PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Análisis e implementación de la herramienta de código libre Pandora FMS para el monitoreo de nodos de telecomunicaciones y servidores en ambientes Windows y Linux, usando el protocolo SNMP y la tecnología instrumental de administración de Windows WMI

TESIS DE GRADO PRESENTADA COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE MASTER EN REDES DE COMUNICACIONES

SANTIAGO DAVID SILVA PROAÑO
sdsilva@outlook.com

DIRECTOR: ING. MARÍA SOLEDAD JIMÉNEZ, M.SC.

QUITO, SEPTIEMBRE DE 2013

Declaración

Yo, SANTIAGO DAVID SILVA PROAÑO, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo los derechos de propiedad intelectual correspondiente a este trabajo, a la Pontificia Universidad Católica del Ecuador, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

_____

Santiago David Silva Proaño

Certificación

Certifico que el presente trabajo fue desarrollado por Santiago David Silva Proaño bajo mi supervisión.

Ing. María Soledad Jiménez, M.Sc.
Director del Proyecto

AGRADECIMIENTOS

Agradezco de todo corazón a mi papá y mi mamá que siempre han confiado en mí y me han brindado su apoyo para alcanzar mis metas, espero algún día devolverles todo el esfuerzo y dedicación que me han dado; a Andreita que desde muchos años ha formado parte de mi vida siendo un impulso diario para dar lo mejor de mí y convertirme en un mejor ser humano. Finalmente a todos quienes han sido parte de mi formación: mis abuelitos, tíos, maestros y amigos sepan que sus enseñanzas y consejos se ven plasmados en lo que soy hoy en día, muchas gracias por todo.

Santiago Silva P.

DEDICATORIA

El presente proyecto está dedicado a mis padres Alba y César que siempre han inculcado en mí la responsabilidad de cumplir mis metas y me han apoyado para que sea sobre todo un ser humano íntegro y de valores. El trabajo aquí desarrollado es un fruto de su trabajo y constante abnegación por sus hijos.

Santiago Silva P.

ÍNDICE GENERAL

LISTA DE FIGURAS ..... **viii**
LISTA DE TABLAS ..... **x**
CAPÍTULO 1 ..... **1**
1. INTRODUCCIÓN **1**
1.1 Conceptos Básicos de la herramienta de monitoreo 2
1.2 Monitoreo de Redes LAN 4
1.3 Protocolo SNMP 5
1.4 Protocolo WMI 6
1.5 Justificación e Importancia del Proyecto 6
1.6 Objetivos 7
1.7 Contenido 7
CAPÍTULO 2 ..... **9**
2. MARCO TEÓRICO **9**
2.1 Protocolo SNMP 9
2.1.1 Historia del Protocolo SNMP 9
2.1.2 Componentes básicos de SNMP 10
2.1.3 Estructura de la base de información de gestión o MIB 12
2.2 *Windows Management Instrumentation*–WMI 13
2.2.1 Historia de WMI 13
2.2.2 Arquitectura WMI 14
2.3 Pandora *Flexible Monitoring System* (FMS) 16
2.3.1 Funcionalidades de Pandora FMS 18
2.3.2 Arquitectura de Pandora FMS 18
2.3.3 Servidores de Pandora FMS 19
2.3.4 Consola Web de Pandora FMS 21
2.3.5 Base de datos de Pandora FMS 22
2.3.6 Agentes Software de Pandora FMS 23
2.3.7 Operación con *traps* SNMP 24
2.3.8 Alertas de *traps* SNMP 26
CAPÍTULO 3 ..... **29**
3. DISEÑO E IMPLEMENTACIÓN DE LA HERRAMIENTA **29**
3.1 Diseño del entorno de trabajo 29
3.2 Implementación de la herramienta de monitoreo 31

3.2.1 Configuración del servidor Ubuntu 12.04 para Pandora FMS 31
3.2.2 Instalación de los componentes del servidor de Pandora FMS 32
3.2.3 Configuración del servidor de Pandora FMS 34
3.2.4 Configuración de la hora del Servidor de Pandora FMS 39
3.2.5 Configuración de Agentes de Monitoreo en Servidores Windows 41
3.2.6 Configuración de Agentes de Monitoreo en Servidores Linux/Unix 43
3.2.7 Configuración del agente Windows mediante consultas WMI 44
3.2.8 Configuración para recepción de TRAPS SNMP 46
3.2.9 Configuración de envío de mensajes SMS 49
3.2.10 Configuración de envío de notificaciones vía e-mail 54
3.3 Configuración del *Router* Cisco para acceder a la información SNMP 56
3.4 Guía de Implementación de la herramienta Pandora FMS 57
CAPÍTULO 4 61
4. ANÁLISIS DE LOS RESULTADOS 61
4.1 Monitoreo de agentes en sistemas operativos Windows 61
4.1.1 Monitoreo mediante consultas WMI 61
4.1.2 Monitoreo mediante consultas al Indicador de Rendimiento 63
4.1.3 Monitoreo mediante lectura del Log de Eventos 65
4.2 Monitoreo de agentes en sistemas operativos Unix/Linux 67
4.3 Monitoreo mediante la consola de Pandora FMS 69
4.4 Generación de Alertas 73
4.5 Generación de reportes sobre los valores monitoreados 74
4.6 Generación de Reporte para CMDB 75
4.6 Comparación con la herramienta de monitoreo *HP OpenView* 77
4.6.1 *Network Node Manager* 78
4.6.2 *Sitescope* 79
4.6.3 *Operations Manager* 79
CAPÍTULO 5 80
5. CONCLUSIONES Y RECOMENDACIONES 80
5.1 Conclusiones 80
5.1 Recomendaciones 81
CAPÍTULO 6 82
6 RECOMENDACIONES PARA TRABAJOS FUTUROS 82
CAPÍTULO 7 83
7 BIBLIOGRAFÍA 83

LISTA DE FIGURAS

Figura 1.1	Servidores de Pandora FMS	Pág. 2
Figura 1.2	Esquema de la infraestructura para monitoreo mediante Pandora FMS	Pág. 3
Figura 1.3	Verificación ICMP sobre un nodo de red	Pág. 4
Figura 1.4	Verificación de puertos sobre un nodo de red	Pág. 5
Figura 2.1	Formato del paquete SNMP	Pág. 11
Figura 2.2	Campos PDU de una trama SNMP	Pág. 11
Figura 2.3	Estructura del árbol de una MIB	Pág. 13
Figura 2.4	Arquitectura WMI	Pág. 14
Figura 2.5	Repositorio CIM mediante WMI *Explorer*	Pág. 15
Figura 2.6	Arquitectura de Pandora FMS	Pág. 19
Figura 2.7	Comprobaciones de red de Pandora FMS	Pág. 20
Figura 2.8	Consola Web de monitoreo de Pandora FMS	Pág. 22
Figura 2.9	Esquema de la Base de Datos de Pandora FMS	Pág. 22
Figura 2.10	*Trap* SNMP en la consola de recepción	Pág. 24
Figura 2.11	Filtrado de *traps* SNMP	Pág. 25
Figura 2.12	Pre-visualización de la alerta a un *trap* SNMP	Pág. 28
Figura 3.1	Descripción del Servidor 1	Pág. 29
Figura 3.2	Descripción del Servidor 2	Pág. 29
Figura 3.3	Entorno de trabajo con Windows para *Active Directory*, DNS y DHCP	Pág. 31
Figura 3.4	Consola de Ubuntu Server 12.04 LTS desde *vmWare WorkStation*	Pág. 31
Figura 3.5	Requisitos para la instalación del Servidor de Pandora FMS	Pág. 32
Figura 3.6	Requisitos para la instalación de la consola de Pandora FMS	Pág. 32
Figura 3.7	Asistente para la Base de Datos *MySQL*	Pág. 33
Figura 3.8	Componentes adicionales de la herramienta de monitoreo	Pág. 33
Figura 3.9	Paquetes de instalación de Pandora FMS	Pág. 34
Figura 3.10	Asistente o *Wizard* de Configuración de Pandora FMS – 1	Pág. 34
Figura 3.11	Asistente o *Wizard* de Configuración de Pandora FMS – 2	Pág. 35
Figura 3.12	Asistente o *Wizard* de Configuración de Pandora FMS – 3	Pág. 35
Figura 3.13	Asistente o *Wizard* de Configuración de Pandora FMS – 4	Pág. 36
Figura 3.14	Asistente o *Wizard* de Configuración de Pandora FMS – 5	Pág. 36
Figura 3.15	Asistente o *Wizard* de Configuración de Pandora FMS – 6	Pág. 37
Figura 3.16	Asistente o *Wizard* de Configuración de Pandora FMS – 7	Pág. 37
Figura 3.17	Consola SSH para borrar el archivo install.php	Pág. 38
Figura 3.18	Página web de bienvenida de Pandora FMS	Pág. 38
Figura 3.19	Configuración del archivo pandora_server.conf	Pág. 38
Figura 3.20 (a)	Configuración de la Zona Horaria del servidor Linux (a)	Pág. 39
Figura 3.20 (b)	Configuración de la Zona Horaria del servidor Linux (b)	Pág. 40
Figura 3.21	Cambio de la zona horaria en la consola de Pandora FMS	Pág. 40
Figura 3.22	Configuración del Agente de Pandora FMS en Windows – 1	Pág. 41
Figura 3.23	Configuración del Agente de Pandora FMS en Windows – 2	Pág. 42
Figura 3.24	Actualización de configuración del agente instalado en Sistema Operativo Windows	Pág. 42
Figura 3.25	Reinicio de los servicios del Agente de Pandora FMS en el equipo Windows	Pág. 43
Figura 3.26	Instalación del Agente Pandora FMS en un servidor con Ubuntu 12.04	Pág. 43
Figura 3.27	Cambios en el archivo de configuración del agente de Pandora FMS	Pág. 44
Figura 3.28	Consola de Administración y Monitoreo de Pandora FMS	Pág. 44
Figura 3.29	Asignación de políticas WMI	Pág. 45
Figura 3.30	Configuración de módulo WMI	Pág. 46
Figura 3.31	Prueba de conexión WMI	Pág. 46
Figura 3.32	Consola de Administración y Monitoreo de Pandora FMS	Pág. 47

Figura 3.33	Configuración del archivo *snmp.conf*	Pág. 47
Figura 3.34	Configuración del archivo *snmptrapd.conf*	Pág. 47
Figura 3.35	Configuración del archivo *snmpd*	Pág. 48
Figura 3.36	Confirmación de reinicio de los servicios de SNMP	Pág. 48
Figura 3.37	Confirmación de servicios de SNMP levantados por Pandora FMS	Pág. 49
Figura 3.38	Administrador de dispositivos de Windows	Pág. 49
Figura 3.39	Detalle del puerto que usa el Modem GSM USB	Pág. 50
Figura 3.40	Propiedades el Modem GSM USB	Pág. 50
Figura 3.41	Detalle de los puertos seriales habilitados para GSM modem	Pág. 51
Figura 3.42	Opciones del software *minicom*	Pág. 52
Figura 3.43	Configuración de *minicom* para envío de SMS	Pág. 52
Figura 3.44	Envío de SMS a través de m*inicom* para conexión serial	Pág. 53
Figura 3.45	Archivo de configuración *smsd.conf*	Pág. 54
Figura 3.46	Configuración con EXPECT para envío de notificaciones vía mail	Pág. 55
Figura 3.47	Topología de simulación en GNS3	Pág. 56
Figura 3.48	Configuración de *traps* SNMP	Pág. 56
Figura 3.49	Consola de recepción de *traps* SNMP	Pág. 57
Figura 4.1	Creación de Módulo WMI	Pág. 61
Figura 4.2	Visualización de información WMI a través de WMI Explorer	Pág. 62
Figura 4.3	Consulta WMI desde WMI Explorer	Pág. 62
Figura 4.4	Módulo de monitoreo	Pág. 62
Figura 4.5	*Performance Monitor* de Windows	Pág. 63
Figura 4.6	Módulo de monitoreo mediante *Performance Monitor*	Pág. 64
Figura 4.7	Selección del contador de *Performance Monitor*	Pág. 64
Figura 4.8	Información de los contadores de *Performance* de Windows	Pág. 65
Figura 4.9	Formato de Módulo para monitoreo del *Log* de Eventos	Pág. 65
Figura 4.10 (a)	Visualización de eventos auto generados mediante línea de comandos	Pág. 66
Figura 4.10 (b)	Propiedades del evento auto generado	Pág. 67
Figura 4.11	Módulo de monitoreo para el *Log* de Eventos	Pág. 67
Figura 4.12	Módulo para consultas SNMP sobre equipos Unix/Linux	Pág. 68
Figura 4.13	Módulo de monitoreo en sistema Unix/Linux mediante SNMP	Pág. 68
Figura 4.14	Consultas SNMP y se respectiva OID para sistemas Unix/Linux	Pág. 68
Figura 4.15	Monitoreo remoto mediante consultas WMI	Pág. 69
Figura 4.16	Revisión del árbol de la MIB	Pág. 70
Figura 4.17	Búsqueda de MIBs para un producto específico del fabricante Cisco	Pág. 71
Figura 4.18 (a)	Visualización de dependencias de las MIBs	Pág. 72
Figura 4.18 (b)	Descarga de la MIB y las dependencias	Pág. 72
Figura 4.19	Monitoreo mediante consulta SNMP	Pág. 73
Figura 4.20	Monitoreo de monitoreo modificado	Pág. 74
Figura 4.21	Notificaciones vía Correo Electrónico y SMS	Pág. 74
Figura 4.22 (a)	Generación de gráfico de consumo de CPU	Pág. 75
Figura 4.22 (b)	Reporte de Valores numéricos de consumo de CPU	Pág. 75
Figura 4.23	Generación de información vía SNMP	Pág. 76
Figura 4.24	Campos administrables para información de equipos monitoreados	Pág. 76
Figura 4.25	Esquema de la Infraestructura de HP *OpenView*	Pág. 77

LISTA DE TABLAS

Tabla 2.1	Componentes en una red con SNMP	Pág. 10
Tabla 2.2	Tipos de Mensajes SNMP	Pág. 11
Tabla 2.3	Estado de Errores para consultas SNMP	Pág. 12
Tabla 2.4	Recomendación para asignación de recursos al servidor de Pandora FMS	Pág. 17
Tabla 2.5	Sistemas Operativos soportados para instalación del agente	Pág. 18
Tabla 2.6	Tipos de mensajes SNMP	Pág. 25
Tabla 3.1	Equipos virtuales creados para el ambiente de pruebas	Pág. 29
Tabla 3.2	Requisitos para Windows 2003	Pág. 30
Tabla 3.3	Requisitos para Ubuntu *Server*	Pág. 30
Tabla 3.4	Detalle de las interfaces de la simulación	Pág. 56
Tabla 4.1	Comparación de Pandora FMS con HP *OpenView*	Pág. 78

CAPÍTULO 1

1. INTRODUCCIÓN

El monitoreo de la infraestructura de servidores, nodos de telecomunicaciones y equipos que están dentro de la red de una empresa o negocio permite conocer si los mismos tienen condiciones de operación óptimas o presentan inconvenientes en su funcionamiento, por ejemplo tener la información en detalle del funcionamiento de los componentes de un servidor puede alertar cuando sea necesario la actualización de memoria RAM, depuración de discos, etc. Este detalle de conocimiento de los equipos está enfocado en mantener bajo control la infraestructura tecnológica que es un aspecto crítico para la operación de organizaciones que dependen de ella. Por esa razón el principal objetivo del monitoreo es adelantarse a los problemas y generar directivas preventivas y correctivas sobre los sucesos que se presenten en una infraestructura tecnológica. Para esto existen en el mercado varias opciones de software de monitoreo mediante soluciones propietarias y de código abierto que ahorran tiempo y recurso humano en la gestión de incidentes, tareas de mantenimiento preventivo y monitorización de elementos críticos para el negocio.

Dentro de las opciones de software para monitoreo podemos encontrar HP *OpenView* de la empresa Hewlett Packard que es un referente en el campo de la monitorización de elementos de red con su producto *Network Node Manager* (NNM) y HP *Operation Manager* (HPOM), estos realizan las tareas de monitorización de nodos y equipos de telecomunicaciones mediante el protocolo SNMP y agentes desplegados en servidores con sistemas operativos Windows y Unix. Sin embargo esta herramienta es propietaria y bajo licencia, lo que restringe su uso en entornos pequeños y de recursos económicos limitados. Ante esto existen opciones de software libre que realizan tareas similares haciendo uso de características propias de los sistemas operativos, como consultas WMI[1] para Sistemas Operativos Windows o consultas SNMP para hardware propietario como *routers* y *switches*. Entre las herramientas de mayor uso podemos encontrar Zenoss, Cacti, Nagios, Zabbix, Pandora FMS, etc.; siendo precisamente esta última la herramienta que será parte del estudio de este trabajo, de la cual se analizará sus potencialidades en un entorno de trabajo de pruebas que se definirá más adelante en este documento.

---

[1]WMI (*Windows Management Instrumentation*) es un protocolo que corre bajo Sistema Operativo Windows y permite realizar consultas del estado del equipo Windows como: consumo de CPU, memoria, disco, etc.

1.1 Conceptos Básicos de la herramienta de monitoreo [1]

Durante el desarrollo de este trabajo se realizó la implementación y configuración sobre la versión 4.0.2 de Pandora FMS y sus respectivas versiones de agentes de monitoreo sobre los servidores Windows y Linux utilizados. Pandora FMS es un software de monitorización de código libre enfocado en servidores y equipos de telecomunicaciones, implementando para ello varios protocolos de gestión de información y consulta sobre los equipos, siendo su principal objetivo ser lo suficientemente flexible como para gestionar y controlar una infraestructura amplia, sin la necesidad de invertir un alto contingente de tiempo, recurso humano y dinero.

Dentro de las características de la herramienta de monitoreo Pandora FMS están las de incorporar herramientas de código libre para la recolección y gestión de la información, estas herramientas se incorporan al servidor de Pandora FMS a través de "servidores" que para el sistema operativo son los servicios encargados de realizar las tareas de monitoreo, esto se lo puede observar en la Figura 1.1, en este capítulo se los menciona de manera general para entender su importancia.

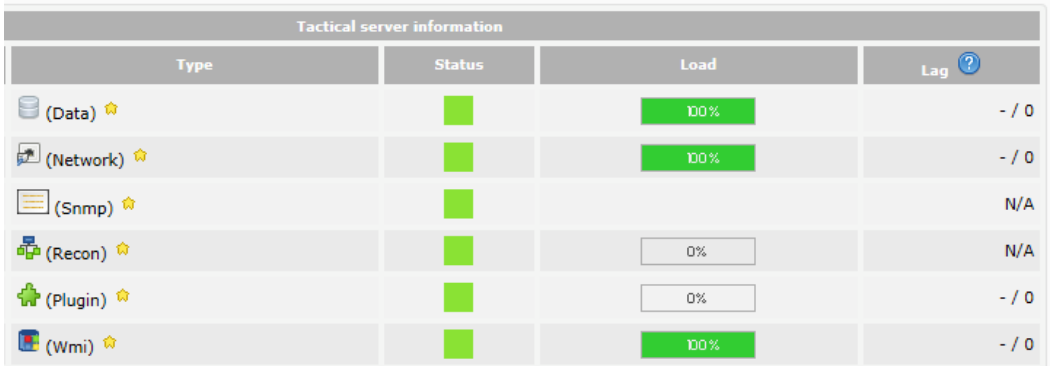


Figura 1.1
Servidores de Pandora FMS

Servidor de Datos.- Es el encargado de procesar la información que envían los agentes instalados en los servidores, para este trabajo se realiza pruebas en equipos con sistemas operativos Windows 2003 SP2 y Linux-Ubuntu 12.04.

Servidor de Red.- Realiza la comprobaciones ICMP y TCP a nivel de red para los nodos que se encuentran configurados en la herramienta.

Servidor SNMP.- Utiliza el demonio de recepción de *traps* llamado *snmptrapd* que se encarga de recibir alertas desde los nodos monitoreados.

Servidor de reconocimiento.- Realiza una comprobación de la red para detectar nuevos elementos. Se lo puede configurar con herramientas como NMAP[2] y TRACEROUTE[3] para detectar la topología y datos específicos como puertos abiertos y el tipo de sistema operativo.

Servidor de plugins.- Permite añadir complementos de desarrollo propio sobre cualquier lenguaje para realizar tareas sobre el servidor de Pandora FMS.

Servidor WMI.- Realiza la consulta de información principalmente sobre servidores Windows y recoge información sobre el sistema operativo y su estado.

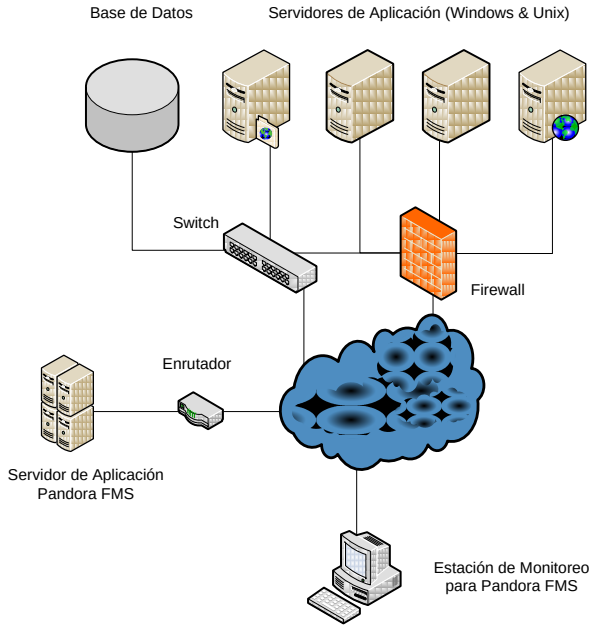


Figura 1.2
Esquema de la infraestructura para monitoreo mediante Pandora FMS

En la Figura 1.2 se puede observar a escala reducida la infraestructura tecnológica que se maneja por las empresas y que requiere de la aplicación de un sistema que alerte sobre los incidentes y permita generar reportes de comportamiento de los elementos para de esta manera planificar el crecimiento, re-potenciación o reemplazo de la infraestructura existente.

---

[2]NMAP es un programa de código abierto que efectúa rastreo de puertos, se usa para evaluar la seguridad de sistemas informáticos, así como para descubrir servicios o servidores en una red informática.

[3]TRACEROUTE es una consola de diagnóstico que permite seguir la pista de los paquetes que vienen desde un host (punto de red). Esta herramienta se llama *traceroute* en sistemas basados en UNIX y *tracert* en sistemas Windows.

La principal ventaja de la herramienta en comparación con otras opciones de código libre es la implementación de agentes de monitoreo que se instalan en los servidores con los siguientes sistemas operativos compatibles: Windows 2000 SP3, Windows 2003, Windows XP, Windows Vista, Windows 7, Windows 2008, SUSE Linux 10, Ubuntu Linux 8.04, Debian Linux, AIX 4.3.3, HPUX 11.x y Solaris 2.6; lo que facilita las tareas de configuración de los equipos mediante un agente configurable en cada servidor.

1.2 Monitoreo de Redes LAN

El monitoreo en Redes LAN que realiza Pandora FMS se ejecuta mediante comprobaciones ICMP y TCP como se observa en la Figura 1.3, por ejemplo para *switches* o *routers* una verificación mediante protocolo ICMP permite conocer si el dispositivo está encendido y trabajando dentro de la red.

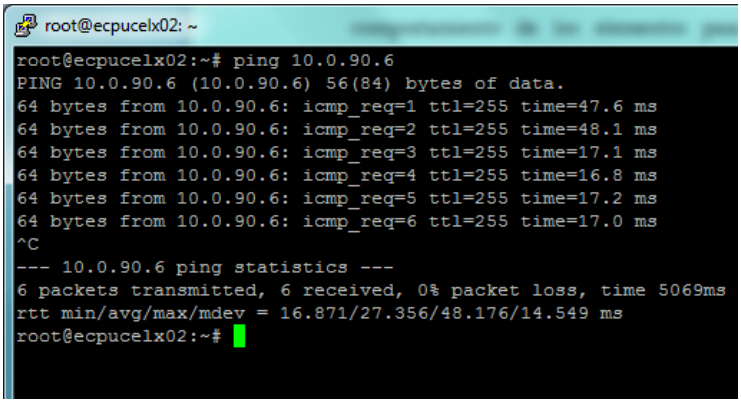


Figura 1.3
Verificación ICMP sobre un nodo de red

Además de una verificación ICMP, como se observa en la Figura 1.3, se puede realizar verificación de los puertos abiertos sobre un dispositivo de red y determinar si los servicios que se acceden están disponibles, para esto se realiza una comprobación de puertos mediante la herramienta NMAP y seguidamente se realiza una prueba de Telnet para verificar si el servicio está operativo sobre el nodo o equipo de la red LAN.

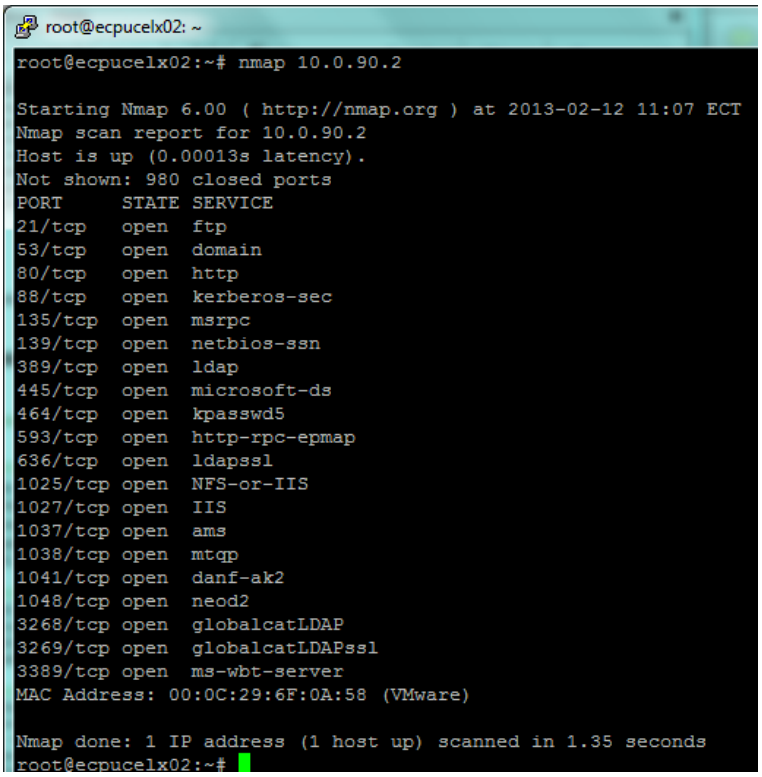


Figura 1.4
Verificación de puertos sobre un nodo de red

En la Figura 1.4 se muestra cómo se realiza la prueba de reconocimiento de puertos sobre un servidor de Directorio Activo y Controlador de Dominio, entre los puertos que se pueden observar está el del protocolo de autenticación *kerberos* (*Puerto 88*), *ldap* (*Puerto 389*) y escritorio remoto (*Puerto 3389*). Toda esta información puede ser gestionada por la herramienta de monitoreo a fin de verificar el buen estado de la infraestructura.

1.3 Protocolo SNMP [2]

El Protocolo SNMP (*Simple Network Management Protocol*) es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el desempeño, buscar y resolver problemas, y planear su crecimiento. Las versiones de SNMP más utilizadas son dos: SNMP versión 1 (SNMPv1) y SNMP versión 2 (SNMPv2);ambas versiones tienen un número de características en común, pero SNMPv2 ofrece mejoras, como por ejemplo, la introducción de tres nuevas operaciones de protocolo: *GetBulk* para que el gestor recupere de una forma eficiente grandes bloques de datos, tales como las columnas de una tabla; *Inform* para que un agente envíe información espontánea al gestor y reciba una confirmación y finalmente *Report* para que el agente envíe

de forma espontánea excepciones y errores de protocolo. SNMP en su última versión (SNMPv3) incorpora cambios significativos con relación a sus predecesores, sobre todo en aspectos de seguridad.

SNMP trabaja en dos modalidades para obtener información de los equipos administrados, la primera es realizando un sondeo o *polling* cada determinado tiempo sobre los elementos administrados para conocer el estado de valores como: CPU, memoria, estado de las tarjetas de red, etc.; el segundo modo de trabajo de SNMP es la captura de alertas o *traps* provenientes de los equipos monitoreados que se lanzan cuando suceden alguna de las siguientes acciones:

- Se produce una caída en la interfaz del equipo.
- Se estropea el ventilador de un *router* o *switch*.
- Se llena un *filesystem* de un servidor.
- Un UPS cambia de estado

1.4 Protocolo WMI [3]

Windows Management Instrumentation (WMI) es la infraestructura para la administración de datos y operaciones en los sistemas operativos basados en Windows. Se pueden escribir scripts WMI o aplicaciones para automatizar tareas en equipos remotos. El Protocolo WMI además suministra datos sobre el sistema operativo e información de otros productos Microsoft instalados en el equipo Windows como por ejemplo *System Center Operations Manager* o *Windows Remote Management*. WMI se puede utilizar en todas las aplicaciones basadas en Windows, y es usado en aplicaciones empresariales y scripts administrativos. Para ello los administradores de sistemas pueden encontrar información sobre el uso de WMI en el *TechNet* de Microsoft.

1.5 Justificación e Importancia del Proyecto

Para la realización de este proyecto se profundizará en el estudio y conocimiento del protocolo SNMP y la tecnología instrumental de administración de Windows WMI, con el propósito de recolectar información de servidores y equipos de telecomunicaciones. Además se hará uso de comprobaciones de red elementales como consultas ICMP, consulta de puertos, etc., para verificar el estado del equipo monitoreado.

Todas estas tareas se las realizará a través de la herramienta de monitoreo Pandora FMS que permite configurar alertas para que los administradores tomen acciones preventivas y correctivas sobre la infraestructura, teniendo para ello información del comportamiento de indicadores como procesador, memoria y espacio en disco.

Una de las características que tiene la herramienta de monitoreo consiste en usar agentes en los servidores con sistemas operativos compatibles, con el objetivo de realizar monitoreo y recolección de información. Esta opción de la herramienta es especialmente útil pues simplifican las tareas de configuración de la herramienta y no depende de los servicios centralizados de la herramienta.

Finalmente la realización de este trabajo permitirá utilizar herramientas de Administración de Red que sean de uso libre y que permitan profundizar en los conocimientos teóricos adquiridos durante las clases de la maestría con un enfoque de aplicación práctica.

1.6 Objetivos

- Explorar las capacidades de la herramienta Pandora FMS para el monitoreo de nodos de telecomunicaciones y servidores.
- Explorar las capacidades del protocolo SNMP para recolección de información y como mecanismo de aviso cuando se envían *traps* o mensajes de alerta a la consola central de monitoreo.
- Realizar la instalación de la herramienta y desarrollar una guía práctica de consulta para futuras implementaciones de la herramienta.
- Explicar los beneficios de la utilización de la herramienta en comparación con la herramienta HP *OpenView* de código propietario.
- Analizar los datos de la base de datos *MySQL* a través de la consola web de la herramienta Pandora FMS para la generación de reportes sobre los equipos monitoreados y creación de un inventario de los equipos (CMDB).
- Generación de reportes usando las opciones que presenta la herramienta Pandora FMS.

1.7 Contenido

Este proyecto en su primer capítulo presenta información en forma resumida respecto de los conceptos básicos y del marco de desarrollo utilizado para alcanzar los objetivos planteados. En el segundo capítulo se presentan todos los fundamentos teóricos necesarios para entender el funcionamiento de la herramienta de monitoreo. En el tercer capítulo se

realiza una explicación detallada del software usado y de todas las configuraciones necesarias para realizar el monitoreo de la infraestructura desplegada. En el cuarto capítulo se explican configuraciones adicionales de la herramienta para un monitoreo más avanzado y específico de la infraestructura, además se analizan los resultados y se hacen conjeturas sobre los mismos, previo al desarrollo de las conclusiones finales. En el capítulo cinco, se presentan los resultados de la implementación de la herramienta en el entorno de pruebas. Finalmente en el capítulo seis se termina con recomendaciones para trabajos futuros sobre esta tesis.

CAPÍTULO 2

2. MARCO TEÓRICO

A lo largo de la tesis se mencionan conceptos y definiciones que permiten entender el desarrollo de este proyecto, por lo que este capítulo se dedica enteramente a aclarar esos fundamentos básicos. Algunos de los puntos a tratarse serán acerca del protocolo SNMP, WMI y conceptos propios de la herramienta de monitoreo utilizada.

2.1 Protocolo SNMP

2.1.1 Historia del Protocolo SNMP [2][4]

El Protocolo SNMP se usa para intercambiar información de gestión entre los dispositivos de la red. Su principal tarea consiste en monitorizar y administrar redes manteniendo un esquema común entre los dispositivos gestionados. Hasta la llegada del Protocolo SNMP, la gestión de red había sido propietaria y los productos se desarrollaban por cada fabricante, complicando la gestión de redes heterogéneas, además, dada la dificultad de desarrollar este tipo de productos y el mercado restringido al que iban dirigidos, los productos eran caros y complejos. Con el crecimiento de la popularidad de TCP/IP, apareció un mercado lo suficientemente atractivo para que la IETF[4] propusiera un estándar de gestión. Una característica de cuidado en estos sistemas es que pueden generar tanto tráfico en la red y tanta información para gestionarse a sí mismos que llegan a convertirse en una carga significativa para la red. Para que una gestión estándar tenga éxito debe basarse en un software simple, de bajo consumo de recursos (procesador, memoria y disco), y costo reducido para poder instalarlo de forma masiva en las redes administradas.

SNMP fue publicado inicialmente en 1989 pero las primeras aplicaciones no aparecieron hasta 1990.SNMP en su Versión 2 apareció en mayo de 1993 añadiendo nuevos comandos para reducir el tráfico de red, especialmente en redes grandes, además ofrece nuevas capacidades de notificación de errores, introduce la definición de nuevos objetos, más contadores, mejores herramientas de gestión y añadidos para garantizar la seguridad y la autenticación. SNMP en su última versión (SNMPv3) data de 2002 y posee cambios significativos con relación a sus predecesores, sobre todo en aspectos de seguridad.

---

[4] *The Internet EngineeringTaskForce* (IETF), es la organización encargada de hacer de Internet un mejor lugar, produciendo información técnica de alta calidad para influenciar la forma en la que las personas desarrollan, usan y administran Internet. Página web: http://www.ietf.org/

2.1.2 Componentes básicos de SNMP [2][4][5]

En entornos de red administrados con el Protocolo SNMP existen componentes que deben existir para que éste funcione y que se detallan en la Tabla 2.1.

Componente	*Descripción*
Dispositivos administrativos	Elementos de una red administrada que contienen un agente SNMP. Tales como *routers*, *switches*, servidores de acceso, computadores, impresoras, *hubs*, *bridges*.
Agente	Es un componente de software que se ejecuta en el dispositivo a gestionar. Es un elemento pasivo y no origina mensajes, al contrario responde a las peticiones del NMS[5]. Únicamente iniciará la comunicación cuando deba comunicar una alarma porque el sistema se ha reiniciado o por fallos de seguridad en el sistema.
NMS(Network Management System)	Ejecuta aplicaciones que supervisan y controlan a los dispositivos administrados mediante SNMP. Estos NMS's conectan con los agentes SNMP para proporcionar volumen de recursos de procesamiento y memoria requeridos para la administración de la red. Cuando un NMS envía una solicitud, el agente devuelve la información solicitada desde el MIB.
Base de información de gestión (MIB)	Los recursos de la red que se pueden gestionar se presentan mediante objetos aunque básicamente son variables. La MIB es una colección de estos objetos.

Tabla 2.1
Componentes en una red con SNMP[6]

La comunicación entre los dispositivos administrados y el NMS requiere de un servicio de comunicación que en el caso de SNMP utiliza un protocolo no orientado a conexión (UDP[7]) para enviar un pequeño grupo de mensajes (PDUs) entre el NMS y los agentes. La utilización de un mecanismo de este tipo asegura que las tareas de administración de red no afectarán al rendimiento global de la red, ya que se evita la utilización de mecanismos de control y recuperación como los de un servicio orientado a conexión como TCP. Los puertos comúnmente utilizados para SNMP son el 161 para mensajes SNMP de consulta y configuración SNMP; y el puerto 162 para alertas tipo *traps* que envía el dispositivo administrado a través del agente.

Los paquetes utilizados para enviar consultas y respuestas SNMP poseen el formato indicado en la Figura 2.1. En los tres campos se contiene información referente a la versión, comunidad y el tipo de dato solicitado.

---

[5] NMS, *Network Management System*: es el encargado de realizar la gestión de la monitorización.
[6]Fuente:
http://es.wikipedia.org/wiki/Simple_Network_Management_Protocol#Componentes_b.C3.A1sicos
[7]UDP, *User Datagram Protocol:* es un protocolo del nivel de transporte basado en el intercambio de datagramas.

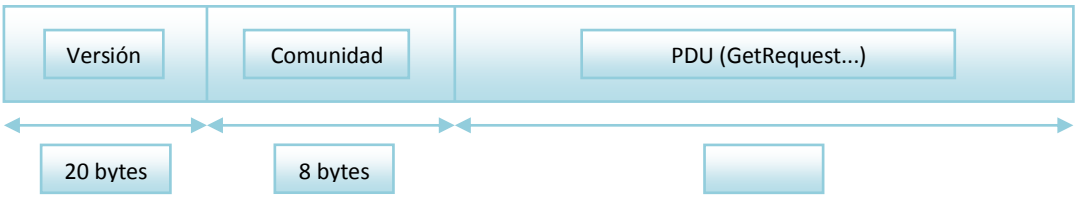


Figura 2.1
Formato del paquete SNMP[8]

- *Versión*: Toma el valor en base de la versión del protocolo SNMP que se está utilizando.
- *Comunidad*: Nombre o palabra clave que se usa para la autenticación. Generalmente existe una comunidad de lectura llamada "*public*" y una comunidad de escritura llamada "*private*".
- *PDU*: Contenido de la unidad de datos del protocolo, que depende de la operación que se ejecute. Se consideran los mensajes que pueden ser del tipo mostrado en la Tabla 2.2.

Mensaje	*Descripción*
GetRequest	Mensaje de solicitud SNMP básico. Enviado por un NMS, solicita información de una sola entrada MIB a un agente.
GetNextRequest	Tipo extendido de mensaje que se utiliza para ver la jerarquía completa de objetos de administración. Se devuelve el valor del siguiente objeto
SetRequest	Es un mensaje para actualizar un valor en el MIB, siempre que exista acceso de lectura.
GetBulkRequest	Solicitud de transferencia de datos tan grande como sea posible. Minimiza el número de intercambios requeridos para obtener una gran cantidad de información. Solo utilizado en versiones 2 y 3 de SNMP.
Trap	Mensaje no solicitado por un NMS que envía el agente cuando detecta cierto evento. Es el tipo de mensaje que inicia el agente cuando se produce una alarma.

Tabla 2.2
Tipos de mensajes SNMP[9]

Los mensajes del campo PDU tienen una estructura predeterminada que incluye los siguientes campos como se muestra en la Figura 2.2.

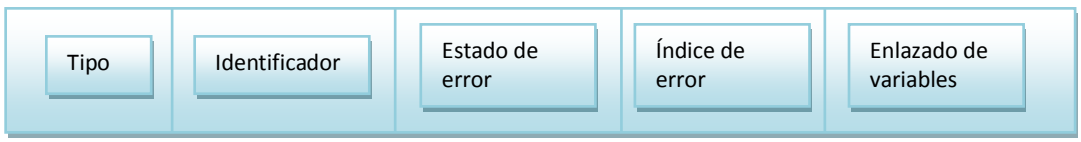


Figura 2.2
Campos PDU de una trama SNMP[10]

[8]"Introducción a SNMP". Estébanez Sevilla. Página 11.
[9]"Introducción a SNMP". Estébanez Sevilla. Página 11.
[10]"Introducción a SNMP". Estébanez Sevilla. Página 13.

- *Tipo*: Cualquier tipo de mensaje de los indicados en Tabla 2.2.
- *Identificador*: Se utiliza para relacionar peticiones y respuestas. El emisor asigna números de manera que cada consulta pendiente al mismo agente es identificada de manera inequívoca de modo que la aplicación SNMP puede correlacionar las respuestas emitidas con las peticiones pendientes y hacer frente a PDU duplicadas por un servicio de transporte inseguro como UDP.
- *Estado e índice de error*: Se emplea para indicar que ha ocurrido una anomalía mientras se procesaba una consulta. Los valores posibles son los indicados en la Tabla 2.3.
- *Enlazado de variables*: Es una serie de nombres de variables con sus valores correspondientes (codificados en ASN.1).

Estado de error	*Nombre*	*Significado*
0	*NoError*	No hay error
1	*tooBig*	Demasiado grande
2	*noSuchName*	No existe esa variable
3	*badValue*	Valor incorrecto
4	*readOnly*	El valor es de solo lectura
5	*genErr*	Error genérico

Tabla 2.3
Estado de Errores para consultas SNMP[11]

2.1.3 Estructura de la base de información de gestión o MIB [4][5]

La base de información de gestión o *Management Information Base* por sus siglas en inglés, es una base de datos a través de la cual se tiene acceso a la información para la gestión de un determinado dispositivo, la información está contenida en la memoria interna del dispositivo en cuestión. MIB es una base de datos completa y bien definida, con una estructura en árbol, adecuada para manejar diversos grupos de objetos (información sobre variables/valores que se pueden adoptar), con identificadores exclusivos para cada objeto. A manera de resumen se define una MIB como una colección de información que está organizada jerárquicamente y que es accedida usando un protocolo de administración de red como SNMP.

[11] "Introducción a SNMP". Estébanez Sevilla. Página 13.

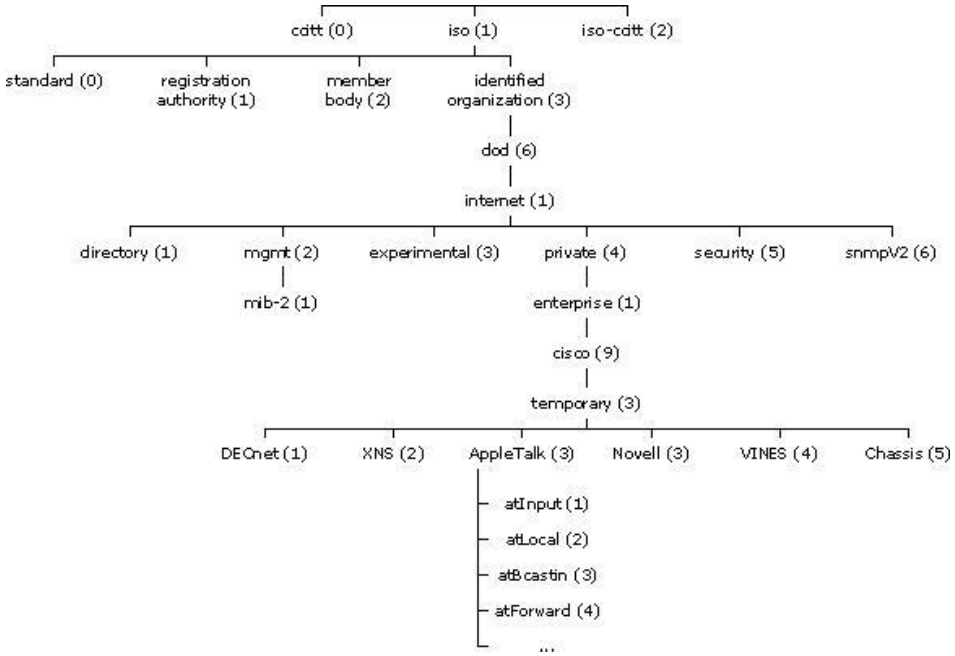


Figura 2.3
Estructura del árbol de una MIB[12]

En la Figura 2.3 se muestra la estructura general del árbol de la MIB, en dónde a medida que se desciende por el mismo es posible encontrar información acerca de los diferentes fabricantes e información sobre los dispositivos administrados.

2.2 *Windows Management Instrumentation*–WMI

2.2.1 Historia de WMI [6]

El *Windows Management Instrumentation* (WMI) es la implementación de Microsoft del *Web Based Enterprise Management* (WBEM) que es una iniciativa de la industria para desarrollar un estándar de tecnología para acceso a información de administración. WMI usa el *Common Information Model* (CIM) que es un estándar de la industria para representar sistemas, aplicaciones, redes, dispositivos y otros equipos administrados. CIM es desarrollado y mantenido por el *Distributed Management Task Force* (DMTF).

WBEM trabaja independientemente del vendedor, protocolo y del estándar de administración, sin reemplazar a estándares de administración como SNMP. Esta tecnología realiza la gestión de recursos como registros, hardware, software, dispositivos, aplicaciones, etc.

---

[12] Fuente: http://es.wikipedia.org/wiki/Simple_Network_Management_Protocol

2.2.2 Arquitectura WMI [6][7]

La arquitectura de WMI posee tres capas: capa de recursos administrados, capa de aplicaciones consumidoras y capa de infraestructura WMI, como se puede observar en la Figura 2.4.

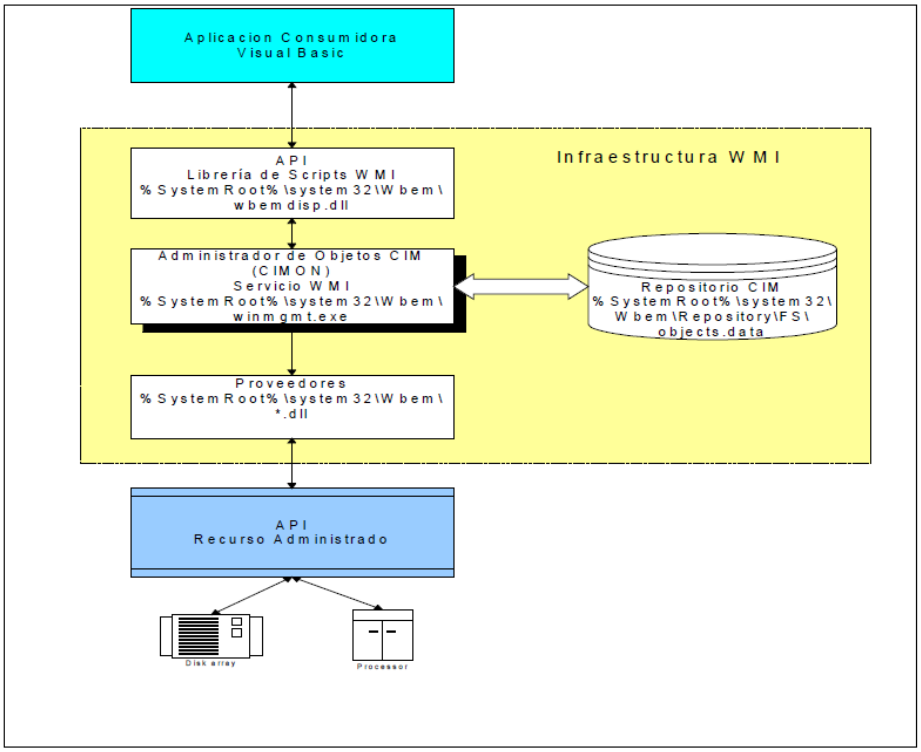


Figura 2.4
Arquitectura WMI[13]

Recursos Administrados.- es cualquier componente físico o lógico, el cual es expuesto y administrado mediante WMI.

Aplicación consumidora.- es una aplicación basada en Windows o un servicio de Windows que procesa los datos solicitados a un objeto administrado, o información provista por el objeto sin ningún tipo de solicitud. Esta aplicación puede realizar diferentes tipos de tareas como: medida de rendimiento, inventario de componentes de equipos, eventos, etc. Una solicitud a un objeto administrado puede ser realizada usando una consulta mediante el lenguaje WQL.

Infraestructura WMI.- está compuesto de tres elementos: el CIMON (Administrador de objetos CIM), el repositorio CIM, y los proveedores. Mediante esta infraestructura los

[13] Fuente: "Sistema para la gestión de computadoras bajo la Plataforma Windows usando WMI a través de páginas web" – Arévalo Leonardo.

datos de configuración y administración se pueden definir, exponer, acceder y obtener su información.

CIMON (Administrador de objetos CIM).- Es el componente principal de la infraestructura WMI porque maneja las interacciones entre las aplicaciones consumidoras y los proveedores. Las aplicaciones consumidoras, proveedores de objetos y extensiones de esquema interactúan con el CIMON por medio de cualquier lenguaje de programación que pueda registrar objetos COM, como C, C++, debido a que CIMON provee una interfaz de programación basada en COM.

Repositorio CIM.- (CIM o Modelo de Información Común) es un esquema orientado a objetos, y no depende de la implementación para describir la información de gestión de la red. La función del repositorio es almacenar datos estáticos, que generalmente son los datos operacionales de WMI como información del contenedor, y almacenar los identificadores de los recursos administrados.

Dentro de la información del repositorio CIM constan las clases a las que se accede mediante consultas WQL. En la Figura 2.5 se indica una de las muchas consultas de información que se pueden hacer al repositorio.

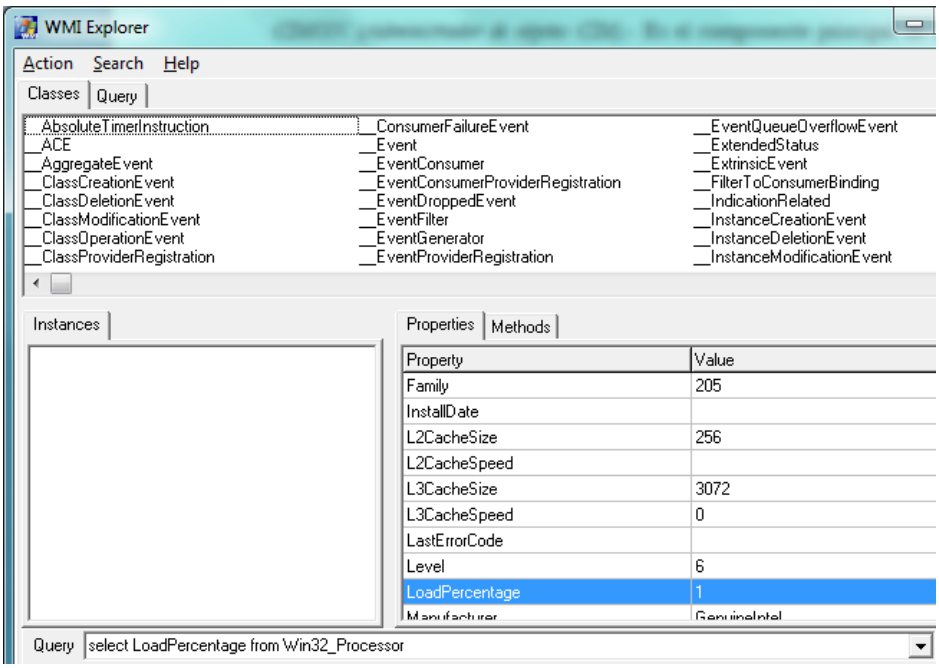


Figura 2.5
Repositorio CIM mediante WMI Explorer[14]

[14]Se usa el software WMI *Explorer* para realizar la consulta de objetos en el repositorio CIM. Fuente: http://www.ks-soft.net/hostmon.eng/wmi/index.htm

2.3 Pandora *Flexible Monitoring System* (FMS) [1]

Pandora FMS es una aplicación de monitorización encargada de vigilar cualquier tipo de sistema y/o aplicación. Permite conocer el estado de cualquier elemento de la infraestructura, es decir se encarga de vigilar el hardware, software, aplicaciones y el sistema operativo, además de detectar si una interfaz de red se ha caído. La herramienta también puede enviar mensajes en el momento que falle cualquier sistema o aplicación dentro de la infraestructura que se realiza el monitoreo. El motivo por el que se ajusta a estas necesidades, es porque está diseñado con herramientas de uso libre, es modular, multiplataforma y fácil de personalizar. Esta herramienta está orientada para los administradores de sistemas, pero puede adaptarse a diferentes entornos de software o hardware.

Sin embargo de todo lo indicado Pandora FMS no es una herramienta de análisis aunque permite recolectar dicha información, pero no está ni diseñada ni pensada para ser eficiente recogiendo esta información. También recoge *logs* de los eventos pero tampoco es su función principal y puede dar lugar a errores si la información de los mismos es muy grande. No es un sistema de detección de intrusos. Finalmente no se considera un sistema de monitorización entiempo real o de monitorización de entornos críticos.

Para la instalación de la herramienta se debe cumplir con los siguientes requerimientos de hardware y software que constan en el manual de la herramienta para la versión 4.0.2.

a. Requisitos mínimos para la consola y el servidor

Aunque puede trabajar sobre cualquier sistema operativo con Perl 5.8 instalado y con *iThreads* habilitados, se recomienda y está soportado únicamente sobre Linux, siendo las distribuciones recomendadas SUSE y Ubuntu/Debian. En la Tabla 2.4 se encuentra detallado los requerimientos de hardware según la cantidad de nodos que se van a monitorear.

Número de Agentes Monitoreados	*Recomendación de CPU y Memoria RAM*	*Recomendación de disco*
Hasta 500 agentes o 5.000 módulos	3GB de RAM y una CPU de un sólo núcleo a 2GHz de reloj. Disco duro rápido, 7200rpm o equivalente	15GB de disco, recomendado 25GB.
Hasta 2.000 agentes o 10.000 módulos	4GB de RAM y una CPU de doble núcleo a 2.5GHz de reloj y disco duro rápido (7.200 rpm o más)	15GB de disco, recomendado 25GB.
Para más de 4.000 agentes	12GB de RAM, una CPU con cuatro núcleos a 3GHZ y disco duro muy rápido (15.000 rpm o más)	15GB de disco, recomendado 25GB.

Tabla 2.4
Recomendación para asignación de recursos al servidor de Pandora FMS[15]

Según el manual de configuración de Pandora FMS no se recomienda utilizar el software en ambientes virtuales, porque los requisitos de acceso a disco son muy estrictos. Pero en caso de ser ese el caso se recomienda asignar disco independiente, así como RAM y CPU. En estos escenarios se recomienda usar discos SAN[16]. Sólo se debería usar una máquina virtual como entornos de prueba, certificación o desarrollo, pero nunca se recomienda usarlos en sistemas de producción.

b. Requisitos mínimos para el agente de monitoreo

El agente de monitoreo puede ejecutarse correctamente sobre cualquier hardware que pueda correr el sistema operativo mínimo requerido, siendo los de la Tabla 2.5 los soportados, además la instalación del agente requiere de un espacio en disco de 40 *MegaBytes*. Finalmente en entornos UNIX se debe tener al menos Perl 5.8 o superior, opcionalmente el paquete *unzip* instalado correctamente y accesible en el *path* del usuario que va a ejecutar el agente, además durante la instalación del agente en ambientes UNIX se puede presentar la necesidad de instalación de paquetes como SNMPD u otras dependencias para que el agente funcione correctamente y reporte a la consola de Pandora FMS.

[15]Fuente: Pandora FMS 4.0.2 Manual de Usuario página 58
[16]La SAN es una red dedicada al almacenamiento que está conectada a las redes de comunicación de una compañía. Además de contar con interfaces de red tradicionales, los equipos con acceso a la SAN tienen una interfaz de red específica que se conecta a la SAN.

Sistema Operativo	*Soportado*
Windows 2000 SP3	Si
Windows 2003	Si
Windows XP	Si
Windows Vista	Si
Windows 7	Si
Windows 2008	Si
SUSE Linux 10	Si
Ubuntu Linux 8.04	Si
Debian Linux	Si
AIX 4.3.3	Si
HPUX 11.x	Si
Solaris 2.6	Si

Tabla 2.5
Sistemas Operativos soportados para instalación del agente[17]

2.3.1 Funcionalidades de Pandora FMS

Pandora FMS es una herramienta de monitorización que aparte de indicarnos si un parámetro está bien o mal, permite cuantificar el estado de ese parámetro como por ejemplo determinando el porcentaje de uso del procesador, a la vez que almacena ese valor sea numérico o alfanumérico para consultas futuras. Pandora trabaja sobre una base de datos, lo cual permite generar informes y estadísticas de los elementos monitoreados. Todas estas funcionalidades se integran en una arquitectura abierta y distribuida, que se puede desplegar sobre sistemas Windows, Linux, Solaris, Cisco IOS, etc.

La herramienta de monitoreo no solo permite obtener información mediante agentes, sino que mediante SNMP y pruebas de red (TCP, ICMP) puede monitorear cualquier sistema de hardware con conectividad TCP/IP. A manera general la herramienta permite monitorear cualquier proceso o sistema, siempre y cuando se puedan obtener valores numéricos o alfanuméricos.

2.3.2 Arquitectura de Pandora FMS

Pandora FMS es modular y descentralizado, siendo su parte más esencial la base de datos (*MySQL* es la base por defecto teniendo la posibilidad de instalar Bases de Datos *PostgreSQL* y Oracle) en donde se almacena la información. En la Figura 2.6 se puede observar el esquema general de la Base de Datos de Pandora FMS.

[17]Fuente: PandoraFMS 4.0.2 Manual de Usuario página 58

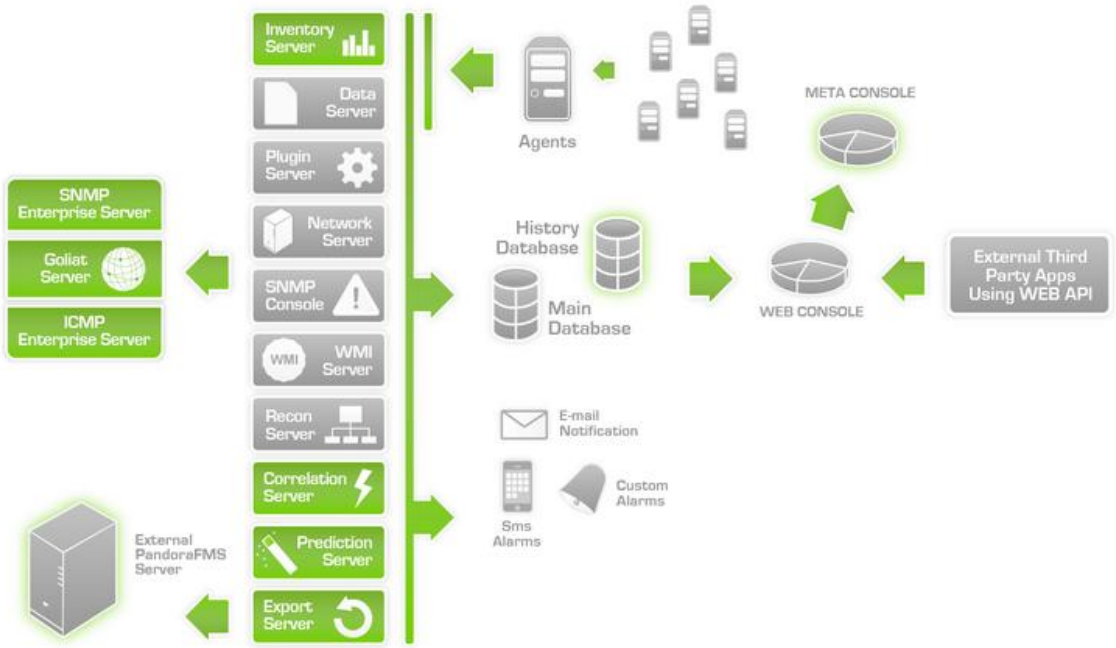


Figura 2.6
Arquitectura de Pandora FMS[18]

Pandora FMS consta de diversos elementos que se encargan de recoger y procesarlos datos de los servidores. Los servidores introducen dichos datos en la base de datos donde la consola se encarga de mostrar dichos datos, para esto se cuenta con agentes que son aplicaciones que corren en los sistemas y recolectan la información para enviarla a Pandora FMS. Existen distintos tipos de servidores, en el siguiente punto se mencionarán algunos de éstos.

2.3.3 Servidores de Pandora FMS

Dentro de la herramienta existen los denominados "servidores" de Pandora FMS, que son servicios de la herramienta que ejecutan diferentes tareas de comprobación y obtención de datos de los equipos que se encuentran monitoreados.

a. Servidor de datos

Es el encargado de procesar la información enviada por los agentes de software. Los agentes envían los datos en formato XML al servidor a través de diferentes medios de envío (FTP, SSH). Periódicamente el servidor verifica si existen nuevos ficheros de datos esperando a ser procesados e ingresados en la base de datos de Pandora FMS.

[18]Fuente: http://www.openideas.info/wiki/index.php?title=File:Arquitectura_mejorada.png

El servidor de datos se ejecuta como un demonio o servicio que se encarga de procesar los paquetes almacenados en su sistema de ficheros. Este servicio sólo trabaja con los datos que provienen de los agentes de software y no realiza ningún tipo de comprobación remota de los mismos como lo hacen los servidores de red o WMI.

b. Servidor de red

El servidor de red se encarga de ejecutar tareas de monitorización remota a través de la red realizando pruebas ICMP, peticiones TCP y peticiones SMTP. En el momento que asignamos un agente a un servidor lo estamos asignando a un servidor de red y no a un servidor de datos. Es importante que las máquinas que ejecutan los servidores de red puedan ver la red para ejecutar las tareas de monitorización asignadas.

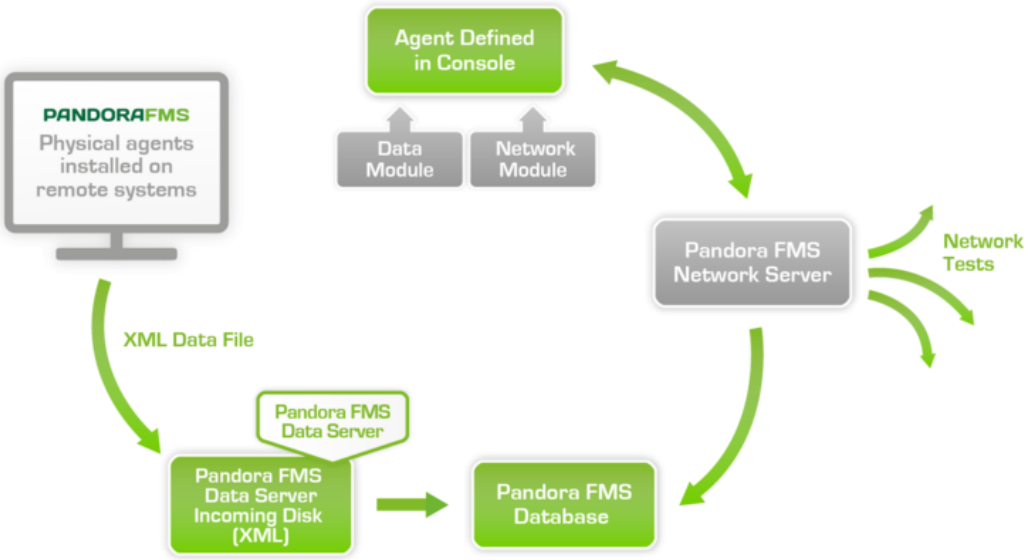


Figura 2.7
Comprobaciones de red de Pandora FMS[19]

c. Servidor de SNMP (también conocido como Consola de Traps SNMP)

El servidor SNMP usa el demonio estándar del sistema de obtención de *traps* (SNMPTRADP). Este demonio o servicio, recibe *traps* SNMP y el servidor SNMP de Pandora los procesa y los guarda en la base de datos. En el momento que procesa y analiza los datos también puede lanzar las alertas asignadas al evento generado.

[19]Fuente:
http://www.openideas.info/wiki/index.php?title=File:Pandora_1.3_Network%26DataServer_Arch.png

d. Servidor de WMI

El servidor WMI es un estándar de Microsoft para obtener información del sistema operativo y de las aplicaciones de entornos Windows. Pandora FMS posee un servidor dedicado para realizar llamadas nativas WMI de forma centralizada. De esta manera se pueden recoger datos de los sistemas Windows de forma remota sin usar ningún agente.

e. Servidor de reconocimiento

El servidor de reconocimiento se usa para explorar regularmente la red y detectar nuevos sistemas en funcionamiento. Al usar aplicaciones de sistema como *nmap*, *xprobe* y *traceroute* dicho servidor permite identificar equipos por su Sistema Operativo, en función de los puertos que tenga abiertos, además establece la topología de red en función de los sistemas que ya conoce.

f. Servidor de complementos (Plugins)

El servidor de complementos se encarga de realizar verificaciones complejas desarrolladas en cualquier lenguaje e integrarlo en la interfaz de Pandora. Esta funcionalidad permite realizar pruebas complejas sobre los sistemas monitoreados, que son lanzadas desde la consola central de Pandora FMS.

g. Servidor de predicción

El servidor de predicción es un componente de Inteligencia Artificial, que realiza una predicción estadística de los datos usando información histórica y permite predecir el valor futuro de una variable. De esta forma el servidor busca determinar si un dato en el momento actual es anómalo, es decir no es correcto, respecto al historial y al análisis que realiza la herramienta.

2.3.4 Consola Web de Pandora FMS

La Consola Web de Pandora FMS es la interfaz de usuario desde la cual se pueden realizar las tareas de administración y operaciones, esto permite a diferentes usuarios con distintos privilegios controlar el estado de los agentes o equipos monitoreados, ingresar o eliminar equipos de la consola de monitoreo, asignar módulos de monitoreo, etc. En general la consola Web de Pandora FMS permite administrar el estado de los equipos monitoreados respecto de los parámetros configurados como se indica en la Figura 2.8.

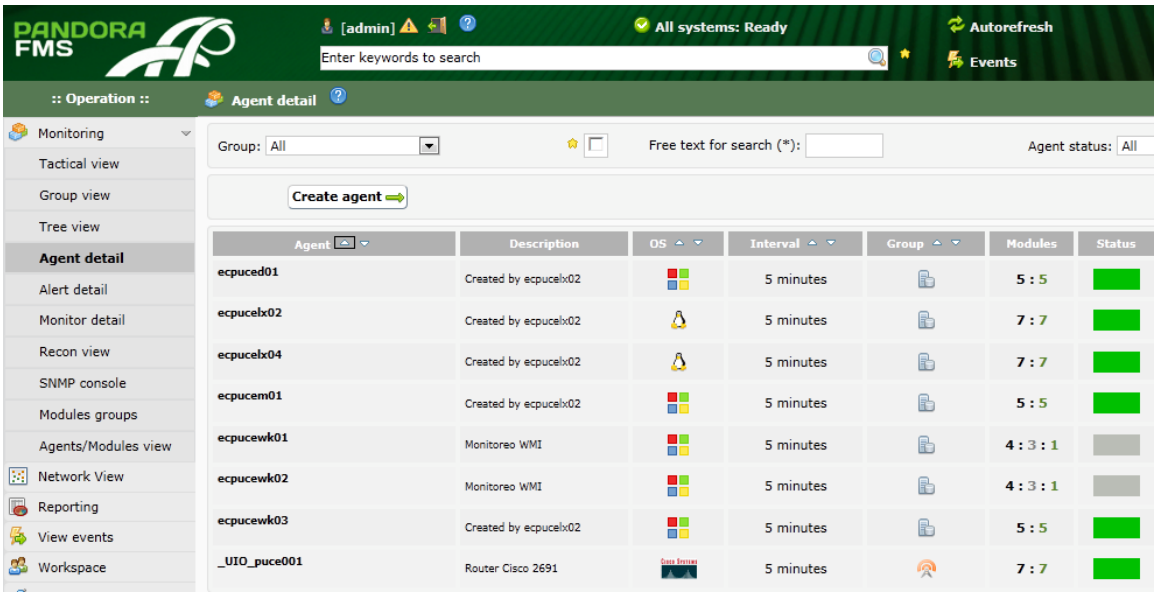


Figura 2.8
Consola Web de monitoreo de Pandora FMS

2.3.5 Base de datos de Pandora FMS

Pandora FMS usa por defecto una base de datos *MySQL*, que recibe datos de manera asíncrona por los agentes de monitoreo desplegados en los equipos de una red LAN. Seguidamente realiza una unión o cohesión temporal de todos los datos y realiza una normalización de dichos datos provenientes de las diferentes fuentes de origen, como se explica en la Figura 2.9.

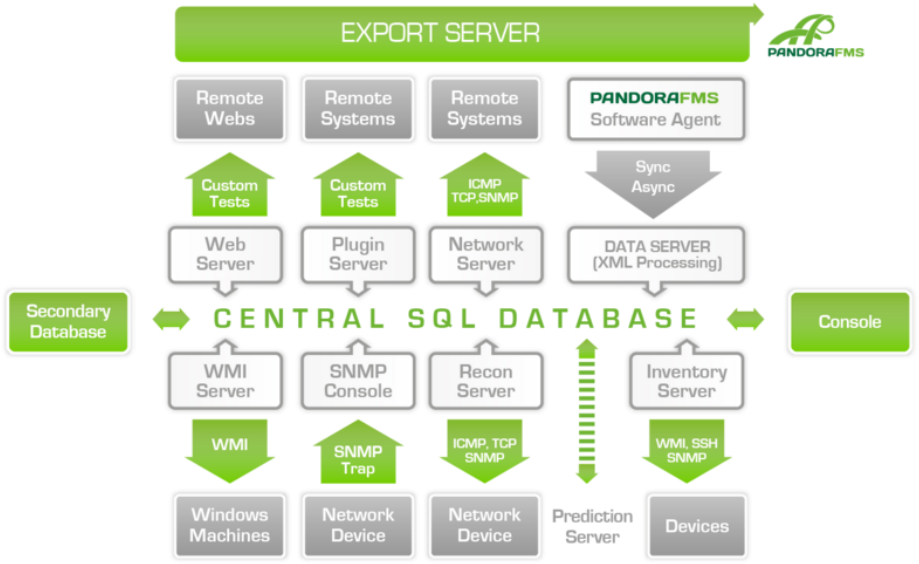


Figura 2.9
Esquema de la Base de Datos de Pandora FMS

Cada módulo de datos de un agente o equipo administrado genera una entrada de datos para cada paquete. Es decir que un sistema real de producción puede tener alrededor de

diez millones de datos. Dichos datos son gestionados de forma automática por Pandora FMS. Como parte de las tareas de mantenimiento automático de la base de datos se lleva una depuración periódica, permitiendo quela herramienta no necesite de ningún tipo de administración de base de datos. Este proceso se lo realiza mediante una depuración periódica de los datos una vez pasado un período de tiempo, que por defecto se programa en 90 días.

2.3.6 Agentes Software de Pandora FMS

Al referirnos a un agente Pandora FMS, podemos distinguir 3 formas de obtener los datos:

- Agente.
- Agente de Software, es una aplicación de Software que corre en una máquina o equipo.
- Agente Físico o de Hardware.

a. Agente

El agente de Pandora FMS es simplemente un elemento organizativo que se crea con la consola Web de Pandora FMS. El cual está asociado a un grupo de módulos o elementos individuales de monitorización. También permite que dicho agente pueda tener de forma opcional asociadas una o más direcciones IP.

b. Agente de Software

El agente de Software es el agente que se instala en una máquina o equipo remoto, el mismo es completamente diferente al del servidor o al de la consola Web de Pandora FMS. El agente de Software permite obtener la información local del equipo donde se está ejecutando.

c. Agente Físico

El grupo de trabajo de la aplicación Pandora FMS ha implementado un agente físico montado sobre un *router Asus* y un autómata *Arduino*. Este equipo junto con los sensores conectados consigue, por el momento, monitorizar características ambientales de humedad, temperatura, luz y presencia. Los sensores son de fácil calibración al ser electrónicos, y sus valores también son fácilmente procesables por Pandora FMS.

2.3.7 Operación con *traps* SNMP

Pandora FMS tiene una consola de recepción de *traps* que visualiza los *traps* que envían los objetos monitorizados y añade alertas a dichos *traps*. Los *traps* SNMP se reciben a través del demonio SNMPD que es iniciado por la herramienta Pandora FMS cuando el servidor se enciende o reinicia. Este servidor, generalmente almacena los *traps* en un log que se ubica en */var/log/pandora/pandora_snmpconsole.log*.

Los *traps* se reciben generalmente en formato "crudo", es decir, con OID's numéricos, a no ser que una MIB instalada en el Sistema Operativo sea capaz de resolverlos. La consola SNMP de Pandora FMS en su versión pagada permite crear reglas para renombrar OID's numéricas a OID's alfanuméricas o convertirlas en simples cadenas de texto descriptivas (p.e: Se ha caído la interfaz) de forma que sea más intuitivo trabajar con los *traps*. Pandora FMS también permite cargar MIB's de *traps* de cualquier fabricante para definir automáticamente esas reglas.

a. Acceso a la consola de recepción de traps

Para acceder a la consola de recepción de *traps* se va a Operación > Consola SNMP, donde aparece la lista de *traps* que se han recibido. Existe un icono (el ojo) que permite desplegar toda la información del *trap*. Aquí se puede ver la información detallada de un *trap* SNMP.

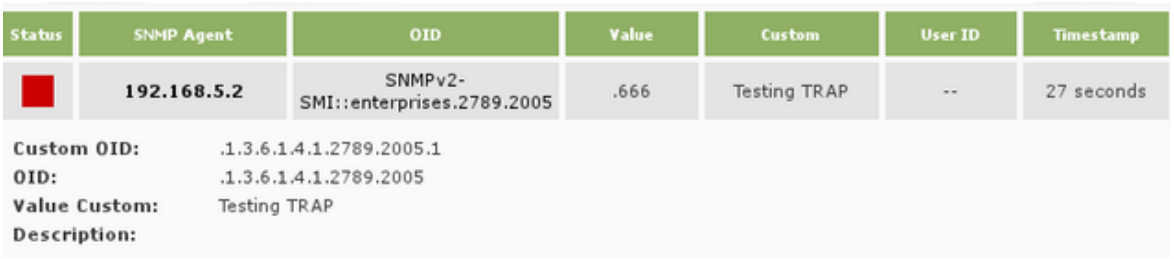


Figura 2.10
Trap SNMP en la consola de recepción

Para cada *trap,* como se observa en la Figura 2.10, existen columnas que poseen información acerca del evento reportado[20], estas proporcionan detalles que permiten identificar el *trap* y determinar su procedencia, cada una de estas columnas se explica en la Tabla 2.6.

[20] En la Figura 2.10 se ha recortado el gráfico por temas de espacio, sin embargo se explica todas las columnas que presenta la herramienta.

Mensaje	*Descripción*
Status	Cuadrado verde si el *trap* se ha validado y rojo si no se ha validado.
SNMP Agent	Agente que ha enviado el *trap*.
OID	OID del *trap* enviado. Un *trap* solo puede enviar un dato en este campo.
Value	Campo *value* del *trap* enviado. Un *trap* solo puede enviar un dato en este campo.
Custom OID, Custom Value	Campos personalizados enviados en el *trap*. Pueden ser datos muy complejos, que tengan una lógica específica en función del dispositivo que envía el *trap*. Un *trap* puede enviar varios datos en este campo.
Time Stamp	Tiempo que ha pasado desde que se ha recibido el *trap*.
Alert	Cuadrado amarillo si se ha lanzado alguna alerta con este *trap* o cuadrado gris, si no se ha lanzado ninguna alerta.
Action	Campo para borrar o validar el *trap*. Además los *traps* tiene un color (visto como color de fondo de la línea del *trap*) diferente según el tipo de *trap*. • Azul: los *traps* de tipo mantenimiento. • Morado: los *traps* de tipo información. • Verde: los *traps* de tipo Normal. • Amarillo: los *traps* de tipo *Warning*. • Rojo: los *traps* de tipo Crítico.

Tabla 2.6
Tipos de mensajes SNMP[21]

b. Filtrado de traps

En la parte superior de la consola de *traps* aparece la opción "*Toogle Filter*". Pulsando sobre dicha opción aparecen o desaparecen los campos para filtrar *traps*. Ver figura 2.11.

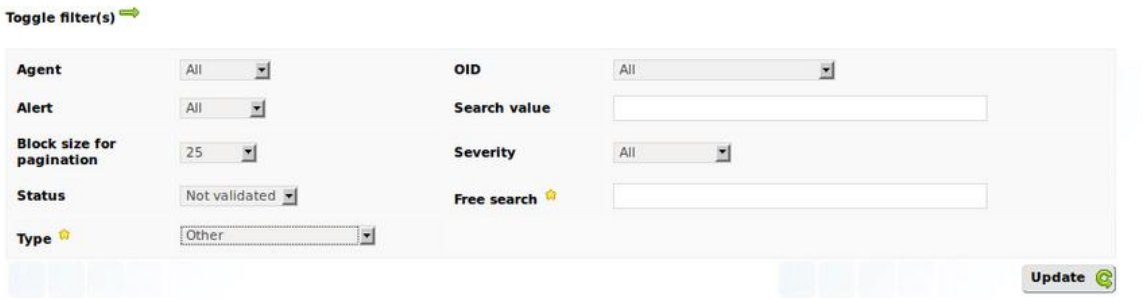


Figura 2.11
Filtrado de *traps* SNMP

Es posible filtrar en la consola de *traps* por campos como el nombre del agente, OID, Alerta, severidad del mensaje, etc. Además de estos campos de búsqueda, está la opción "*Block size of pagination*", que permite definir el número de *traps* que habrá en cada página.

[21]Fuente: PandoraFMS 4.0.2 Manual de Usuario página 306

2.3.8 Alertas de *traps* SNMP

Dentro de Pandora FMS es posible asociar una alerta a un *trap* SNMP para que Pandora informe si se ha producido un evento específico. Esta opción se encuentra en el menú de SNMP -> Alertas SNMP. Los *traps* SNMP por sí solos no generan alertas específicas y deben usar el sistema de generación de alertas de la herramienta. Cada alerta generada por *traps* SNMP se procesa como una alerta independiente de forma que no existen alertas de recuperación (no se puede recuperar un *trap*), ya que en caso de que se reciba otro *trap* para advertir de que algo deja de estar mal, hay que tratarlo como un *trap* diferente.

a. Añadir una alerta

Las alertas de *traps* SNMP tienen varios campos que pueden ser utilizados para buscar datos en el *trap* SNMP. Los campos que se pueden usar, tanto por separado como por combinación son:

- *Description*: Combo para escribir una descripción de la alerta.
- *OID*: OID principal del *Trap*. Es una expresión regular. Si no se usa una expresión regular se buscará la cadena exacta, si se quiere buscar un trozo del OID, se debe usar una expresión regular, de forma que si queremos buscar, por ejemplo: 1.21.34.2.3 en un OID más largo, podemos usar la expresión regular .*1.21.34.2.3.*
- *Custom Value/OID*: Esto busca en los campos "Value" del *trap*, así como en los campos "Custom OID" y "Custom Value", es decir, en el resto de campos del TRAP. Aquí funciona, igualmente, la búsqueda por expresión regular. Por ejemplo si se tiene un *trap* que envía la cadena "Testing TRAP 225"se puede buscar cualquier *trap* con la subcadena "Testing TRAP" con la expresión regular "Testing.*TRAP.*"
- *SNMP Agent*: IP del agente que envía el *trap*. De igual forma, se puede usar una expresión regular o una subcadena.
- *Trap type*: Filtra por tipo de trap pudiendo ser: Cold start, Warm start, Link down, Link up, Authentication failure u Other. La mayoría de los *traps* generados suelen ser de tipo "Other"; si no especifica nada, buscará cualquier tipo de *trap*.
- *Single value*: Filtra por el valor del *trap*. Esto solo hace referencia al valor simple del OID principal, no de cualquier OID secundario.
- *Custom OID/Data #1*: Es una expresión regular. Se deben usar selectores para seleccionar la parte de la expresión regular que se cargará en la macro _snmp_f1_ que

luego se podrá utilizar en los campos Field #1 (Alias, name), Field #2 (Single Line) y Field #3 (Full Text).

- *Custom OID/Data #2*: Es una expresión regular. Se deben usar selectores para seleccionar la parte de la expresión regular que se cargará en la macro _snmp_f2_ que luego se podrá utilizar en los campos Field #1 (Alias, name), Field #2 (Single Line) y Field #3 (Full Text).
- *Custom OID/Data #3*: Es una expresión regular. Se deben usar selectores para seleccionar la parte de la expresión regular que se cargará en la macro _snmp_f3_ que luego se podrá utilizar en los campos Field #1 (Alias, name), Field #2 (Single Line) y Field #3 (Full Text).
- *Custom OID/Data #4*: Es una expresión regular. Se deben usar selectores para seleccionar la parte de la expresión regular que se cargará en la macro _snmp_f4_ que luego se podrá utilizar en los campos Field #1 (Alias, name), Field #2 (Single Line) y Field #3 (Full Text).
- *Custom OID/Data #5*: Es una expresión regular. Se deben usar selectores para seleccionar la parte de la expresión regular que se cargará en la macro _snmp_f5_ que luego se podrá utilizar en los campos Field #1 (Alias, name), Field #2 (Single Line) y Field #3 (Full Text).
- *Custom OID/Data #6*: Es una expresión regular. Se deben usar selectores para seleccionar la parte de la expresión regular que se cargará en la macro _snmp_f6_ que luego se podrá utilizar en los campos Field #1 (Alias, name), Field #2 (Single Line) y Field #3 (Full Text).
- *Field 1*: Campo para poner el parámetro del comando de la alarma Field 1. Éste es el campo que se utilizará en el caso de elegir generar un evento, o el mail de destino en caso de elegir una acción de email (si queremos sobrescribir el mail que tenga por defecto en la acción).
- *Field 2*: Campo para poner el parámetro del comando de la alarma Field 2. En el caso de enviar un email, p.e. será el *subject* del mensaje. Si se deja en blanco utilizaría lo que hubiera definido en la acción.
- *Field 3*: Campo para poner el parámetro del comando de la alarma Field 3. En el caso de enviar un email, sería el texto del mensaje. Si se deja en blanco utilizaría lo que hubiera definido en la acción.
- *Min. Number of Alerts*: Campo donde se define el mínimo número de *traps* que tienen que llegar para que salte la alarma.

- *Max. Number of Alerts*: Campo donde se define el número máximo de veces que se ejecutará la acción.
- *Time Threshold*: Campo donde se define el tiempo que debe pasar antes de resetear el contador de alarmas. Este contador es el que se usa para el campo *Min. Number of alerts*.
- *Priority*: Combo donde se establece la prioridad de la alarma.
- *AlertAction*: Combo donde se elije la acción que va a ejecutar la alerta. Si se elije un evento, el evento normal de generación de alerta no se generará.

En el ejemplo de la Figura 2.12 se define una alerta para la recepción de un *trap* asociado a una OID de prueba, que en este caso es la .1.3.6.1.4.1.2789.2005.666. Aquí se puede observar que en el capo *Type* se indica que se ha lanzado una alerta, de esta forma es posible generar alertas cuando se detecten eventos específicos en la consola de recepción de *traps*, lo facilita la revisión de eventos como caída de enlaces o consumos de CPU y memoria que sobrepasen los umbrales pre-establecidos u de óptimo funcionamiento de los dispositivos.

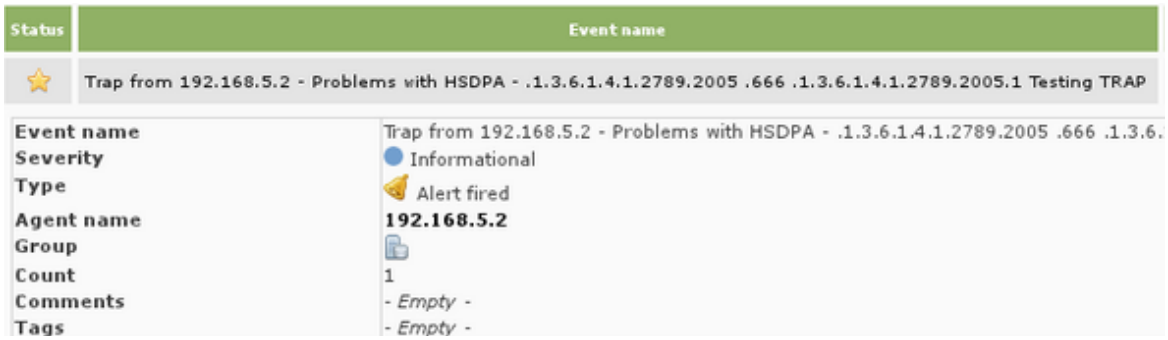


Figura 2.12
Pre-visualización de la alerta a un *trap* SNMP

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN DE LA HERRAMIENTA

Para el desarrollo de este trabajo ha sido necesario crear un ambiente de pruebas en donde se ha desplegado la herramienta de monitoreo y se realiza las configuraciones necesarias que permiten efectuar el análisis de la herramienta. En el presente capítulo se indicarán las configuraciones realizadas y los pasos seguidos para levantar el ambiente de trabajo.

3.1 Diseño del entorno de trabajo

Para la realización de este proyecto se optó por utilizar *vmWare Workstation* 8.0 y crear máquinas virtuales, esto permite aprovechar al máximo los recursos de las máquinas físicas y da la posibilidad de obtener respaldos completos de las máquinas sobre las que se trabaja. Para el ambiente de trabajo se utilizó siguientes equipos físicos:

Servidor 1	
Modelo	*: HP ProBook 4520s*
CPU	*: Intel(R) Core(TM) i3 @ 2.4GHz x 4*
Memoria RAM	*: 8.0 GB DDR3*
Disco	*: 500 GB*
Sistema Operativo	*: Microsoft Windows 7 Professional SP1*

Figura 3.1
Descripción del Servidor 1

Servidor 2	
Modelo	*: Acer Aspire One*
CPU	*: Intel(R) Atom(TM) CPU N570 @ 1.66GHz x 2*
Memoria RAM	*: 2.0 GB DDR3*
Disco	*: 200 GB*
Sistema Operativo	*: Microsoft Windows 7 Professional SP1*

Figura 3.2
Descripción del Servidor 2

Para la creación de los servidores virtuales se minimizó los recursos asignados a los servidores de tal manera que se pueda aprovechar los recursos del Servidor 1, en la Tabla 3.1 se especifica las máquinas virtuales creadas y los respectivos recursos asignados.

Equipo	*CPU Asignado*	*Memoria Asignado*	*Disco Asignado*	*Sistema Operativo – Rol*
ecpuced01	1 procesador	1 GB	40 GB	Windows – Directorio Activo
ecpucelx02	1 procesador	3 GB	30 GB	Ubuntu 12.04 – Pandora FMS *Server*
ecpucelx04	1 procesador	1 GB	20 GB	Ubuntu 12.04 –Pandora Agente
ecpucem01	1 procesador	512 MB	40 GB	Windows – Servidor SMTP
ecpucewk03	1 procesador	1 GB	60 GB	Windows – Pandora Agente

Tabla 3.1
Equipos virtuales creados para el ambiente de pruebas

Los recursos asignados a las máquinas virtuales se basan en los requerimiento mínimos recomendados para sistemas Windows 2003 y Ubuntu que se indican en las Tabla 3.2 y Tabla 3.3 respectivamente, sin embargo se asignó recursos adicionales según el comportamiento de los equipos durante la fase de implementación y pruebas. Para el servidor *ecpucelx02* que es donde se instaló la herramienta de monitoreo Pandora FMS se siguió las recomendaciones de la Tabla 2.4.

Componente	*Requisito*
Equipo y procesador	PC con procesador a 133 MHz mínimo; procesador a 550 MHz o superior recomendado; compatibilidad con hasta cuatro procesadores en un servidor
Memoria	128 MB de RAM mínimo; 256 MB o más recomendado; 4 GB máximo
Disco duro	1,2 GB para instalación de red; 2,9 GB para instalación de CD
Unidad	Unidad de CD-ROM o DVD-ROM
Pantalla	VGA o hardware compatible con redirección de consola mínimo; Super VGA compatible con 800 x 600 o monitor de resolución superior recomendado

Tabla 3.2
Requisitos para Windows 2003[22]

Componente	*Requisito*
Equipo y procesador	AMD o Intel Procesador de 64-32bits
Memoria	512 MB de RAM mínimo;
Disco duro	4 Gb de espacio en HDD (Incluido swap)
Unidad	Unidad de CD-ROM o DVD-ROM
Pantalla	Tarjeta Gráfica VGA, monitor con resolución de 800x600

Tabla 3.3
Requisitos para Ubuntu Server[23]

En el Servidor 2 se instaló la herramienta GNS3[24] sobre la cual corre la simulación del *router* CISCO 2691 que se utiliza para la consulta de información y recepción de *traps* mediante el protocolo SNMP como se lo verá más adelante en este capítulo.

Finalmente para la creación de la infraestructura de pruebas cada uno de los equipos creados cumple una función específica dado que se ha procurado distribuir los roles de los equipos; es así que para el entorno Windows se ha creado un dominio, como se observa en la Figura 3.3, al que todos los servidores con este Sistema Operativo se unirán y se podrá

[22] Fuente: http://technet.microsoft.com/es-es/windoñwsserver/bb430827.aspx
[23] Fuente: http://isft179-ubuntuserver.blogspot.com/2011/09/requisitos-para-instalar-ubuntu-server.html
[24] GNS3 es un simulador gráfico que permite la simulación de redes complejas con equipos de fabricantes como CISCO, Juniper, etc. Fuente: http://www.gns3.net/

trabajar con una configuración centralizada de cuentas de acceso que es de mucha utilidad cuando se trabaja con consultas WMI.

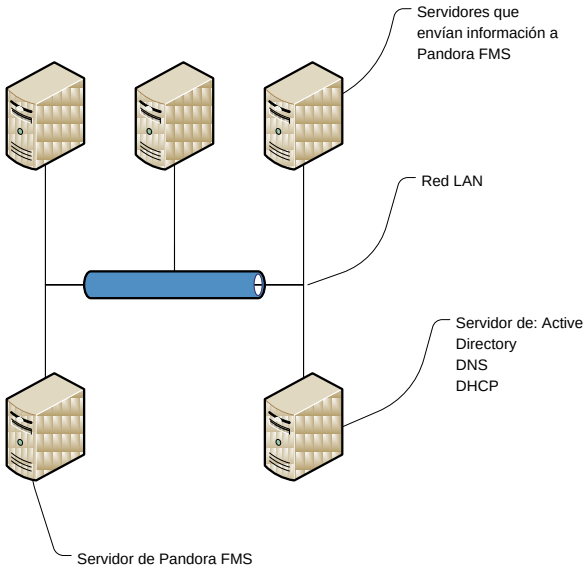


Figura 3.3
Entorno de trabajo con servidor Windows para *Active Directory*, DNS y DHCP

3.2 Implementación de la herramienta de monitoreo

3.2.1 Configuración del servidor Ubuntu 12.04 para Pandora FMS

El servidor de Pandora FMS se instaló bajo Ubuntu Server 12.04 LTS[25]sobre el cual se deben ejecutar los comandos indicados en la Figura 3.4 para actualizar el Sistema Operativo y proceder con la instalación de la herramienta de monitoreo. Adicionalmente se recomienda instalar el paquete de acceso SSH al servidor de monitoreo para continuar el proceso de instalación de forma remota.

```
sudo apt-get update
sudo apt-get install openssh-server openssh-client
```

Figura 3.4
Consola de Ubuntu Server 12.04 LTS desde vmWare WorkStation

Como paso final se recomienda habilitar el acceso de la cuenta *root* al servidor de Ubuntu, lo cual permite realizar tareas de configuración más avanzadas y acceder a lugares restringidos del Sistema Operativo, para esto se deben ejecutar el siguiente comando:

```
sudo passwd root
```

[25]Desde la página oficial del proyecto Ubuntu se realiza la descarga del software.
http://www.ubuntu.com/download/server

El asistente pedirá la clave del usuario con el que se está trabajando para a paso seguido solicitar la clave de acceso para el usuario *root*.[26]

3.2.2 Instalación de los componentes del servidor de Pandora FMS

Para la instalación de la herramienta se debe cumplir con los pre-requisitos que se indican en el manual de Pandora FMS 4.0.2para la instalación del servidor y consola respectivamente, que se explican en el Capítulo 2. Para la instalación de cada paquete se debe ejecutar el siguiente comando y se recomienda realizar la instalación individual de cada paquete con el fin de verificar que el proceso se ha ejecutado sin problemas.

```
sudo apt-get install #nombre del paquete#
Ejemplo: sudo apt-get install snmp
```

Para la instalación del servidor de Pandora FMS se deben instalar los requisitos como consta en la Figura 3.5, además se debe considerar que el paquete de *wmi-client* se debe descargar e instalar manualmente por separado. Este paquete es el que permite realizar consultas WMI hacia los servidores Windows y se debe instalar la versión más reciente del mismo para conectarse a servidores Windows de versiones más actuales.

```
snmp snmpd libtime-format-perl libxml-simple-perl libxml-twig-perl libdbi-perl
libnetaddr-ip-perl libhtml-parser-perl  xprobe nmap libmail-sendmail-perl
traceroute libio-socket-inet6-perl libhtml-tree-perl libsnmp-perl snmp-mibs-
downloader libio-socket-multicast-perl
```

Figura 3.5
Requisitos para la instalación del Servidor de Pandora FMS

Para la instalación de la consola se instalan los componentes de la Figura 3.6, entre los que se incluye el motor de Base de Datos que usa la aplicación de Monitoreo a la Consola.

```
php5 libapache2-mod-php5 apache2 mysql-server php5-gd php5-mysql php-pear
php5-snmp php-db php-gettext graphviz mysql-client php5-curl php5-xmlrpc php5-
ldap dbconfig-common
```

Figura 3.6
Requisitos para la instalación de la consola de Pandora FMS

Durante la instalación de *MySQL* se lanzará un Asistente de Instalación o *Wizard,* como se observa en la Figura 3.7, en el que se pedirá la clave de acceso *root* al motor de la Base de Datos.

---

[26]Fuente: http://www.debianadmin.com/enable-and-disable-ubuntu-root-password.html

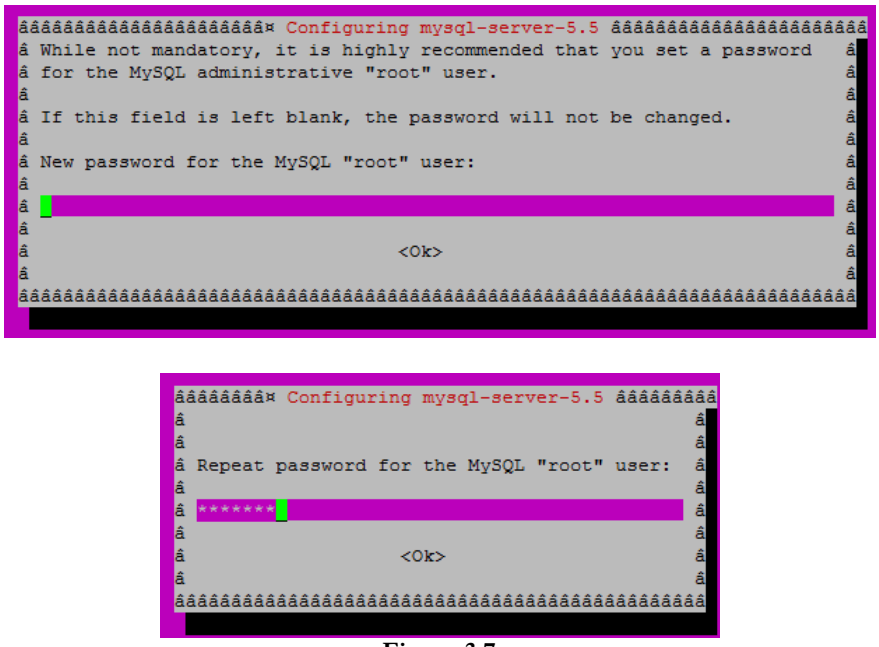


Figura 3.7
Asistente de la instalación para la Base de Datos *MySQL*

Como paso siguiente en la instalación del software y paquetes, es necesario realizar la descarga de los paquetes indicados en la Figura 3.8 para ser instalados en el servidor de Pandora FMS y que se incluyen en la documentación electrónica del proyecto.

```
libnet-traceroute-perl_1.10-1_all.deb
libnet-traceroute-pureperl-perl_0.10-1_all.deb
php-xml-rpc_1.5.2-1_all.deb
libwmiclient1_1.3.14-3_amd64.deb
```

Figura 3.8
Componentes adicionales de la herramienta de monitoreo

Para ello se ejecuta el siguiente comando con el nombre de cada uno de los paquetes a ser instalado desde la consola SSH del servidor de Pandora FMS.

```
sudo dpkg -i #nombre del paquete#
```

```
Ejemplo: sudo dpkg -i libnet-traceroute-perl_1.10-1_all.deb
```

Durante este proceso se pueden presentar errores en la instalación de algún paquete, por lo que será necesario ejecutar el comando indicado para corregir cualquier dependencia o prerrequisito que necesite el paquete y el Sistema Operativo lo buscará automáticamente en Internet.

```
apt-get -f install
```

Para finalizar el proceso de instalación del software de Pandora FMS se debe realizar la instalación de los siguientes paquetes en el orden indicado mediante el comando DPKG.

```
pandorafms.console_4.0.2.deb
pandorafms.server_4.0.2.deb
pandorafms.agent_unix_4.0.2.deb
```

Figura 3.9
Paquetes de instalación de Pandora FMS

Con esto se culmina el proceso de instalación del software y se puede continuar con la configuración de la herramienta mediante un Asistente que se lanza desde una página web.

3.2.3 Configuración del servidor de Pandora FMS

Para empezar con la configuración de Pandora FMS se debe ingresar en la siguiente dirección desde un navegador web y seguir el asistente de configuración.

```
http://ip_servidor_pandora/pandora_console
```

A continuación se presentará las capturas de las pantallas del Asistente de Configuración con una explicación de lo que se debe realizar para avanzar con el proceso.



Figura 3.10
Asistente o *Wizard* de Configuración de Pandora FMS – 1

En la Figura 3.10 se presenta la pantalla de bienvenida en la que se realiza una verificación de las configuraciones y en caso de tener todo correcto se podrá avanzar dando *click* en *Next*.

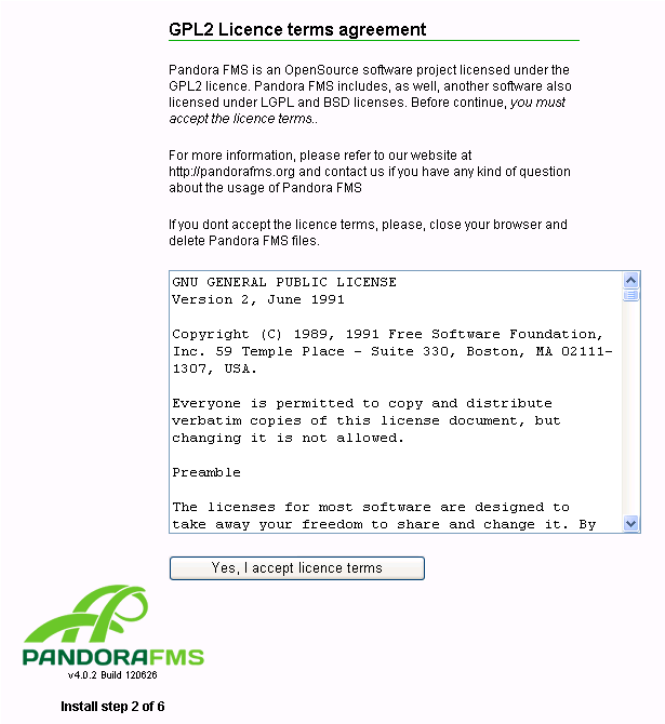


Figura 3.11
Asistente o *Wizard* de Configuración de Pandora FMS – 2

En la Figura 3.11 se presenta el contrato de la licencia GPL2 de uso de Pandora FMS, se acepta el contrato y se avanza a la siguiente pantalla.

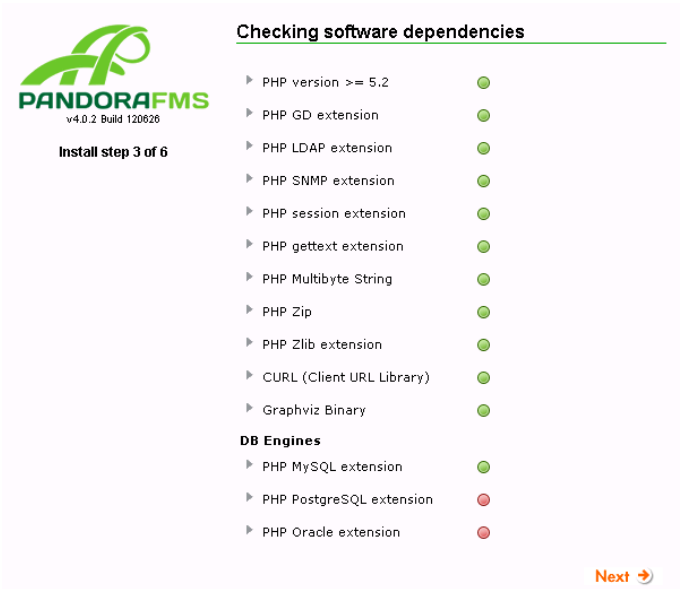


Figura 3.12
Asistente o *Wizard* de Configuración de Pandora FMS – 3

En la Figura 3.12 se observa que se realiza un chequeo de todas las dependencias y del software necesario para el funcionamiento de Pandora FMS, en la Figura están todos en

verde a excepción de la parte final en donde se indica el motor de base de datos escogido para la herramienta es *MySQL*.

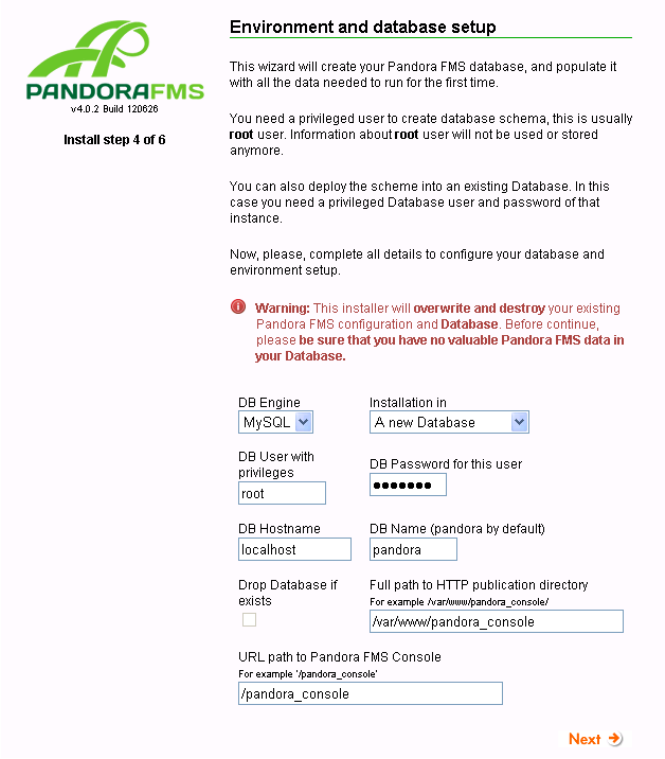


Figura 3.13
Asistente o *Wizard* de Configuración de Pandora FMS – 4

En la Figura 3.13 está la ventana en la que se pide colocar la contraseña de usuario *root* que se definió en la instalación del motor de base de datos *MySQL*, se coloca la clave indicada y se avanza con el proceso de configuración.

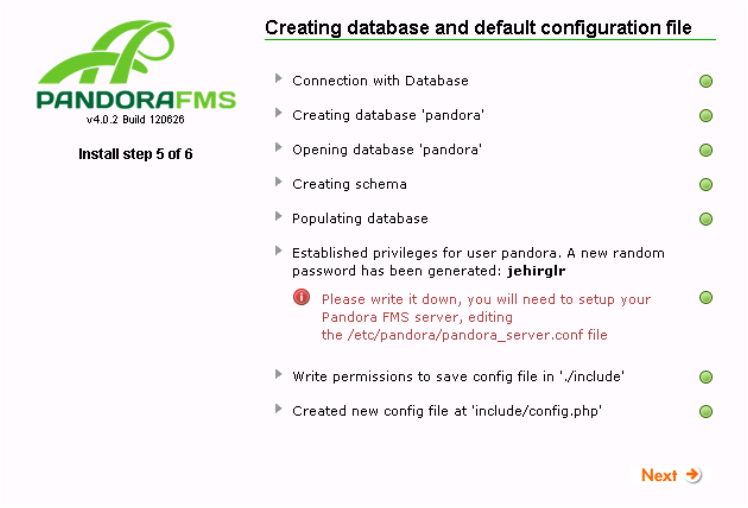


Figura 3.14
Asistente o *Wizard* de Configuración de Pandora FMS – 5

En la Figura 3.14 el Asistente realiza el proceso de creación de la base de datos y se genera la contraseña de *root* de la base de datos que más adelante debe ser colocado en el archivo de configuración de Pandora FMS.

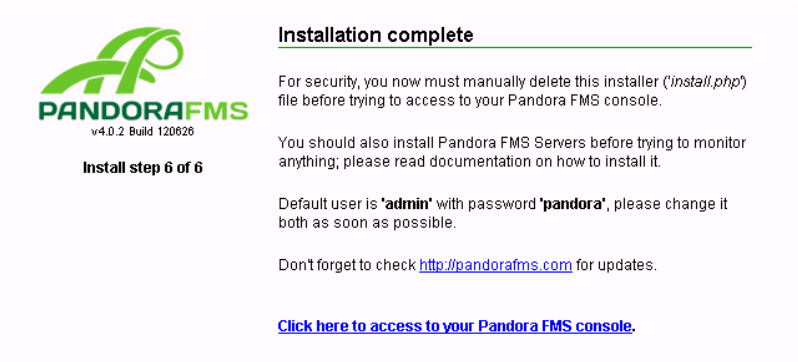


Figura 3.15
Asistente o *Wizard* de Configuración de Pandora FMS – 6

En la Figura 3.15 se puede observar la pantalla final del asistente que indica que se debe borrar el archivo de Instalación de Pandora FMS e introducir el usuario y contraseña por defecto de la herramienta.

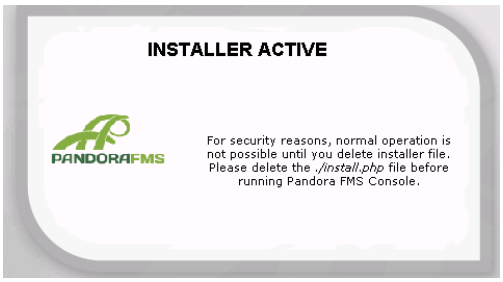


Figura 3.16
Asistente o *Wizard* de Configuración de Pandora FMS – 7

En la Figura 3.16 se accede a la URL del servidor de Pandora FMS *http://ip_servidor_pandora/pandora_console* que muestra que el servidor está activo y que se debe borrar el archivo de Instalación de Pandora FMS. Este archivo se encuentra en la ruta */var/www/pandora_console/install.php*. Mediante una consola SSH al servidor se accede al servidor de Pandora FMS y se realiza la eliminación del archivo de la forma que se indica en la Figura 3.17, posterior a lo cual es posible acceder a la consola web de Pandora FMS.

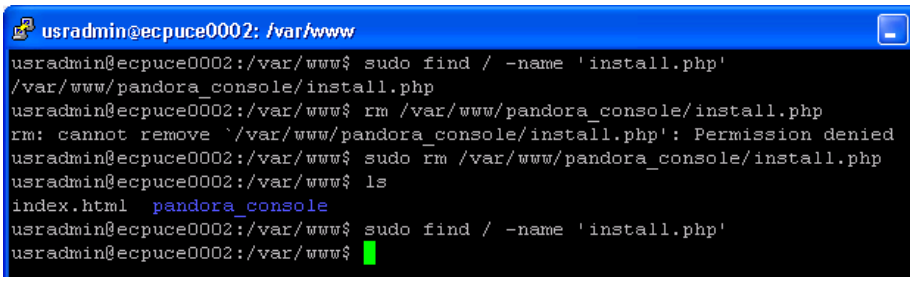


Figura 3.17
Consola SSH para borrar el archivo *install.php*

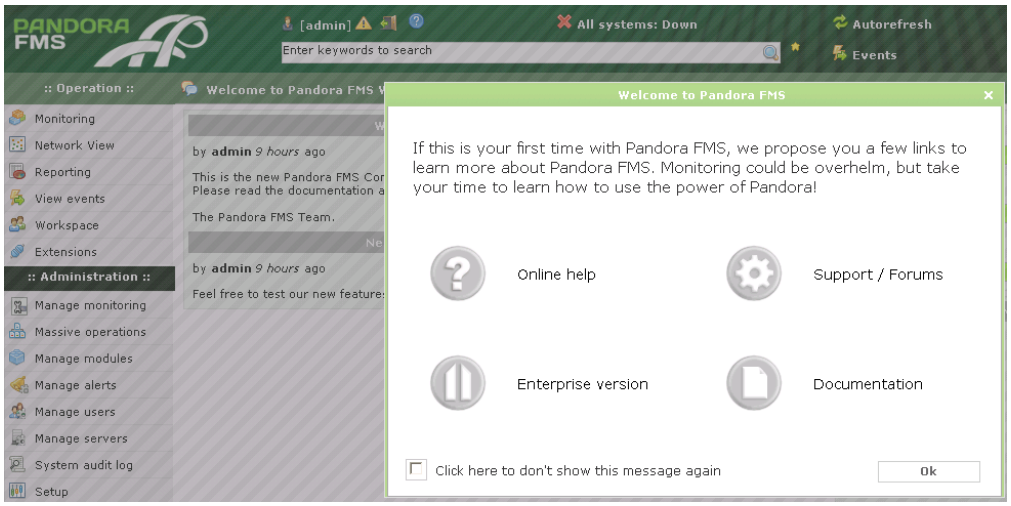


Figura 3.18
Página web de bienvenida de Pandora FMS

En la Figura 3.18 se observa que se ha podido acceder a la consola principal de administración y configuración. En este punto se debe verificar que el acceso al servicio web de Pandora FMS está activo, sin embargo los servicios principales aún no se encuentran levantados y es necesario realizar la modificación del archivo de configuración de Pandora FMS para que los servicios se levanten correctamente, este archivo se encuentra en la siguiente ruta */etc/pandora/pandora_server.conf*.

```
# Servername: Name of this server
# if not given, it takes hostname. It's preferable to setup one
# because machine name could change by some reason.
servername ecpucelx02
.
.
.
# dbpass: Database password
dbpassjehirglr
.
.
.
# wmiserver : 1 or 0. Set to 1 to activate WMI server with this setup
# DISABLED BY DEFAULT
wmiserver 1
```

Figura 3.19
Configuración del archivo *pandora_server.conf*

En la Figura 3.19 se indican las líneas en las que se debe realizar los cambios del nombre del equipo y se debe colocar la clave de acceso *root* a la Base de Datos *MySQL* que se creó durante el proceso de configuración. Finalmente se habilita el servicio de consulta WMI desde el servidor de Pandora FMS para obtener información desde servidores Windows, el valor por defecto viene desactivado. El resto de configuraciones deberán permanecer intactas sin realizar alteraciones en las mismas. Una vez realizado los cambios descritos en el archivo de configuración, es necesario subir los servicios de Pandora FMS por primera vez, para ello se ejecuta los siguientes comandos:

```
sudo /etc/init.d/pandora_server start
sudo /etc/init.d/tentacle_serverd start
sudo /etc/init.d/pandora_agent_daemon start
```

Finalizados estos comandos el servidor de Pandora FMS está listo para recolectar información desde los servidores en los que se instale el agente local o se configure los servicios de monitoreo sin agente como SNMP o WMI, el proceso de instalación de los agentes se los explicará más adelante en este capítulo.

3.2.4 Configuración de la hora del Servidor de Pandora FMS

El servidor de Pandora FMS realiza consultas a los servidores y equipos monitoreados en un determinado intervalo y asocia los resultados obtenidos a una marca de tiempo. Por esto es importante que la hora en los equipos monitoreados y en el servidor de Pandora sean las mismas para de esta forma tener un control preciso de las alertas y el momento exacto en que fueron generadas. Como primer paso se debe configurar la zona horaria en el servidor sobre el que está corriendo nuestro servicio de Pandora FMS, para lo quese ejecuta el siguiente comando que lanza el Asistente de Configuración.

```
sudo dpkg-reconfigure tzdata
```

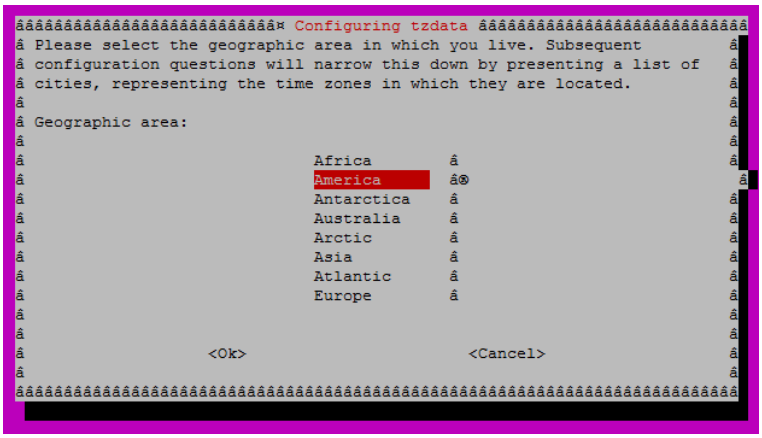


Figura 3.20 (a)
Configuración de la Zona Horaria del servidor Linux (a)

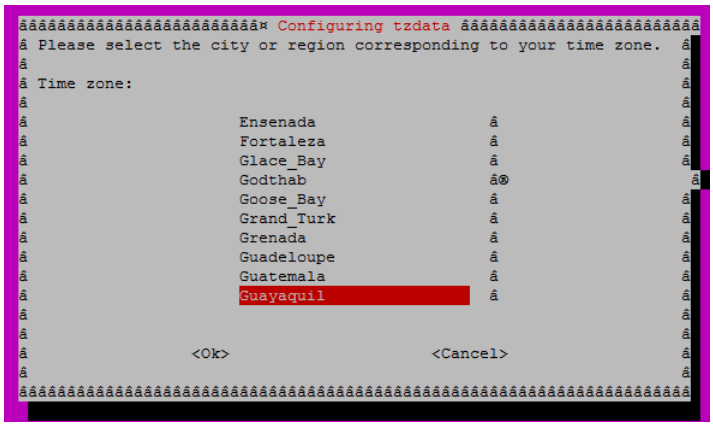


Figura 3.20 (b)
Configuración de la Zona Horaria del servidor Linux (b)

En la Figura 3.20(a) y Figura 3.20(b) se realiza el proceso de configuración de la Zona Horaria que para este trabajo es America\Guayaquil. Además se debe realizar un cambio similar en la consola de Administración Web de Pandora FMS en la opción de *Setup* del Menú de Administración. En la Figura 3.21 se indica el lugar en el que se debe realizar la configuración.

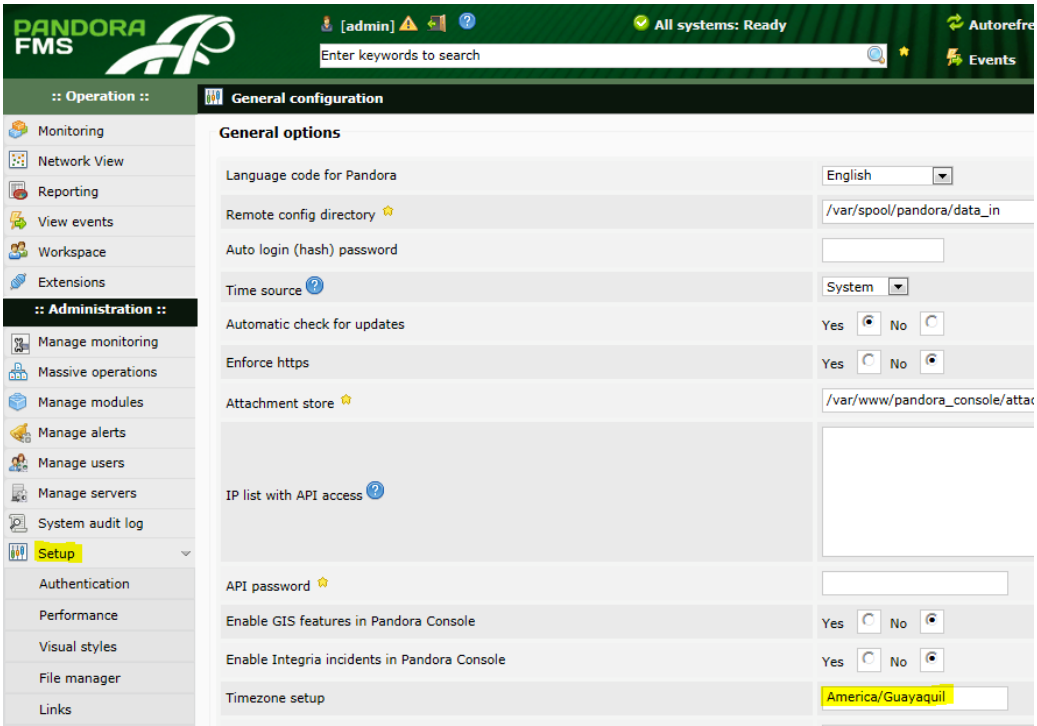


Figura 3.21
Cambio de la zona horaria en la consola de Pandora FMS

Finalmente se realiza la sincronización de la hora con un servidor NTP[27] externo, que para el este trabajo se ha escogido el NTP oficial de Ecuador que es inocar.ntp.ec[28]

```
sudo ntpdate inocar.ntp.ec
```

3.2.5 Configuración de Agentes de Monitoreo en Servidores Windows

Para la instalación del Agente en un servidor Windows se descargó el paquete de instalación desde la página web del proyecto y se siguió el Asistente de Instalación del agente. Para el presente trabajo se usó la versión 4.0.2 del agente sobre un servidor con Windows 2003 R2 - SP2.

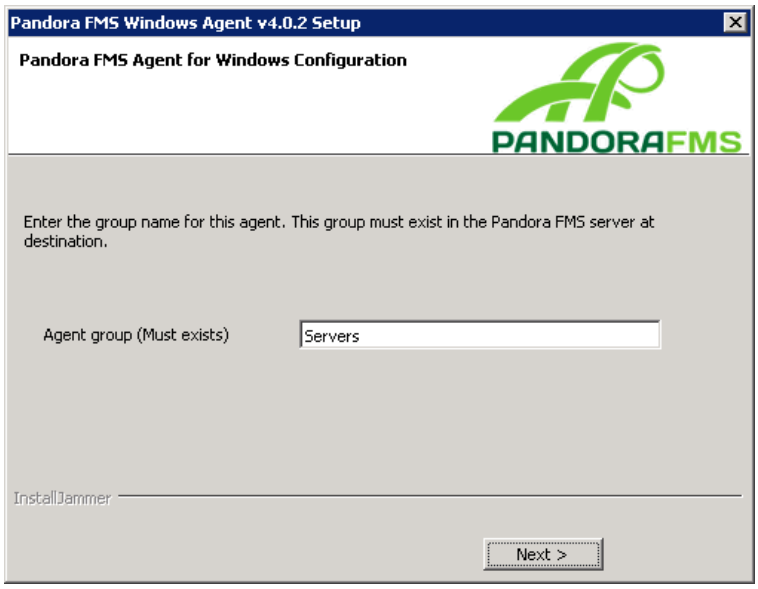


Figura 3.22
Configuración del Agente de Pandora FMS en Windows – 1

En la Figura 3.22 se debe indicar el grupo del que el servidor va a formar parte, este grupo debe estar previamente creado en el servidor de Pandora FMS. En la Figura 3.23 se ingresa la dirección o nombre del servidor de Pandora FMS, que se ha definido con la IP 10.0.90.11.

---

[27] Network Time Protocol (NTP) es un protocolo de Internet para sincronizar los relojes de los sistemas informáticos. Fuente: http://es.wikipedia.org/wiki/Network_Time_Protocol
[28]Fuente: http://inocar.ntp.ec/sincronizacion.html

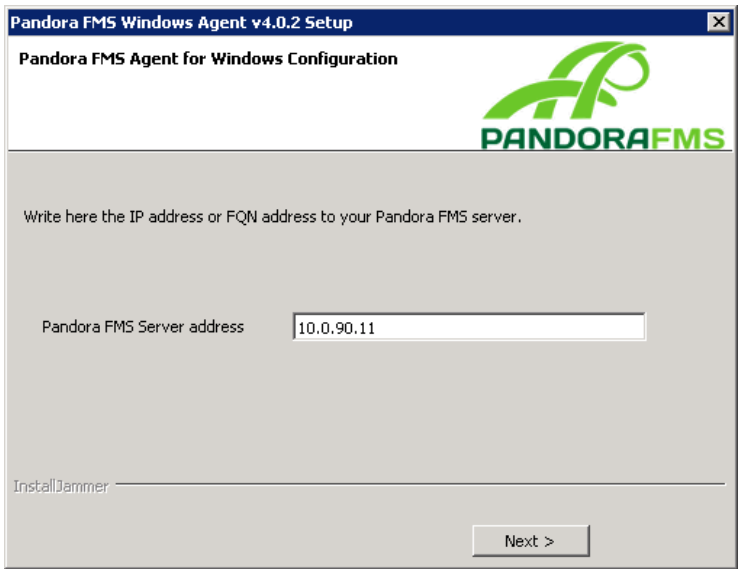


Figura 3.23
Configuración del Agente de Pandora FMS en Windows – 2

Para finalizar el proceso de configuración del agente es necesario modificar el archivo de configuración de Pandora FMS que se ubica en *C:\Program Files\pandora_agent\pandora_agent.conf*, en el cual se coloca la IP del servidor de Pandora y se asigna un nombre al servidor monitoreado como se lo hace en la Figura 3.24, posterior a esto se guarda el archivo y se reinician los servicios mediante la línea de comandos indicada en la Figura 3.25.

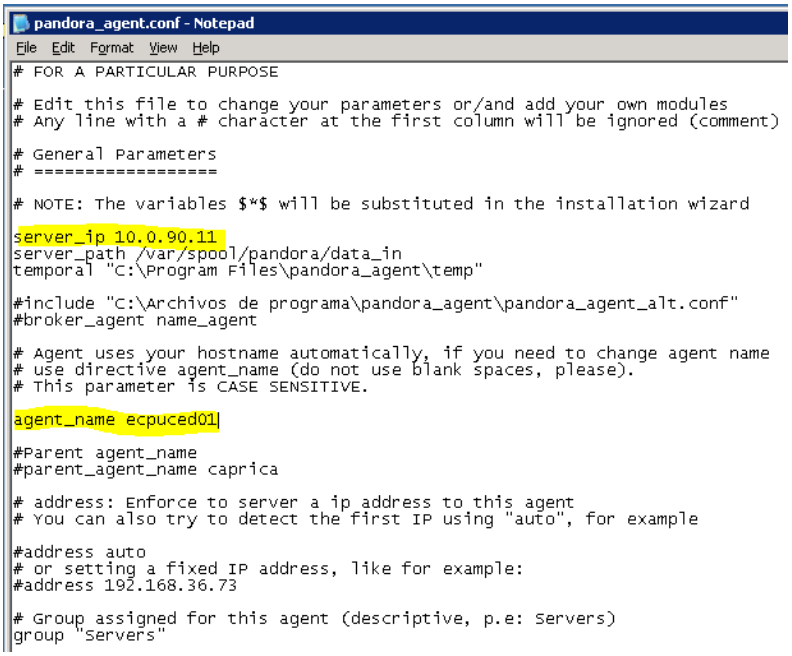


Figura 3.24
Actualización de configuración del agente instalado en Sistema Operativo Windows

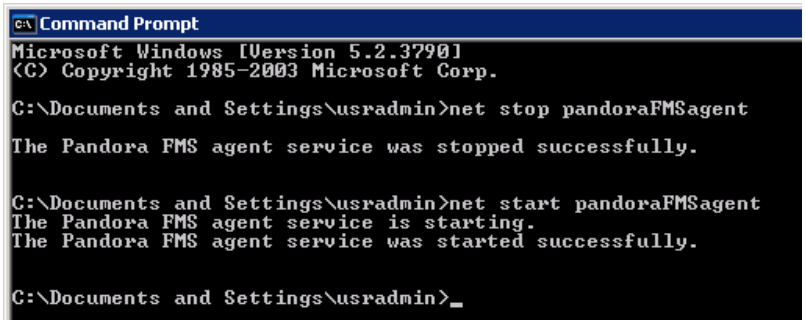


Figura 3.25
Reinicio de los servicios del Agente de Pandora FMS en el equipo Windows

3.2.6 Configuración de Agentes de Monitoreo en Servidores Linux/Unix

Para la instalación del Agente en un servidor Linux/Unix se debe contar con el paquete de instalación correspondiente a la versión del Sistema Operativo, que en este caso es un Servidor Ubuntu 12.04. Como primer su subirá al servidor el instalador del agente de Pandora FMS, seguidamente se debe conectar al servidor con un cliente SSH para realizar la instalación del agente de Pandora FMS, para esto se ejecuta el comando de la Figura 3.26 que inicia la instalación del agente en el servidor Linux

```
sudo dpkg -i pandora.agente_unix_4.0.2.deb
```



Figura 3.26
Instalación del Agente Pandora FMS en un servidor con Ubuntu 12.04

En caso de presentarse algún error en la instalación, se puede corregir el proceso ejecutándose el siguiente comando que permite corregir las dependencias que necesite el instalador del agente de Pandora FMS. Este comando se lo puede ejecutar solamente después de haber ejecutado el comando indicado arriba.

```
sudo apt-get -f install
```

Seguidamente se debe modificar el archivo de configuración de Pandora FMS que está en la ruta */etc/pandora/pandora_agent.conf* y realizar los cambios indicados en la Figura 3.27, esto permitirá que el agente envíe información al servidor de monitoreo de Pandora FMS y que sea reconocido con un nombre único dentro de la herramienta de monitoreo.

```
.
.
.
# General Parameters
# ==================
server_ip    10.0.90.11
.
.
.
# By default, agent takes machine name
agent_name     ecpucelx04
.
.
.
```

Figura 3.27
Cambios en el archivo de configuración del agente de Pandora FMS

Finalmente es necesario reiniciar los servicios del agente de Pandora FMS en el servidor Linux/Unix, para lo cual se ejecuta lo siguiente:

```
sudo /etc/init.d/pandora_agent_daemon restart
```

Con esto el servidor Linux/Unix empezará a reportar en la consola de monitoreo de Pandora FMS, como se observa en la Figura 3.28. Con esta configuración se recolecta información del *performance* del equipo según la configuración por defecto, que sin embargo puede ser expandida según los cambios que se hagan al archivo de configuración del agente.

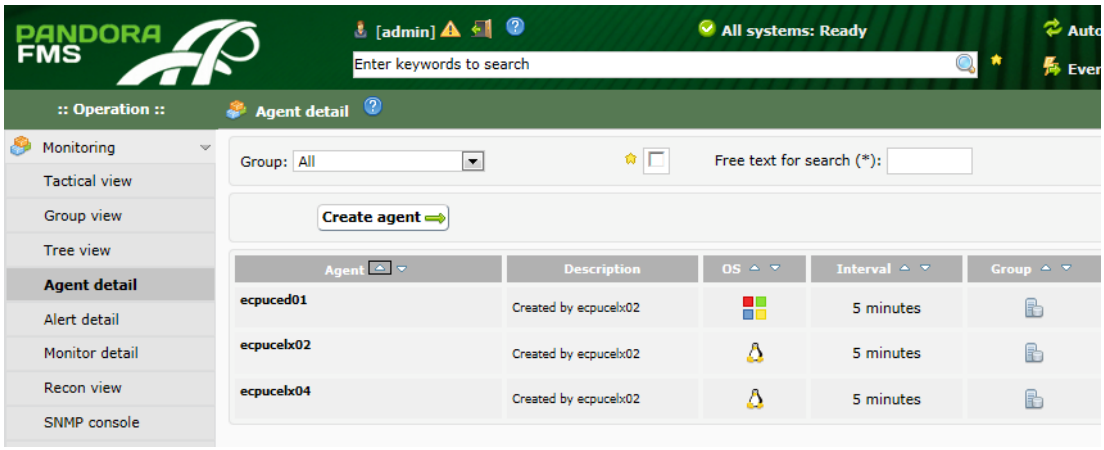


Figura 3.28
Consola de Administración y Monitoreo de Pandora FMS

3.2.7 Configuración del agente Windows mediante consultas WMI

Una de las principales características que incorpora Pandora FMS es la posibilidad de realizar consultas de información a servidores Windows sin la necesidad de instalar un agente para la recolección de información del equipo. Esto resulta particularmente útil en sistemas antiguos o de criticidad muy alta en los que no se pueda instalar ningún software adicional. En el proceso de configuración del servidor realizado con anterioridad en este

capítulo, y específicamente en la Figura 3.19, se dejó activando el servicio WMI, el cual permite realizar un monitoreo sin agente, para esto se necesita de una cuenta autorizada en el servidor de Directorio Activo que realice dichas consultas.

Por defecto la herramienta de monitoreo implementa ciertos módulos de monitoreo de *CPU load*, *Free RAM* y *Windows version* mediante un grupo de políticas conocidas como *Basic WMI monitoring*, como se observa en la Figura 3.29.

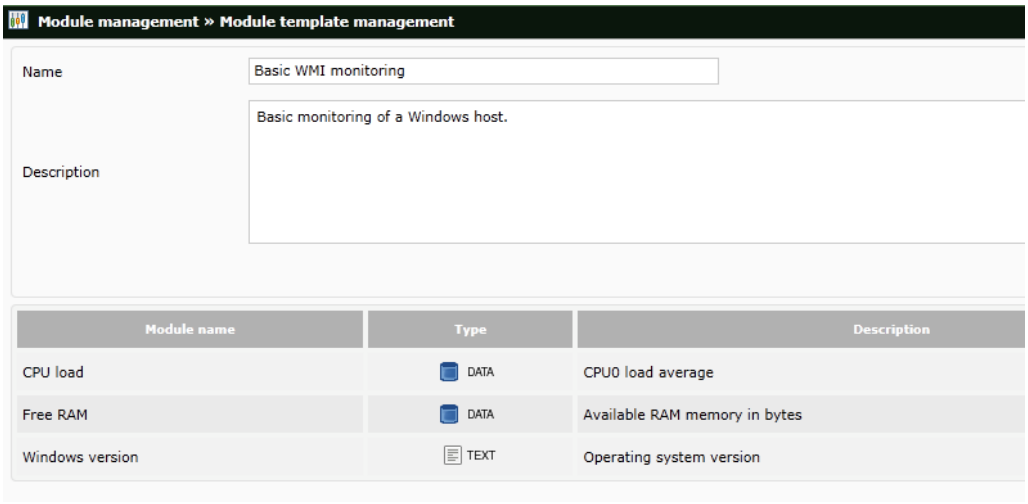


Figura 3.29
Asignación de políticas WMI

En la Figura 3.30 se puede observar el módulo de monitoreo WMI que se ejecuta para determinar la carga del CPU, para esto ejecuta la siguiente sentencia al servidor Windows monitoreado:

```
SELECT LoadPercentage from Win32_Processor WHERE DeviceID = "CPU0"
```

Además se deben llenarlos campos de *Username* y *Password* que corresponden a la cuenta autorizada para acceder a información del servidor a través de consultas WMI, que corresponde a un usuario dentro del Directorio Activo con perfil de Administrador.

De esta forma se puede realizar un monitoreo sin agente, sobre la base de información que nos proporciona WMI, mas adelante en este capítulo se indicará la forma de navegar sobre la base de información y escoger los parámetros que más se ajusten con las necesidades de monitoreo.

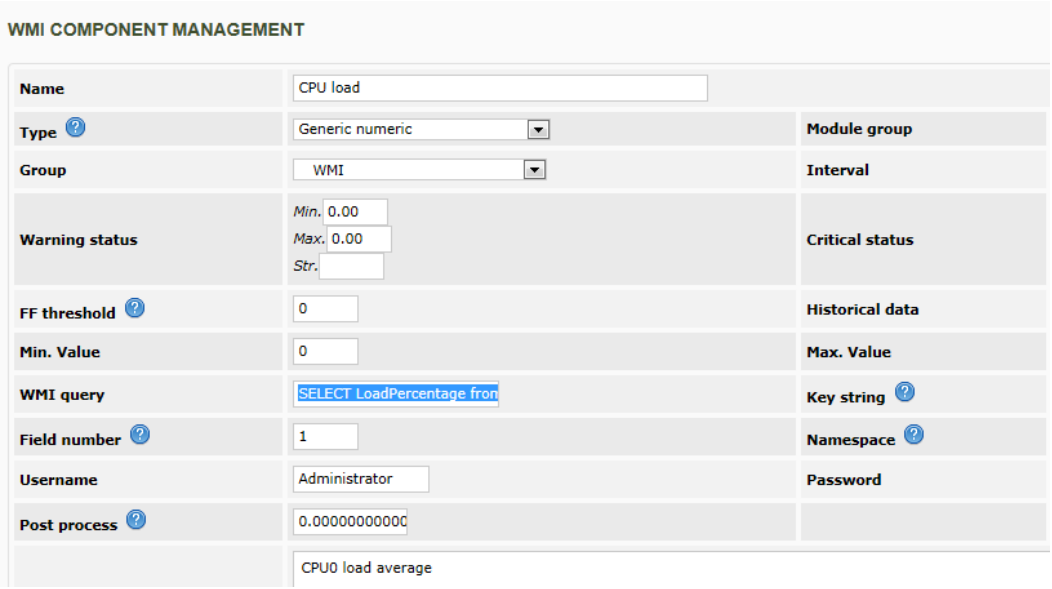


Figura 3.30
Configuración de módulo WMI

Como paso de verificación es necesario realizar una consulta WMI desde el servidor de Pandora FMS, para lo cual desde una consola en el servidor se ejecuta el siguiente comando:

```
wmic -U userwmi@puce.edu%Catolica1 //10.0.90.2 "SELECT LoadPercentage from
Win32_Processor WHERE DeviceID = 'CPU0'"
```

Como respuesta al comando se obtiene lo indicado en la Figura 3.31, que confirma que el servicio WMI está trabajando de manera correcta.

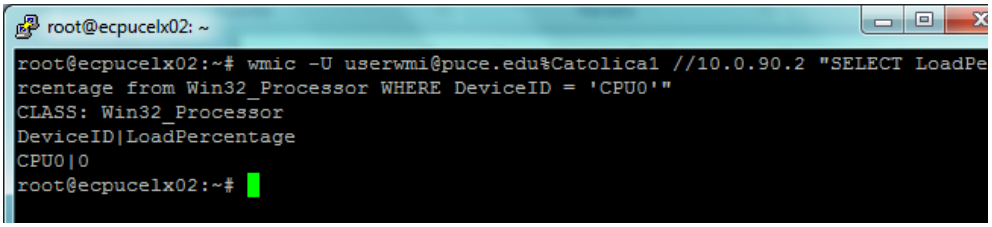


Figura 3.31
Prueba de conexión WMI

3.2.8 Configuración para recepción de TRAPS SNMP

Una de las principales características de la herramienta de monitoreo es la recepción de *traps* SNMP que se generan en los dispositivos de telecomunicaciones y servidores con clientes SNMP. Este monitoreo se lo realiza de forma asincrónica permitiendo que la herramienta reciba la alerta en el momento que se produzca el evento. La consola de recepción de *traps* SNMP es como la que se muestra en la Figura 3.32.

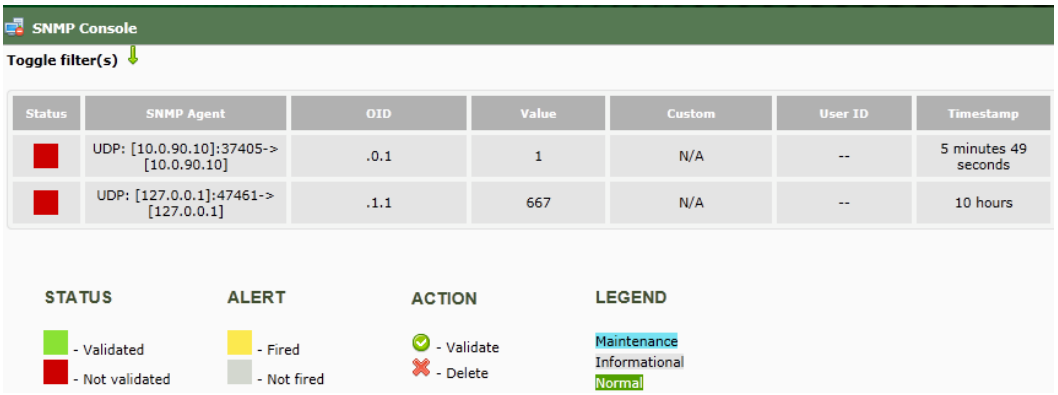


Figura 3.32
Consola de Administración y Monitoreo de Pandora FMS

Para contar con esta funcionalidad es necesario tener instalado el paquete NET-SNMP en el servidor de Pandora FMS y modificar los siguientes archivos de configuración para la recepción de *traps*, que se indican en las Figuras 3.33, 3.34 y 3.35.

```
/etc/snmp/snmpd.conf
```

```
#Informacion de contacto para consultas SNMP
#este archivo de configuracion contiene una configuración basica
syslocation Universidad Catolica del Ecuador
syscontact  Santiago Silva "sdsilva@outlook.com"

#se definen las IP's por las que se van a escuchar los traps SNMP
rocommunity public 127.0.0.1
rocommunity public 10.0.90.10

master       agentx
```

Figura 3.33
Configuración del archivo *snmp.conf*

```
/etc/snmp/snmptrapd.conf
```

```
##############################################################################
#
# EXAMPLE-trap.conf:
# An example configuration file for configuring the Net-SNMP snmptrapd
# agent.
#
##############################################################################
#
# This file is intended to only be an example.  If, however, you want
# to use it, it should be placed in /etc/snmp/snmptrapd.conf.
# When the snmptrapd agent starts up, this is where it will look for
# it.
#
# All lines beginning with a '#' are comments and are intended for you
# to read.  All other lines are configuration commands for the agent.
#
# PLEASE: read the snmptrapd.conf(5) manual page as well!
#
authCommunity log public
```

Figura 3.34
Configuración del archive *snmptrapd.conf*

```
/etc/default/snmpd
```

```
# This file controls the activity of snmpd and snmptrapd

# Don't load any MIBs by default.
# You might comment this lines once you have the MIBs downloaded.
export MIBS=

# snmpd control (yes means start daemon).
SNMPDRUN=yes

# snmpd options (use syslog, close stdin/out/err).
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux -p /var/run/snmpd.pid'

# snmptrapd control (yes means start daemon).  As of net-snmp version
# 5.0, master agentx support must be enabled in snmpd before snmptrapd
# can be run.  See snmpd.conf(5) for how to do this.
TRAPDRUN=yes

# snmptrapd options (use syslog).
TRAPDOPTS='-Lsd -p /var/run/snmptrapd.pid'

# create symlink on Debian legacy location to official RFC path
SNMPDCOMPAT=yes
```

Figura 3.35
Configuración del archivo *snmpd*

Seguidamente se debe realizar el reinicio del agente SNMP y verificar que se encuentra levantado correctamente.

```
/etc/init.d/snmpd restart
/etc/init.d/snmpd status
```

Es necesario además reiniciar el equipo para que el servidor de Pandora FMS levante el servicio SNMP y se pueda empezar a recibir las alertas SNMP. En la Figura 3.36 se realiza la verificación los servicios SNMP y que estos se encuentren activos.

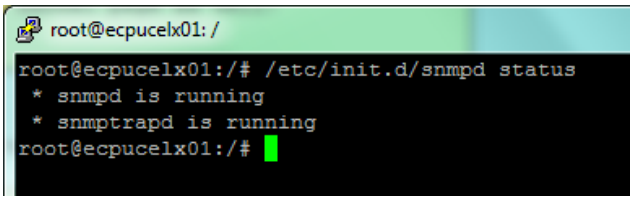


Figura 3.36
Confirmación de reinicio de los servicios de SNMP

Posterior al reinicio del equipo se debe verificar que el servicio SNMP ha sido levantado por Pandora FMS, para esto se debe ejecutar el siguiente comando:

```
psaux | grep snmptrapd
```

La salida de este comando debe tener similitud con lo indicado en la Figura 3.37, en donde se observa que el servicio *snmptrapd* hace referencia a rutas de Pandora FMS. Desde estas rutas la herramienta accederá a la información de los *traps* SNMP.

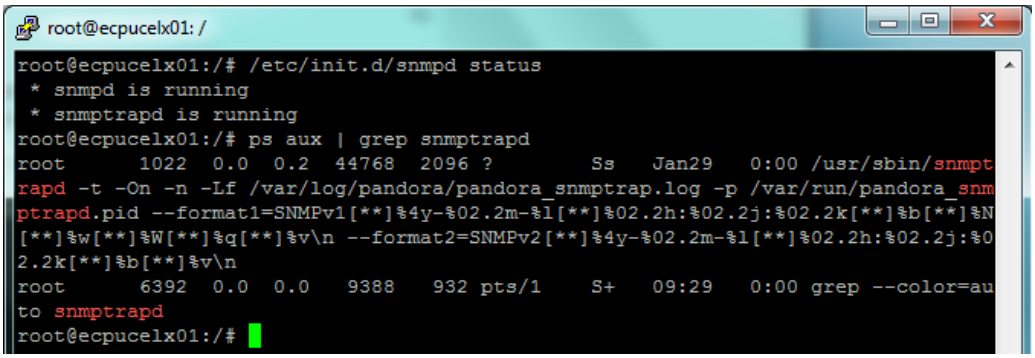


Figura 3.37
Confirmación de servicios de SNMP levantados por Pandora FMS

3.2.9 Configuración de envío de mensajes SMS

3.2.9.1 Configuración del modem para trabajar con puertos seriales

Para el envío de notificaciones SMS a través de la herramienta de monitoreo se utiliza el modem ZTE MF 100 que tiene la opción de conectarse de forma serial y enviar mensajes SMS. Para esto es necesario deshabilitar una característica específica de los módems USB que emulan una unidad de CD-ROM y que se ejecuta automáticamente para la instalación del software cliente del dispositivo. [8]

Desde una máquina Windows se conecta el modem y se deja que se instale automáticamente el dispositivo, posterior a esto desde el Administrador de Dispositivos como se indica en la Figura 3.38, se accede a la sección de *Modems* donde se despliega los dispositivos conectados a la máquina.

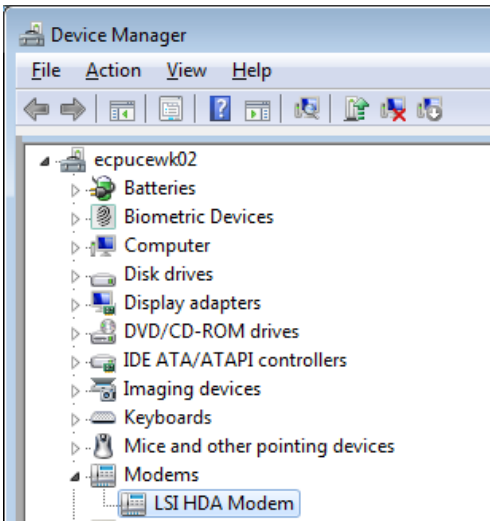


Figura 3.38
Administrador de dispositivos de Windows

Desde este lugar se debe acceder a las propiedades del dispositivo para verificar el puerto en el que está trabajando el modem y la velocidad de conexión, para esto se da *click* derecho sobre el objeto y en la opción Propiedades se visualizará una pantalla como la Figura 3.39 y se obtiene la información para realizar la conexión serial al dispositivo.

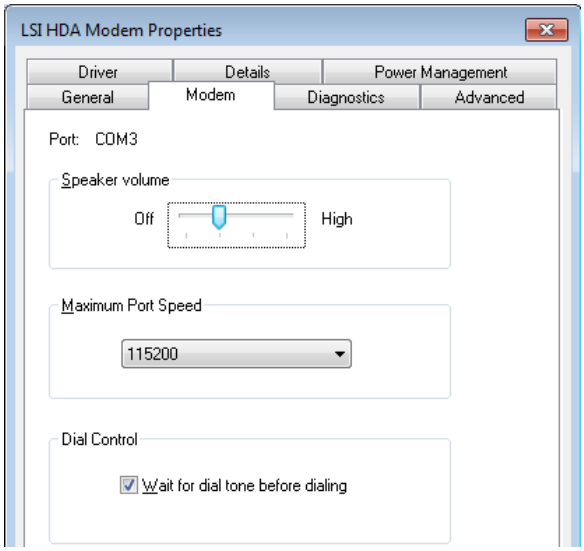


Figura 3.39
Detalle del puerto que usa el Modem GSM USB

Con la información obtenida de la Figura 3.39 se realiza la conexión serial mediante *HyperTerminal*[29] al Modem USB ingresando los valores presentados en la Figura 3.40.

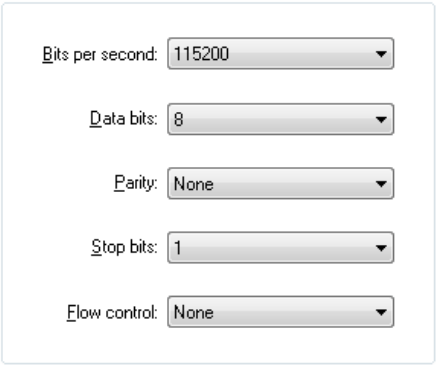


Figura 3.40
Propiedades el Modem GSM USB

Ahora es posible desactivar la ejecución automática de la unidad de CD-ROM que se ejecuta cada vez que el dispositivo se conecta a una PC. Para esto se ingresa a la consola del

[29] *HyperTerminal* es un programa presente en los Sistemas Windows hasta Windows XP que permite conexiones seriales a dispositivos que tengan consolas Terminal basadas en líneas de comando.

dispositivo mediante una consola de *Hyperterminal* y se deshabilita la opción de carga automática de la unidad de CD-ROM ingresando el siguiente comando:

```
AT+ZCDRUN=8
```

Y el modem debe responder lo siguiente:

```
Close autorun state result(0:FAIL 1:SUCCESS):1
```

En caso que se necesite habilitar nuevamente la opción de carga automática de la unidad de CD-ROM del modem, se debe ingresar el siguiente comando:

```
AT+ZCDRUN=9
```

Y el modem debe responder lo siguiente:

```
Open autorun state result(0:FAIL 1:SUCCESS):1
```

3.2.9.2 Configuración del modem en Ubuntu 12.10

Realizada la configuración del modem en el punto anterior, el siguiente paso es conectar el modem USB a la máquina Linux para la configuración del envío de alertas SMS. Para esto se conecta el modem a la máquina y se realiza la siguiente verificación desde la consola.

```
dmesg | grep tty
```

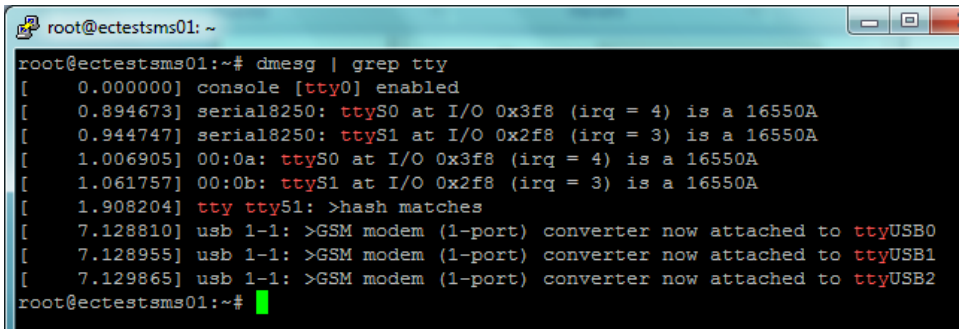


Figura 3.41
Detalle de los puertos seriales habilitados para GSM modem

En la Figura 3.41 se puede observar que el modem GSM ha tomado el uso de los puertos seriales ttyUSB0, ttyUSB1 y ttyUSB2. Con esta información el siguiente paso consiste en realizar el envío de un mensaje SMS a través del modem usando una consola serial para lo que se usará la herramienta *minicom*[9]. Para instalar esta herramienta se ejecuta el siguiente comando desde la consola:

```
apt-get install minicom
```

Para realizar la configuración del programa se ingresa el siguiente comando y se despliega un menú como en la Figura 3.42.

```
minicom -s
```

Se deben escoger las siguientes opciones para la configuración del puerto serial de forma que queden las opciones como se observa en la Figura 3.43.

```
A - Serial device: /dev/ttyS0
E - Bps/Par/Bits: 115200 8N1
F - Hardware Flowñ Control : No
```

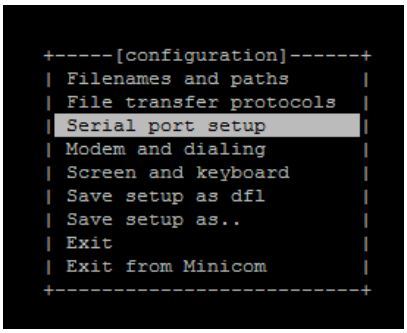


Figura 3.42
Opciones del software *minicom*

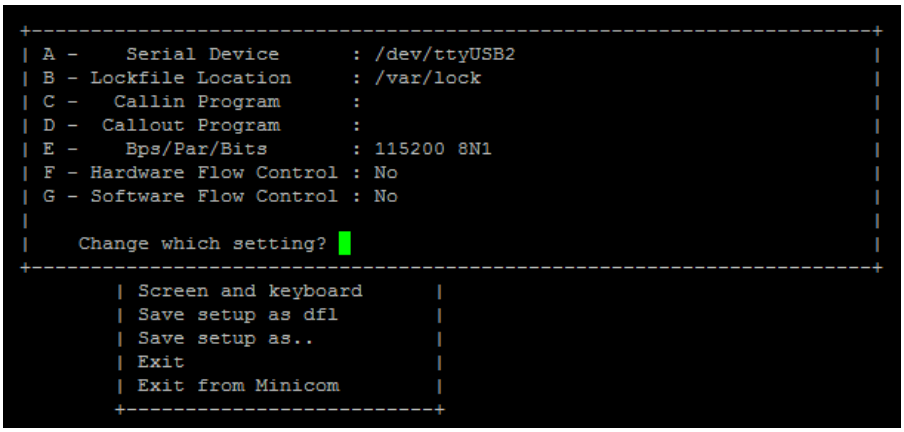


Figura 3.43
Configuración de *minicom* para envío de SMS

Dentro de la consola es posible realizar el envío del SMS ingresando los comandos marcados en negrilla como se observa en la Figura 3.44, después de los cuales se recibirá una respuesta de confirmación de parte del modem, al final de la secuencia de comandos se habrá realizado el envío del mensaje y este llegará al número indicado en la línea 5 de la figura. [10]

1. **AT**
2. OK
3. **AT+CMGF=1**
4. OK
5. **AT+CMGW="+593992388692"**
6. ***>"Se escribe el mensaje de prueba"***
7. *Se debe presionar CTRL + Z para terminar el mensaje*
8. +CMSS=3
9. *Se recibe la siguiente respuesta con el número del mensaje (en el ejemplo 3) que debe ser el mismo número para el siguiente comando.*
10. **AT+CMSS=3**
11. +CMSS: 16
12. OK

Figura 3.44
Envío de SMS a través de *minicom* para conexión serial

3.2.9.3 Configuración para el envío de SMS usando SMSTools3[11]

Para el envío de SMS se va a usar el programa SMSTools3 que se descarga desde la página web del proyecto[30], la versión utilizada será la smstools3-3.1.15.Previo a esto desde el servidor basado en Ubuntu se deben correr los siguientes comandos:

```
apt-get install make
apt-get install build-essential
```

Ejecutados estos comandos se puede descargar el paquete de instalación del paquete mediante el comando *wget* de la página web del proyecto:

```
wget http://smstools3.kekekasvi.com/packages/smstools3-3.1.15.tar.gz
```

En el siguiente paso se extrae el paquete en una ruta desde la que se pueda lanzar la instalación que para este caso puede ser: */usr/local/src*

```
tar -zxvf smstools3-3.1.14.tar.gz -C /usr/local/src
```

Posterior a la extracción o descompresión se realiza la instalación del mismo mediante los siguientes comandos:

```
cd /usr/local/src
cd smstools3
make
makeinstall
```

En el archivo */etc/smsd.conf* es necesario configurar el puerto serial y la velocidad a la que está conectado el Modem USB.

---

[30] Fuente: http://smstools3.kekekasvi.com

```
# Example smsd.conf. Read the manual for a description

devices = GSM1
logfile = /var/log/smsd.log
loglevel = 7

[GSM1]
device = /dev/ttyUSB2
incoming = yes
#pin = 1111
```

Figura 3.45
Archivo de configuración smsd.conf

Como siguiente paso se crean los archivos de trabajo de la herramienta *smstools3* mediante los siguientes comandos:

```
mkdir -p /var/spool/sms
cd /var/spool/sms
mkdir -p checked failed incoming outgoing sent
mkdir -p /var/log/sms
```

Se dan los permisos de escritura completos sobre los directorios creados de manera que la aplicación tenga acceso total para su correcto funcionamiento.

```
chmod -R 777 /var/spool/sms
```

Ahora es necesario iniciar el servicio desmstools3.

```
/etc/init.d/sms3 start
```

Si se necesita que el servicio de smstools3 arranque automáticamente después de cada reinicio del sistema operativo se debe ejecutar el siguiente comando:

```
update-rc.d sms3 defaults
```

Finalmente para verificar las configuraciones de envío de SMS, desde una consola de Linux se ejecuta el siguiente comando y se verifica en un dispositivo celular la recepción del mensaje de prueba.

```
sendsms +593992388692 "test"
```

3.2.10 Configuración de envío de notificaciones vía e-mail

Adicional al envío de mensajes de texto se realiza el envío de notificaciones vía correo electrónico, para lo que se ha implementado un servidor de correo interno *POP3* sobre Windows 2003. La configuración del servidor de correo no se explicará en este documento pues se asume que se cuenta con una infraestructura de envío de notificaciones vía correo electrónico. Dentro del servidor de la herramienta de monitoreo se instala el software

EXPECT que nos permite automatizar la tarea de envío de las notificaciones vía *telnet* en el servidor *SMTP* de correo electrónico.

```
apt-get install expect
```

Posterior a la instalación del software *EXPECT* se escribe el *script* que será el encargado de conectarse al servidor de *SMTP* y realizar el envío del mensaje de correo electrónico de alerta ingresando los valores de la cuenta de destino, el título del mensaje y el cuerpo del mensaje.

```
#!/usr/bin/expect -f
set arg1 [lindex $argv 0]
set arg2 [lindex $argv 1]
set arg3 [lindex $argv 2]
set timeout 1
spawn telnet ecpucem01.puce.edu 25
expect "220"
send "helo ecpucem01.puce.edu\r"
expect "250"
send "MAIL FROM: pandora@puce.edu\r"
expect "250"
send "RCPT TO: $arg1\r"
expect "250"
send "DATA\r"
expect "354"
send "Subject: $arg2\r"
send "$arg3 \r\r"
send ".\r"
expect "delivery"
send "quit"
quit
```

Figura 3.46[31]
Configuración con EXPECT para envío de notificaciones vía mail

Se debe cambiar los permisos sobre el archivo para permitir su ejecución mediante el siguiente comando:

```
chmod 700 /root/smtpmail
```

Ahora se debe realizar el envío de un mensaje de prueba ingresando los siguientes valores descritos en el comando y se podrá verificar que se recibe el mensaje en cualquier cliente de servicio de correo electrónico.

```
/root/smtpmail pandora@puce.edu "Alerta Pandora FMS" "Body Message"
```

[31]Fuente: Pandora FMS 4.0.2 Manual de Usuario página 354

3.3 Configuración del *Router* Cisco para acceder a la información SNMP

En la sección 3.2.8 se describió la configuración para la recepción de *traps* en la consola de monitoreo, sin embargo es necesario habilitar el envío de estas notificaciones desde los dispositivos monitoreados, que en este caso se utilizará un *Router* Cisco emulado sobre GNS3 de la serie 2691, la topología de la emulación es la que se indica en la Figura 3.47 y los datos de las interfaces se explican en la Tabla 3.4.

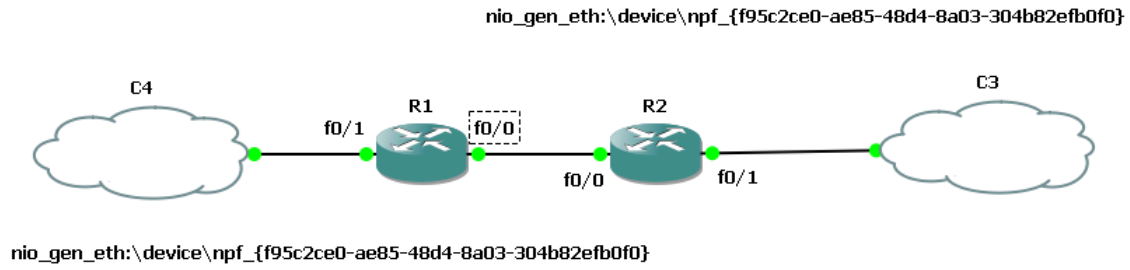


Figura 3.47
Topología de simulación en GNS3

Dispositivo	*Interface*	*Direccionamiento IP*
C4	*Tarjeta de red 1 del servidor de simulación	n/a
R1	F0/1	10.0.90.6
	F0/0	192.168.20.10
R2	F0/0	192.168.20.11
	F0/1	10.0.91.6
C3	*Tarjeta de red 2 del servidor de simulación	n/a

Tabla 3.4
Detalle de las interfaces de la simulación

Para acceder a la información mediante el protocolo SNMP y recibir las alertas o *traps* es necesario ingresar los siguientes comandos desde el modo privilegiado de configuración del *Router,* como se indica en la Figura 3.48.

```
snmp-server community public RO
snmp-server community restPUCE RW 60
access-list 60 permit 10.0.90.11
access-list 60 permit 10.0.90.53
snmp-server host 10.0.90.11 inform version 2c public
snmp-server host 10.0.90.11 public
snmp-server enable traps
snmp-server enable traps ospf cisco-specific state-change shamlink
```

Figura 3.48
Configuración de *traps* SNMP

La línea 1 habilita el acceso de solo lectura a la comunidad *Public* en el dispositivo mediante SNMP, la línea 2 habilita el acceso de escritura y lectura a la comunidad del tipo *Private* y restringe el acceso mediante una lista de control de acceso. En las líneas 3 y 4 se especifica

las direcciones IP de los dispositivos que tiene acceso a la comunidad *Private*. En las líneas 5 y 6 se indica cual es el servidor de envío de notificaciones SNMP, la comunidad y la versión del protocolo utilizado. Finalmente en las líneas 7 y 8 se habilita el envío de las notificaciones SNMP en forma de *traps*.[12][13]

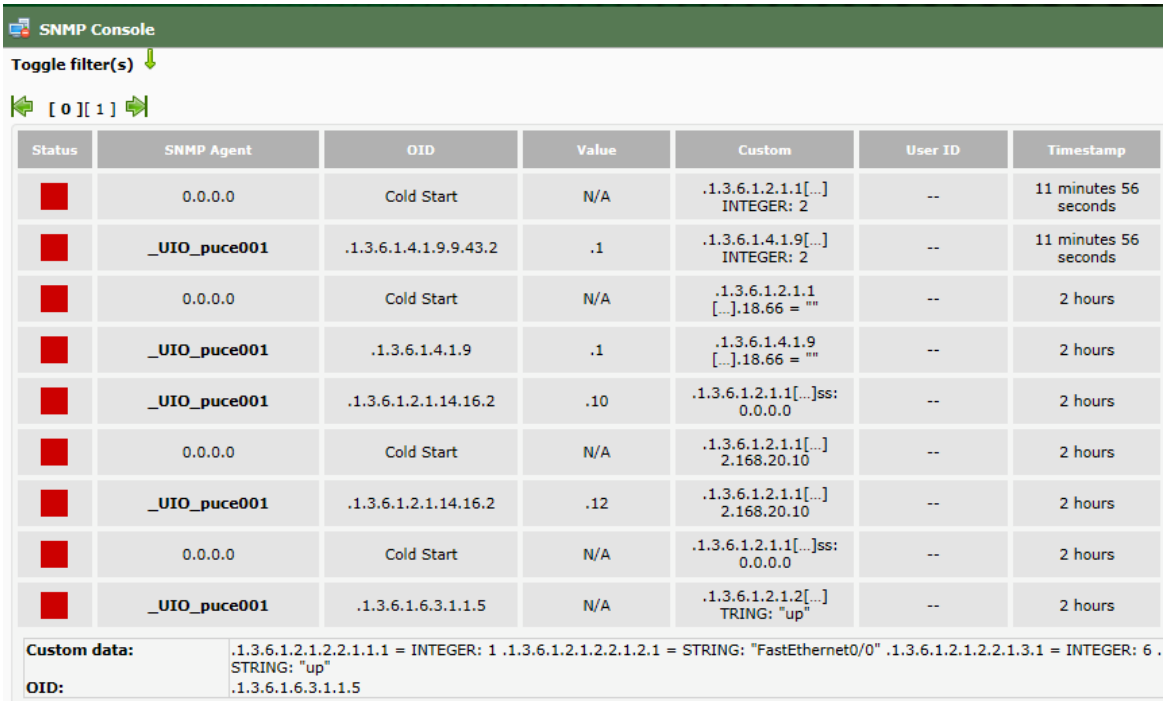


Figura 3.49
Consola de recepción de *traps* SNMP

Como paso final se puede abrir la consola de recepción de *traps* SNMP en la herramienta de monitoreo y verificar que se reciben eventos como en la Figura 3.49 ante un evento en el que se subió y bajó la interfaz "*FastEthernet0/0*".

3.4 Guía de Implementación de la herramienta Pandora FMS

Como parte de este trabajo se presenta una guía reducida para la implementación de la herramienta de monitoreo Pandora FMS en base a lo desarrollado en el Capítulo 3, para encontrar más detalles acerca de algún punto se puede revisar minuciosamente este capítulo.

Paso 1: Configuración del servidor Ubuntu 12.04 para Pandora FMS

Se debe configurar el acceso remoto al servidor Linux/Unix y habilitar el usuario *root* mediante los siguientes comandos:

```
sudo apt-get update
sudo apt-get install openssh-server openssh-client
sudo passwd root
```

Paso 2: Instalación de los componentes del servidor de Pandora FMS

Se deben instalar los paquetes de pre-requisistos para la distribución Ubuntu/Debian según lo indicado en el siguiente comando:

```
sudo apt-get install #nombre del paquete#
Ejemplo: sudo apt-get install snmp
```

Los paquetes que se instalan son los siguientes:

```
snmp snmpd libtime-format-perl libxml-simple-perl libxml-twig-perl libdbi-perl
libnetaddr-ip-perl libhtml-parser-perl  xprobe nmap libmail-sendmail-perl
traceroute libio-socket-inet6-perl libhtml-tree-perl libsnmp-perl snmp-mibs-
downloader libio-socket-multicast-perl
```

```
php5 libapache2-mod-php5 apache2 mysql-server php5-gd php5-mysql php-pear
php5-snmp php-db php-gettext graphviz mysql-client php5-curl php5-xmlrpc php5-
ldap dbconfig-common
```

Durante la instalación de *MySQL* se lanzará un Asistente de Instalación o *Wizard*en el que se pedirá la clave de acceso *root* al motor de la Base de Datos. Seguidamente se deben instalar los siguientes paquetes:

```
libnet-traceroute-perl_1.10-1_all.deb
libnet-traceroute-pureperl-perl_0.10-1_all.deb
php-xml-rpc_1.5.2-1_all.deb
libwmiclient1_1.3.14-3_amd64.deb
```

Para instalar los paquetes se ejecuta el siguiente comando con el nombre de cada uno de ellos desde la consola SSH del servidor de Pandora FMS.

```
sudo dpkg -i #nombre del paquete#
```

Durante el proceso de instalación se pueden presentar errores con algún paquete, por lo que será necesario ejecutar el comando indicado para corregir cualquier dependencia o pre-requisito que necesite el Sistema Operativo.

```
apt-get -f install
```

Para finalizar el proceso de instalación del software de Pandora FMS se realizará la instalación de los siguientes paquetes en el orden indicado mediante el comando DPKG.

```
pandorafms.console_4.0.2.deb
pandorafms.server_4.0.2.deb
pandorafms.agent_unix_4.0.2.deb
```

Con esto se finaliza el proceso de instalación del software y se puede continuar con la configuración de la herramienta mediante un Asistente que se indica en el Paso 3.

Paso 3: Configuración del servidor de Pandora FMS

Para empezar con la configuración de Pandora FMS se debe ingresar en la siguiente dirección desde un navegador web y seguir el asistente de configuración.

```
http://ip_servidor_pandora/pandora_console
```

A continuación se lanzará el *Wizard* de configuración del servidor de Pandora FMS en donde se debe copiar la clave que genera el asistente para la Base de Datos y que posteriormente se coloca en el archivo de configuración del servidor de Pandora FMS. Al finalizar el asistente se ingresa nuevamente en la URL del servidor de Pandora FMS *http://ip_servidor_pandora/pandora_console* que muestra que el servidor está activo y que se debe borrar el archivo de Instalación de Pandora FMS. Este archivo se encuentra en la ruta */var/www/pandora_console/install.php*.

Antes de subir los servicios de Pandora FMS se debe realizar ciertas modificaciones en el archivo de configuración para que los servicios se levanten correctamente, este archivo se encuentra en la siguiente ruta */etc/pandora/pandora_server.conf*.

```
# Servername: Name of this server
# if not given, it takes hostname. It's preferable to setup one
# because machine name could change by some reason.
servername ecpucelx02
.
.
.
# dbpass: Database password
dbpassjehirglr
.
.
.
# wmiserver : 1 or 0. Set to 1 to activate WMI server with this setup
# DISABLED BY DEFAULT
wmiserver 1
```

En este archivo de configuración se debe colocar la clave de acceso *root* a la Base de Datos *MySQL* que se creó durante el *Wizard* de configuración. Además se debe habilitar el servicio de consultas WMI desde el servidor de Pandora FMS que por defecto viene desactivado. El resto de configuraciones deberán permanecer intactas sin realizar cambios en las mismas.

Seguidamente se levantan los servicios de Pandora FMS por primera vez, para ello se ejecuta los siguientes comandos:

```
sudo /etc/init.d/pandora_server start
sudo /etc/init.d/tentacle_serverd start
sudo /etc/init.d/pandora_agent_daemon start
```

Finalizado estos comandos el servidor de Pandora FMS está listo para recolectar la información de los agentes monitoreados.

Paso 4: Configuración de la hora del Servidor de Pandora FMS

En este paso se debe realizar la sincronización de la hora del servidor de Pandora FMS para que los mensajes recibidos de los equipos monitoreados tengan la misma marca de tiempo. Para esto se deberá configurar la Zona Horaria en la consola de Pandora FMS y sincronizar la hora con un servidor NTP externo.

CAPÍTULO 4

4. ANÁLISIS DE LOS RESULTADOS

En este capítulo se analizan los resultados de implementar la herramienta de monitoreo Pandora FMS en el ambiente de pruebas. Para esto se han ejecutado modificaciones en los agentes de monitoreo y realizado pruebas para determinar su comportamiento cuando se presentan consumos elevados de recursos, en el caso del agente SNMP en el *router* Cisco 2691 se analizará la recepción de *traps y* la consulta de información en el equipo. Además se verificará el sistema de generación de alertas usado por Pandora FMS v mediante correos electrónicos y mensajes SMS a celulares.

4.1 Monitoreo de agentes en sistemas operativos Windows

Para el monitoreo de servidores Windows se debe instalar el agente según lo indicado en la sección 3.2.5, posterior a esto se recolectará información de los módulos desplegados por defecto por el agente de monitoreo, adicional a estas políticas es posible incrementar módulos de monitoreo que toman información mediante consultas WMI, consultas al Indicador de Rendimiento de Windows y lectura del log de eventos. Aparte de estos métodos se pueden realizar consultas SNMP instalando un agente en los servidores Windows, conexión OBDC para Bases de Datos, etc.; sin embargo este apartado se enfocará en la configuración y recolección de información mediante los tres primeros métodos descritos y se explicará su utilidad y eficacia para la recolección de información.

4.1.1 Monitoreo mediante consultas WMI

El agente de Pandora FMS instalado realiza consultas WMI sin la necesidad de utilizar herramientas externa y requiere de la invocación de dos parámetros para obtener la información de monitoreo: *module_wmiquery* y *module_wmicolumn*.

```
module_begin
module_name Services
module_type generic_data_string
module_wmiquery Select Name from Win32_Service
module_wmicolumn Name
module_end
```

Figura 4.1
Creación de Módulo WMI

En la Figura 4.1 se ha creado un módulo de monitoreo de nombre *Services* que consulta información en la tabla de *Win32_Service*. El sistema operativo Windows incorpora una gran cantidad de información que puede ser consultada y accedida, sin embargo para esto es necesario realizar una investigación y consulta de la información. Para el efecto se puede

utilizar el software *WMI Explorer*[32] que permite realizar consultas WMI a equipos Windows.

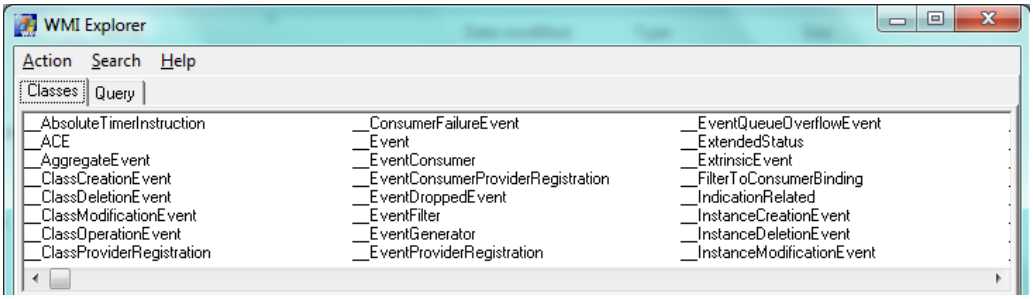


Figura 4.2
Visualización de información WMI a través de WMI Explorer

En la Figura 4.2 se tiene la interfaz del explorador WMI que nos permite navegar por las diferentes clases del repositorio de información que será consultada por nuestro módulo de monitoreo. En este caso se realizará una consulta WMI para conocer el año configurado en el sistema operativo.

```
select Year from Win32_LocalTime
```

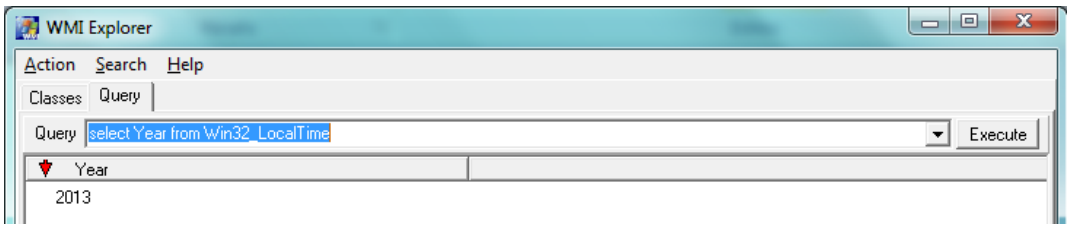


Figura 4.3
Consulta WMI desde WMI Explorer

En la Figura 4.3 se observa que la consulta WMI realizada ha lanzado como resultado el dato del año en curso. A partir de este momento es posible crear un módulo de monitoreo en el agente de Pandora FMS instalado en el equipo Windows. Para esto se necesita modificar el archivo de configuración que se explicó en la sección 3.2.5 en la Figura 3.24 y añadir las siguientes líneas y reiniciar el agente.

```
module_begin
module_name Año_SO
module_type generic_data_string
module_wmiquery select Year from Win32_LocalTime
module_wmicolumn Year
module_end
```

Figura 4.4
Módulo de monitoreo

[32]Fuente: http://www.ks-soft.net/hostmon.eng/wmi/index.htm

En la Figura 4.4. en la línea 3 del código presentado se ha declarado al dato monitoreado del tipo *generic_data_string* que se usa para datos del tipo texto, adicionalmente para datos del tipo numérico se debe usar *generic_data*. Mediante la creación de este tipo de módulos el agente ejecuta consultas que son llevadas a la consola central de monitoreo e informan sobre el estado de nuestro sistema, entre algunas de las clases que contienen información relevante que puede ser consultada y monitoreada tenemos:

- Información del equipo - *Win32_ComputerSystem*
- Información del Sistema Operativo - *Win32_OperatingSystem*
- Información del procesador - *Win32_Processor*
- Información de los discos físicos - *Win32_DiskDrive*
- Información sobre las particiones - *Win32_DiskPartition*
- Información sobre las particiones lógicas - *Win32_LogicalDisk*
- Información de la memoria - *Win32_PhysicalMemory, Win32_PhysicalMemoryArray*
- Información de Red - *Win32_NetworkAdapter, Win32_NetworkAdapterConfiguration*

4.1.2 Monitoreo mediante consultas al Indicador de Rendimiento

Los sistemas operativos Windows incorporan una herramienta de monitoreo de rendimiento o *performance* desde la cual es posible extraer información de componentes como el procesador, memoria, disco, interfaces de red, etc.

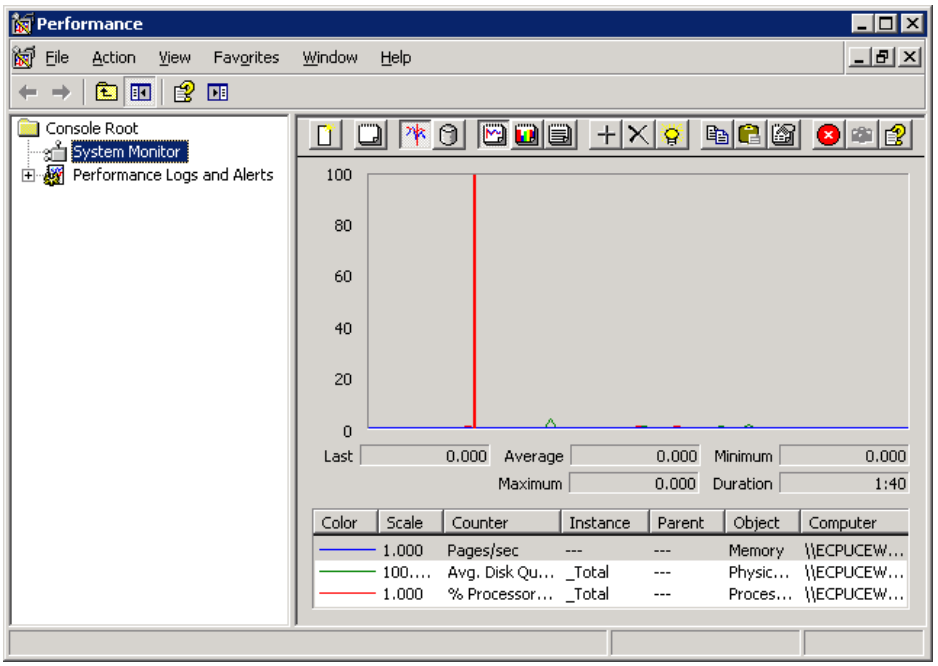


Figura 4.5
Performance Monitor de Windows

Para la generación de módulos de monitoreo es necesario tomar en cuenta que la obtención de información es dependiente del idioma instalado, es decir que si nuestro sistema operativo está en idioma inglés, el módulo creado no servirá para sistemas operativos en un idioma diferente. A continuación se presenta el ejemplo de un módulo de monitoreo que extrae información del contador de *performance* de Windows.

```
module_begin
module_name Memory
module_type generic_data
module perfcounter \Memory\% Committed Bytes In Use
module_end
```

Figura 4.6
Módulo de monitoreo mediante *Performance Monitor*

En la Figura 4.6 se tiene un módulo de monitoreo que extrae información del porcentaje de uso de memoria física que tiene el sistema operativo, para esto se debe hacer referencia al contador que se indica en la Figura 4.7 y que es parte de los Sistemas Operativos Windows 2000 SP4 en adelante.

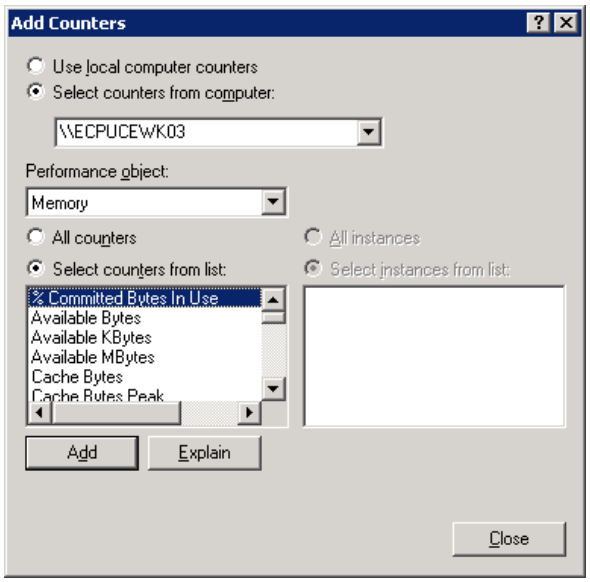


Figura 4.7
Selección del contador de *Performance Monitor*

Este método de monitoreo presenta algunas ventajas en comparación con el método WMI porque está orientado a obtener datos del rendimiento del Sistema Operativo y además ofrece una breve explicación de los contadores, como indica la Figura 4.8, lo que facilita la tarea de crear contadores que sean de utilidad para el diagnóstico y análisis del sistema operativo.

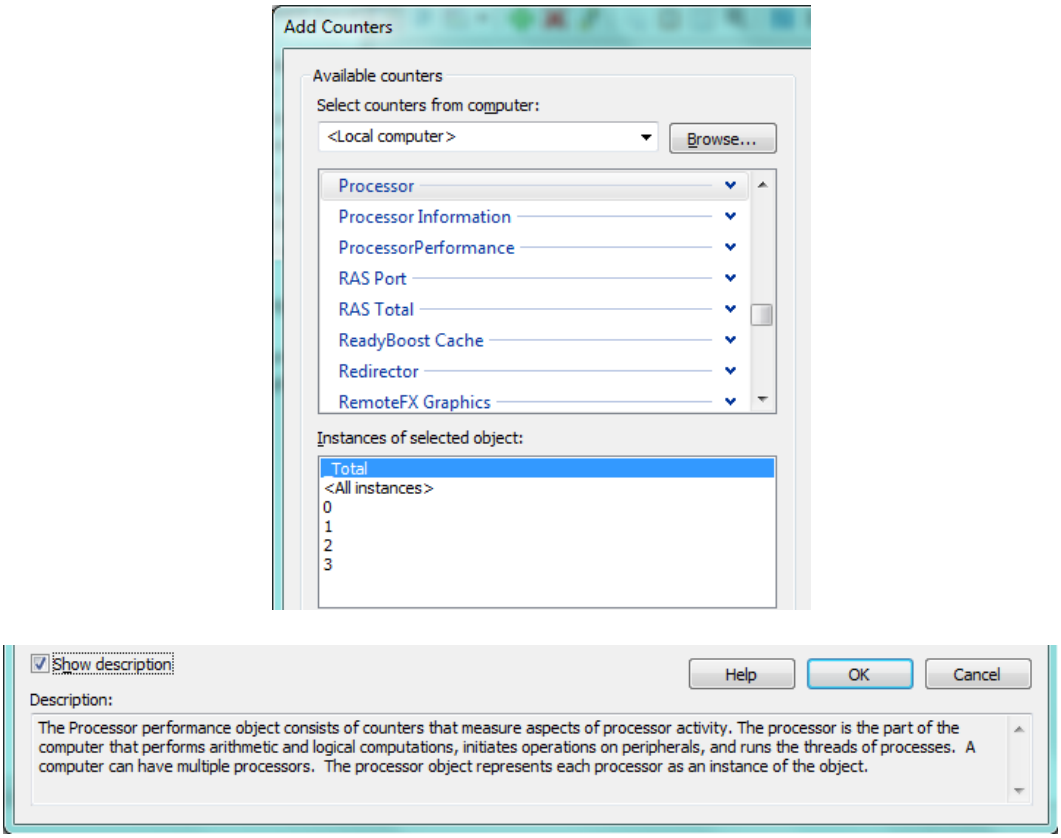


Figura 4.8
Información de los contadores de *Performance* de Windows

4.1.3 Monitoreo mediante lectura del Log de Eventos

El monitoreo del Log de Eventos en sistemas operativos Windows resulta de mucha utilidad porque existen registros que pueden alertar sobre posibles fallas de disco duro, intentos inválidos de autenticación, problemas en aplicaciones, etc. Para esto se realiza un filtrado en función de la fuente y el tipo de evento, el formato general de este módulo es el siguiente:

```
module_begin
module_name MyEvent
module_type async_string
module_logevent
module_source <logName>
module_eventtype <event_type/level>
module_eventcode <event_id>
module_application <source>
module_pattern <text substring to match>
module_description
module_end
```

Figura 4.9
Formato de Módulo para monitoreo del Log de Eventos

El módulo de monitoreo sólo toma en cuenta aquellos eventos que tengan lugar durante la última vez que se ejecutó el agente. En la Figura 4.9 se muestra el esquema para la creación

del módulo, los valores que se introducen son los siguientes y se debe tener cuidado con la forma como se escriben puesto que son sensibles a mayúsculas y minúsculas.

- *module_source*: Origen del evento (System, Application, Security). Este campo esobligatorio.
- *module_eventtype*: Tipo de evento (error, information...). Es un campo opcional.
- *module_pattern*: Patrón a buscar (subcadena). Es un campo opcional.
- *module_eventcode*: Es el ID numérico del evento. Es un campo opcional.
- *module_application*: Aplicación origen del evento, ojo, no confundir con *module_source* que indica el nombre de la fuente o fichero log de donde se buscan los eventos.

Para generar un ejemplo de monitoreo de *logs*, se puede introducir el siguiente comando desde la línea de comandos de Windows y se generar una entrada en el Log de Eventos.

```
eventcreate /S localhost /U PUCE\usradmin /P Catolica1 /T ERROR /ID
324 /L Application /d "My custom error"
```

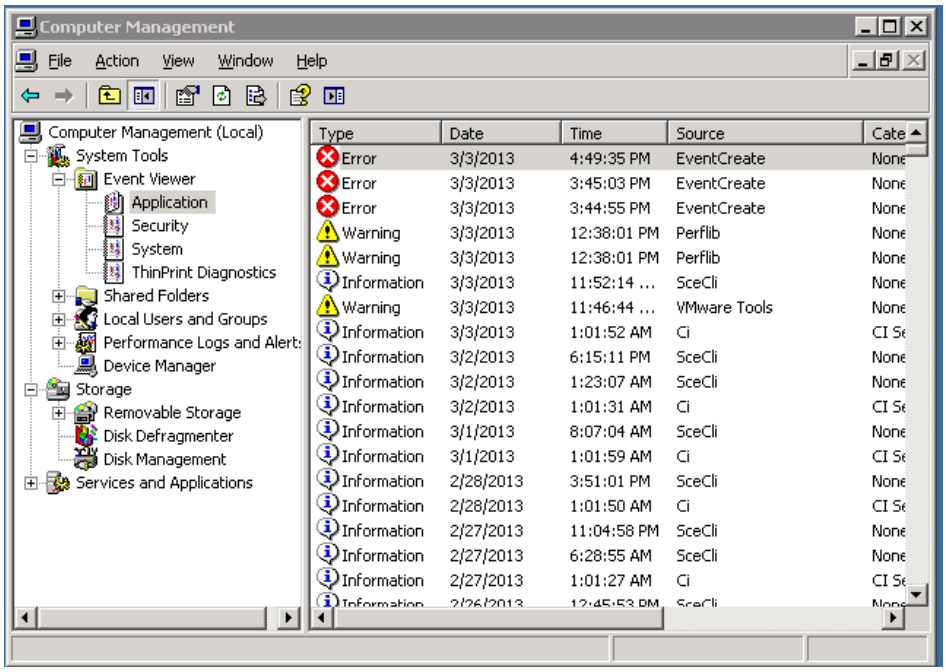


Figura 4.10 (a)
Visualización de eventos auto generados mediante línea de comandos

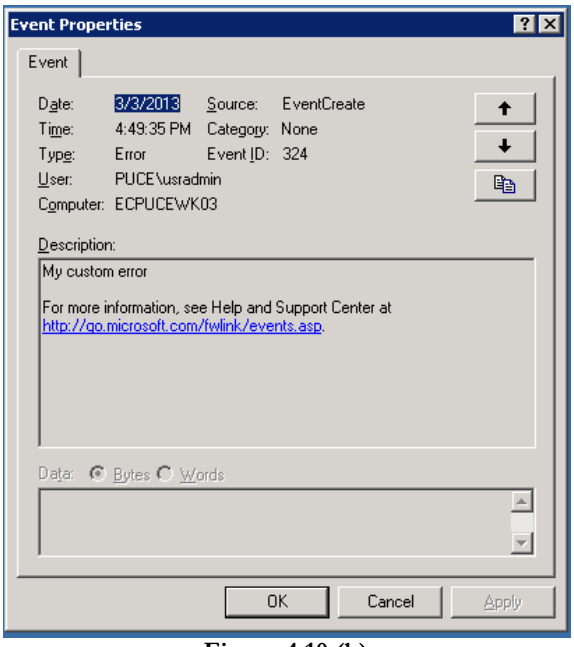


Figura 4.10 (b)
Propiedades del evento auto generado

Con los datos del evento reportado, según lo que se indica en la Figura 4.10 (a) y Figura 4.10 (b), se puede crear un módulo que revise periódicamente si existe una entrada con los datos configurados.

```
module_begin
module_name Evento_Test
module_type async_string
module_description EventLog de Prueba
module_logevent
module_source EventCreate
module_eventcode 324
module_end
```

Figura 4.11
Módulo de monitoreo para el Log de Eventos

En la Figura 4.11 se presenta el módulo implementado para el monitoreo del Log de Eventos, sin embargo después de las pruebas realizadas se ha observado que la herramienta detecta el último evento generado del total de eventos y no el último evento que haga referencia al código puesto en la línea *module_eventcode*; esta novedad se ha reportado en la página web del proyecto.

4.2 Monitoreo de agentes en sistemas operativos Unix/Linux

En sistemas Unix/Linux la recolección de información se la realiza ejecutando comandos hacia el sistema operativo, este método es equivalente a ejecutar sentencias desde una consola de terminal y obtener valores numéricos que son enviados a la herramienta de monitoreo. Para esto se crean módulos con la siguiente sintaxis:

```
module_begin
module_name Load Average
module_type generic_data
module_exec cat /proc/loadavg | cut -d' ' -f1
module_description Average process in CPU (Last minute)
module_end
```

Figura 4.12
Módulo de monitoreo en sistema Unix/Linux

En la Figura 4.12 se cuenta con un módulo que determina el promedio de consumo del procesador durante el último minuto, para esto ejecuta el comando descrito en la línea 4 *module_exec*. Como otro método para obtener información sobre sistemas Unix/Linux se tiene la posibilidad de realizar consultas SNMP a los equipos, para esto el agente de monitoreo ejecutará la consulta SNMP sobre el equipo, para lo cual se puede crear módulos como los descritos en la Figura 4.13.

```
module_begin
module_name SNMP get
module_type generic_data
module_exec snmpget 10.0.90.11 -v 2c -c public .1.3.6.1.4.1.2021.10.1.3.1| awk
'{print $4}'
module_end
```

Figura 4.13
Módulo de monitoreo en sistema Unix/Linux mediante SNMP

Entre algunas de las consultas que pueden proporcionar información de interés para monitoreo de sistemas Unix/Linux se tienen las expuestas en la Figura 4.14. [14]

```
Carga del CPU
1 minute Load:                              .1.3.6.1.4.1.2021.10.1.3.1
5 minute Load:                              .1.3.6.1.4.1.2021.10.1.3.2
15 minute Load:                             .1.3.6.1.4.1.2021.10.1.3.3

Estadísticas de Memoria
Total Swap Size:                            .1.3.6.1.4.1.2021.4.3.0
Available Swap Space:                       .1.3.6.1.4.1.2021.4.4.0
Total RAM in machine:                       .1.3.6.1.4.1.2021.4.5.0
Total RAM used:                             .1.3.6.1.4.1.2021.4.6.0
Total RAM Free:                             .1.3.6.1.4.1.2021.4.11.0

Estadísticas de disco
Path where the disk is mounted:             .1.3.6.1.4.1.2021.9.1.2.1
Path of the device for the partition:       .1.3.6.1.4.1.2021.9.1.3.1
Total size of the disk (Bytes):             .1.3.6.1.4.1.2021.9.1.6.1
Available space on the disk:                .1.3.6.1.4.1.2021.9.1.7.1
Used space on the disk:                     .1.3.6.1.4.1.2021.9.1.8.1
Percentage of space used on disk:           .1.3.6.1.4.1.2021.9.1.9.1
Percentage of inodes used on disk:          .1.3.6.1.4.1.2021.9.1.10.1

System Uptime:                              .1.3.6.1.2.1.1.3.0 [14]
```

Figura 4.14
Consultas SNMP y se respectiva OID para sistemas Unix/Linux

4.3 Monitoreo mediante la consola de Pandora FMS

La herramienta de monitoreo Pandora FMS incorpora la posibilidad de obtener información de los agentes monitoreados realizando consultas remotas a los mismos y obtener la información de interés para el administrador de la infraestructura, para esto se pueden realizar consultas del tipo SNMP, WMI y verificaciones de red. En la Figura 4.15 se presenta la ventana de creación de módulos de monitoreo del tipo WMI remoto, en las que se debe ingresar los valores del Nombre del módulo; los umbrales para determinar el momento que se produzcan alertas del tipo *Warning* y *Critical*; IP de destino del agente Windows; *Username* y *Password* autorizado para acceder a la información y que en este ejemplo es un usuario de dominio Windows con permisos suficientes; y finalmente el tipo de sentencia WMI que obtendrá la información.

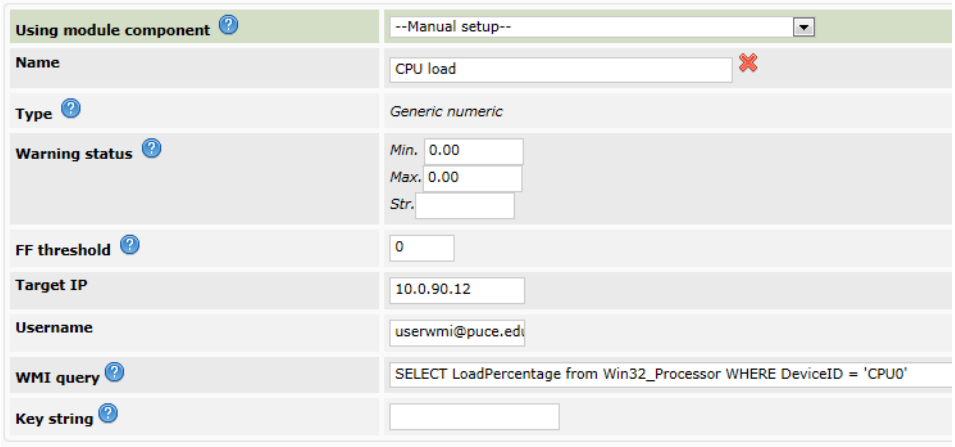


Figura 4.15
Monitoreo remoto mediante consultas WMI

De esta manera es posible obtener información de sistemas Windows en los que no se pueda instalar agentes de software. Esta técnica depende de la configuración individual en cada agente para la obtención del valor monitoreado y que el servicio WMI del servidor de Pandora FMS esté activo para que pueda recolectar la información.

Una segunda alternativa para la recolección de información es el uso de consultas SNMP hacia equipos que tengan habilitado este protocolo, previo a la generación de este tipo de módulos es necesario investigar la información que se puede consultar dentro del árbol de información MIB que tiene un determinado dispositivo.[33] Para esto se utilizará la herramienta *MIBBrowser*[34] que permite navegar sobre el árbol de información de SNMP y

[33] En la Figura 4.14 se muestras algunas OID's para sistemas Unix/Linux
[34] Fuente: http://www.ks-soft.net/hostmon.eng/wmi/index.htm

cargar MIBs de fabricantes y observar la información que se tiene disponible para monitoreo y consulta.

En la Figura 4.16 se observa la ventana principal de la herramienta *MIBBrowser* en la que se ha cargado las MIBs del fabricante CISCO para la consulta de información del dispositivo monitoreado. Previo a esto se debe conocer el nombre de la imagen del IOS del dispositivo y realizar la consulta en la página web de CISCO:

```
http://tools.cisco.com/ITDIT/MIBS/MainServlet
```

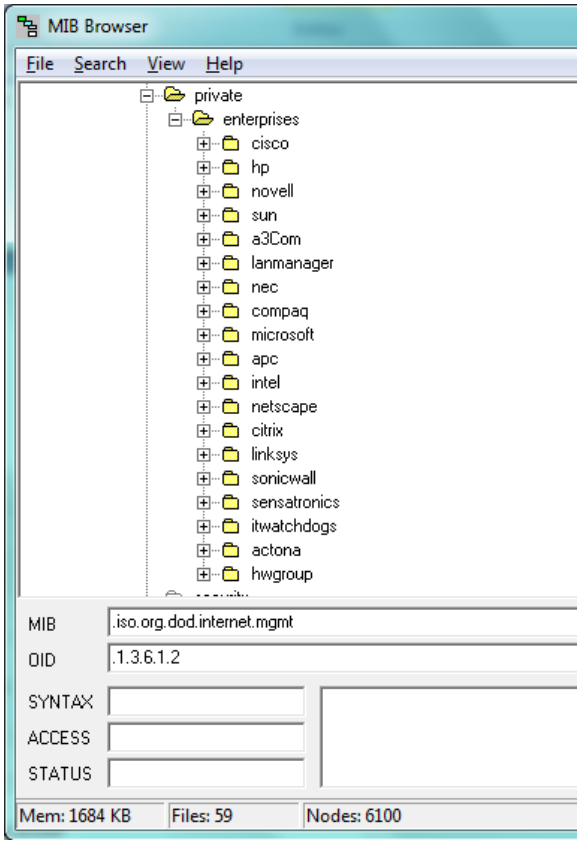
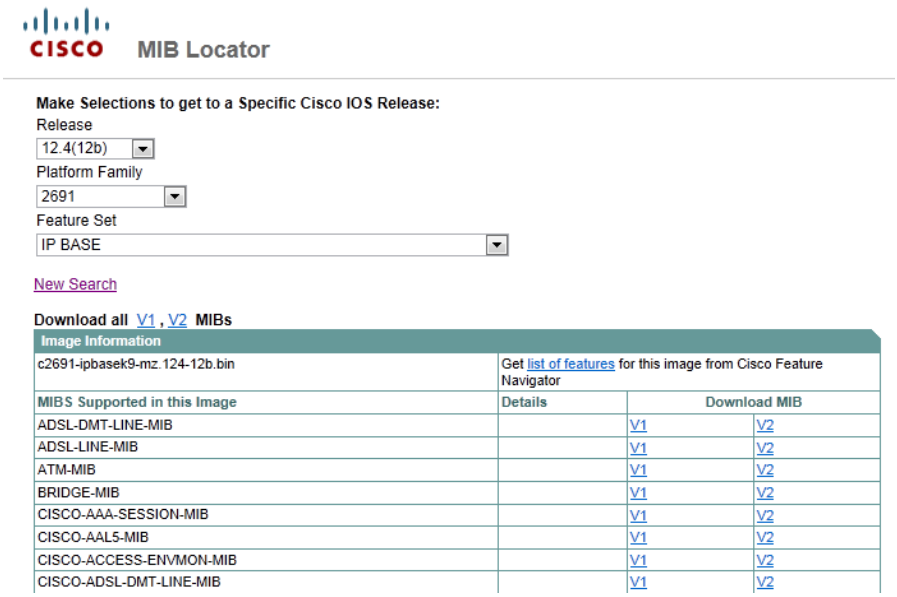


Figura 4.16
Revisión del árbol de la MIB

La información que debe ser ingresada para la búsqueda puede ser consultada ejecutando el siguiente comando que despliega la información del IOS instalado en el equipo.

```
show version
```



MIB Locator

Make Selections to get to a Specific Cisco IOS Release:

Release: 12.4(12b)

Platform Family: 2691

Feature Set: IP BASE

New Search

Download all V1 , V2 MIBs

Image Information			
c2691-ipbasek9-mz.124-12b.bin	Get list of features for this image from Cisco Feature Navigator		
MIBS Supported in this Image	**Details**	**Download MIB**	
ADSL-DMT-LINE-MIB		V1	V2
ADSL-LINE-MIB		V1	V2
ATM-MIB		V1	V2
BRIDGE-MIB		V1	V2
CISCO-AAA-SESSION-MIB		V1	V2
CISCO-AAL5-MIB		V1	V2
CISCO-ACCESS-ENVMON-MIB		V1	V2
CISCO-ADSL-DMT-LINE-MIB		V1	V2

Figura 4.17
Búsqueda de MIBs para un producto específico del Fabricante Cisco

Con la información de la versión del equipo, se coloca en la página web de CISCO como se muestra en la Figura 4.18 (a) los datos que permiten localizar las MIBs del dispositivo. Las MIBs por lo general requieren de otras MIBs para desplegar información, esto se conoce como dependencias. De las MIBs presentadas en la Figura 4.17 se tiene por ejemplo la MIB "TCP-MIB" que para ser utilizada necesita que sean cargadas sus dependencias para funcionar correctamente, para esto se accede a la siguiente dirección web de CISCO:

```
http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2
```

En esta web se puede localizar todas las dependencias de la MIB como se indica en la Figura 4.18 (a), posterior a esto es posible descargar la MIB y las dependencias como se indica en la Figura 4.18 (b).

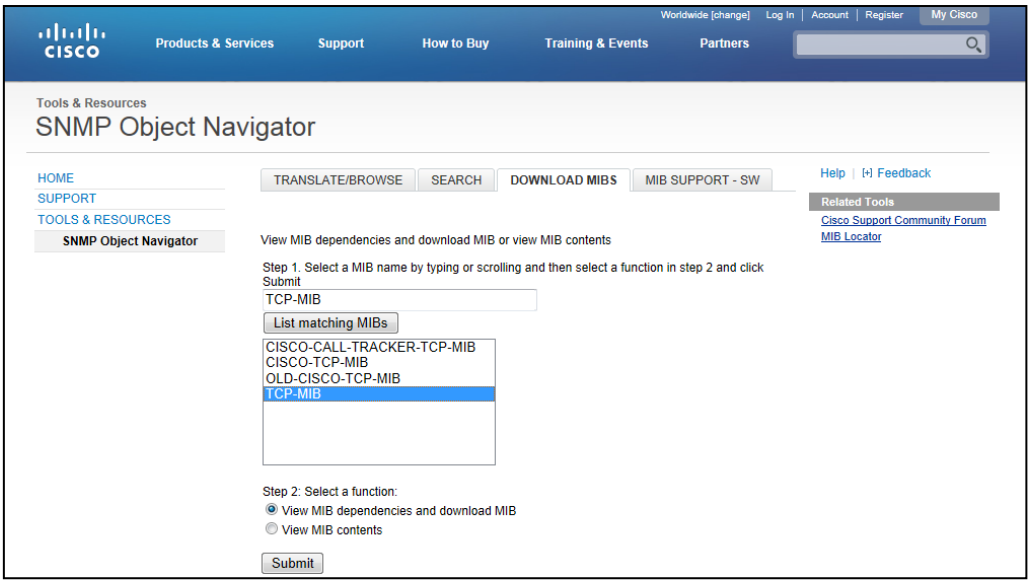


Figura 4.18 (a)
Visualización de dependencias de las MIBs

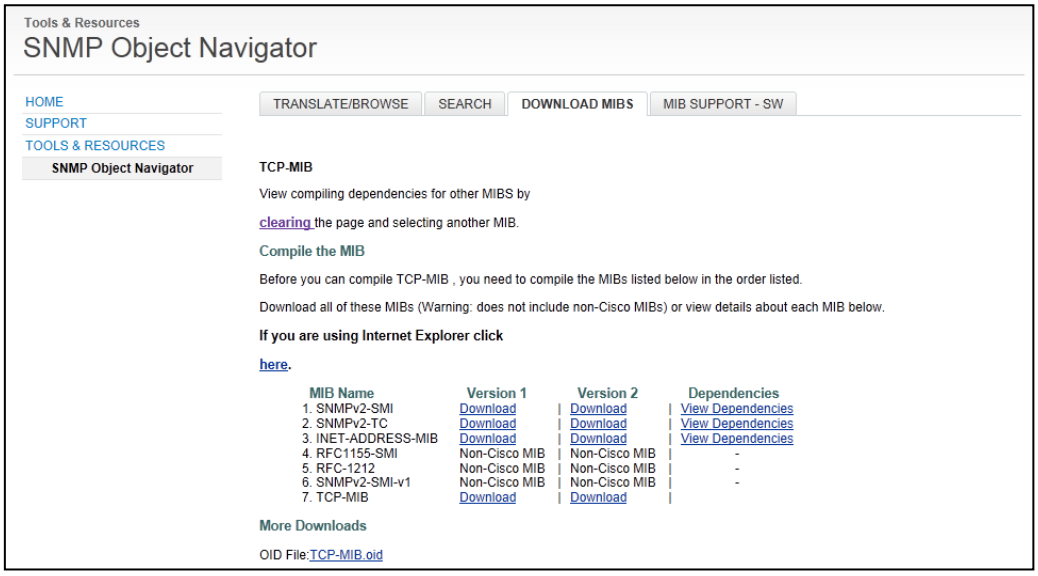


Figura 4.18 (b)
Descarga de la MIB y las dependencias

Una vez concluido este proceso se considera tarea del administrador navegar sobre el árbol SNMP y determinar la información que se va a monitorear. Con esto se puede pasar a Pandora FMS y generar el módulo de monitoreo basado en la OID, en el caso del ejemplo indicado en la Figura 4.19 se monitorea consumo de CPU en un dispositivo *Catalyst*, en el que se ha especificado que se está usando la versión 2c del protocolo SNMP y la comunidad es del tipo *public*.

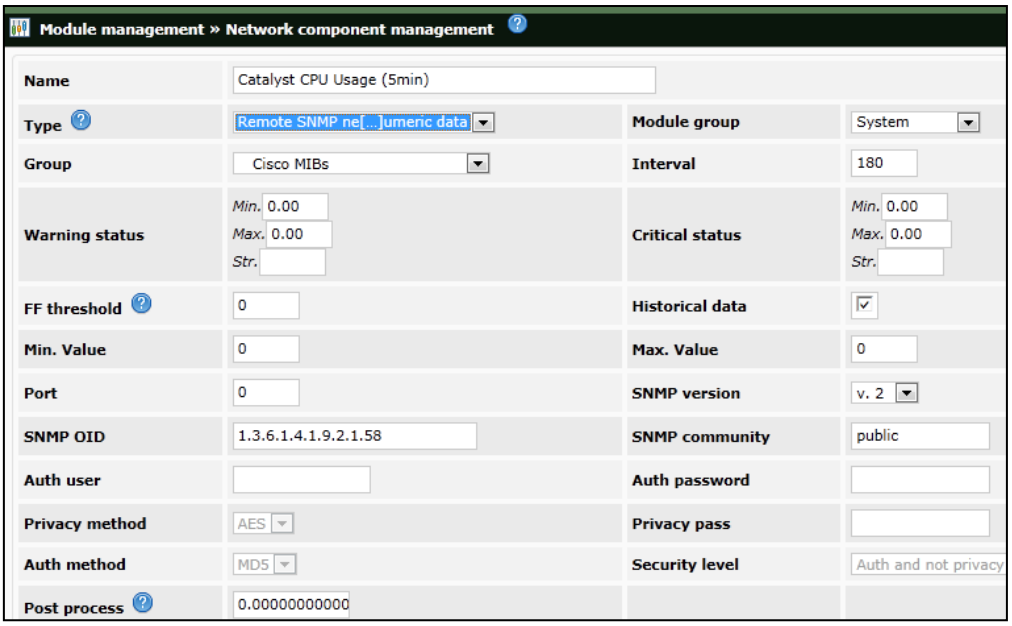


Figura 4.19
Monitoreo mediante consulta SNMP

4.4 Generación de Alertas

Una de las principales tareas que debe ejecutar cualquier herramienta de monitoreo es la de enviar notificaciones cuando los umbrales preestablecidos para el monitoreo han sido sobrepasados. Por ejemplo si el procesador ha excedido un consumo del 90% se debe enviar notificaciones que permitan tomar acciones por parte del administrador, la herramienta de monitoreo Pandora FMS permite automatizar tareas como envío de SMS´s y correo electrónico. Para el efecto se deben generar las acciones "*Actions*" que se ejecutan cuando se produce un evento del tipo *Warning* o *Critical*. Para esto se configura el servidor donde corre la herramienta de monitoreo (Puntos 3.2.9 y 3.2.10) para que mediante un comando se pueda ejecutar el envío de las notificaciones. El detalle de creación de las alertas se puede localizar en el manual de la herramienta para la versión 4.0.2.

Para la ejecución de este trabajo se configuró una alerta de consumo de Procesador o CPU sobre un servidor que tiene configurado un agente de monitoreo, en el cual se realizó una modificación en el valor máximo de CPU y se ubicó en 105%, como se indica en la Figura 4.20, esto debido a que se pudo detectar mediante un monitoreo del Sistema Operativo que el valor del 100% se superaba y la alerta no reconocía valores superiores a éste.

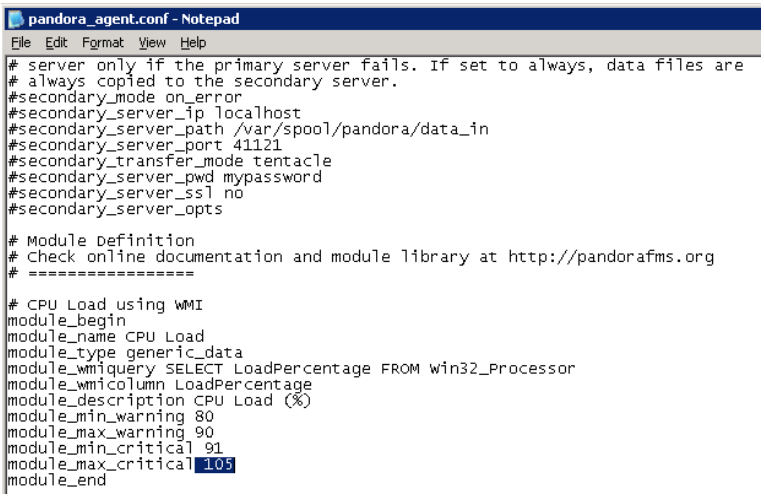


Figura 4.20
Monitoreo de monitoreo modificado

Posterior a esto se realiza una prueba de carga sobre el servidor y se verificó que se reciben las alertas generadas desde la herramienta de monitoreo con el detalle de la alerta, el estado y el nombre del equipo que presenta la alerta, estas alertas llegan mediante correo electrónico y notificaciones SMS según lo que se presenta en la Figura 4.21.

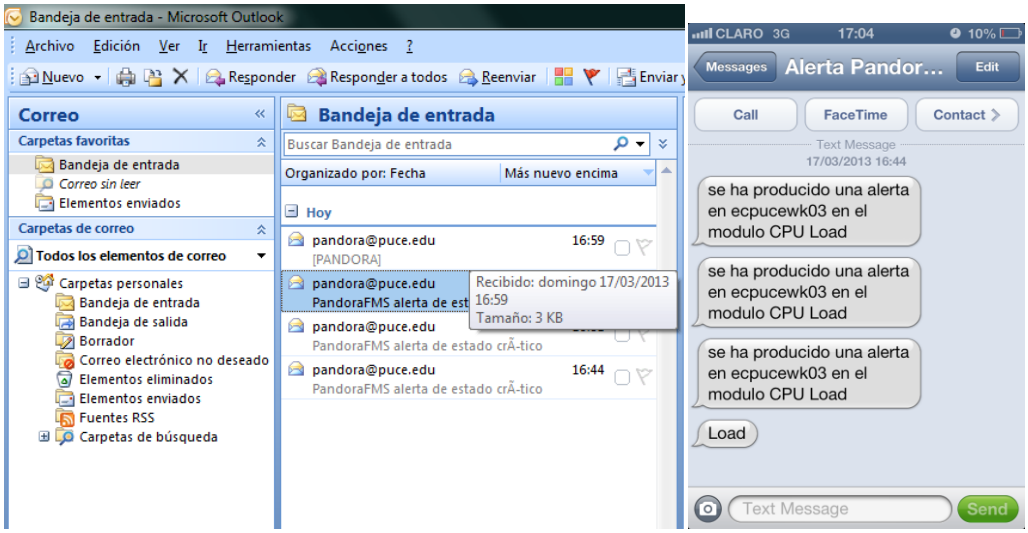


Figura 4.21
Notificaciones vía Correo Electrónico y SMS

4.5 Generación de reportes sobre los valores monitoreados

La generación de reportes en la herramienta se limita a obtener histogramas de los datos almacenados en la Base de Datos *MySQL* para analizar el comportamiento y determinar las tendencias de crecimiento o decaimiento en el uso de un determinado recurso. En la Figura 4.22 (a) se observa que se puede generar un gráfico que indica el consumo de CPU y en la Figura 4.22 (b) se obtiene el detalle de los valores numéricos generados con la marca de tiempo en el momento en que fueron tomados. Estos datos sin embargo no son de un

monitoreo en tiempo real y son obtenidos con espaciamiento de tiempo de 5 minutos que se ha considerado como un tiempo apropiado para realizar las mediciones sobre los agentes monitoreados.

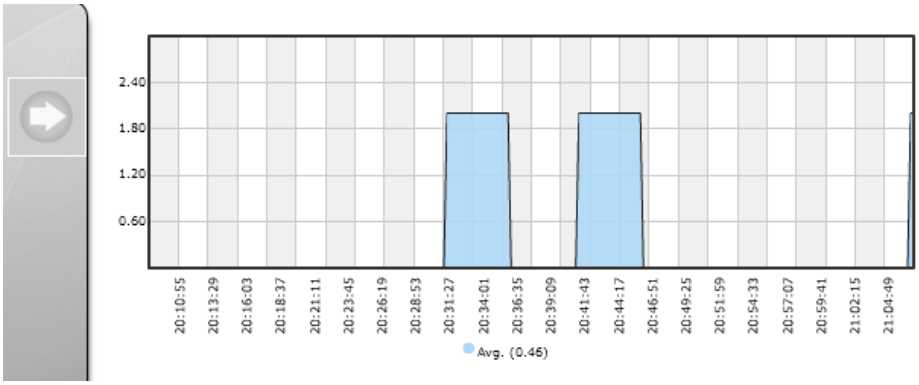
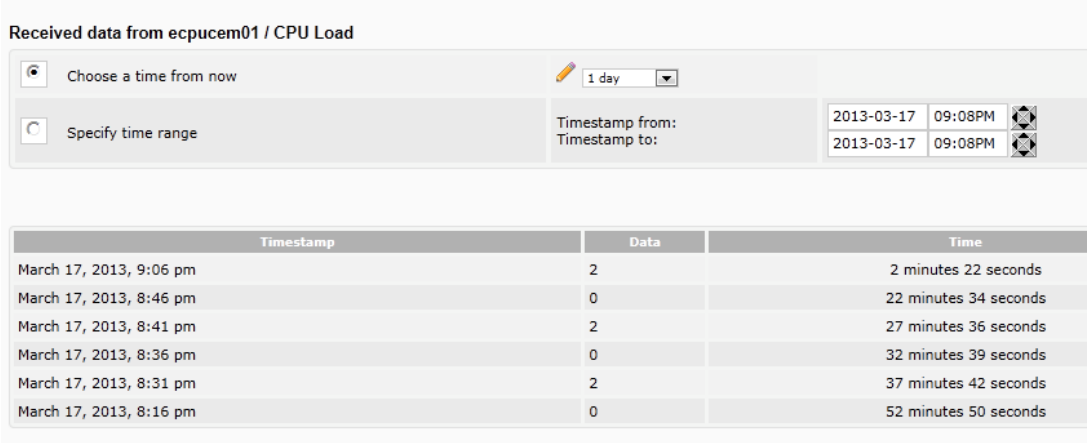


Figura 4.22(a)
Generación de gráfico de consumo de CPU



Timestamp	Data	Time
March 17, 2013, 9:06 pm	2	2 minutes 22 seconds
March 17, 2013, 8:46 pm	0	22 minutes 34 seconds
March 17, 2013, 8:41 pm	2	27 minutes 36 seconds
March 17, 2013, 8:36 pm	0	32 minutes 39 seconds
March 17, 2013, 8:31 pm	2	37 minutes 42 seconds
March 17, 2013, 8:16 pm	0	52 minutes 50 seconds

Figura 4.22 (b)
Reporte de Valores numéricos de consumo de CPU

4.6 Generación de Reporte para CMDB[35]

La herramienta de monitoreo Pandora FMS puede ser usada como una Base de Datos para la gestión de la configuración ó CMDB por sus siglas en inglés, porque permite almacenar información de los equipos monitoreados como: CPU, memoria, disco y Sistema Operativo; adicionalmente para sistemas Windows se puede realizar consultas WMI hacia los equipos monitoreados para extraer información que alimenta la CMDB como el nombre del fabricante del equipo, fecha de instalación, etc. En la Figura 4.23 se ha configurado tres módulos WMI que consultan información del servidor *ecpucewk01* y dan algunos detalles del equipo que se almacena en la Base de Datos de Pandora FMS.

---

[35] CMDB *Configuration Management DataBase* o Base de Datos de la Gestión de Configuración es un repositorio de información donde se relacionan todos los componentes de un sistema de información, ya sean hardware, software, documentación, etc.

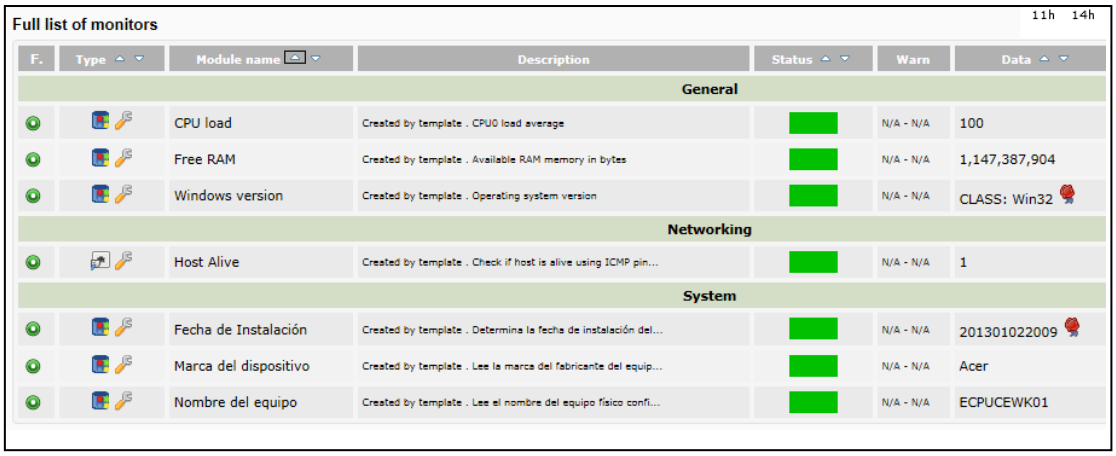


Figura 4.23
Generación de información vía WMI

La herramienta Pandora FMS además posee campos configurables en los que se puede colocar información acerca del Número de Serie, Departamento y un Identificador adicional. Esto se lo puede observar en la Figura 4.24, en donde se colocó información acerca del equipo *ecpucewk01*.

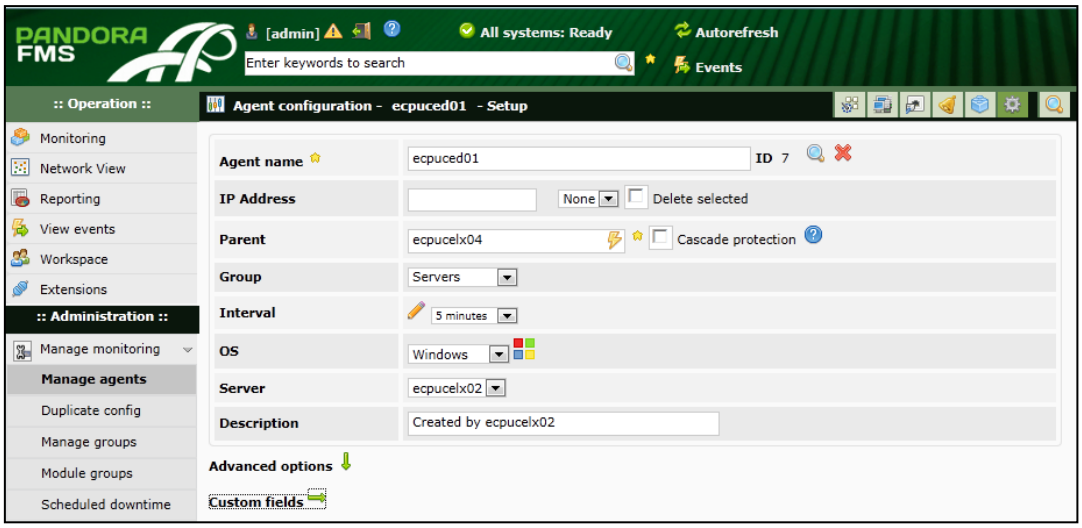


Figura 4.24
Campos administrables para información de equipos monitoreados

Con todo lo indicado en este punto se ha podido determinar que la herramienta puede adaptarse para cumplir las necesidades de alimentar una CMDB mediante los ajustes que se hagan para recolectar información de los equipos monitoreados, sin embargo no puede considerarse en si como una CMDB.

4.6 Comparación con la herramienta de monitoreo *HP OpenView*[36]

Una de las principales herramientas para monitoreo de infraestructura bajo licencia desarrollada por la empresa Hewlett Packard es el *HP OpenView,* que realiza tareas de monitoreo sobre agentes desplegados en los sistemas operativos soportados y que adicionalmente realiza tareas de monitoreo mediante el protocolo SNMP. Para esto *HP OpenView* utiliza un servidor central donde se ejecuta la aplicación y la base de datos Oracle de la herramienta de monitoreo; adicionalmente se cuenta con una consola central de configuración conocida como *Operations Manager*, donde se realizan las tareas de configuración de las políticas de monitoreo, que para el caso de Pandora FMS se conocen como módulos de monitoreo, que una vez asignados a los agentes monitoreados se despliegan en los servidores. En la Figura 4.25 se puede observar un esquema general del funcionamiento de *HP OpenView*.

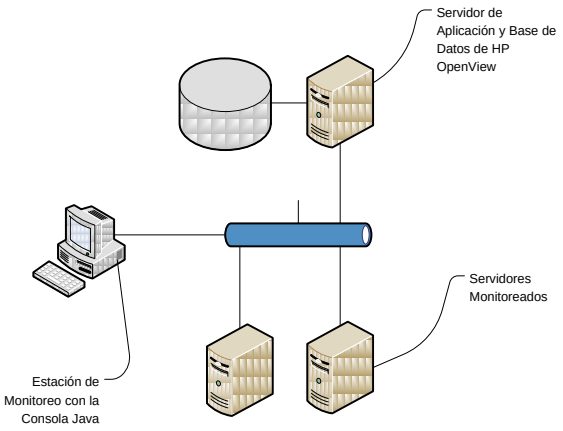


Figura 4.25
Esquema de la Infraestructura de HP OpenView

Entre las similitudes que se encuentran con la herramienta Pandora FMS se observó que ambas herramientas hacen uso de agentes desplegados sobre los nodos monitoreados para obtener información que posteriormente es procesada en la herramienta central de monitoreo, esto resulta de mucha utilidad pues se libera al servidor central de realizar las tareas de consultas y delega las mismas a los agentes. Adicionalmente para el monitoreo de nodos de telecomunicaciones *OpenView* utiliza una tercera herramienta conocida como *HP Network Node Manager* que realiza consultas a los nodos mediante protocolo SNMP. Dado que la herramienta HP *OpenView* presenta varios módulos o paquetes de software para el

[36] La información acerca de HP OpenView se la obtuvo de la experiencia del autor de la tesis en esta herramienta, no se incluyen citas bibliográficas pues no existe información de manera oficial y de acceso público en la página web de HP.

monitoreo, en la Tabla 4.1 se presenta un detalle de las tareas de monitoreo que realiza cada uno en comparación a Pandora FMS.

	Network Node Manager	*SiteScope*	*Operations Manager*	*Pandora FMS*
Gráficas	Si	Si	Si	Si
Estadísticas	Si	Si	Si	Si
Predicción de Estadísticas	No	No	No	Si
Autodescubrimiento	Si	Si	Si	Si
Agentes	No	No	Si	Si
SNMP	Si	No	No	Si
Scripts	No	No	Si	Si
Plugins	No	No	Si	Si
Alertas	Si	Si	Si	Si
Aplicación Web	Si	Si	Si	Si
Monitorización Distribuida	No	No	Si	Si
Licencia	Pagada	Pagada	Pagada	GPL y pagada para la versión Enterprise
Mapas	Si	Si	Si	Si
Seguridad	Si – Usando SNMP V2c	No	Si	No
Manejo de Eventos	Si	Si	Si	Si

Tabla 4.1
Comparación de Pandora FMS con HP *OpenView*[37]

Como se analiza en la Tabla 4.1 existen varios módulos dentro de la herramienta HP *OpenView* que realizan diferentes tareas de monitoreo, a continuación se explicarán cada una de estas herramientas.

4.6.1 *Network Node Manager*

Se encarga de la gestión de la red mediante el manejo de ICMP/SNMP con las siguientes funcionalidades principales:

- Descubrimiento de equipos de comunicación (SW, *Routers*, *Firewalls*, PBX, APs, ACs)
- Generación automática de topología en L2 y L3.
- Monitoreo de *performance* y disponibilidad de los equipos de comunicación mediante indicadores y gráficos online.
- Gestión de incidentes de status y rendimiento de la red.
- Gestión de *traps* SNMP y *Syslogs* enviados por los equipos.

[37] Información resumida de los manuales de la herramienta HP *OpenView* y Pandora FMS

- Disparo de acciones automáticas en base a los incidentes generados.

4.6.2 *Sitescope*

Se encarga de la gestión y monitoreo de la Infraestructura TI (Servidores, aplicaciones, dispositivos especiales) con tecnología "sin agente" mediante una conexión ICMP, SSH, *Netbios*, WMI, con las siguientes funcionalidades principales:

- Monitoreo de disponibilidad de equipos.
- Monitoreo de performance de servidores.
- Monitoreo de servicios y procesos Windows.
- Monitoreo de procesos UNIX.
- Monitoreo de *VMware*.
- Monitoreo de aplicaciones WEB, Base de Datos, URLs, etc.

4.6.3 *Operations Manager*

Se encarga de la gestión de la infraestructura TI y monitoreo mediante agentes e implementación de políticas según necesidades específicas. Las principales funcionalidades son:

- Monitoreo detallado de Performance de Servidores.
- Monitoreo de *Logs*.
- Despliegue de políticas en base a necesidades específicas.
- Generación de gráficas de performance y disponibilidad mediante Performance Manager.
- Consolidación de eventos provenientes de NNMi y *Sitescope* mediante la integración con el agente de *Operations*.

Del análisis realizado a ambas herramientas se observa que en ambas realizan consultas periódicas para tener un estado global del sistema; y que ambas cuentan con herramientas que permiten generar reportes gráficos y de tablas de datos para su posterior análisis, sin embargo su principal diferencia radica en la configuraciones precargadas que tiene *HP OpenView* para realizar monitoreo de esquemas específicos como ambientes Unix, infraestructura de correo electrónico como *Microsoft Exchange* o *Active Directory*; en contraste la herramienta Pandora FMS en su versión *Opensource* no posee estas características y sus posibilidades de configuración dependen exclusivamente del administrador de la herramienta.

CAPÍTULO 5

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- En la ejecución de este trabajo se determinó que la herramienta Pandora FMS es una opción de software libre que integra el monitoreo mediante agentes, consultas WMI, consultas SNMP y verificaciones de red (ICMP, Telnet) en una sola consola de monitoreo, que permite conocer el estado de la infraestructura monitoreada. Siendo esa integración de características de monitoreo su principal ventaja pues no requiere de módulos adicionales o integración con herramientas de terceros.
- Para el monitoreo de servidores Pandora FMS incorpora un agente que se instala en los equipos, este mecanismo que es similar al que implementa la herramienta HP *OpenView*, es de gran utilidad pues descentraliza la tarea de monitoreo y restringe las tareas al servidor central de monitoreo a la recolección de los datos captados por el agente local instalado en la máquina.
- Durante el desarrollo de este proyecto se ha trabajado bastante con los conceptos del protocolo SNMP y la tecnología WMI, que en el fondo son mecanismos de acceso a información de los equipos monitoreados. Mediante estos dos métodos es posible extraer casi cualquier dato que permita conocer el estado de los equipos monitoreados.
- Una de las principales funcionalidades de la herramienta de monitoreo es la generación de alertas cuando se produzcan eventos críticos en los equipos monitoreados (P.E. el consumo de CPU está por encima del 90%) y enviar notificaciones como mensajes de texto y correos electrónicos, que son de gran ayuda cuando se posee una infraestructura de gran tamaño y que debe ser monitoreada durante las 24 horas del día. Mediante esta funcionalidad se garantiza que el administrador de red pueda tomar acciones correctivas sobre los equipos que presenten inconvenientes en el momento que se produzca la falla sin necesidad de estar siempre frente a la consola de monitoreo.
- Durante el desarrollo del trabajo se pudo observar que existe gran cantidad de información y parámetros que pueden ser monitoreados en una red, sin embargo es importante reducir los parámetros monitoreados a los que entreguen información relevante sobre el estado de nuestros equipos como CPU, memoria, estado de los discos y en caso de ser posible información sobre el hardware. De no ser posible se debe al menos procurar que las alertas que se generen en la herramienta Pandora FMS

sean sobre los parámetros más críticos de la infraestructura que puedan provocar indisponibilidad.

5.1 Recomendaciones

- Dentro del manejo de la herramienta de Monitoreo es recomendable realizar una carga manual de los equipos para guardar un estricto control de los equipos que ingresan. De igual manera es recomendable asignar un grupo de políticas pre-establecidas por tipo de sistema operativo para de esa forma guardar uniformidad en los servidores ingresados en la herramienta.
- El envío de notificaciones vía SMS y correo electrónico debe estar limitado a equipos críticos dentro de la infraestructura para evitar congestionar los buzones de los responsables con alertas que por la cantidad pueden ser simplemente ignoradas.
- De trabajarse en redes que estén formadas por varias VLANs y protegidas por firewalls, es recomendable realizar pruebas de conexión y garantizar que se tiene plena conectividad entre los dispositivos monitoreados y el servidor central.

Ñ2CAPÍTULO 6

6 RECOMENDACIONES PARA TRABAJOS FUTUROS

- En trabajos futuros que hagan uso de lo expuesto aquí se recomienda trabajar sobre la Base de Datos de la aplicación de Pandora FMS para generar reportes directamente sobre la misma y obtener reportes personalizados sobre el estado de la infraestructura.
- La herramienta de monitoreo Pandora FMS posee una versión de pago que se conoce como Pandora FMS *Enterprise,* que incorpora varias ventajas sobre la versión libre como la posibilidad de cargar MIB's para visualización de los mensajes SNMP, módulos de monitoreo pre-configurados, etc. Un análisis de esta versión de la herramienta es importante realizarlo para determinar qué ventajas se pueden obtener y si conviene optar por la compra de la misma.
- Uno de los elementos de mayor esfuerzo en este trabajo fue la recepción de las alertas SNMP o *traps*, por esto se recomienda enfocar trabajos futuros en el análisis de las alertas que envían los dispositivos cuando se producen eventos de caída de interfaces del equipo o consumo alto de recursos del equipo.

CAPÍTULO 7

7 BIBLIOGRAFÍA

1. "Pandora FMS 4.0.2 Manual de Usuario". 1era. Edición (España). 20 de junio 2012.[Online]. Available: http://hivelocity.dl.sourceforge.net/project/pandora/Pandora%20FMS%204.0.2/Final/Documentation/PandoraFMS_4.0.2_Manual_ES.pdf

2. L. Hernández. "Tutorial de NET-SNMP". Escuela Técnica Superior de Ingenieros de Telecomunicación – Universidad de Las Palmas de Gran Canaria. 2006.

3. "Windows Management Instrumentation (Windows)". [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa394582(v=vs.85).aspx

4. E. Sevilla. "Introducción a SNMP". [Online]. Available: http://www.unainet.net/documents/SNMP.pdf

5. "Simple Network Management Protocol". [Online]. Available: http://es.wikipedia.org/wiki/Simple_Network_Management_Protocol#Componentes_b.C3.A1sicos

6. R. Chaganti. WMI Query Language via PowerShell. [Online]. Available: http://www.ravichaganti.com/blog/?p=1979

7. L. Arévalo. "Sistema para la gestión de computadoras bajo la Plataforma Windows usando WMI a través de páginas Web". Escuela Politécnica Nacional. Quito – Ecuador. 2010.

8. M. Elsabagh. "Automatic CD-ROM ejection of ZTE 3G modem". 2010. [Online]. Available: http://melsabagh.blogspot.com/2010/01/zte-modem-and-vista-problem.html#!/2010/01/zte-modem-and-vista-problem.html

9. "Serial Console Howto". 2013. [Online]. Available: https://help.ubuntu.com/community/SerialConsoleHowto

10. Q. Plummer. "How to Send an SMS Using Hyperterminal". [Online]. Available: http://www.ehow.com/how_7486315_send-sms-using-hyperterminal.html

11. A. Rajarja. “SMS Server Tools 3 Installation”. 2012. [Online]. Available: https://github.com/antonraharja/playSMS/wiki/SMS-Server-Tools-3-Installation

12. P. Welcher. “Configuring SNMP in Cisco Routers”. 1999. [Online]. Available: http://www.netcraftsmen.net/resources/archived-articles/370-configuring-snmp-in-cisco-routers.html

13. “Cisco IOS Network Management Command Reference”. 2013. [Online]. Available: http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_19.html

14. “Linux SNMP OID’s for CPU, Memory and Disk Statistics”. 2006. [Online]. Available:http://www.debianadmin.com/linux-snmp-oids-for-cpumemory-and-disk-statistics.html

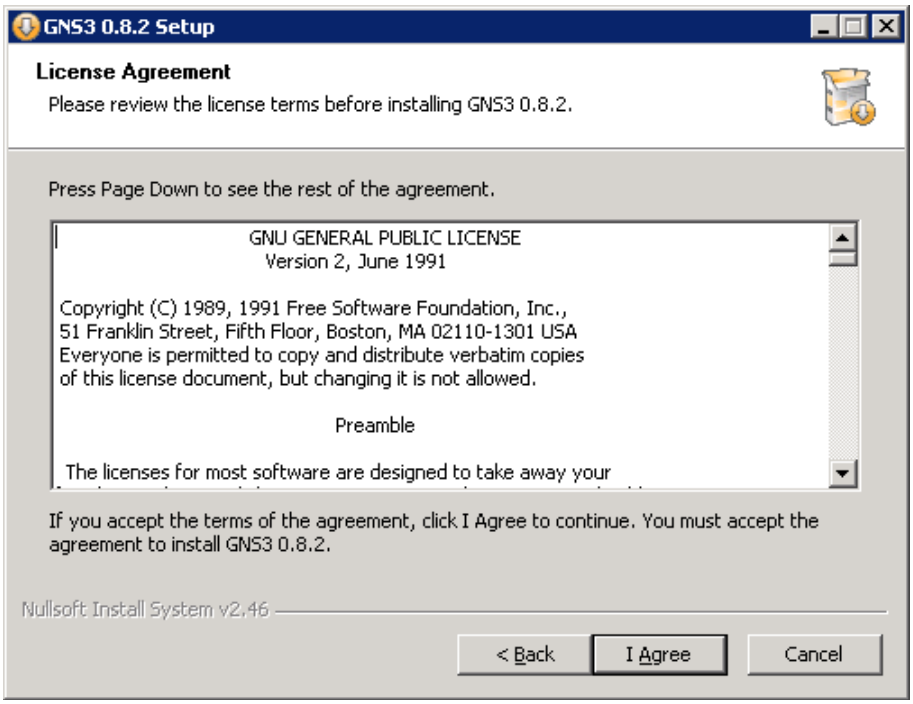
ANEXOS

PROCESO DE INSTALACIÓN DE GNS3

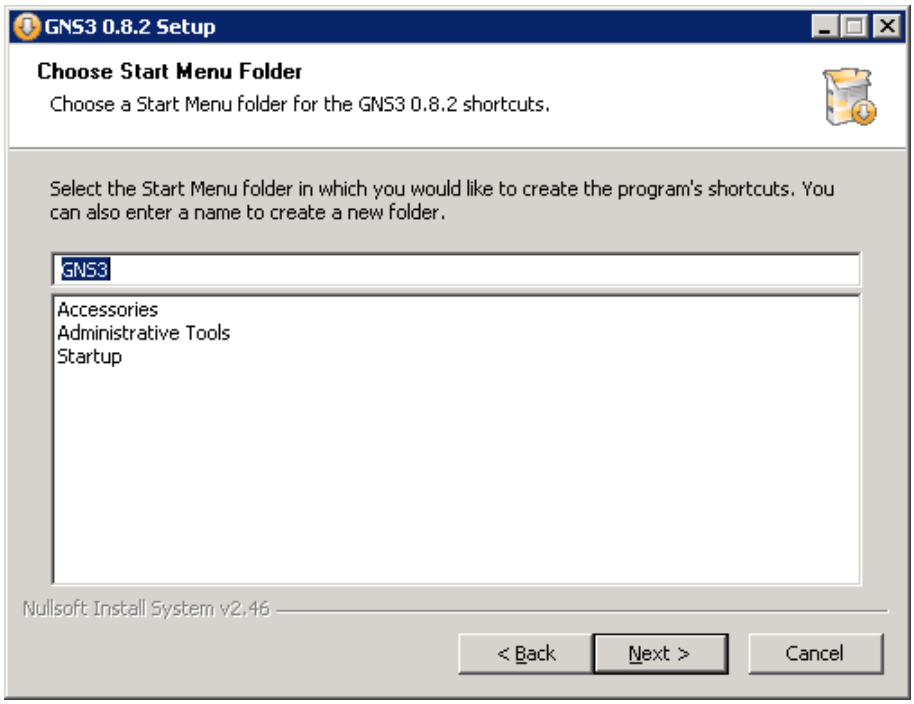
El proceso de instalación empieza descargando el software de GNS3 desde la página web: http://www.gns3.net/. Posterior a esto se sigue el proceso de instalación en dónde se debe instalar los paquetes completos.



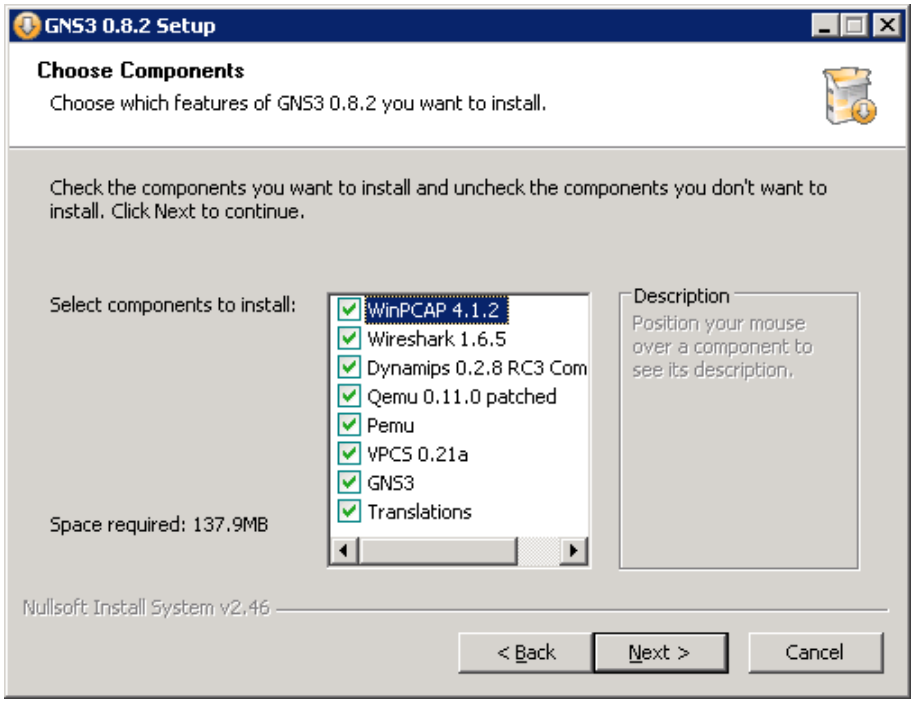
Referencia: Se da *click* en siguiente y se continúa el proceso de instalación.



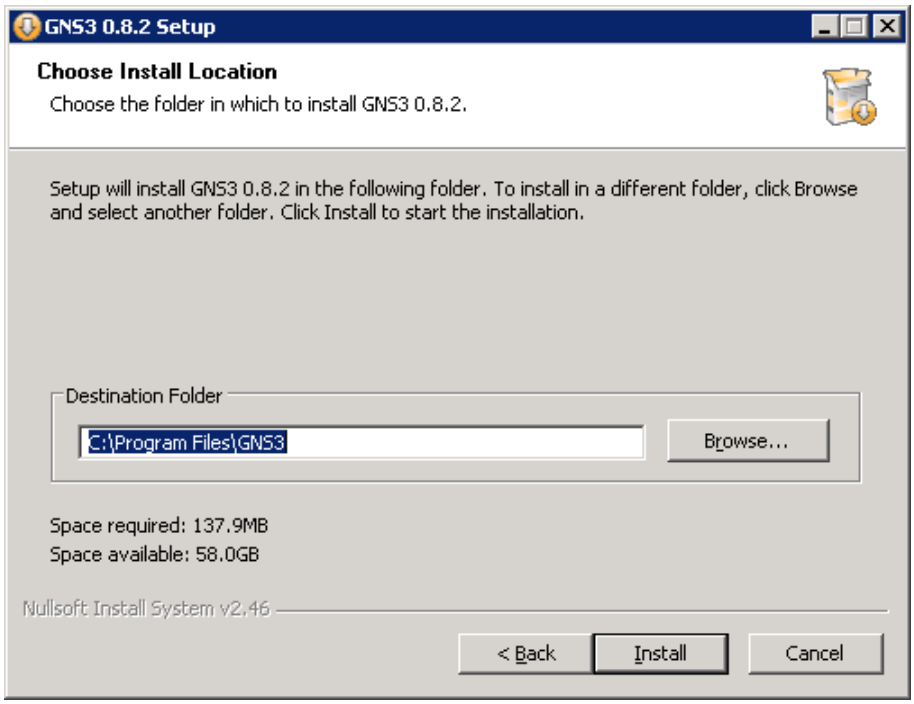
Referencia: Se da *click* en "*I Agree*" y se continúa el proceso de instalación.



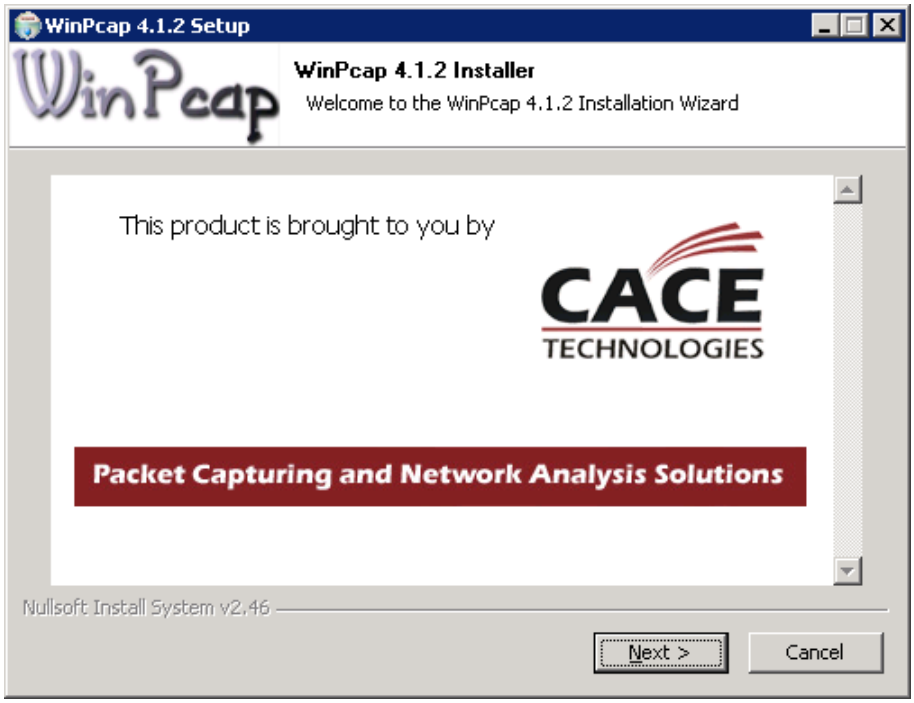
Referencia: Se debe escoger el nombre de la carpeta a crearse en el menú de programas.



Referencia: Se debe escoger los paquetes a instalarse, en el caso de una instalación desde cero se deberán escoger todos los paquetes.



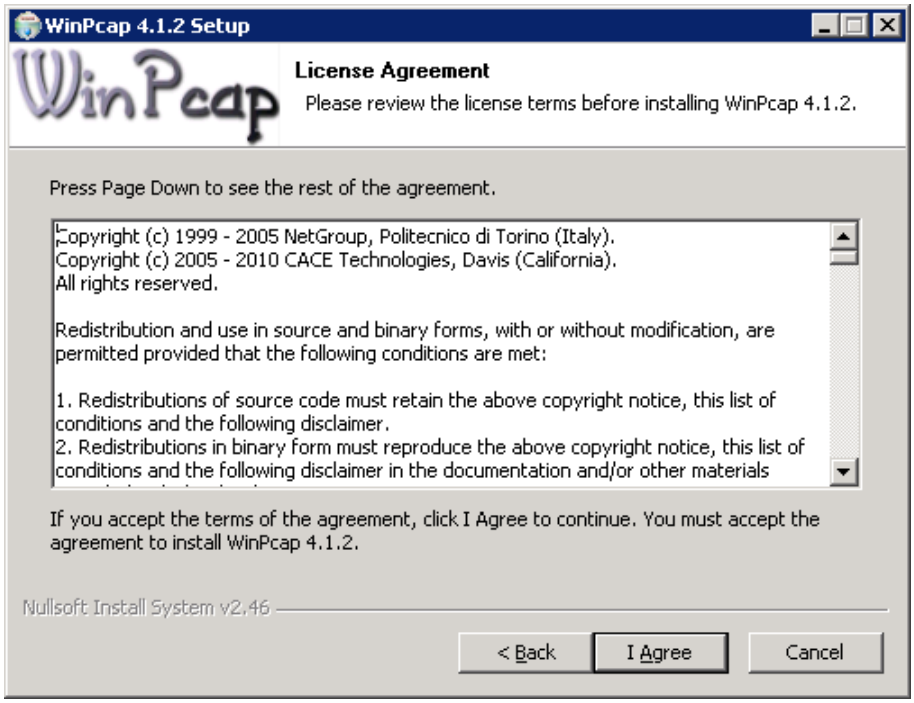
Referencia: Se escoge la ruta por defecto para realizar la instalación.



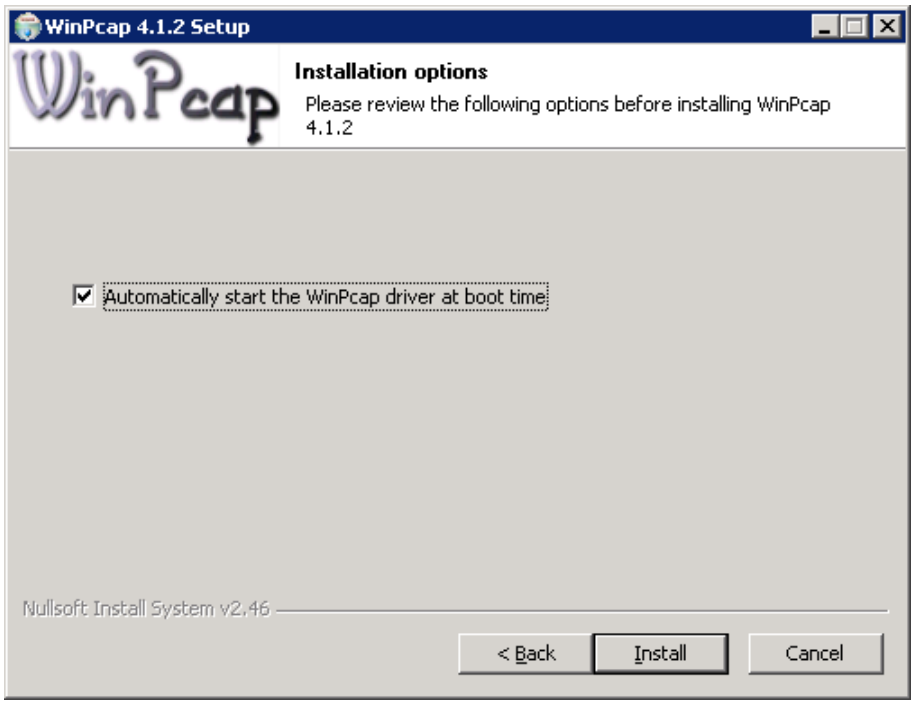
Referencia: Antes de instalar los paquetes de GNS3, nos lanzará la instalación de *WinPcap* que forma parte de los pre-requisitos.



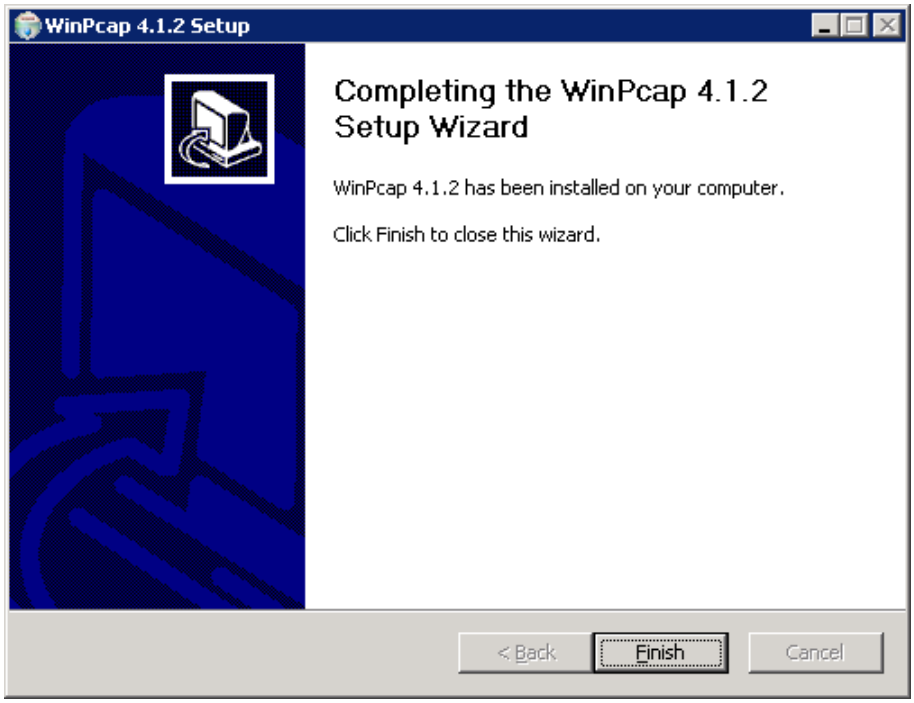
Referencia: Se empieza la instalación de *WinPcap*.



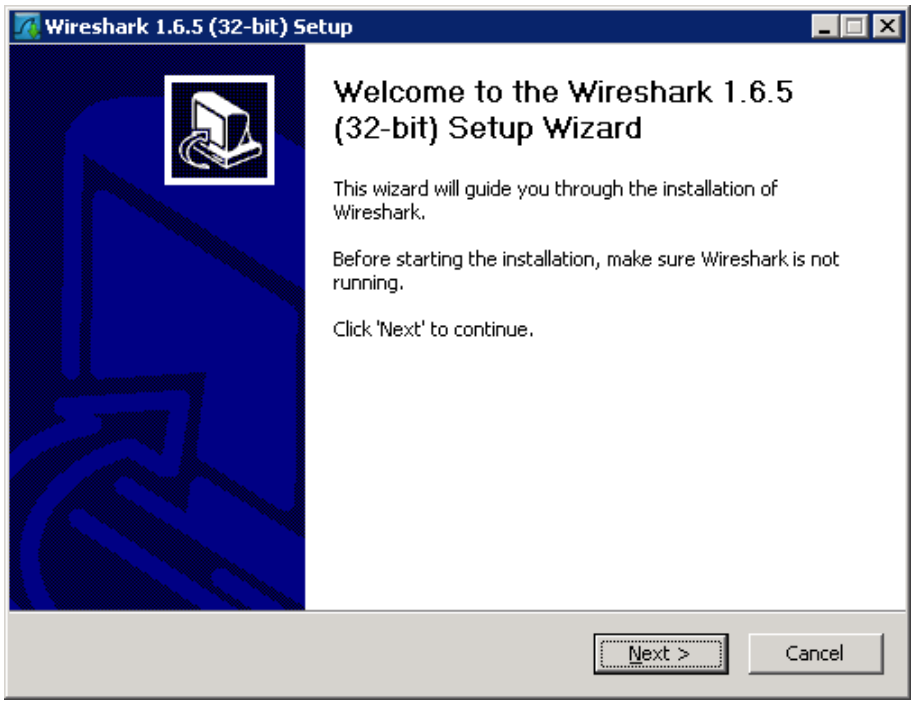
Referencia: Se da click en "*I Agree*" pañ2ra aceptar la licencia de instalación del software *WinPcap*.



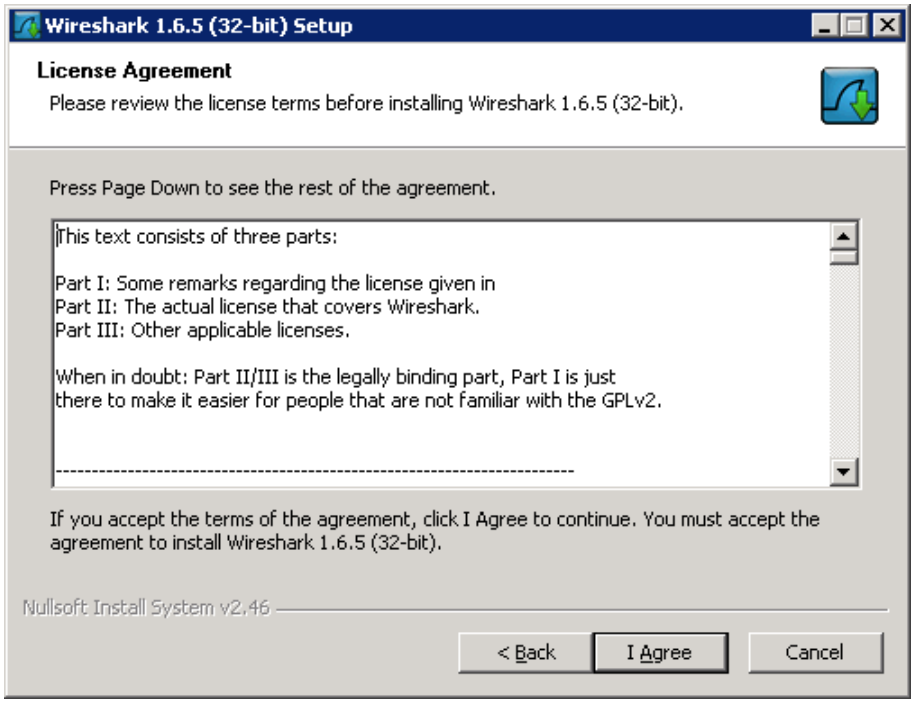
Referencia: Deberemos señalar la opción por defecto que empieza los servicios de *WinPcap* en cada ocasión que se levanta el Sistema Operativo.



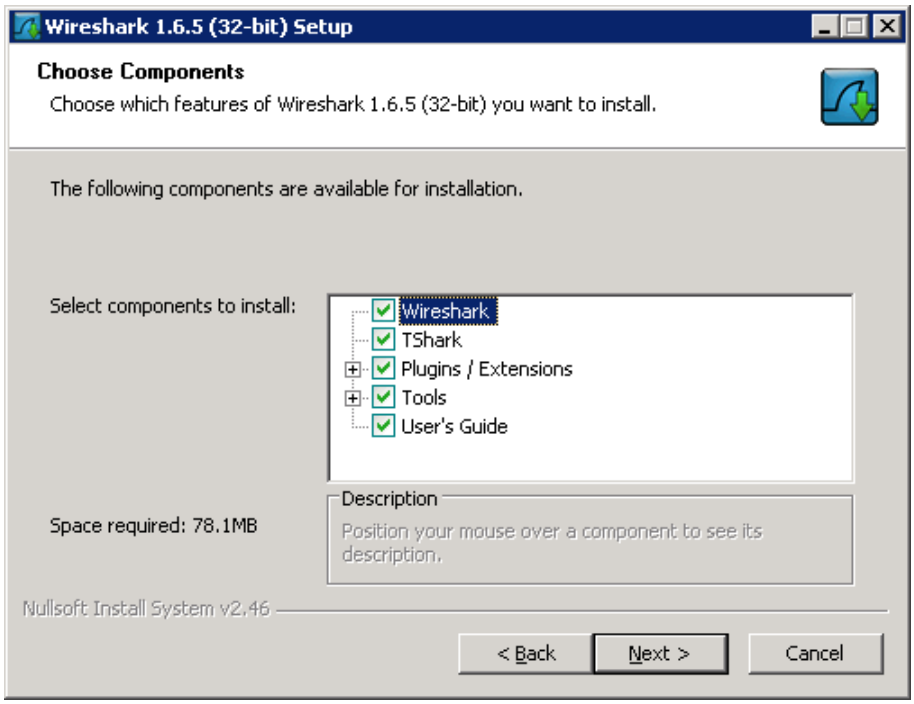
Referencia: El proceso de instalación de *WinPcap* se ha finalizado.



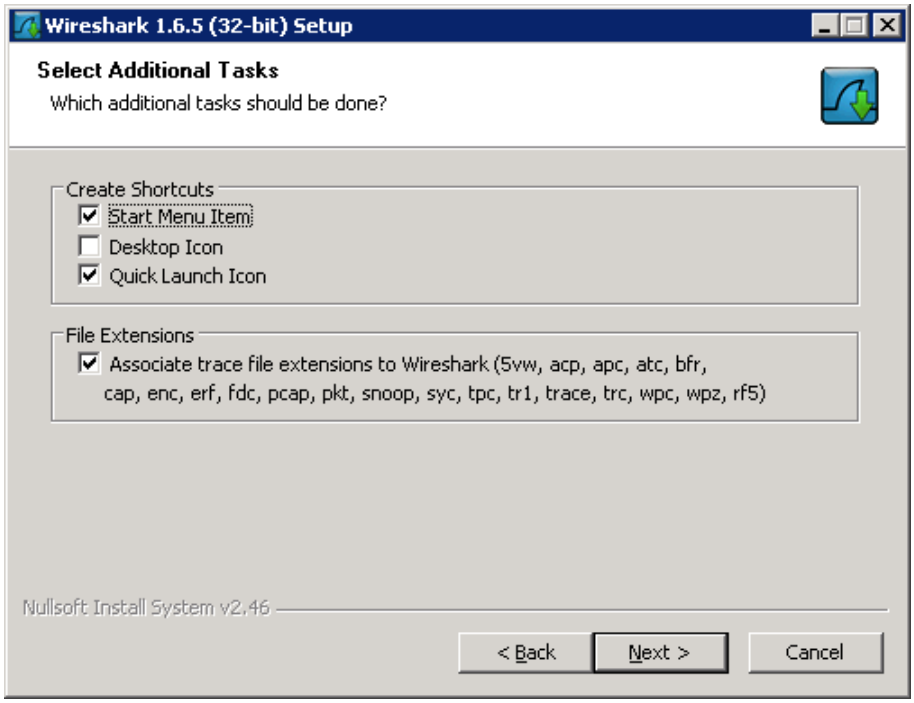
Referencia: Se empieza la instalación de *Wireshark*.



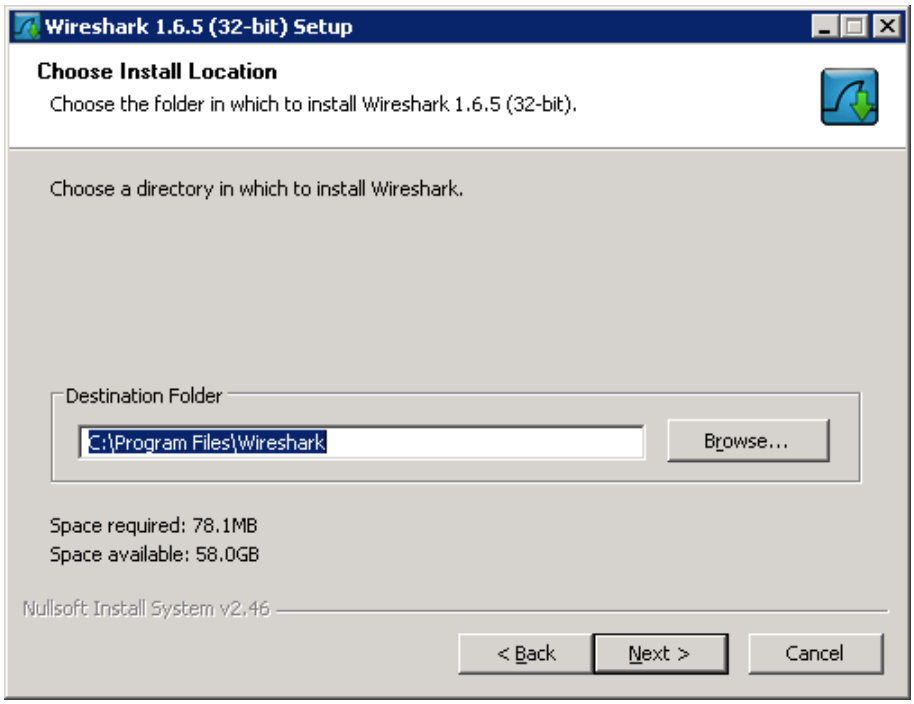
Referencia: Se da *click* en "*I Agree*" para aceptar la licencia de instalación del software *Wireshark*.



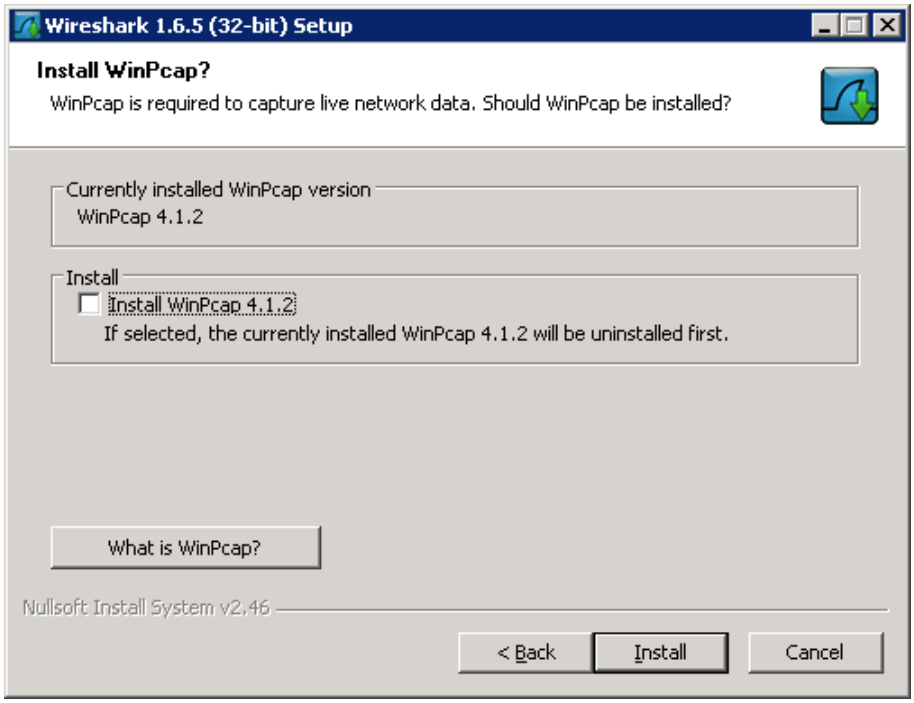
Referencia: Se escogen todos los componentes disponibles.



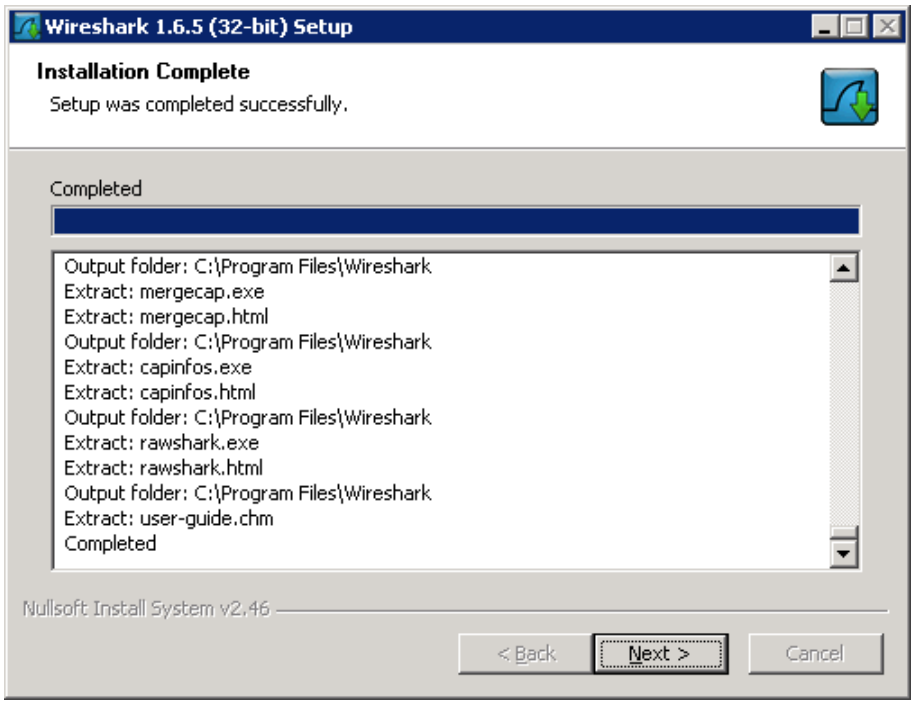
Referencia: Se dejan las opciones por defecto y se continúa con el proceso de instalación.



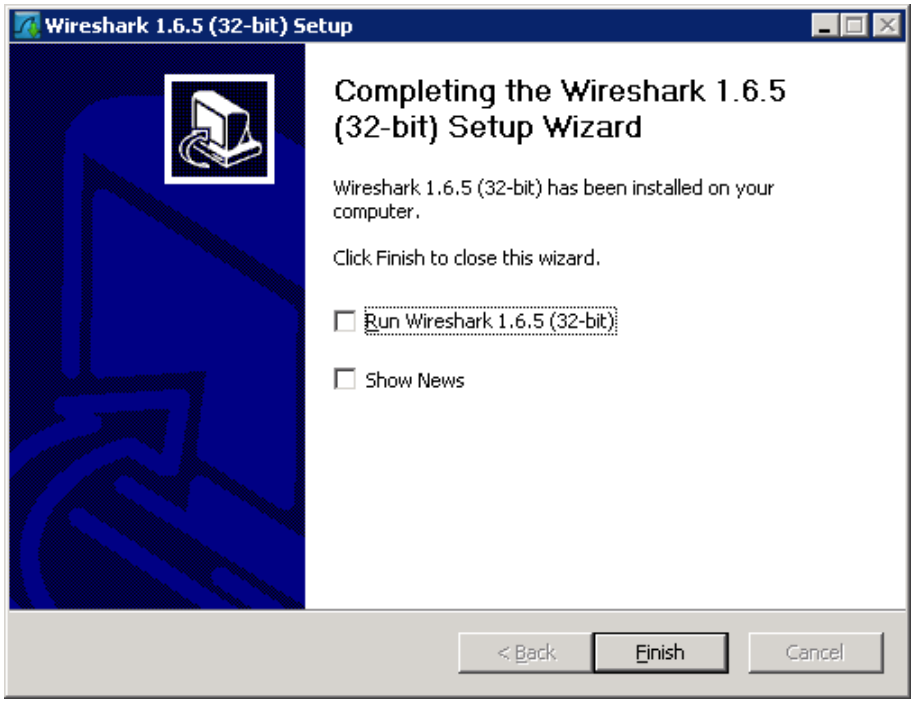
Referencia: Se escoge el directorio para la instalación de *Wireshark*.



Referencia: En esta opción como ya se tiene instalado *WinPcap* no se escoge esta opción y se continúa con la instalación.

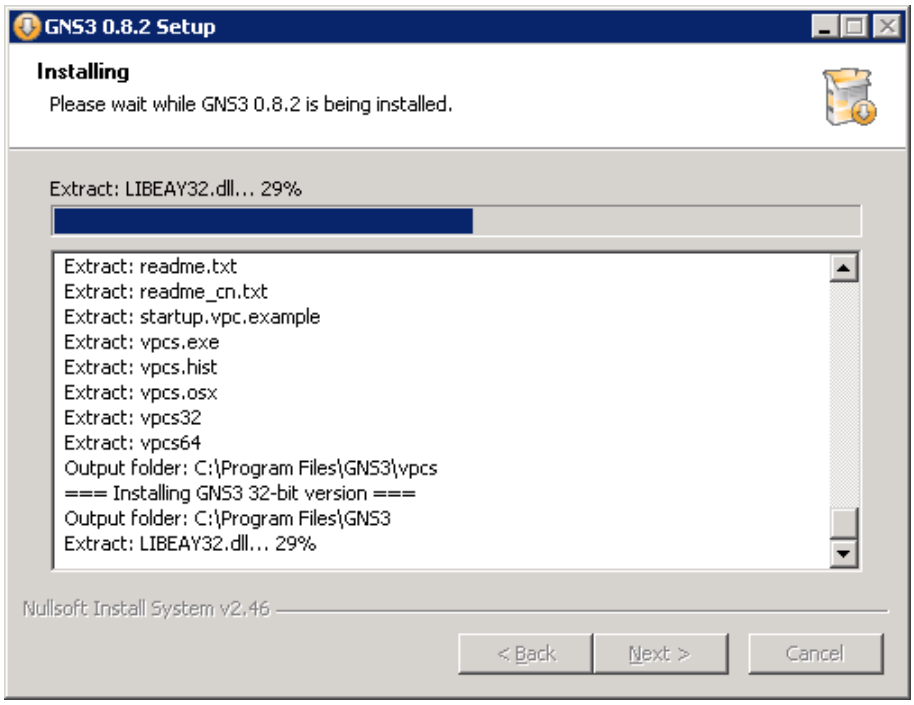


Referencia: Proceso de instalación de *Wireshark*.



Referencia: Finalización de la instalación de *Wireshark*.

Terminado este proceso se continúa con la instalación de la herramienta GNS3

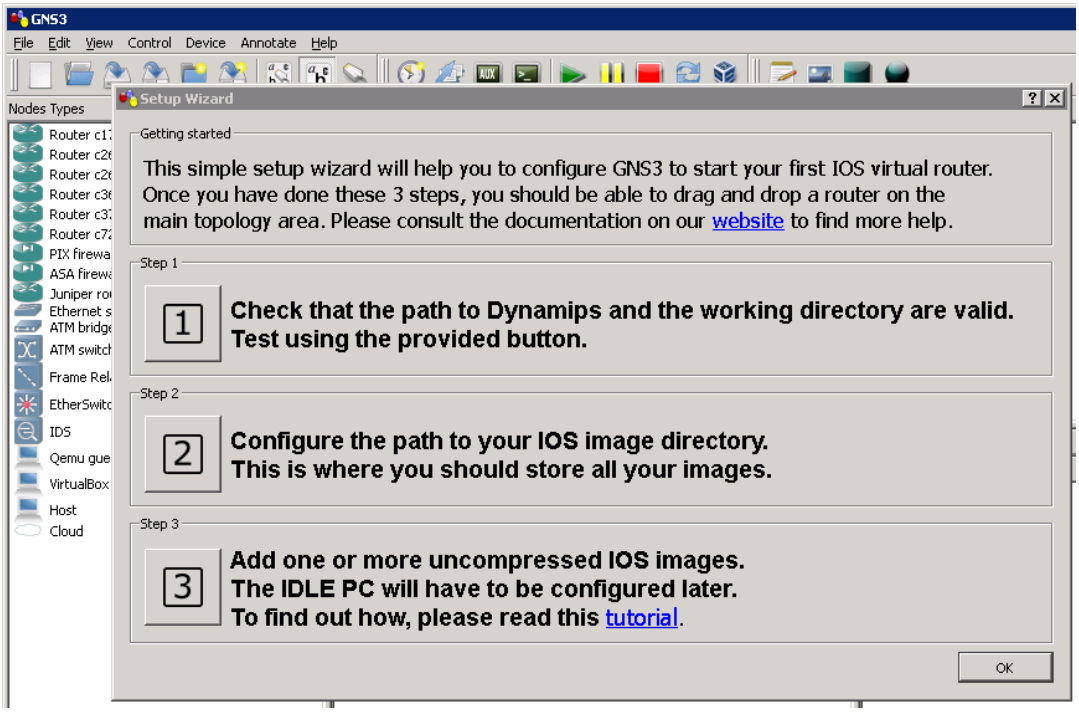


Referencia: Se empieza el proceso de instalación de GNS3



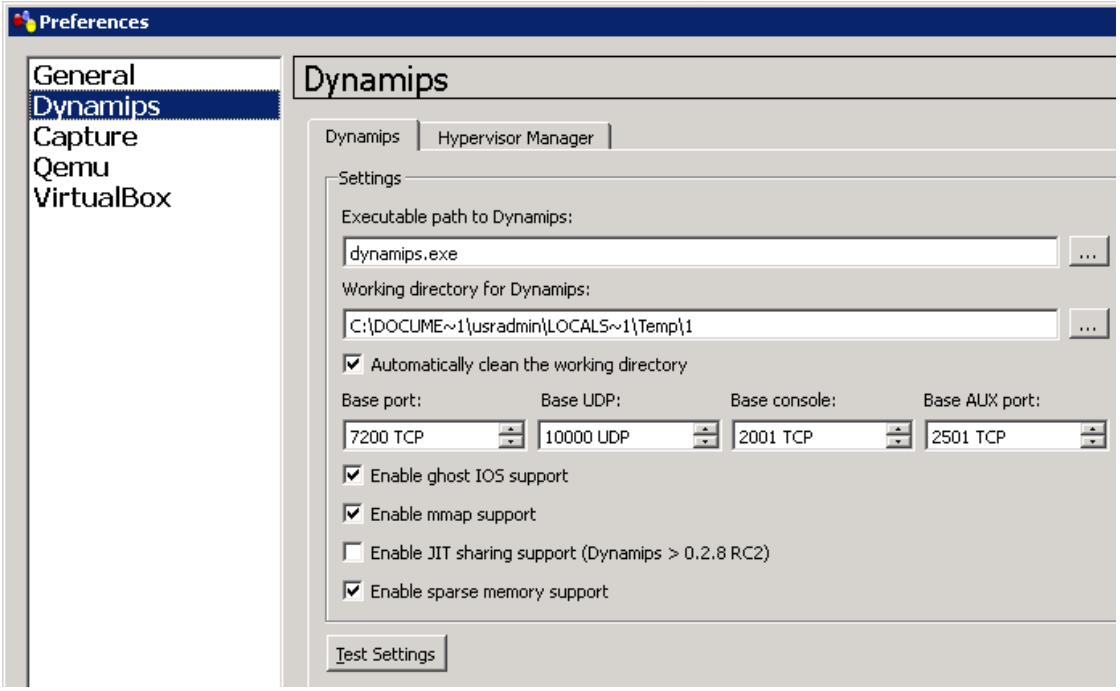
Referencia: El proceso concluye y se puede iniciar la herramienta.

Dentro de la ventana de bienvenida de la herramienta, se indica los pasos a seguir para la configuración de la herramienta.

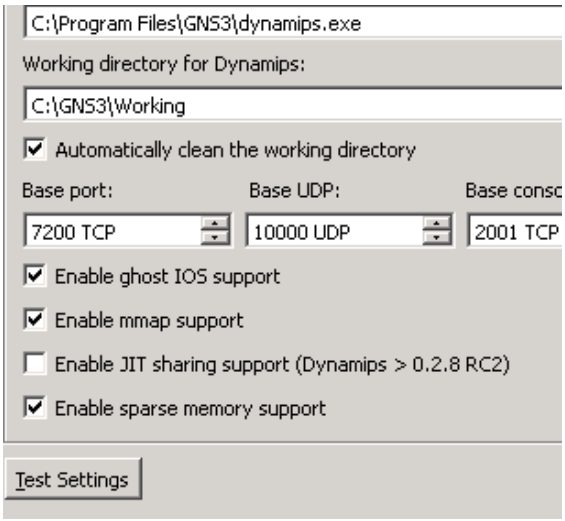


En el directorio C: del sistema operativo se crea una carpeta GNS3 para guardar las imágenes del IOS de los *routers* CISCO emulados. Para este trabajo se utilizará la imagen de un *Router* Cisco modelo 2600. Adicionalmente se crea una carpeta de trabajo de la forma C:\GNS3\Working para guardar la información de la simulación creada.

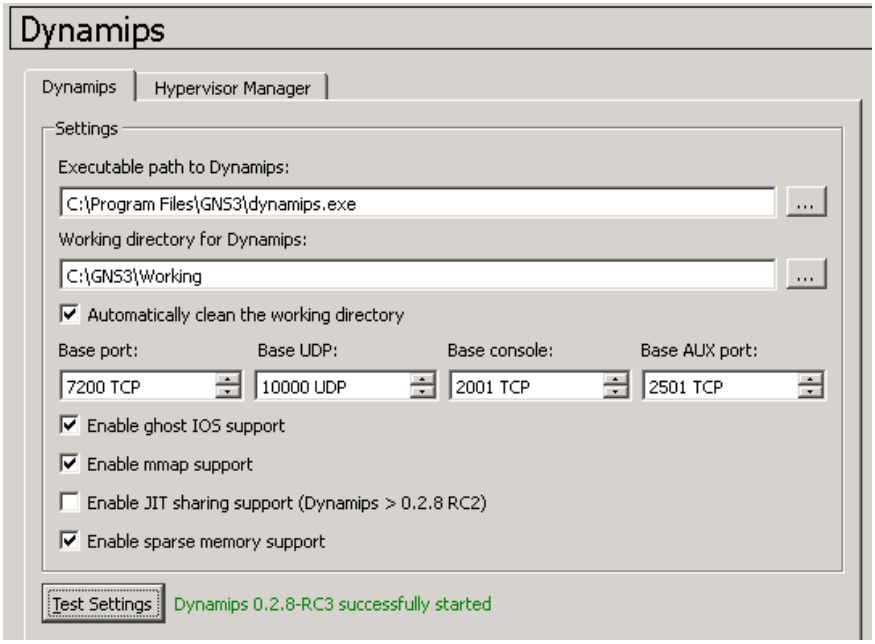
El siguiente paso consite en configurar la opción *Dynamips* desde el Menú *Edit – Preferences*.



Referencia: Configuración por defecto de *Dynamips*.

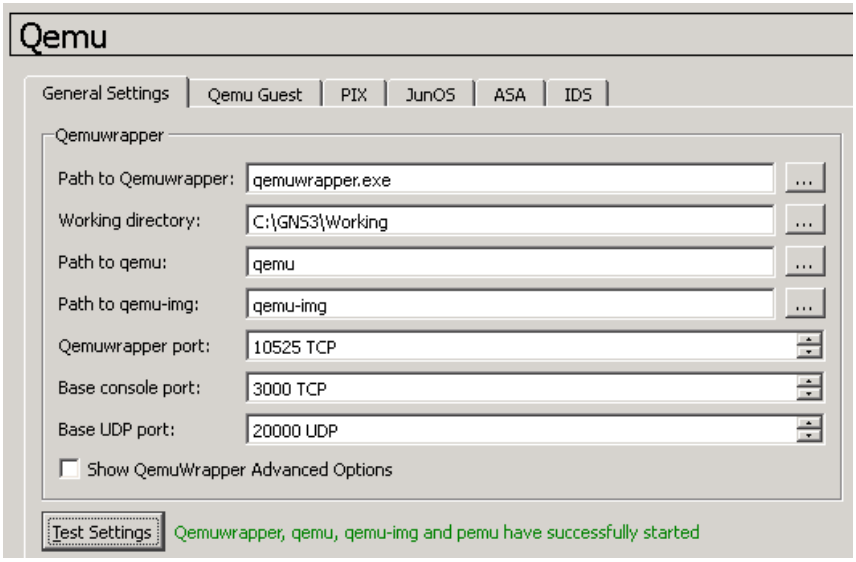


Referencia: Configuración después de los cambios

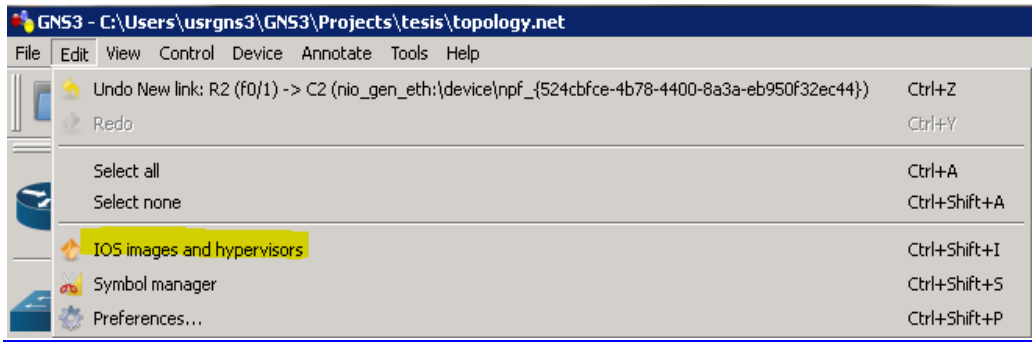


Referencia: Se realiza la prueba de la configuración "*Test Settings*" que es positiva

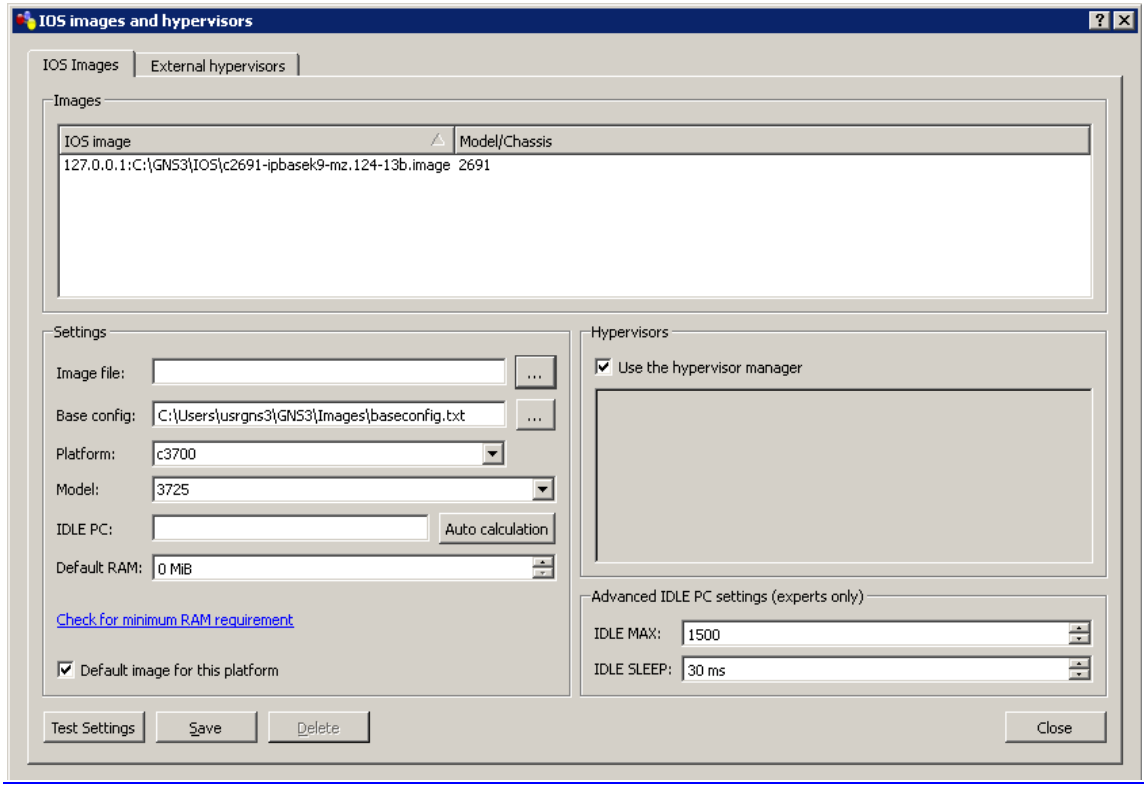
Ahora se configura la opción *Qemu* desde el Menú *Edit – Preferences*. En esta opción se ha cambiado el directorio de trabajo "*Working Directory*" y se realiza una prueba de "*Test Settings*" para verificar que la configuración es correcta.



Ahora el siguiente paso consiste en cargar la imagen del IOS del Router emulado



Referencia: Se debe escoger en el menú Edit -> IOS images and hypervisors



Referencia: se escoge la imagen del router emulado y se escoge la opción de "*Test Settings*" para verificar que la imagen es correcta y no tiene errores. Posterior a eso ya es posible realizar la emulación del router desde la ventana principal de GNS3.

ARCHIVOS DE CONFIGURACIÓN DE PANDORA FMS

Nota: Se indican los archivos en el orden que aparecen en el documento.

/etc/pandora/pandora_server.conf

```
##################################################################
#############
# Pandora FMS Server Parameters
# Pandora FMS, the Flexible Monitoring System.
# Version 4.0.2
# Licensed under GPL license v2,
# (c) 2003-2011 Artica Soluciones Tecnologicas
# http://www.pandorafms.com
# Please change it for your setup needs
##################################################################
#############

# Servername: Name of this server
# if not given, it takes hostname. It's preferable to setup one
# because machine name could change by some reason.

servername ecpucelx02

# incomingdir:  Defines directory where incoming data packets
are stored
# You could set directory relative to base path or absolute,
starting with /

incomingdir /var/spool/pandora/data_in

# log_file: Main logfile for pandora_server
# You could set file relative to base path or absolute, starting
with /

log_file /var/log/pandora/pandora_server.log

# Log file for Pandora FMS SNMP console. Its generated by
NetSNMP Trap daemon

snmp_logfile /var/log/pandora/pandora_snmptrap.log

# Error logfile: aux logfile for pandora_server errors (in
Daemon mode)
# You could set file relative to base path or absolute, starting
with /

errorlog_file /var/log/pandora/pandora_server.error

# daemon: Runs in daemon mode (background) if 1, if 0 runs in
foreground
# this could be also configured on commandline with -D option

# daemon 1

# dbengine: mysql, postgresql or oracle (mysql by default)
```

```
#dbengine mysql

# Database credentials. A VERY important configuration.
# This must be the same credentials used by your Pandora FMS
Console
# but could be different if your console is not running in the
same
# host than the server. Check your console setup in
/include/config.php

# dbname: Database name (pandora by default

dbname pandora

# dbuser:  Database user name (pandora by default)

dbuser pandora

# dbpass: Database password

dbpass yrdvnjwr

# dbhost: Database hostname or IP address

dbhost localhost

# dbport: Database port number
# Default value depends on the dbengine (mysql: 3306,
postgresql: 5432, oracle: 1521)

#dbport 3306

# By default, parent agent will not update
#update_parent 0

# verbosity: level of detail on errors/messages (0 default, 1
verbose, 2 debug.... 10 noisy)
# -v in command line (verbose) or -d (debug). Set this to 10
when try to locate problems and
# set to 0 or 1 on production enviroments.

verbosity 1

# Master Server, 1 if master server (normal mode), 0 for slave
mode (slave in multi-server setup)

master 1

# Activate Pandora SNMP console (depending on snmptrapd)
# only available on Pandora Network server

snmpconsole 1

# Activate (1) Pandora Network Server

networkserver 1
```

```
# Activate (1) Pandora Data Server

dataserver 1

# Activate (1) Pandora FMS Recon server

reconserver 1

# pluginserver : 1 or 0. Set to 1 to activate plugin server with
this setup
pluginserver 1

# Pandora FMS Plugin exec tool filepath (by default at /usr/bin)

plugin_exec /usr/bin/timeout

# predictionserver : 1 or 0. Set to 1 to activate prediction
server with this setup
# DISABLED BY DEFAULT
predictionserver 0

# wmiserver : 1 or 0. Set to 1 to activate WMI server with this
setup
# DISABLED BY DEFAULT
wmiserver 1

# Network timeout (in seconds) for timeout in network
connections for Network agents

network_timeout 4

# Server keepalive (in seconds)

server_keepalive 45

# Server Threshold: defines number of seconds of main loop (in
sec)

server_threshold 5

# Network threads: Do not set too high (~40). Each threads make
a network module check.

network_threads 4

# icmp_checks x : defines number of pings for each icmp_proc
module type. at least one of
# that ping should be 1 to report 1

icmp_checks 1

# tcp specific options :
# tcp_checks: number of tcp retries if first attempt fails.
# tcp_timeout: specific timeout for tcp connections
```

```
tcp_checks 1
tcp_timeout 30

# snmp specific options :
# snmp_checks: number of snmp request retries if first attempt
fails.
# snmp_timeout: specific timeout for snmp request.

snmp_checks 1
snmp_timeout 5

# snmp_proc_deadresponse 1 (default): Return DOWN if cannot
contact
# or receive NULL from a SNMP PROC module.

snmp_proc_deadresponse 1

# plugin_threads: Specify number of plugin server threads for
processing plugin calls

plugin_threads 1

# plugin_timeout: Specify number of seconds calling plugin exec
waiting for response
# after this time, call is aborted and result is "unknown".

plugin_timeout 15

# wmi_timeout : specific timeout for wmi request.

wmi_timeout 10

# wmi_threads: Specify number of WMI server threads for
processing WMI remote calls

wmi_threads 2

# recon_threads. Each thread will scan a different scantask.

recon_threads 2

# dataserver_threads: Number of threads for data server (XML
processing threads)

dataserver_threads 1

# mta_address: External Mailer (MTA) IP Address to be used by
Pandora FMS internal email capabilities
#mta_address 192.168.50.1

mta_address localhost

# mta_port: MTA port (default 25)
#mta_port 25
```

```
# mta_user MTA User (if needed for auth, FQD or simple user,
depending on your server)
#mta_user myuser@mydomain.com

# mta_pass MTA Pass (if needed for auth)
#mta_pass mypassword

# mta_auth MTA Auth system (if needed, support: LOGIN, PLAIN,
CRAM-MD5, DIGEST-MD)
#mta_auth LOGIN

# mta_from Email address that sends the mail, by default is
pandora@localhost
#           probably you need to change it to avoid problems
with your antispam
#mta_from Pandora FMS <pandora@mydomain.com>

# xprobe2: Optional package to detect OS types using advanced
TCP/IP
# fingerprinting tecniques, much more accurates than stadard
nmap.
# If not provided, nmap is used insted xprobe2

xprobe2 /usr/bin/xprobe2

# nmap: If provided, is used to detect OS type with recon server
using
# advanded OS fingerprint technique. Xprobe2 gives more accurate
results
# Nmap is also used to do TCP port scanning in detected host.

nmap /usr/bin/nmap

# snmpget: Needed to do SNMP checks. By default is on
/usr/bin/snmpget

snmpget /usr/bin/snmpget

# Location of the braa binary needed by the Enterprise SNMP
Server (/usr/bin/braa by default) (PANDORA FMS ENTERPRISE ONLY).
braa /usr/bin/braa

# Default group id for new agents created with Pandora FMS Data
Server
autocreate_group 2

# Set to 1 if want to autocreate agents with Pandora FMS Data
Server,
# set to 0 to disable

autocreate 1

# max_log_size: Specify max size of Pandora FMS server log file
(1MB by default). If
# log file grows above this limit, is renamed to
"pandora_server.log.old".
```

```
max_log_size 65536

# max_queue_files (250 by default)
# When server have more than max_queue_files in incoming
directory, skips the read
# the directory to avoid filesystem overhead.

max_queue_files 500

# Use the XML file last modification time as timestamp.
# use_xml_timestamp 1

# Pandora FMS will autorestart itself each XXX seconds, use this
if you experience problems with
# shutting down threads, or other stability problems.

# auto_restart 86400

# Pandora FMS will restart after restart_delay seconds on
critical errors.

# restart 0
# restart_delay 60

# More information about GIS Setup in
/usr/share/pandora_server/util/gis.README
# Flag to activate GIS (positional information for agents and
maps)
# by default it is desactivated
#activate_gis 0

# Radius of error in meters to consider two gis locations as the
same location.
#location_error 50

# Recon reverse geolocation mode [disabled, sql, file]
#   disabled    The recon task doesn't try to geolocate the ip
discovered.
#   sql         The recon task trys to query the SQL database to
geolocate the
#               ip discovered
#   file        The recon task trys to find the geolocation
information of the
#               ip discovered in the file indicated in the
#                recon_reverse_geolocation_file parameter

# recon_reverse_geolocation_mode disabled

# Recon reverse geolocation file. This is the database with the
reverse
# geolocation information using MaxMind GPL GeoLiteCity.dat
format).
#recon_reverse_geolocation_file
/usr/local/share/GeoIP/GeoIPCity.dat
```

```
# Radius (in meters) of the circle in where the agents will be
place randomly
# when finded by a recon task. Center of the circle is guessed
# by geolocating the IP.
#recon_location_scatter_radius 1000

# Pandora Server self-monitoring (embedded agent) (by default
disabled)
#self_monitoring 1

# Update parent from the agent xml
#update_parent 1
#
#
# This enable realtime reverse geocoding using Google Maps
public api.
# This requires internet access, and could have performance
penalties processing GIS
# information due the connetion needed to resolve all GIS input.
# NOTE: If you dont pay the service to google, they will ban
your IP in a few days.

# google_maps_description 1

# This enable realtime reverse geocoding using Openstreet Maps
public api.
# This requires internet access, and could have performance
penalties processing GIS
# information due the connetion needed to resolve all GIS input.
# You can alter the code to use a local (your own) openstreet
maps server.

# openstreetmaps_description 1

# Enable (1) or disable (0) Pandora FMS Event Web Server
(PANDORA FMS ENTERPRISE ONLY).
webserver 1

# Number of threads for the Web Server (PANDORA FMS ENTERPRISE
ONLY).
web_threads 1

# Enable (1) or disable (0) Pandora FMS Inventory Server
(PANDORA FMS ENTERPRISE ONLY).
inventoryserver 1

# Number of threads for the Web Server (PANDORA FMS ENTERPRISE
ONLY).
inventory_threads 1

# Enable (1) or disable (0) Pandora FMS Export Server (PANDORA
FMS ENTERPRISE ONLY).
exportserver 0

# Number of threads for the Export Server (PANDORA FMS
ENTERPRISE ONLY).
```

```
export_threads 1

# Enable (1) or disable (0) Pandora FMS Event Server (PANDORA
FMS ENTERPRISE ONLY).
eventserver 1

# Event Server event window in seconds (3600 by default)
(PANDORA FMS ENTERPRISE ONLY).
event_window 3600

# Enable (1) or disable (0) Pandora FMS Enterprise ICMP Server
(PANDORA FMS ENTERPRISE ONLY).
icmpserver 1

# Number of threads for the Enterprise ICMP Server (PANDORA FMS
ENTERPRISE ONLY).
icmp_threads 2

# Enable (1) or disable (0) Pandora FMS Enterprise SNMP Server
(PANDORA FMS ENTERPRISE ONLY).
snmpserver 0

# Number of threads for the Enterprise SNMP Server (PANDORA FMS
ENTERPRISE ONLY).
snmp_threads 2

# Block size for block producer/consumer servers, that is, the
number of modules
# per block (25 by default) (PANDORA FMS ENTERPRISE ONLY).
block_size 25
```

C:\Program Files\pandora_agent\pandora_agent.conf

```
# Base config file for Pandora FMS Windows Agent
# (c) 2006-2010 Artica Soluciones Tecnologicas
# Version 4.0.2

# This program is Free Software, you can redistribute it and/or
modify it
# under the terms of the GNU General Public Licence as published
by the Free Software
# Foundation; either version 2 of the Licence or any later
version
# This program is distributed in the hope that it will be
useful, but
# WITHOUT ANY WARRANTY, without ever the implied warranty of
MERCHANTABILITY or FITNESS
# FOR A PARTICULAR PURPOSE

# Edit this file to change your parameters or/and add your own
modules
# Any line with a # character at the first column will be
ignored (comment)

# General Parameters
# ==================

# NOTE: The variables $*$ will be substituted in the
installation wizard

server_ip 10.0.90.11
server_path /var/spool/pandora/data_in
temporal "C:\Program Files\pandora_agent\temp"

#include "C:\Archivos de
programa\pandora_agent\pandora_agent_alt.conf"
#broker_agent name_agent

# Agent uses your hostname automatically, if you need to change
agent name
# use directive agent_name (do not use blank spaces, please).
# This parameter is CASE SENSITIVE.

agent_name ecpucem01

#Parent agent_name
#parent_agent_name caprica

# address: Enforce to server a ip address to this agent
# You can also try to detect the first IP using "auto", for
example

#address auto
# or setting a fixed IP address, like for example:
#address 192.168.36.73

# Group assigned for this agent (descriptive, p.e: Servers)
```

```
group "Servers"

# This limits operation if temporal dir has not enough free
disk.
#temporal_min_size 1024

# Delay start execution X second before start to minonitoring
nothing
#startup_delay 30

# Interval is defined in seconds
interval 300

# tranfer_modes: Possible values are local, tentacle (default),
ftp and ssh.
transfer_mode tentacle
server_port 41121

# In case of using FTP or tentacle with password. User is always
"pandora"
#server_pwd pandora

# Debug mode do not copy XML data files to server.
# debug 1

# ODBC connections. Check documentation for more information.
# Configuring "ExampleDSN" DSN. Notice that this DSN connection
must be configured
# under Control panel -> Administrative tools -> ODBC -> DSN
# odbc_ExampleDSN_username UserNameForDsn
# odbc_ExampleDSN_password Password1234

# If set to 1 allows the agent to be configured via the web
console
# (only works on enterprise version).
# remote_config 1

# Set XML encoding (ISO-8859-1 by default).
#encoding ISO-8859-1

# If set to 1 start Drone Agent's Proxy Mode
# proxy_mode 1

# Max number of simmultaneus connection for proxy (by default
10)
# proxy_max_connection 10

# Proxy timeout (by default 1s)
# proxy_timeout 1

# Enable or disable XML buffer.
xml_buffer 0

# Secondary server configuration
# ==============================
```

```
# If secondary_mode is set to on_error, data files are copied to
the secondary
# server only if the primary server fails. If set to always,
data files are
# always copied to the secondary server.
#secondary_mode on_error
#secondary_server_ip localhost
#secondary_server_path /var/spool/pandora/data_in
#secondary_server_port 41121
#secondary_transfer_mode tentacle
#secondary_server_pwd mypassword
#secondary_server_ssl no
#secondary_server_opts

# Module Definition
# Check online documentation and module library at
http://pandorafms.org
# =================

# CPU Load using WMI
module_begin
module_name CPU Load
module_type generic_data
module_wmiquery SELECT LoadPercentage FROM Win32_Processor
module_wmicolumn LoadPercentage
module_description CPU Load (%)
module_min_warning 80
module_max_warning 90
module_min_critical 91
module_max_critical 100
module_end

# Number processes
module_begin
module_name Number processes
module_type generic_data
module_exec tasklist | gawk "NR > 3 {print$0}" | wc -l
module_description Number of processes running
module_min_warning 175
module_max_warning 249
module_min_critical 250
module_max_critical 300
module_end

# Free Memory
module_begin
module_name FreeMemory
module_type generic_data
module_freepercentmemory
module_description Free memory (%).
module_min_warning 21
module_max_warning 30
module_min_critical 0
module_max_critical 20
module_end
```

```
# Log events
module_begin
module_name System Events (TermService)
module_type async_string
module_logevent
module_description Log Events coming from Terminal Service
module_source System
module_application TermService
module_end

module_begin
module_name Security Events (Invalid Login)
module_type async_string
module_description Security log events for invalid login attempt
module_logevent
module_source Security
module_eventcode 529
module_end

# Check if Dhcp service is enabled
module_begin
module_name DHCP Enabled
module_type generic_proc
module_service Dhcp
module_description Check DCHP service enabled
module_end

#Antivirus monitoring
#This modules checks the antivirus is running on your system, if
there is and antivirus
#This module gets the last date the signature file was updated
and send this date to pandora.
module_begin
module_name Antivirus Last Update
module_type async_string
module_precondition =~ avguard.exe cmd.exe /c tasklist | grep
avguard.exe | gawk "{print $1}"
module_exec dir "%ProgramFiles%\Avira\AntiVir Desktop\aevdf.dat"
| grep aevdf.dat | gawk "{print $1\" \"$2}"
module_description Last update for Antivirus Signature file
module_end

# Example plugin to retrieve drive usage
module_plugin cscript.exe //B
"%ProgramFiles%\Pandora_Agent\util\df.vbs"

# Free space on disk C: (%)
#module_begin
#module_name FreeDiskC
#module_type generic_data
#module_freepercentdisk C:
#module_description Free space on drive C: (%)
#module_min_warning 31
#module_max_warning 40
#module_min_critical 0
#module_max_critical 30
```

```
#module_end

# CPU usage percentage
#module_begin
#module_name CPUUse
#module_type generic_data
#module_cpuusage all
#module_description CPU# usage
#module_min_warning 70
#module_max_warning 90
#module_min_critical 91
#module_max_critical 100
#module_end

# Free space on disk D: (%)
# module_begin
# module_name FreeDiskD
# module_type generic_data
# module_freepercentdisk D:
# module_description Free space on drive D: (%)
# module_end

# Sample of Windows inventory module (ONLY ENTERPRISE)!
#module_begin
#module_name Inventory
#module_interval 7
#module_type generic_data_string
#module_inventory CPU RAM CDROM Video HD Patches Software
Services NIC
#module_description Inventory
#module_end

# Example plugin to retrieve last 5 min events in log4x format
# module_plugin cscript.exe //B
"%ProgramFiles%\Pandora_Agent\util\logevent_log4x.vbs"
Aplicacion System 300

# Sample on how to get a value from registry
# This returns the last time user launch microsoft Windows
update
#module_begin
#module_name Windows_Update_LastRun
#module_type generic_data_string
#module_exec getreg LM
"SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto
Update" SetupWizardLaunchTime
#module_description Last date and time user launch microsoft
Windows update
#module_end

# Example of a remote TCP check
#module_begin
#module_name Google Port 80
#module_type generic_proc
#module_tcpcheck http://www.google.com
#module_port 80
```

```
#module_timeout 5
#module_description Check local port 80
#module_end

# Example of regexp matching
#module_begin
#module_name PandoraAgent_log
#module_type generic_data_string
#module_regexp C:\archivos de
programa\pandora_agent\pandora_agent.log
#module_description This module will return all lines from the
specified logfile
#module_pattern .*
#module_end

# Get processor time from Performance Counter (SPANISH only,
check your
# locale string) using the Windows Performance tool to
# identify proper PerCounter strings. Check documentation for
detailed steps.
#module_begin
#module_name Processor_Time
#module_type generic_data
#module_perfcounter \Procesador(_Total)\% de tiempo de
procesador
#module_end

# Example of module exec, used to know about the memory used by
pandora process
# grep.exe and gawk.exe are included in the util directory of
the agent.
#module_begin
#module_name PandoraFMS RAM
#module_type generic_data
#module_exec tasklist | grep Pandora | gawk "{ print $5 }" | tr
-d "."
#module_end

# Example of module exec, used get number of active terminal
services sessions
# Works on Windows 2003. In Windows XP the query.exe and
quser.exe files were
# moved to %WINDIR%\system32\dllcache. If XP, copy the exe to
%WINDIR%\system32
#module_begin
#module_name Active TS Sessions
#module_type generic_data_string
#module_exec query session | grep Activ | gawk "{ print $2 }"
|wc -l
#module_description Number of active TS Sessions
#module_end

# Example of watchdog process opening it if it gets closed
# NOTE: This need to enable "Service can interactuate with the
deskop" option
```

```
# in the Pandora FMS Service configuration (Windows Service
Control management).
#module_begin
#module_name TaskManager
#module_type generic_proc
#module_proc taskmgr.exe
#module_description This keeps taskmgr always running in the
system
#module_async yes
#module_watchdog yes
#module_start_command c:\windows\system32\taskmgr.exe
#module_end

# Example of watchdog service opening it if it gets closed
#module_begin
#module_name ServiceVNC_Server
#module_type generic_proc
#module_service winvnc
#module_description Service VNC Server watchdog/service
#module_async yes
#module_watchdog yes
#module_end

# Example UDP server to be able to execute remote actions such
# as starting or stopping process.
#udp_server 1
#udp_server_port 4321
#udp_server_auth_address 192.168.1.23
#process_firefox_start firefox
#process_firefox_stop killall firefox
#service_messenger 1

# Example of preconditions
#module_begin
#module_name Test Precondicion
#module_type generic_data
#module_precondition < 10 cmd.exe /c echo 5
#module_precondition > 10 cmd.exe /c echo 15
#module_precondition = 10 cmd.exe /c echo 10
#module_precondition != 10 cmd.exe /c echo 5
#module_precondition =~ 10 cmd.exe /c echo 10
#module_precondition (5,15) cmd.exe /c echo 10
#module_freepercentmemory
#module_description Precondition test module
#module_end

# Example of postconditions
#module_begin
#module_name Test Postcondicion
#module_type generic_data
#module_condition < 10 cmd.exe /c echo min >> c:\log.txt
#module_condition > 3 cmd.exe /c echo max >> c:\log.txt
#module_condition = 5 cmd.exe /c echo equal >> c:\log.txt
#module_condition != 10 cmd.exe /c echo diff >> c:\log.txt
#module_condition =~ 5 cmd.exe /c echo regexp >> c:\log.txt
#module_condition (3,8) cmd.exe /c echo range >> c:\log.txt
```

```
#module_exec echo 5
#module_description Postcondition test module
#module_end
```

/etc/pandora/pandora_agent.conf

```
# Base config file for Pandora FMS agents
# Version 4.0.2, GNU/Linux
# Licensed under GPL license v2,
# Copyright (c) 2003-2009 Artica Soluciones Tecnologicas
# http://www.pandorafms.com

# General Parameters
# ==================

server_ip  10.0.90.11
server_path     /var/spool/pandora/data_in
temporal /tmp
logfile /var/log/pandora/pandora_agent.log
#include /etc/pandora/pandora_agent_alt.conf
#broker_agent name_agent

# Interval in seconds, 300 by default
interval        300

# Debug mode only generate XML, and stop after first execution,
# and does not copy XML to server.
debug           0

# Optional. UDP Server to receive orders from outside
# By default is disabled, set 1 to enable
# Set port (41122 by default)
# Set address to restrict who can order a agent restart (0.0.0.0
= anybody)
#
udp_server 0
udp_server_port 41122
udp_server_auth_address 0.0.0.0

# By default, agent takes machine name
agent_name     ecpucelx02

#Parent agent_name
#parent_agent_name caprica

# Agent description
#description This is a demo agent for Linux

# Group assigned for this agent (descriptive, p.e: Servers)
group Servers

# address: Enforce to server a ip address to this agent
# You can also try to detect the first IP using "auto", for
example
#address auto
# or setting a fixed IP address, like for example:
#address 192.168.36.73

# Autotime: Enforce to server to ignore timestamp coming from
this
```

```
# agent, used when agents has no timer or it's inestable. 1 to
enable
# this feature
#autotime 1

# Timezone offset: Difference with the server timezone
#timezone_offset 0

# Agent position paramters
# Those parameters define the geographical position of the agent

# gis_exec: Call a script that returns a string with a fixed
# format of latitude,longitude,altitude
# i.e.: 41.377,-5.105,2.365

#gis_exec /tmp/gis.sh

# This sets the GIS coordinates as fixed values:
# latitude
#latitude 0
# longitude
#longitude 0
# altitude
#altitude 0

#GPS Position description
#position_description Madrid, centro

# By default agent try to take default encoding defined in host.
#encoding  UTF-8

# Listening TCP port for remote server. By default is 41121 (for
tentacle)
# if you want to use SSH use 22, and FTP uses 21.
server_port     41121

# Transfer mode: tentacle, ftp, ssh or local
transfer_mode tentacle

# Server password (Tentacle or FTP). Leave empty for no password
(default).
#server_pwd mypassword

# Set to yes/no to enable/disable OpenSSL support for Tentacle
(disabled by default).
#server_ssl no

# Extra options for the Tentacle client (for example,
server_opts "-v -r 5").
#server_opts

# delayed_startup defines number of seconds before start
execution
# for first time when startup Pandora FMS Agent
#delayed_startup 10
```

```
# Pandora nice defines priority of execution. Less priority
means more intensive execution
# A recommended value is 10. 0 priority means no Pandora CPU
protection enabled (default)
#pandora_nice 0

# Cron mode replace Pandora FMS own task schedule each XX
interval seconds by the use
# of old style cron. You should add to crontab Pandora FMS agent
script to use this mode.
# This is disabled by default, and is not recommended.  Use
Pandora FMS internal scheduler
# is much more safe
#cron_mode

# If set to 1 allows the agent to be configured via the web
console (Only Enterprise version)
#remote_config 1

# If set to 1 start Drone Agent's Proxy Mode
# proxy_mode 1

# Max number of simmultaneus connection for proxy (by default
10)
# proxy_max_connection 10

# Proxy timeout (by default 1s)
# proxy_timeout 1

# Number of threads to execute modules in parallel
#agent_threads 1

# User the agent will run as
#pandora_user root

# Enable or disable XML buffer.
# If you are in a secured environment and want to enable the XML
buffer you
# should consider changing the temporal directory, since /tmp is
world writable.
#xml_buffer 0

# Minimum available bytes in the temporal directory to enable
the XML buffer
#temporal_min_size 1024

# Secondary server configuration
# ==============================

# If secondary_mode is set to on_error, data files are copied to
the secondary
# server only if the primary server fails. If set to always,
data files are
# always copied to the secondary server.
#secondary_mode on_error
#secondary_server_ip localhost
```

```
#secondary_server_path /var/spool/pandora/data_in
#secondary_server_port 41121
#secondary_transfer_mode tentacle
#secondary_server_pwd mypassword
#secondary_server_ssl no
#secondary_server_opts

# Module Definition
# =================

# System information

# vmstat syntax depends on linux distro and vmstat command
version, please check before use it
module_begin
module_name cpu_user
module_type generic_data
module_interval 1
module_exec vmstat 1 2 | tail -1 | awk '{ print $13 }'
module_max 100
module_min 0
module_description User CPU Usage (%)
module_min_warning 70
module_max_warning 90
module_min_critical 91
module_max_critical 100
module_end

#Get load average
module_begin
module_name Load Average
module_type generic_data
module_exec cat /proc/loadavg | cut -d' ' -f1
module_description Average process in CPU (Last minute)
module_end

#Get free memory in MB
module_begin
module_name Cache mem free
module_type generic_data
module_exec free -m | grep buffers/cache | awk '{print $4}'
module_description Free cache memory in MB
module_min_warning 500
module_max_warning 600
module_min_critical 100
module_max_critical 499
module_end

#Count total number of processes
module_begin
module_name proctotal
module_type generic_data
module_exec ps -A | tail --lines=+5 | wc -l
module_description Total processes
module_min_warning 150
module_max_warning 249
```

```
module_min_critical 250
module_max_critical 300
module_end

# Process information

module_begin
module_name sshDaemon
module_type generic_proc
module_exec ps -Af | grep sshd | grep -v "grep" | wc -l
module_description Check ssh service
module_end

# Async data example

module_begin
module_name LastLogin
module_type async_string
module_exec last | head -1
module_description Monitor last user loggin
module_end

# Module that get the number or cron file
# This module uses a precondition, if cron is running the
# module will check the folder /etc/cron.d to get the number of
files
module_begin
module_name Cron task files
module_type async_string
module_precondition =~ .*cron.* ps aux | grep cron
module_exec ls -l /etc/cron.d | awk 'NR>1 {print $0}' | wc -l
module_description Number of cron task files
module_end


# Plugin example

# This plugin detects all disk partitions and monitor the free
spaces

module_plugin pandora_df

# This parses /var/log/syslog file, under the module name
"syslog"
# And search for "ssh" string into it, sending only that
information.

module_plugin grep_log /var/log/syslog Syslog ssh

# Get disk space free in MB
#module_begin
#module_name disk_root_free
#module_type generic_data
#module_exec df -kh / | tail -1 | awk '{ print 100-$5 }'
#module_max 100
#module_min 0
```

```
#module_description Free disk Percentage of root partition
#module_min_warning 70
#module_max_warning 90
#module_min_critical 91
#module_max_critical 100
#module_end

# This module uses postprocess feature to unit conversion
#module_begin
#module_name memused
#module_type generic_data
#module_exec free -k | grep buffers/cache | awk '{print $3}'
#module_postprocess 0,000976
#module_description Used memory in KB postprocessed to be in MB
#module_end

# Plugin for inventory on the agent (Only Enterprise)

# module_plugin inventory 1 cpu ram video nic hd cdrom software
init_services filesystem users process ip route

# Example of preconditions
#module_begin
#module_name Test Precondicion
#module_type generic_data
#module_precondition < 10 echo 5
#module_precondition > 10 echo 15
#module_precondition = 10 echo 10
#module_precondition != 10 echo 5
#module_precondition =~ 10 echo 10
#module_precondition (5,15) echo 10
#module_freepercentmemory
#module_description Precondition test module
#module_end

# Example of postconditions
#module_begin
#module_name Test Postcondicion
#module_type generic_data
#module_condition < 10 echo min >> /tmp/log.txt
#module_condition > 3 echo max >> /tmp/log.txt
#module_condition = 5 echo equal >> /tmp/log.txt
#module_condition != 10 echo diff >> /tmp/log.txt
#module_condition =~ 5 echo regexp >> /tmp/log.txt
#module_condition (3,8) echo range >> /tmp/log.txt
#module_exec echo 5
#module_description Postcondition test module
#module_end
```

/etc/snmp/snmpd.conf

```
#Informacion de contacto para consultas SNMP
#este archivo de configuracion contiene una configuracion basica
syslocation Universidad Catolica del Ecuador
syscontact  Santiago Silva "sdsilva@outlook.com"

#se definen las IP's por las que se van a escuchar los traps
SNMP
rocommunity public 127.0.0.1
rocommunity public 10.0.90.11

master      agentx
```

/etc/snmp/snmptrapd.conf

```
##############################################################################
#
# EXAMPLE-trap.conf:
#   An example configuration file for configuring the Net-SNMP
snmptrapd agent.
#
##############################################################################
#
# This file is intended to only be an example.  If, however, you
want
# to use it, it should be placed in /etc/snmp/snmptrapd.conf.
# When the snmptrapd agent starts up, this is where it will look
for it.
#
# All lines beginning with a '#' are comments and are intended
for you
# to read.  All other lines are configuration commands for the
agent.

#
# PLEASE: read the snmptrapd.conf(5) manual page as well!
#
authCommunity log public
```

/etc/default/snmpd

```
# This file controls the activity of snmpd and snmptrapd

# Don't load any MIBs by default.
# You might comment this lines once you have the MIBs
downloaded.
export MIBS=

# snmpd control (yes means start daemon).
SNMPDRUN=yes

# snmpd options (use syslog, close stdin/out/err).
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux -p
/var/run/snmpd.pid'

# snmptrapd control (yes means start daemon).  As of net-snmp
version
# 5.0, master agentx support must be enabled in snmpd before
snmptrapd
# can be run.  See snmpd.conf(5) for how to do this.
TRAPDRUN=yes

# snmptrapd options (use syslog).
TRAPDOPTS='-Lsd -p /var/run/snmptrapd.pid'

# create symlink on Debian legacy location to official RFC path
SNMPDCOMPAT=yes
```

/etc/smsd.conf

```
# Example smsd.conf. Read the manual for a description

devices = GSM1
logfile = /var/log/smsd.log
loglevel = 7

[GSM1]
device = /dev/ttyUSB2
incoming = yes
#pin = 1111
```

ARCHIVOS DE CONFIGURACIÓN DE LOS ROUTERS SIMULADOS

Router _UIO_puce001

```
!
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname _UIO_puce001
!
boot-start-marker
boot-end-marker
!
enable password cisco
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
no ip domain lookup
!
!
!
!
!
ip tcp synwait-time 5
!
!
interface FastEthernet0/0
 ip address 192.168.20.10 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 10.0.90.6 255.255.255.0
 duplex auto
 speed auto
!
router ospf 2
 log-adjacency-changes
 network 10.0.90.0 0.0.0.255 area 0
 network 192.168.20.0 0.0.0.255 area 0
!
!
no ip http server
no ip http secure-server
!
access-list 60 permit 10.0.90.11
```

```
access-list 60 permit 10.0.90.53
snmp-server community public RO
snmp-server community restPUCE RW 60
snmp-server enable traps snmp authentication linkdown linkup
coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps ds1
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps flash insertion removal
snmp-server enable traps ds3
snmp-server enable traps envmon
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
snmp-server enable traps ds1-loopback
snmp-server enable traps atm subif
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps entity
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-
trans-change
snmp-server enable traps ospf cisco-specific state-change
shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change
shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps pim neighbor-change rp-mapping-change
invalid-pim-message
snmp-server enable traps pppoe
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps syslog
snmp-server enable traps l2tun session
snmp-server enable traps vtp
snmp-server host 10.0.90.11 inform version 2c public
snmp-server host 10.0.90.11 public
!
!
control-plane
```

```
!
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0
 password cisco
 login
line vty 1 4
 login
!
!
end
```

Router _UIO_puce002

```
!
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname _UIO_puce002
!
boot-start-marker
boot-end-marker
!
enable password cisco
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
no ip domain lookup
!
!
!
!
!
ip tcp synwait-time 5
!
!
interface FastEthernet0/0
 ip address 192.168.20.11 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 10.0.91.6 255.255.255.0
 duplex auto
 speed auto
!
router ospf 2
 log-adjacency-changes
 network 10.0.91.0 0.0.0.255 area 0
 network 192.168.20.0 0.0.0.255 area 0
!
!
no ip http server
no ip http secure-server
!
access-list 60 permit 10.0.90.11
access-list 60 permit 10.0.90.53
snmp-server community public RO
snmp-server community restPUCE RW 60
```

```
snmp-server enable traps snmp authentication linkdown linkup
coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps ds1
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps flash insertion removal
snmp-server enable traps ds3
snmp-server enable traps envmon
snmp-server enable traps icsudsu
snmp-server enable traps isdn call-information
snmp-server enable traps isdn layer2
snmp-server enable traps isdn chan-not-avail
snmp-server enable traps isdn ietf
snmp-server enable traps ds0-busyout
snmp-server enable traps ds1-loopback
snmp-server enable traps atm subif
snmp-server enable traps cnpd
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps entity
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-
trans-change
snmp-server enable traps ospf cisco-specific state-change
shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change
shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps pim neighbor-change rp-mapping-change
invalid-pim-message
snmp-server enable traps pppoe
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps syslog
snmp-server enable traps l2tun session
snmp-server enable traps vtp
snmp-server host 10.0.90.11 inform version 2c public
snmp-server host 10.0.90.11 public
!
!
control-plane
!
!
!
```

```
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0
 password cisco
 login
line vty 1 4
 login
!
!
end
```