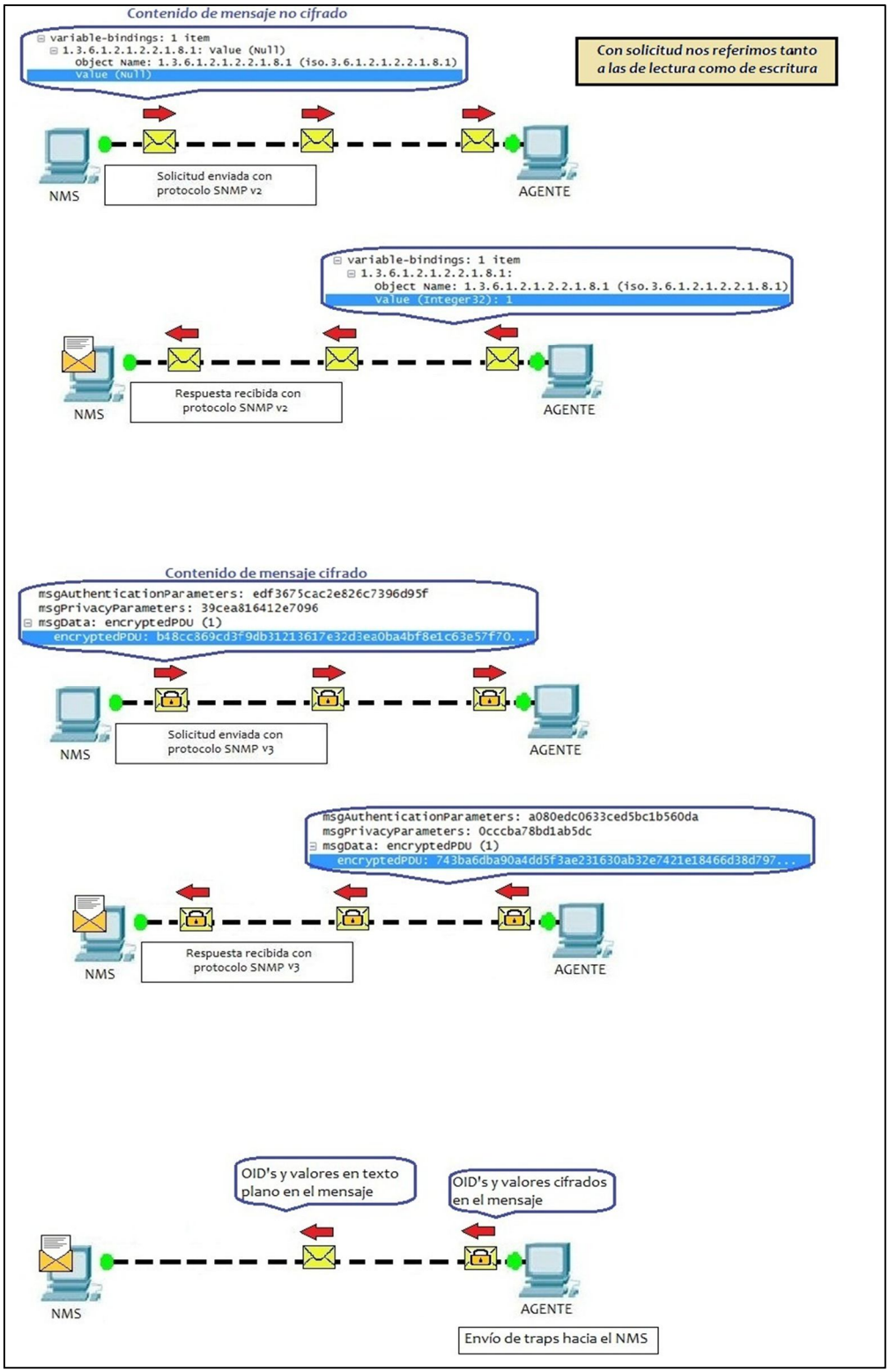


Figura 4.18- Resumen análisis de resultados de escenarios

CONCLUSIONES

1. En la realización de este proyecto nos dimos cuenta que no basta para una persona que realice operaciones de administración usando SNMPv3, que sepa configurar los agentes y usar las aplicaciones, sino que debe conocer el funcionamiento del protocolo en general y cómo interpretar los OIDs de los mensajes. Para el análisis y estudio de estos paquetes, tuvimos que estudiar primero los OIDs, los cuales no fue suficiente buscar en los repositorios; ya que muchos de estos OIDs vienen indexados con otros datos, y los tipos de indexaciones varían dependiendo de la raíz o tipo de rama. Es decir, para poder comprender el manejo de las solicitudes y los paquetes SNMP en sí, es necesario conocer el funcionamiento de los OIDs y su contenido, sólo así se podrá tener una total comprensión de la gestión por medio de solicitudes,

ya que tener el conocimiento del contenido de cada paquete que es enviado o recibido en nuestro NMS nos ayuda a una mejor administración de la red y a prevenir cualquier intrusión.

2. Las primeras versiones de SNMP introducían seguridades muy básicas tales como los nombres de comunidad (que actuaban como contraseñas de acceso a los objetos de la MIB de un agente, el identificador de los mensajes SNMP que permitía vincular a una solicitud un mensaje de respuesta y obviamente, las políticas de acceso a la MIB dadas por VACM a la comunidad. Todas estas funcionalidades eran fácilmente vulnerables, sobretodo porque las comunidades y los datos de los mensajes viajaban en texto plano a través de la red y cualquiera podía capturar el tráfico con alguna herramienta y hacer uso de las comunidades para obtener y modificar datos de forma fraudulenta. SNMPv3 supera todas estas debilidades previas y con su esquema de usuarios provee un esquema de seguridad robusto para las operaciones de administración.

3. Si bien es cierto en las consolas de las aplicaciones se ingresan contraseñas para autenticar y cifrar los contenidos de los mensajes generados en representación de algún usuario, éstas en ningún momento son visibles y a partir de ellas se calculan las llaves que en realidad son las que permiten la autenticación y cifrado de los mensajes en los agentes; y tampoco ninguna de las llaves viajan con los mensajes, sino valores producidos por los protocolos de seguridad que solamente los extremos de la comunicación son capaces de entender y usar para autenticar y descifrar los mensajes. Además como se vio durante el desarrollo de este documento, las llaves son localizadas mediante algoritmos seguros, lo que permite tener realmente una comunicación aislada de uno a uno entre el NMS y cada MD y en el caso que uno de ellos sea vulnerado no se perjudica a ninguno de los otros.

4. El cifrado de los mensajes, es la característica más atractiva del protocolo ya que permite tener la seguridad que las operaciones administrativas no serán observadas por nadie de forma no autorizada. Solo por dar un ejemplo en el caso de AES, es prácticamente imposible recuperar la llave en el caso que se

tuviera un documento original y su equivalente cifrado y el tratar de hacerlo llevaría demasiado tiempo, literalmente, siglos.

5. En las pruebas de solicitudes a los agentes monitoreados, se observó que los usuarios son visibles en el paquete, a pesar que éste tenga cifrado el *scopedPDU.* Sin embargo, esto no constituye una debilidad, ya que un atacante necesitaría primero conocer la llave localizada de autenticación del dispositivo remoto, y aunque la conociera, necesitaría luego estar dentro de la ventana de tiempo del motor autoritativo para que el mensaje fuera aceptado y por último, conocer la llave de privacidad del agente. Como vemos, es mucho más complicado vulnerar la seguridad del protocolo, que cuando se tiene una simple comunidad en las sus versiones previas.

6. Este documento, ha demostrado que el protocolo SNMPv3 es muy sólido, confiable y seguro para las operaciones de administración de la red y su uso, pero no por eso difícil de implementar y usar.

Es más, a pesar que el protocolo SNMPv3 ya tiene algunos años como estándar, no se ha implementado mayoritariamente en las empresas, y este documento pretende facilitar la comprensión del mismo para que eso cambie.

RECOMENDACIONES

1. Es recomendable tener actualizados los sistemas operativos de los hosts sobre los cuales van a funcionar tanto las entidades SNMPv3 administradoras de red, como las que serán objeto de monitoreo. En nuestro caso, uno de los requisitos fundamentales para el funcionamiento del software WhatsUp Gold en su última versión es tener el sistema operativo Windows 7, por lo que se tuvo que actualizar desde Windows XP al host que sería el administrador (NMS).

2. Para poder desplegar SNMPv3 en equipos Cisco tales como switches y routers, es necesario que el archivo de imagen mínimo en la memoria flash de los dispositivos sea del tipo ADVIPSERVICES (*advanced ip services)* para poder permitir autenticación con MD5 o SHA y cifrado con DES. Mejor todavía si la imagen es del tipo ADVENTERPRISE *(advanced enterprises)* para poder permitir el cifrado de los datos con AES128. No olvidar que generalmente en las redes corporativas los equipos están funcionando con los archivos de imagen por defecto IPBASE, por lo que se debe cambiar la imagen de los dispositivos usando un servidor TFTP, procedimiento indicado en el anexo 3.

3. La última versión de SNMP mejora considerablemente las seguridades de las anteriores versiones, lo que se analizó con detalle en todo el desarrollo de este documento y en las conclusiones. Por lo tanto, es aconsejable deshabilitar el uso de comunidades de lectura y escritura y reemplazarlo por el esquema de usuarios, preferiblemente de nivel de seguridad authPriv y con las configuraciones de control de acceso que sean pertinentes.

4. Es importante que las contraseñas de autenticación y cifrado que se ingresen para los diferentes usuarios tanto en entidades administradoras como monitoreadas, tengan como mínimo 8 caracteres y que no sean palabras de diccionario, es decir se debe tratar de combinar caracteres alfanuméricos, mayúsculas, minúsculas, caracteres especiales, etc. Esto contribuye a que sea difícil adivinar alguna contraseña, por lo tanto, las llaves de autenticación y cifrado permanecerán seguras. Además es muy importante cambiar las contraseñas regularmente como buena práctica de seguridad.

5. Evitar configurar la misma contraseña para autenticar y cifrar los mensajes, siempre es preferible que sean diferentes entre sí.

6. Ya que el engineID sirve para localizar las llaves de autenticación y privacidad, es importante que éste no esté basado en direcciones IP que cambian constantemente o en la MAC del dispositivo, ya que alguien que conozca la forma en que se forma el engineID puede deducirlo. A pesar que un atacante además necesitaría las llaves

maestras para realmente poder hacer algún ataque, es preferible configurar un engineID propio que no sea sencillo de adivinar o dejar que el agente lo genere a partir de un valor aleatorio, como se vio lo hace por defecto el agente Ubuntu en la sección 4.1.2.

7. La seguridad en la administración de red tiene muchos pilares y el uso de SNMPv3 es solamente uno de ellos. Es importante también como apoyo a este protocolo, el uso de firewalls, listas de control de acceso en routers de la red, el uso de switches en vez de hubs, seguridades de puerto en switches, entre otras medidas.

8. Finalmente, SNMPv3 no solamente cuenta con el modelo de seguridad USM, hay otros dos modelos de seguridad recientemente definidos por la IETF, uno basado en transporte SSH y otro en transporte (D)TLS que son tan o un poco más robustos que el modelo USM, el cual de por sí es muy seguro. Es recomendable realizar futuros trabajos de investigación como el presente, basados en ambos modelos, reforzando así la seguridad de la administración de la red.

ANEXOS

1. ARCHIVO DE CONFIGURACIÓN SNMPD.CONF DEL AGENTE UBUNTU

```
###########################################################################
#                           AGENT BEHAVIOUR                               #
###########################################################################
#  Esta sección sirve para definir los puertos de escucha para el servicio  #
#  snmp, sean tcp o ud                                                      #
###                                                                       ###
#  Esta configuración permite escuchar por conexiones entrantes en todas las
#  interfaces no solamente la de loopback, por el puerto udp 161 para IPv4 e
#  IPv6.
#
        agentAddress udp:161,udp6:[::1]:161
#
#  La siguiente directiva especifica el modo en que el engineID se genera
#  para este motor SNMP. Para un tipo 1 es a partir de una dirección IPv4,
#  tipo 2 de una dirección IPv6, tipo 3 usando la MAC-address y que elegimos
#  en este caso; por mencionar algunos.
```

```
###########################################################################
#                                ACCESS CONTROL                           #
###########################################################################
# Esta sección define las comunidades y/o usuarios SNMPv3, además del   #
# control de acceso a las variables de la MIB local a la que podrán acceder  #
###                                                                       ###
#
# La siguiente configuración define tres vistas las cuales se vincularán a la
# comunidad o usuario para permitirle el acceso exclusivamente a los objetos
# pertenecientes a las ramas MIB descritas por el OID asociado a cada una.
# La vista internet permite el acceso a toda la rama internet de la MIB;
# solamente onlyMib2 a la rama mib-2 y onlySystem a variables relacionadas
# al sistema, como system que es una de las ramas de la mib-2 y hrSystem
# que es una de las ramas del árbol host.
#
        view internet included .1.3.6.1
        view onlyMib2 included .1.3.6.1.2.1
        view onlySystem included .1.3.6.1.2.1.1
        view onlySystem included .1.3.6.1.2.1.25.1
#
#  Comunidad v2c de lectura y escritura (para fines demostrativos) que
#  procesará solicitudes de la red especificada por la IP y máscara y que
```

```
#  podrá solamente recuperar objetos de la vista onlySystem
#
        rwcommunity c0Mmuni7yAdm  192.168.56.0/24 -V onlySystem
#
#
#  Para la configuración de los usuarios de la versión 3, es suficiente con las
#  directivas rouser y rwuser con vistas adecuadas, que deberían cubrir la
#  mayoría de requerimientos. Este modo de configuración es por defecto
#  desde la versión 5.3 de Net-SNMP en adelante.
#
#
#  Definición de un usuario  SNMPv3 sólo lectura, de nivel de seguridad
#  noAuthNoPriv, que no autentica ni cifra los mensajes enviados y que podrá
#  solamente recuperar objetos de la vista onlySystem
#
        rouser Invitado noauth -V onlySystem
#
#
#  Usuario SNMPv3 sólo lectura, de nivel de seguridad AuthNoPriv, que
#  autentica pero no cifra los mensajes enviados  y que podrá solamente
#  recuperar objetos de la vista mib-2
#
```

```
        rouser Supervisor auth -V onlyMib2
#
#
#  Usuario SNMPVv3 de lectura y escritura, de nivel de seguridad authPriv,
#  que autentica y cifra los mensajes enviados y que recupera todos los
#  objetos que se hallan contenidos bajo el árbol internet.
#
        rwuser Root priv -V internet
#
#  Usuario SNMPv3 de lectura y escritura, de nivel de seguridad authPriv,
#  que autentica y cifra los traps enviados hacia la pc administradora de red
#  originados por cualquiera de las variables bajo el árbol internet. Exclusivo
#  para envío de notificaciones, no para consultas.
#
        rwuser Notificador priv -V internet
#
###################################################################
#                          SYSTEM INFORMATION                     #
###################################################################
#  Información del sistema, objetos que al ser configurados serán de sólo    #
#  lectura  para solicitudes entrantes                                        #
###                                                                        ###
```

```
#
#  Información de localización del equipo monitoreado y contacto en caso de
#  fallos.
#
        sysLocation Laboratorio Simulacion Telecomunicaciones
        sysContact  Ochoa Elsa – Marcelo Venegas

###########################################################################
#                                ACTIVE MONITORING                        #
###########################################################################
# Monitoreo activo que en caso de fallos envía notificaciones v2c y v3 hacia #
# uno o varios agente remotos                                              #
###                                                                      ###
#
#  Habilita el envío de notificaciones para fallos del agente Net-SNMP,
#  linkUp y fallos de autenticación en solicitudes entrantes al agente.
#
        defaultMonitors          yes
        linkUpDownNotifications  yes
        authtrapenable 1
#
#
```

Trap2sink envía una trap v2c en la comunidad c0Mmuni7yAdm al destino
especificado en el caso de alguna falla y trapsess hace lo mismo pero el
trap enviado es por el usuario v3 Notificador, con nivel de seguridad
authPriv por lo que su contenido está cifrado.
#

trap2sink 192.168.56.3 c0Mmuni7yAdm

trapsess -u Notificador -l authPriv 192.168.56.3

2. RESULTADOS DE MAPEO CONTRASEÑA A LLAVE USANDO SHA (ANEXO RFC3414)

Password to Key Sample Results using SHA

The following shows a sample output of the password to key algorithm for a 20-octet key using SHA.

With a password of "maplesyrup" the output of the password to key algorithm before the key is localized with the SNMP engine's snmpEngineID is:

'9f b5 cc 03 81 49 7b 37 93 52 89 39 ff 78 8d 5d 79 14 52 11'H

After the intermediate key (shown above) is localized with the snmpEngineID value of:

'00 00 00 00 00 00 00 00 00 00 00 02'H

the final output of the password to key algorithm is:

'66 95 fe bc 92 88 e3 62 82 23 5f c7 15 1f 12 84 97 b3 8f 3f'H

3. ACTUALIZACIÓN IMAGEN DEL ROUTER MEDIANTE SERVIDOR TFTP.

En nuestro caso, usamos la aplicación SolarWinds TFTP Server para realizar todo el procedimiento; esta aplicación define en el directorio C:\TFTP-Root una carpeta en la cual debemos copiar el nuevo archivo de imagen (ADVIPSERVICES o ADVENTERPRISE) que deseamos configurar al router para su posterior funcionamiento.

Es recomendable respaldar el archivo de imagen actual del router en el servidor TFTP antes de borrarlo, para luego copiar el nuevo. Esto lo hicimos mediante la siguiente secuencia de comandos:

```
copy flash: tftp:
Source filename []? C2800nm-ipbase-mz.124-3i.bin
Address or name of remote host []? IP-SERTVIDOR-TFTP
```

Luego de ingresar estos comandos, se enviará desde la memoria flash del router su archivo de imagen actual hacia la computadora que funciona como servidor TFTP, específicamente a la carpeta TFTP-Root.

Una vez terminado el proceso, se borra el archivo de imagen actual del router para liberar espacio a la nueva imagen que enviaremos desde el servidor TFTP. Se lo hace mediante el comando:

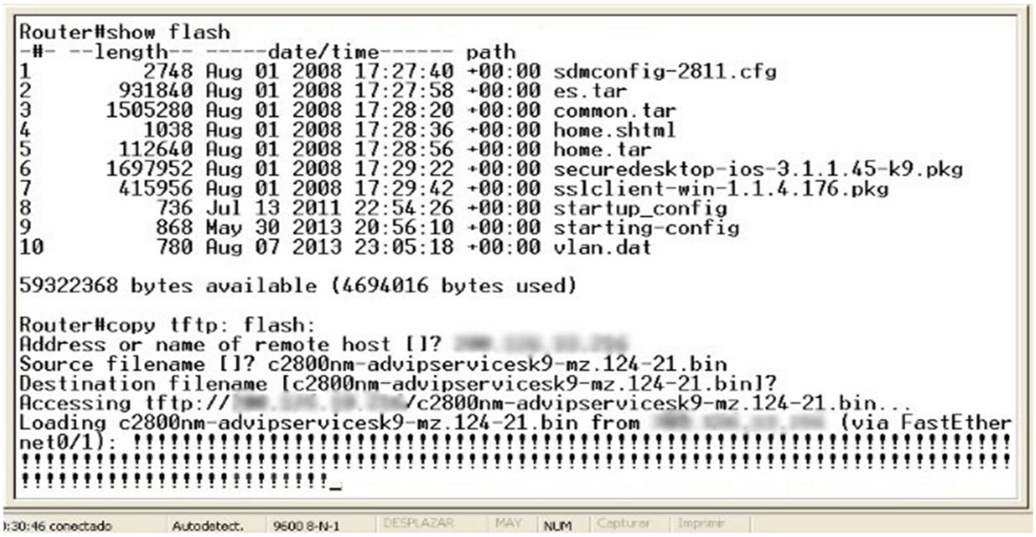
```
delete flash:c2800nm-ipbase-mz.124-3i.bin
```

De todas formas, hay que cerciorarse que el tamaño de la nueva imagen no supere el tamaño disponible total luego de haber hecho el borrado.

Finalmente, copiamos la nueva imagen desde nuestro servidor TFTP hacia la flash del router, mediante la secuencia de comandos:

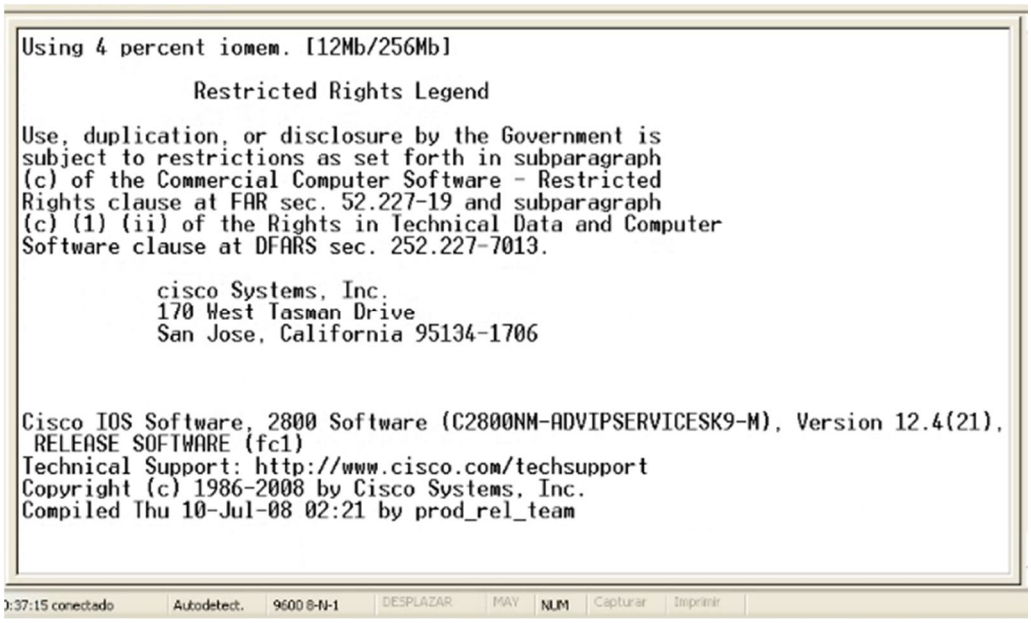
```
copy tftp: flash:

Source filename []? C2800nm-advipservicesk9-mz.124-21.bin

Address or name of remote host []? IP-SERTVIDOR-TFTP
```

El proceso se muestra en la siguiente figura:



Transferencia nueva imagen del router a la flash.

Terminado el proceso se ingresa el comando `reload` al router para reiniciarlo desde la nueva imagen, como se puede apreciar en la siguiente figura:



Inicio del router a partir de la nueva imagen cargada a la flash.

Con esta nueva imagen en la flash, podremos usar autenticación y cifrado (solamente con DES) en los mensajes SNMPv3. Si quisiéramos mejorar aún más el nivel de seguridad en el cifrado de los mensajes, se elige una imagen del tipo ADVENTERPRISES para poder cifrar los mensajes con AES.

BIBLIOGRAFÍA

[1] Monique Morrow, Thomas P. Nadeau, Bernhard Neumair, Rajiv Ramaswami, Kumar N. Sivarajan, John Strassner, Kateel Vijayananda… & James D. McCabe (2009). *Network Management – Know it All*.(1) Burlington, MA: Elsevier.

[2] Antoni Barba Martí. (1999). *Gestión de Red*. Recuperado de: http://librosdeelectronica.blogspot.com/2012/01/gestion-de-red.html

[3] Douglas R. Mauro & Kevin J. Schmidt. (2005). *Essential SNMP 2nd Edition*. Recuperado de: http://it-ebooks.info/book/367/

[4] Jianguo Ding. (2010). *Advances in Network Management*. Recuperado de: http://es.scribd.com/doc/68398925/Advances-in-Network-Management

[5] H3C Technologies Co. s.f. Manual de SNMP. *Tecnologías de Libro Blanco SNMP*. Recuperado de: http://www.h3c.com/portal/Products___Solutions/Technology/System_Management/Technology_White_Paper/200805/606347_57_0.htm

[6] Manual WhatsUp Gold. Recuperado de: http://docs.ipswitch.com/NM/72_WhatsUpGoldv16.1/03_Help/1033/index.htm

[7] Cisco. *Simple Network Management Protocol.* (Octubre 2012). Tutorial de SNMP. Recuperado de: http://docwiki.cisco.com/wiki/Simple_Network_Management_Protocol

[8] Cisco. The Traps Sent with SNMP-Server Enabled Traps Configured. Tutorial de configuración de traps SNMP. Recuperado de:

http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a008021de3e.shtml

[9] Cisco. *Understanding Simple Network Management Protocol (SNMP) Traps.* (Octubre 2006). Tutorial de Traps SNMP. Recuperado de: http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a0080094aa5.shtml

[10] Cisco. snmp-server enable traps through snmp-server enable traps ospf cisco-specific retransmit. Referencia de comandos. Recuperado de: http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_18.html

[11] Jorge Moreno Carreres. (2006). *Esc. Técnica Superior de Ingenieros de Telecomunicación* (Tutorial de Net-SNMP , Univ. de Las Palmas de Gran Canaria). Recuperado de: http://www.personales.ulpgc.es/nramos.dit/?q=system/files/Tutorial_de_NET-SNMP.pdf

[12] Oracle Integrated Lights Out Manager (ILOM) 3.0. *Referencia para la administración de protocolos*. (Julio 2011)(Tutorial de Oracle). Recuperado de: http://docs.oracle.com/cd/E19860-01/PDF/E23694-01.pdf

[13] Morris Dworkin (Diciembre 2001). Instituto Nacional de Estándares y Tecnología. *Recommendation for Block Cipher Modes of Operation* (Seguridad de Computadoras, Estados Unidos de América) http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

[14] Revisión 2. Recuperado de: http://www.unainet.net/documents/SNMP.pdf

[15] RFC3412. Message Processing and Dispatching for the Simple Network Management Protocol (SNMP). https://datatracker.ietf.org/doc/rfc3412/?include_text=1

[16] RFC2579. Textual Conventions for SMIv2. https://datatracker.ietf.org/doc/rfc2579/?include_text=1

[17] RFC3415. View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). https://datatracker.ietf.org/doc/rfc3415/?include_text=1

[18] RFC3411. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. https://datatracker.ietf.org/doc/rfc3411/?include_text=1

[19] RFC3414. User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) https://datatracker.ietf.org/doc/rfc3414/?include_text=1

[20] RFC3826. The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User- based Security Model. https://datatracker.ietf.org/doc/rfc3826/?include_text=1

[21] Patricia Victoria Aguilar. (Febrero 2007). Administración de Redes. (Monografía). Recuperado de: http://www.monografias.com/trabajos43/administracion-redes/administracion-redes2.shtml#mib

[22] Inés E. Inchauspe. (2001). Monitoreo de Redes. (Manuscrito). Universidad Nacional de Luján. Argentina. Recuperado de: http://www.tyr.unlu.edu.ar/tyr/TYR-trab/monitred/TF-Inchauspe.htm

[23] Nicolás Botero Arana. (2005). *Modelo de Gestión de Seguridad con Soporte a SNMP*. (Proyecto de Graduación, Pontificia Universidad Javeriana). Recuperado de:

http://www.javeriana.edu.co/biblos/tesis/ingenieria/Tesis190.pdf

[24] Valarezo Saldarriaga y Simisterra Huila. (2011). *Implementación de un Sistema de Gestión y Administración de Redes - Basados en el Protocolo Simple de Monitoreo de Redes SNMP en la Red*. (Proyecto de Graduación). ESPOL. Recuperado de:

http://www.dspace.espol.edu.ec/handle/123456789/16202

[25] Víctor Hugo Hinojosa Jaramillo, Luis Alberto Madruñero Padilla, Luis Vicente Ortega Pilco. (Noviembre 2001). Sistema de Gestión de Red: *Sistema de Características Resumidas para la Gestión de Redes (Net-Manager).* (Proyecto de Graduación, Univ. Técnica del Norte). Recuperado de:

https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&cad=rja&sqi=2&ved=0CEwQFjAE&url=http%3A%2F%2Frepositorio.utn.edu.ec%2Fbitstream%2F123456789%2F577%2F1%2FTesisFinal.doc&ei=znd5UdnPBsTV0QG63oGYAQ&usg=AFQjCNEnCGLPIBe5ZoEHMukTTUwPxeSlyw&sig2=CEP6nkHitsYJXZBHIv5eiQ&bvm=bv.45645796,d.dmQ

[26] Maestría en Seguridad Informática. Seguridad en Redes I. (César Páez, Luis Marroquín y Marcelo Menal). Recuperado de: http://www.youtube.com/watch?v=R_fPUSPeJSo

[27] Enciclopedia Net-SNMP. Documento FAQ:Agent 27. Net-SNMP. http://www.net-snmp.org/wiki/index.php/FAQ:Agent_27

[28] Enciclopedia Wikipedia. Block Cipher mode of operation. http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

[29] Enciclopedia Wikipedia. Protocolo. Recuperado de: http://es.wikipedia.org/wiki/Protocolo_(inform%C3%A1tica)

[30] Foro el hacker.net. http://foro.elhacker.net/criptografia/funciones_de_hash-t100025.0.html

[31] Blog linux. http://blog.desdelinux.net/wireshark-analiza-el-trafico-de-tu-red/

[32] Textual Conventions of snmpUsmMIB Objects. Recuperado de: http://net-snmp.sourceforge.net/docs/mibs/snmpUsmMIB.html#SnmpAdminString

[33] Repositorio de OIDs. http://www.oid-info.com

[34] Definición de Host-Resources-MIB. Recuperado de: http://www.simpleweb.org/ietf/mibs/modules/IETF/txt/HOST-RESOURCES-MIB

[35] Ejemplo valores de Host-Resources-MIB. Repositorio. Recuperado de: http://www.observium.org/wiki/HOST-RESOURCES-MIB