Universidad Carlos III de Madrid

Ingeniería Técnica de Telecomunicaciones:
Sistemas de Telecomunicación

PROYECTO FIN DE CARRERA

DESCON 2

Escuela Politécnica Superior
Octubre de 2009

Autor: Manuel Fernández Fernández
Tutor: Rafael Calzada Pradas

Agradecimientos

Durante todo el tiempo que he estado realizando este Proyecto Fin de Carrera y mis años en la Universidad, he conocido a muchas personas durante esta etapa con las que he compartido experiencias inolvidables.

Algunas de ellas como a mi novia Virgi, a la que le agradezco con todo mi corazón la ayuda y la comprensión que ha tenido conmigo durante esta época, en las que hemos vivido infinidad de situaciones.

También conocí a tres maravillosos amigos como Miguel Baza, Manuel Martinez e Ivan Fernández que me enseñaron como se puede aprender y pasárselo bien a la vez.

Quiero dar las gracias a mi tutor Rafael Calzada al que he de agradecer todas sus enseñanzas y su paciencia que han hecho que este Proyecto pueda salir adelante, así como al departamento de Informática de la UC3M, del que he sido becario.

No podía tampoco olvidar a toda mi familia y amigos que siempre han estado apoyándome, pero en especial, ya que han sido los que mas han tenido que "soportar" mis quejas, a mi hermano Javi y a mis padres.

Gracias a todos

Resumen

Este proyecto tiene como objetivo la realización de una aplicación web que facilite la desconexión temprana de los equipos comprometidos en redes académicas como la de la Universidad Carlos III. Esta red es un entorno abierto en el que se pretende, reducir al mínimo las restricciones de conectividad para facilitar las labores de investigación y docencia.

Para conseguir de dicho objetivo, este proyecto propone tres puntos de actuación :

- Agregación de la información de seguridad.
- Estimación del estado de seguridad y detección temprana de los equipos conectados de la red.
- Cuarentena de equipos comprometidos.

Para la realización del proyecto la implementación de la aplicación Web usa el lenguaje de programación JAVA. Esta aplicación tendrá las siguientes características:

Facilidad de uso de la aplicación y facilidad de acceso a la información.

Palabras clave:

Agregación de información de seguridad, detección temprana, cuarentena de equipos comprometidos, aplicación web.

Índice general

Índice de figuras

Capítulo 1

Introducción

Este proyecto surgió por la necesidad de conseguir un entorno abierto y seguro en redes académicas, como la de la Universidad Carlos III, que tienen la necesidad de que haya pocas restricciones de conectividad, para facilitar las labores de investigación y docencia. Además, los usuarios de este tipo de redes necesitan tener control absoluto de los equipos que emplean para conectarse a la red, hasta el punto de instalar el sistema operativo y las aplicaciones que consideren necesarias o interesantes. En algunos casos los usuarios utilizan sus equipos personales, generalmente dispositivos móviles (portátiles, pdas, etc), para acceder a los contenidos y servicios que la universidad les ofrece, haciendo que la heterogeneidad sea la nota dominante en cuanto a las combinaciones de acceso a la red. Esta situación provoca que aumente la exposición a ataques a equipos conectados a la red la Universidad de ahora en adelante UC3M.

Para ello, este proyecto denominado Descon2, intenta conseguir la mejor relación entre ofrecer un ambiente abierto y seguro, entendiéndose como tal un sistema que permita repeler los ataques desde el exterior, proteger al usuario de otros usuarios, así como de sí mismo. Ya que en una red de este tamaño no todos los usuarios tienen el mismo nivel de consciencia sobre la importancia de la seguridad informática.

Para lograr esto, Descon2 pretende ofrecer una agregación de toda la información de seguridad disponible, para así conseguir una detección temprana de los equipos comprometidos y facilitar su posterior puesta en cuarentena.

Tras analizar el problema sobre como abordar la gestión de seguridad de los dispositivos conectados a la entidad corporativa se determinó que habría dos formas de afrontarlo:

1.-Enfoque preventivo: Consiste en prevenir la infección mediante la obligación de disponer elementos de seguridad (actualizaciones, anti-virus y cortafuegos) actualizados y correctamente configurados en los equipos de los usuarios y servidores corporativos.

2.-Enfoque reactivo: Cualquier equipo tiene permitido el acceso a la red, pero si se detecta algún comportamiento anómalo se le denegara el acceso hasta que se corrija dicho comportamiento.

Es obvio que es "mejor prevenir que curar", por lo que siempre deberá prevenirse como nos indica la primera opción, pero también es cierto que esto no puede garantizarse en todos los casos, debido a los requerimientos indicados anteriormente, como los de las labores de docencia y de investigación, así como el distinto nivel de consciencia sobre la seguridad. Además la prevención implica la instalación de programas con la posibilidad de que no estén disponibles para todas las plataformas, lo que supone un problema ya que no todos los equipos tendrán el mismo nivel de seguridad, y además no podrá auditarse dicho nivel.

1.1. Objetivos principales de Descon2

Descon2 es un proyecto que nace para solucionar las carencias de seguridad que hay en este tipo de redes académicas y para conseguirlo se marca los siguientes objetivos:

- Agrupar toda la información de seguridad, para poder detectar situaciones de riesgo con la mayor rapidez posible.
- Analizar toda la información de seguridad y estimar el estado de seguridad.

- Facilitar la puesta en una red de cuarentena a los equipos comprometidos y darles los medios necesarios (guías, actualizaciones, etc) para que solucionen su problema de seguridad y puedan conectarse de manera normal a la red UC3M.

- Dar la posibilidad a los administradores de las subredes existentes de conocer el estado de compromiso de los equipos de su red.

Todos estos objetivos son para conseguir el objetivo principal, que es conseguir una red robusta en cuanto a seguridad informática se refiere, con las mínimas restricciones de acceso de los usuarios para que el personal docente e investigador pueda realizar su trabajo sin problemas añadidos.

1.2. Estructura del documento

Capitulo 1. Introducción. Explicación del contexto y los motivos del nacimiento de este proyecto y los objetivos que se pretenden conseguir.

Capitulo 2. Estado de la Cuestión. Breve descripción de las distintas soluciones que ofrece el mercado al problema que se enfrenta este proyecto fin de carrera.

Capitulo 3. Objetivos y requisitos del sistema. Explicación formal y detallada de los objetivos buscados así como de los requisitos establecidos para conseguirlos.

Capitulo 4 . Diseño e implementación del portal. En este capitulo se describe el entorno de desarrollo y la implementación de cada uno de los componentes del sistema.

Capitulo 5. Evaluación y pruebas En este capitulo, se podrá ver los casos en los que se han desarrollado las pruebas y un análisis de los resultados de estas pruebas.

Capitulo 6. Fases y duración del proyecto. Organización del proyecto durante la realización de este en el que se mostrara detalles como las fases, la duración de las tareas y los costes de este proyecto.

Capitulo 7. Manual de usuario. Muestra del manual de usuario de la aplicación Web que pretende servir de ayuda a administradores y usuarios del sistema.

Capitulo 8. Lineas futuras. Futuras lineas de trabajo como continuar Descon2 para añadirle nuevas funcionalidades en el futuro.

Anexos Esta memoria se complementa con una serie de Anexos que recogen los siguientes temas:

- A. Protocolo SNMP.

  En este anexo se recogen los conceptos fundamentales que deben tenerse en cuenta a la hora de diseñar un sistema de gestión de red.

- B. SSL en Tomcat.

  Se recoge la información, de configuración del protocolo de capa de conexión segura(SSL) en el servidor de aplicaciones Web Tomcat.

- C. Cuestionario Usabilidad.

  Cuestionario que sirve para medir cualitativamente y cuantitativamente la Usabilidad de la interfaz Web de este proyecto.

Capítulo 2

Estado de la cuestión

2.1. Seguridad en redes de comunicación

2.1.1. Orígenes

Durante siglos la historia de la humanidad ha estado vinculada a la propagación de epidemias. Estas provocaron innumerables muertes en todo el mundo, y de esta forma lograron cambiar el poder de países así como de cambiar el curso de guerras. Con el nacimiento de la informática, en el siglo XX, el problema surgió en esta nueva disciplina. Nacieron los virus, troyanos, gusanos y todo tipo de software malintencionado (o malware) que tienen como objetivo infiltrarse o dañar un sistema sin el consentimiento de su dueño.

En los inicios la propagación de los virus en ordenadores fue limitada en gran parte a la transmisión manual a través de soportes físicos como diskettes y CD's. Por lo tanto era mucho más fácil detectar y eliminar virus, antes de que pudieran hacer cualquier daño de gran escala a un sistema informático.

2.1.2. Propagación de virus en redes empresariales

El gran auge de Internet, de las redes empresariales o académicas ha cambiado la situación totalmente. Un malware puede ahora propagarse muy rápidamente a partir de un sistema a otro a través de una red.

La preocupación por la propagación del malware empezó en 1988 cuando Robert Tappan Morris, difundió un virus a través de ArpaNet, (precursora de Internet) logrando infectar 6.000 servidores conectados a la red en solo unos pocos minutos. Años después en 2001 la difusión de los gusanos Code Red (código Rojo) y Nimda infectaron millones de ordenadores y servidores provocando un coste económico muy alto tanto a empresas como a la sociedad.

Estos sucesos provocaron un pánico ante la falta de seguridad y estabilidad de los sistemas. Se empezaron a analizar las causas y se observó que se podía establecer una analogía con la propagación de las epidemias en los humanos. Este hecho aportó que los estudios matemáticos establecidos en el estudio del control de enfermedades infecciosas, sean validos también para la ciencia de la seguridad informática.

2.1.3. Modelado de las epidemias

Una epidemia se puede modelar como un proceso estocástico con una población cerrada y homogénea. Esta población se divide en tres clases:

- **Susceptibles:** Miembro de la población con una probabilidad de ser infectado.
- **Infecciosos:** Miembro que tiene la infección y la puede propagar a la población.
- **Eliminados o Inmunes:** Miembros eliminados de de la población o miembros que no pueden ser contagiados.

  En términos informáticos esto quiere decir que los dispositivos estén:

  - Actualizados, osea con todas las actualizaciones o parches instalados.
  - Infectados pero corregidos, mediante una re instalación o limpieza.
  - Inmunes ya que la infección no les afecte debido:
    - Utilizan un sistema operativo diferente.
    - No tienen instalada la aplicación vulnerable.

Este modelo cuyo esquema se muestra en la figura 2.1 se basa en tres probabilidades para el cambio de estados:

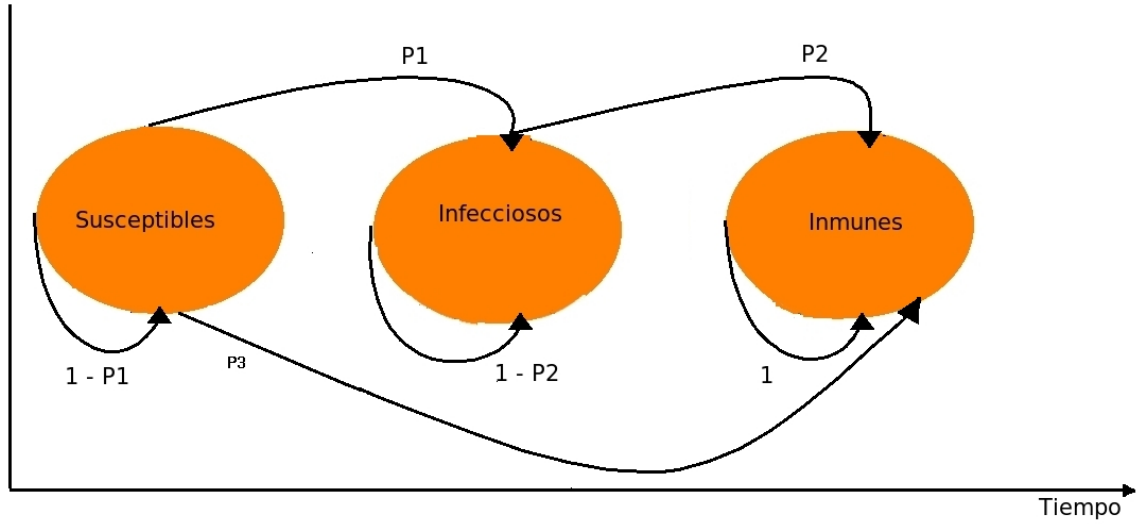


Figura 2.1: Transiciones entre estados

1. P1 es la probabilidad de que un sistema susceptible pase al estado de infectado.
2. P2 es la probabilidad de que el sistema infectado, pase al estado inmune o eliminado.
3. P3 probabilidad de que un sistema susceptible pase a inmune. Por ejemplo que un fabricante publique un parche a una vulnerabilidad que no ha sido explotada en este sistema.

Si no se cambian los estados las probabilidades son las complementarias. En el caso de un sistema sea inmune para una infección, este permanece en el estado de inmunidad para siempre (probabilidad = 1).

Al analizar los tres estados se llegó a la conclusión que cuando un equipo está en el estado de infección, supone un riesgo muy alto para los demás miembros de la población. Por ello se planteo la teoría de cuarentena dinámica, (Weibo Gong [1]), esta se basa en el principio de "**culpabilidad hasta que se demuestre lo contrario** ".

La cuarentena dinámica consiste en cambiar los equipos que generan un trafico sospechoso en la red de explotación a una red denominada de cuarentena para que el personal de seguridad de la red lo inspeccione lo mas rápido posible. Si el personal de seguridad no ha inspeccionado el equipo después de un tiempo breve el equipo vuelve a la red de

explotación. De esta manera se evita que equipos no comprometidos estén bloqueados en la red de cuarentena durante un tiempo demasiado largo.

La cuarentena dinámica sirve para solucionar los compromisos de seguridad del sistema sin poner en riesgo a la mayoría de la población. También sirve para intentar disminuir la velocidad de propagación de las epidemias para así poder minimizar daños. En la figura 2.2 obtenida del articulo de Weibo Gong [1] se muestra una gráfica ilustrativa de como disminuye la velocidad de propagación mediante el uso de cuarentena dinámica.

La importancia de la cuarentena dinámica es que permite frenar la velocidad de la propagación, dando de tiempo que:

- Publicación del parche o actualización.
- Detectar el malware mediante un antivirus.
- Bloquear la propagación por otros medios como por ejemplo mediante un "cortafuegos".

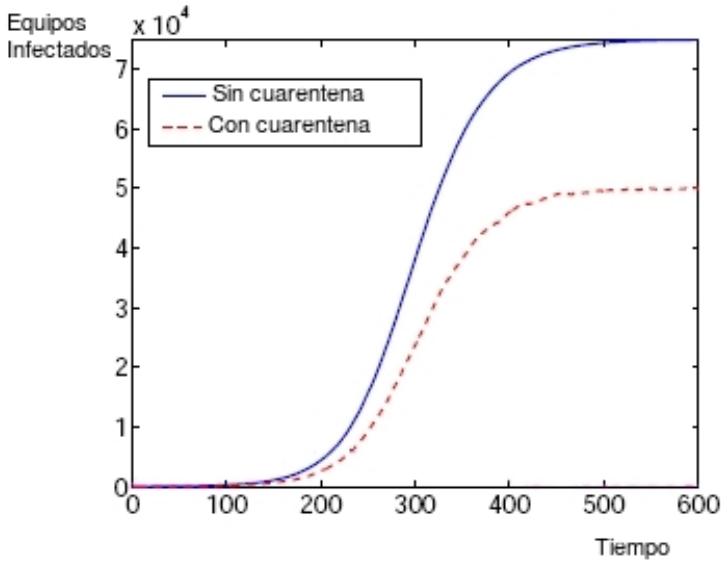


Figura 2.2: Ejemplo cuantitativo de como disminuye la velocidad de propagación en sistemas con cuarentena.

2.2. Soluciones existentes en el mercado

Antes de empezar a diseñar el sistema desde cero, se analizaron las distintas soluciones que ofrece el mercado a este problema planteado y se observaran las ventajas e inconvenientes de los distintos sistemas.

Dentro de los productos evaluados, las dos primeras soluciones comerciales se basan en un enfoque preventivo, mientras que la ultimas analizadas están basadas en software libre y tienen un enfoque reactivo.

2.2.1. Microsoft Connection Manager

Se trata de una plataforma gratuita de Microsoft, que ahora se distribuye bajo la denominación de NAP (Network Access Protección)[2][3], cuyo objetivo es asegurar que cualquier equipo terminal conectado a la red cumpla con el modelo de seguridad definido por la organización que use este sistema.

El nivel de acceso a la red viene determinado por nivel de cumplimiento del modelo de seguridad. Así, podemos configurar que todos los equipos que tengan un nivel de seguridad aceptable (por ejemplo estén actualizados y con el patrón de anti-virus actualizado a día de ayer), puedan acceder a la red sin restricciones, puesto que se pondrán al día en un tiempo breve, y no suponen una amenaza grave a la seguridad del entorno informático de la institución. Los equipos que no cumplan con la política de seguridad son conectados a una red de cuarentena, en la que sólo tienen permiso para acceder a las actualizaciones del sistema y anti-virus permitiéndoles corregir su estado y cumplir con la política de seguridad. Cada cierto tiempo se revisa el nivel de cumplimiento para comprobar si han corregido las deficiencias y en caso afirmativo, son transferidos a la red de explotación.

Este sistema preventivo está orientado a los sistemas Microsoft Windows que son una gran parte de los equipos de la red y son los equipos mas conflictivos ya que son los que más número de ataques reciben, debido a que son el sistema operativo con mayor difusión mundial como se puede ver en la siguiente figura 2.3 según el informe publicado

por MarketShare [4].

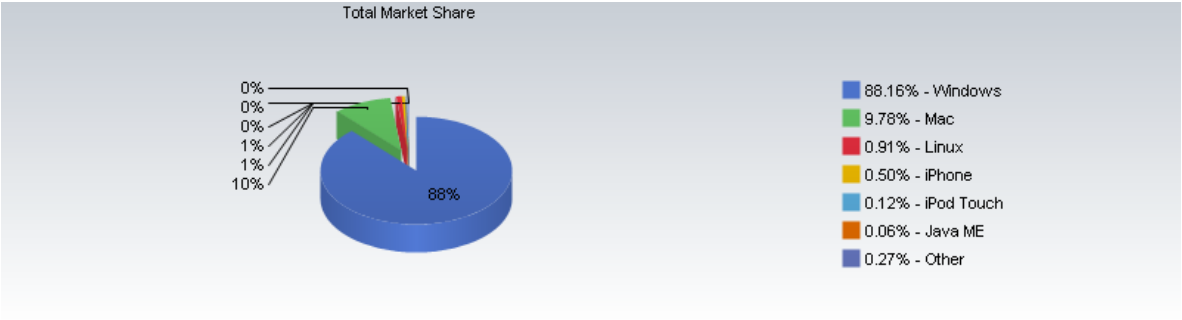


Figura 2.3: Tasa de utilización de los distintos sistemas operativos existentes.

Ventajas:

- Se garantiza el mismo nivel de seguridad para todos los equipos.
- Dispone de una red de cuarentena
- Microsoft esta invirtiendo mucho en este sistema, lo que aporta un respaldo notable.

Inconvenientes:

- Soló disponible para sistemas Microsoft Windows.
- Instalación de software de auditoría en los sistemas de los usuarios.
- Implica la implantación del directorio Active Directory de Microsoft que, no necesariamente esta implantando en todas las organizaciones.

En la figura 2.4 se muestra el esquema Microsoft Connection Manager.

2.2.2. Cisco Network Admission Control

La propuesta de Cisco se basa en el estándar 802.1x [5] para realizar el control de admisión de red basada en puertos.

Permite la autenticación de dispositivos conectados a un puerto LAN, estableciendo una conexión punto a punto o impidiendo el acceso por ese puerto si la autenticación falla. El protocolo 802.1X está disponible en ciertos conmutadores de red y puede configurarse para autenticar nodos que están equipados con software suplicante.De esta

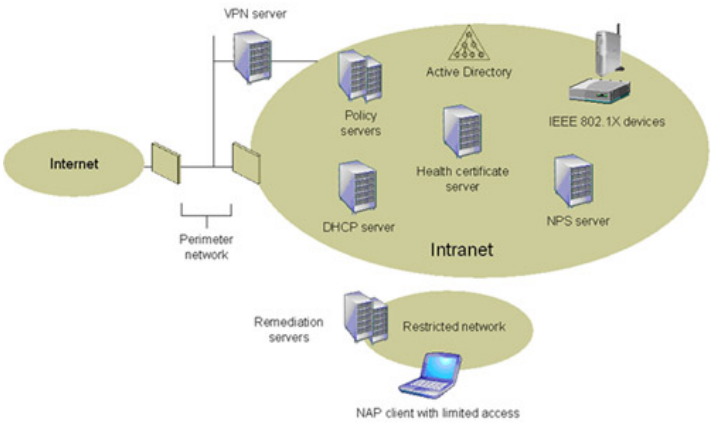


Figura 2.4: Microsoft Connection Manager

forma se elimina el acceso no autorizado a la red en el nivel de la capa de enlace de datos.

La autenticación es realizada normalmente por un tercero, como un servidor de RADIUS/LDAP.

Los sistemas potencialmente peligrosos son confinados en una red de cuarentena, implementada como una VLAN con restricciones en el encaminador. Cisco ha establecido acuerdos con los principales fabricantes de anti-virus y otros productos de seguridad asegurando que su sistema tenga el respaldo necesario para competir con los productos de otros fabricantes.

Para la comprobación del nivel de riesgo se emplea el Cisco Security Agent[6], que permite la integración de módulos de Cisco, así como de otras compañías.

Para el funcionamiento de este sistema es necesario la instalación del Cisco Security Agent en el equipo del usuario y Cisco sólo ofrece el soporte para la plataforma Microsoft Windows, dejando el software para las demás plataformas a otros fabricantes.

Ventajas:

- Dispone de una red de cuarentena implementada en una VLAN.
- Eliminación del acceso no autorizado a la red en la capa de datos.

Inconvenientes:

- Cisco sólo ofrece soporte para Microsoft Windows.

- Instalación de software de auditoría en los sistemas de los usuarios.
- Necesidad de equipos en la red con soporte 802.1X.

En la figura 2.5 se puede observar como trabaja NAC y el antivirus Trend Micro.

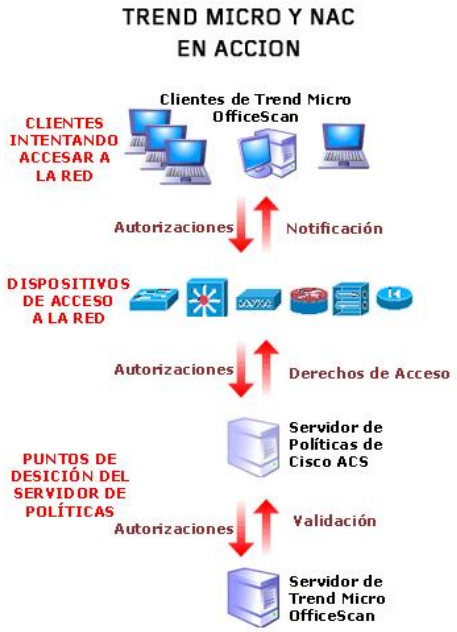


Figura 2.5: Cisco Network Admission Control y Trend Micro OfficeScan

2.2.3. Netsquid

Netsquid [7] es un sistema de protección reactivo, que consiste en ubicar un sistema Linux entre los conmutadores y el encaminador de cada subred. Este sistema Linux actúa como conmutador, pero también ejerce de cortafuegos empleando iptables. [1]

La detección de los equipos comprometidos se basa en el sistema de detección de intrusiones (IDS) Snort, [8] [2] utilizando un conjunto de reglas denominado bleeding-rules[9]. Estas reglas suelen ser muy recientes, por lo que permite la detección de virus/gusanos de nueva aparición. El sistema bloquea temporalmente el acceso a aque-

[1] Iptables es el nombre de la herramienta de espacio de usuario mediante la cual el administrador puede definir políticas de filtrado del tráfico que circula por la red, realizando la función de cortafuegos.

[2] Snort es un sniffer de paquetes y un detector de intrusos basado en red (se monitoriza todo un dominio de colisión). Este sistema implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida. Además existen herramientas de terceros para mostrar informes en tiempo real (ACID) o para convertirlo en un Sistema de detección y prevención de intrusiones.

llos sistemas que hayan generado algún tipo de alerta de seguridad. Durante la desconexión, el tráfico Web es redirigido a una página Web donde se informa al usuario que su sistema ha sido desconectado por motivos de seguridad.

Este sistema de reglas es susceptible de dar falsos positivos, indicando que un sistema está comprometido cuando realmente esta generando tráfico legítimo. Otro punto crítico de este producto es que el sistema Linux se presenta como un punto único de fallo, ya que es el sitio por donde pasa todo el trafico.

Ventajas:

- Utilización conjunto de bleeding rules.
- No es necesario instalar software adicional en el usuario.
- Soporte para todas las plataformas.

Inconvenientes:

- Susceptible a dar falsos positivos
- El sistema Linux es el único punto de fallo del sistema, siendo sensible a vulnerabilidades y sobre todo con problemas con el suministro eléctrico.

2.2.4. Snort Inline y SnortSam

Snort inline [10] es básicamente una versión modificada de Snort que acepta paquetes que provienen del cortafuegos iptables y IPFW vía libipq(linux), en vez de la librería libpcap. Esto permite el uses nuevas tipo de reglas (drop, sdrop, reject) y configurar el cortafuegos iptables/IPFW cuales de los paquetes deberían ser eliminados, reenviados, modificados, o permitidos basándose en una regla de Snort. Esto permite cambiar el modo de funcionamiento del IDS ("Intrusión Detección System") de Snort para transformarlo en IPS ("Intrusión Prevention System"), ya que descarta o aborta todas las conexiones y tramas maliciosas. Su principal problema es la alta probabilidad de que sucedan falsos positivos y se descarte trafico valido.

SnortSam es un plugin para Snort. Este plugin permite automatizar el bloqueo de direcciones IP de manera automática para una larga lista de distintos cortafuegos existentes en el mercado. SnortSam por si mismo consiste en dos bloques el plugin de salida sin Snort y un agente inteligente que corre en el cortafuegos. El agente permite diferentes tipos de capacidades que permiten automatizar los mecanismo de bloqueo.

En la figura 2.6 podemos observar el esquema de funcionamiento de SnortSam.

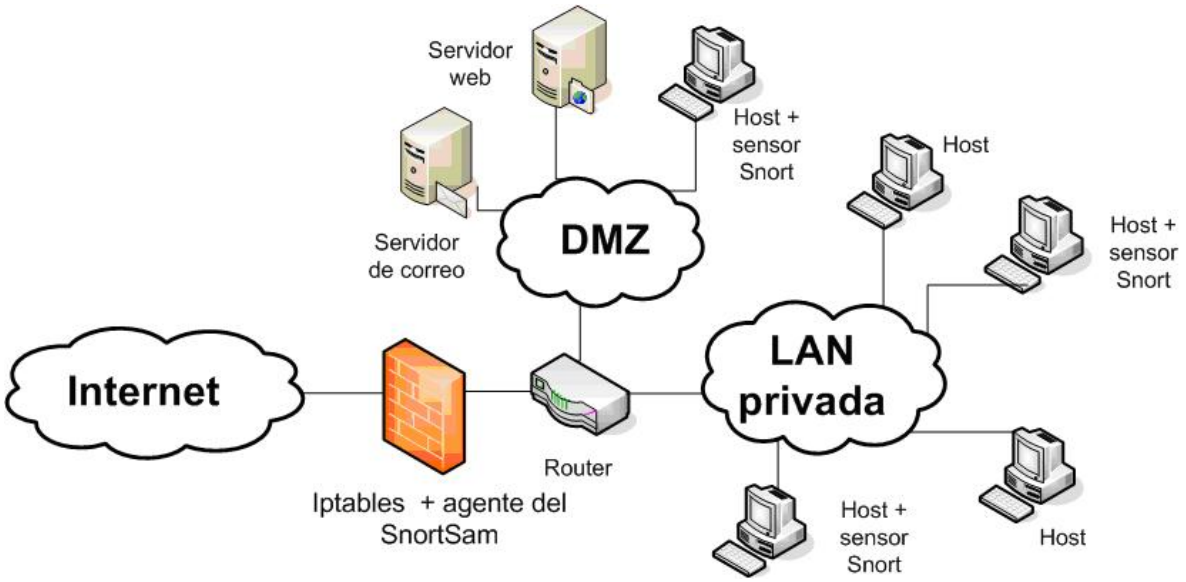


Figura 2.6: SnortSam

Ventajas de los dos sistemas:

- Actualización constante en materia de seguridad.
- Capacidad de personalizacion y creación de reglas.

Inconvenientes de los dos sistemas:

- Recursos para ver las alertas, porque es necesario mirar los logs ya que genera muchos falsos positivos.

2.2.5. Prelude

Prelude [11] es un sistema de detección de intrusiones distribuido que permite agregar la información de varios orígenes, para ello Prelude usa el estándar IDMEF [12][3] para la comunicación de los eventos. Sin embargo no dispone de funcionalidades de cuarentena y Security Event Correlator [16] le falta bastante desarrollo, aunque es un proyecto muy activo y en evolución.

Ventajas:

- Recolección de información de seguridad de diferentes orígenes usando IDMEF.
- No es necesario instalar software adicional en el usuario.
- Soporte para todas las plataformas.

Inconvenientes:

- Falta de desarrollo.
- No dispone de cuarentena.

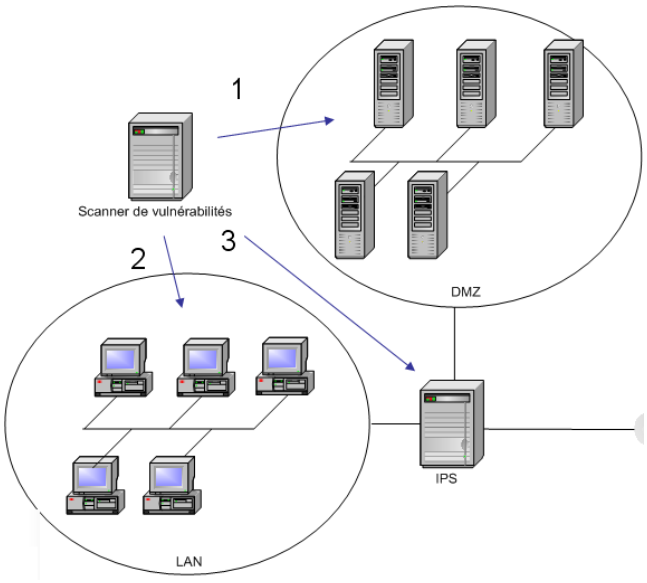


Figura 2.7: Prelude

[3]IDMF es un formato de intercambio de mensajes de alertas de intrusión

2.3. Discusión

La Universidad Carlos III cuenta con una red de más de 9.000 direcciones IP asignadas, en la que se presenta una gran variedad de dispositivos de acceso (SmartPhones, portátiles, etc) y de sistemas operativos (Microsoft Windows, Linux, MacOS X, etc). Esta variedad, junto con las características de una red académica, hacen que sea muy difícil garantizar el cumplimiento de una política de ajuste y mantenimiento de la seguridad de los equipos personales.

Las propuestas de Microsoft y Cisco, se descartaron por que implicaban a los usuarios a instalar software en sus equipos. Software, que aunque estaba disponible para la mayoría de los usuarios, dejaba al margen las otras plataformas. Netsquid se descartó por la fragilidad de ser un sistema con un punto de fallo critico.

Snort-inline y SnortSam es una muy buena alternativa, avalada por la popularidad e implantación de Snort, aunque su carencia de servicios de cuarentena, impide notificar al usuario que su equipo ha sido comprometido y que debería realizar acciones encaminadas a corregir dicha situación.

Prelude es un sistema de detección de intrusiones distribuido, que ha integrado Security Event Correlator [16] y resuelve de manera satisfactoria la detección de sistemas comprometidos, pero no toma ningún tipo de acción para con los sistemas comprometidos.

Descon2 nace para intentar paliar las carencias que suponen los otros sistemas a las necesidades de la red de la universidad Carlos III pero con la capacidad de poderse utilizar en otros tipos de redes sin mucho esfuerzo debido a su estructura modular.

Para ello se ha diseñado en una arquitectura genérica y adaptable que permita agregar todo tipo de información importante para el administrador de la red. Además una vez que se determine que un sistema esta comprometido, se le pondrá en cuarentena avisando al usuario de su situación y ofreciéndole una serie de guías de como puede solucionar su problema de seguridad.

El proyecto actualmente (Octubre 2009) esta siendo explotado por el equipo del

servicio de Informática y mas concretamente por el área de seguridad y comunicaciones de la Universidad Carlos III.

Capítulo 3

Análisis y Requisitos del Sistema

En este capítulo se realiza en primer lugar, una descripción formal de los objetivos impuestos a este proyecto, así como los requisitos necesarios que se han de tener en cuenta a la hora de diseñar un sistema de control y prevención de intrusiones.

3.1. Objetivos

Tras analizar en el capítulo anterior las soluciones existentes en el mercado se decidió establecer unos objetivos que debe cumplir Descon2.

- Automatizar el proceso de detección y desconexión de los equipos existentes en la red.
- Agrupar toda la información de seguridad disponible, para conseguir una buena decisión en los procesos de detección y desconexión.
- Facilitar a los usuarios la corrección del compromiso de seguridad de su sistema. Ofreciéndoles los medios necesarios (guías actualizaciones, etc) para que solucionen su problema de seguridad en el menor tiempo posible.
- Poner en cuarentena los sistemas presuntamente comprometidos, para intentar reducir las consecuencias de posibles infecciones masivas.
- Localización de un equipo en la red cableada así como inalambrica.

- Disponer de unas estadísticas fiables.

3.2. Requerimientos del sistema.

Para cumplir con los objetivos anteriormente descritos y con los inconvenientes existentes en las soluciones de la competencia (Capitulo: 2.2) se han establecido una serie de requisitos se detallan los requerimientos que debería tener el sistema:

3.2.1. Requisitos no funcionales

- **Interfaz Web** con la que interactuar con el sistema.
- **Servidor de aplicaciones Web Tomcat**.
- **Base de datos** donde almacenar la información.
- **Un lenguaje de programación orientado a objetos** con las siguientes capacidades:
  - Capacidad de creación múltiples hilos de ejecución.
  - Integración con la creación de paginas web de contenido dinámico.
  - Independencia de la plataforma de ejecución.
  - Independencia de la plataforma de desarrollo.
  - Integración con un base de datos.
  - Integración con WEKA.
- **Posibilidad de integración con equipamiento de red de diferentes fabricantes**.
- **Posibilidad de poder portar sistema a otra institución**.
- **Escalabilidad:** capacidad de actuar no solo en el marco actual, de protocolos, de necesidades de usuarios etc.., si no los que puedan existir en el futuro.

3.2.2. Requisitos transaccionales o funcionales internos

- **Integrar el sistema con las fuentes de información de seguridad existentes en el mercado así como las que puedan aparecer en el futuro.** Estas aplicaciones pueden ser:

  1. Sistemas antivirus (Trend micro, Norton Security etc..)
  2. Sistemas de detección de intrusiones (Snort, Network Virus Wall, etc..)
  3. Cortafuegos de red (Cisco PIX, Firewall-1, Iptables, etc..)
  4. Recopilación de información de logs

- **Localización de equipos**
- **Automatización de las desconexiones.**
- **Notificación automática** de los sucesos, vía correo electrónico.
- **Muestra de las alertas actuales en el sistema, así como de fechas anteriores.**
- **Generación de informes.**

3.2.3. Requisitos de entorno operacional

- **Soporte de la carga de trabajo**, que pueden suponer unos 9000 equipos generando alertas en el sistema.
- **Necesidad un sistema escalable y distribuido.**
- **Disponibilidad 24 horas y 7 días a las semana**.
- **Concurrencia de usuarios**.
- **Evolución continua**.
- **Seguridad** limitación de la población de los usuarios finales que pueden tener acceso a la aplicación.

3.2.4. Requisitos de interfaz

- **Identificación del usuario.**
- **Necesidad de un correcto funcionamiento con los principales navegadores Web**.
- **Necesidad de confidencialidad de la información** para ello se deben emplear técnicas de codificación y cifrado para la información del usuario.

3.2.5. Requisitos de personalización

- Capacidad de mostrar únicamente la información correspondiente al usuario.

3.3. Análisis de los elementos de la estructura de DESCON2

Descon2 estará compuesto por de varios elementos, cada uno de ellos con una labor concreta que pueden ser diseñados e implementados independientemente, facilitando el cumplimiento de los requisitos establecidos.

3.3.1. Alertas

Son la unidad básica del sistema y se generan en los sensores. Para el control del sistema administrador fija dos parámetros son, el tiempo de vida de la alarma y los coste totales de las alarmas para un determinado equipo, para cubrir diferentes tipos de ataques como por ejemplo, los intentos fallidos de conexión de los usuarios mediante SSH. Este tipo de evento tendría un coste bajo, y sin embargo un tiempo de vida lo suficientemente largo, como para detectar ataques de fuerza bruta.

3.3.2. Sensores

La arquitectura del sistema se compone de sensores, que son programas especializados que monitorizan ciertos eventos, considerados relevantes por el administrador de seguridad y lo notifican al colector de alertas, empleando el formato IDMEF[12].

3.3.3. Colector de alertas

El colector recoge las alertas y evalúa el nivel de seguridad. Cuando el colector de alertas considera que un sistema está presuntamente comprometido, informa que dicho sistema se le debe enviar a la red de cuarentena.

Para la realización del colector es necesario analizar el tratamiento de la información recolectada para clasificar si un sistema está siendo atacado, o está realizando algún tipo de acción maliciosa para los demás usuarios del sistema.

Para ello Descon2 recopilara información de los sensores disponibles en el sistema obteniendo variables que estén relacionadas con la decisión. A partir de las cuales evaluara, explícitamente, una función que, tomará dos valores (0/1 No comprometido/Comprometido).

Para la descripción de este análisis se utilizara la siguiente nomenclatura:

- Cada posible equipo: caso, instancia o ejemplo; (k).
- Su conjunto de respuestas: observación o dato; x(k) (muestra).
- Cada componente de las respuestas: variable o rasgo; xn.
- Las posibilidades (Comprometido/ No Comprometido): hipótesis Hi.
- La función Fw: decisor o clasificador
- El resultado : decisión o clasificación; Dj

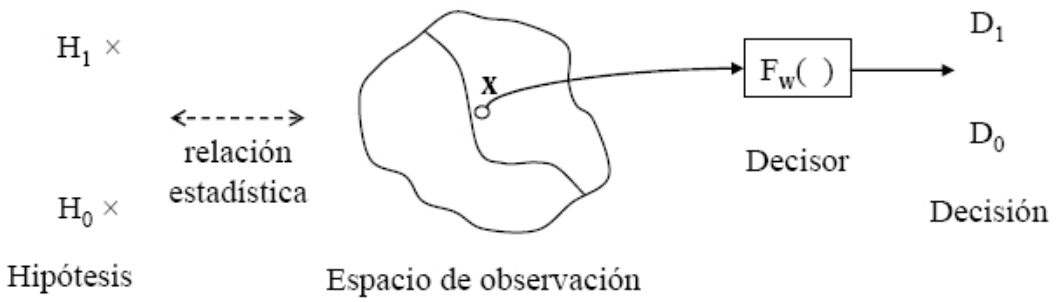


Figura 3.1: La decisión equivale a trazar una frontera en el espacio de observación

La información necesaria para la decisión sera:

- Probabilidades "a priori": Pr(Hi)
- Verosimilitudes: p(x/Hi)
- Parámetros de coste: Cji, se toman positivos,

  siendo:

$$Cji > Cii$$

  , para todo

$$j! = i$$

  se diseña el decisor F.

La obtención de la frontera de decisión se evaluara una función W ($WX \gtreqless Umbral$) estimada con la información y conocimiento que nos dan los sistemas utilizados previos a Descon2.

Con los datos disponibles y al aplicar la función W devolverá un valor que comprado con nuestro umbral determinara la decisión. El valor del umbral en nuestro diseño inicial viene determinado por un valor fijo, pero en el futuro se espera que pueda variarse dinamicamente para poder tener una mejor frontera de decisión.

Para la determinación de cual debe ser el umbral de decisión al sistema se le aplicaran unos datos etiquetados [1] que servirán para entrenar el sistema con lo que se obtendrán unos parámetros de la función W.

Esta función W deber ser una función que establezca una región de decisión lo suficientemente genérica para que sea una buena frontera de decisión para los datos que venga en el futuro.

Para comprobar la calidad del decisor diseño en Descon2 se establecerán dos parámetros:

- **Probabilidad de Falsa Positivo:** Esta probabilidad se refiere a marcar un dispositivo como infectado y no lo este.

[1] Datos etiquetados, o datos de entrenamiento son datos de los cual ya conocemos la decisión correcta.

- **Probabilidad de perdida:** En este caso marcamos un equipo como no infectado y su estado es realmente de infeccioso.

Con estos dos parámetros se puede obtener uno más global al que se llamará probabilidad de error que determina la probabilidad del que sistema falle, que se calculará de la siguiente forma:

Perror=P(H0) x P Falsa Alarma + P(H1) x P Perdida

P(H0)=Probabilidad de que suceda la hipótesis H0.

P(H1)=Probabilidad de que sucede la hipótesis H1.

Para ampliación de la información de esta sección esta en la bibliografía referente al Tratamiento Digital de Señales [17]

Implementación de un colector de alarmas inteligente

La implementación de un colector de alarmas inteligente se realizara con la información que aportan las alarmas y el conocimiento que aportan reglas y filtros, como el que posee el IDS Snort, del que se obtienen datos y con una aplicación de minería de datos obtendremos la información estadística para realizar el análisis.

Este proyecto realizara un colector de alarmas simple con una frontera de decisión fija, en futuro desarrollos se implementara una frontera dinámica utilizando Weka [18] que es un entorno para análisis del conocimiento.

Esto permitirá ajustar la función W y el umbral para minimizar la probabilidad de error del sistema.

Se comentará algo más sobre este tema en el apartado dedicado a lineas futuras.

3.3.4. Localización de equipos

Para poder poner a un equipo en cuarentena, antes habrá que conocer el punto que se encuentra en nuestra red. La localización se hace mediante la dirección IP del equipo comprometido y en función del lugar donde este localizado se aplicara el mecanismo de protección correspondiente.

Los lugares donde puedan estar localizados se determinar por los rangos de las direcciones IP.

Estos rangos de direcciones pueden ser:

- **Direcciones de fuera del rango de la universidad** que acceden a la red a través del encaminador de acceso a Internet.
- **Direcciones Internas** estas direcciones pertenecen a sistemas conectados a la red de la institución, por lo que habrán de considerarse las diferentes posibilidades.
  - **Conexiones desde la red cableada** el equipo comprometido usa la conexión de cable desde las distintas dependencias de la institución.
  - **Conexiones desde la red inalámbrica** el equipo usa los puntos de acceso WIFI disponibles para la validación de usuario.Las direcciones pertenecen a rangos específicos y se asignan mediante DHCP.
  - **Conexiones mediante Red privada virtual o VPN** el sistema utiliza un túnel IPSec y se emplea un servidor Radius para la validación del usuario. Las direcciones pertenecen a rangos específicos y se asignan mediante DHCP.

3.3.5. Interfaz de interacción con el usuario

Para la interacción con el usuario se estableció el uso de una aplicación Web debido a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software.

Análisis de la WebApp Descon2

En esta sección se mostrara las metas y objetivos (determinados durante la formulación).

La jerarquía de usuarios Para la organización de los usuarios en Descon2 se estableció a tres tipos de usuarios:

1. **Administrador de red:** Son los encargados de controlar la red principal de la institución.

2. **Administrador departamental:** Se encarga de redes concretas dentro de la universidad, por lo cual tienen una necesidades distintas a los administradores de red ya que solo deben tener la información correspondiente a su red.

3. **Administrador de Descon2** Es el encargado del control de usuarios y mantenimiento de Descon2.

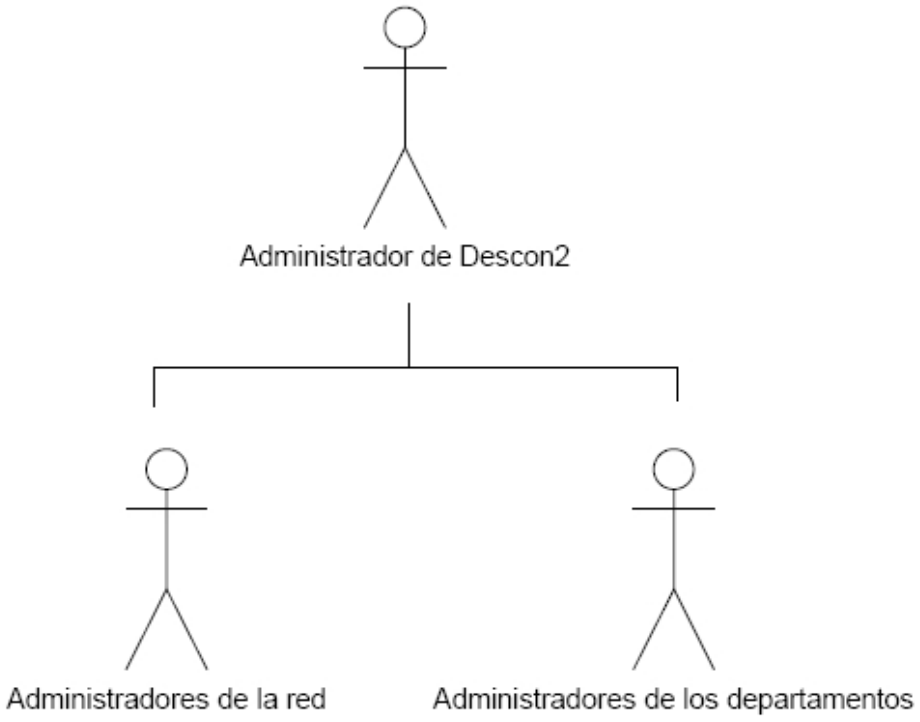


Figura 3.2: Jerarquía de usuarios

Diagramas de uso Para cada uno de los distintos tipo de usuario de la aplicación hay un modelo de caso de uso.

1. Administrador de red (Figura 3.3)
2. Administrador departamental (Figura 3.4)
3. Administrador de Descon2 (Figura 3.5)

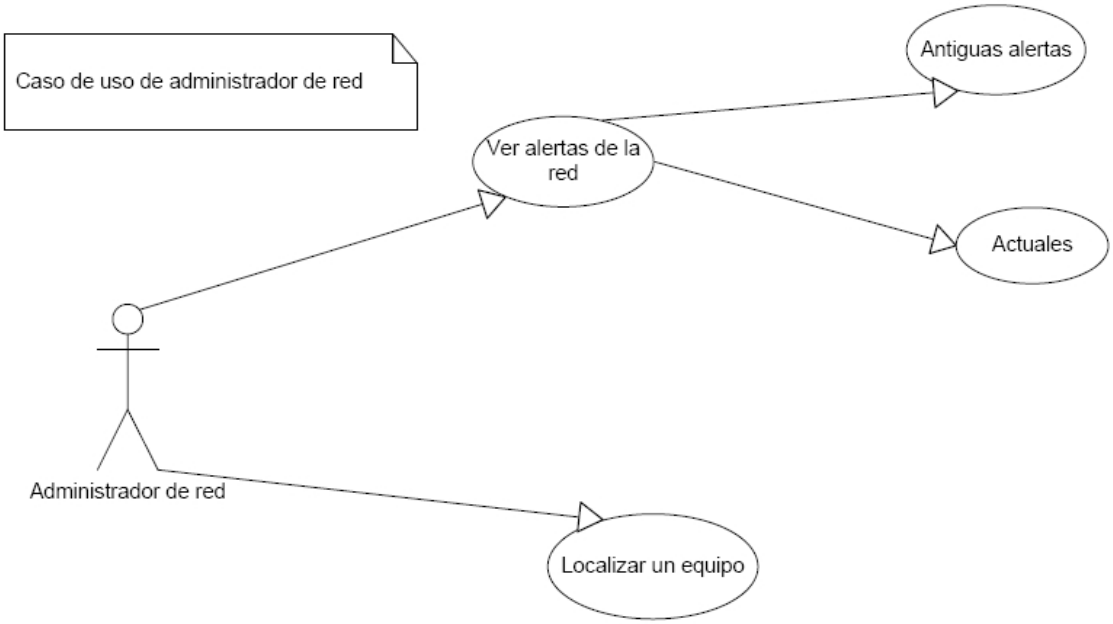


Figura 3.3: Diagrama de uso del administrador de red

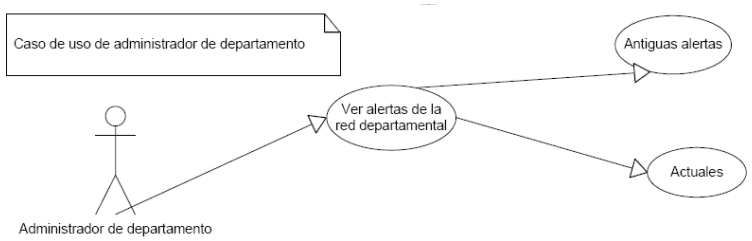


Figura 3.4: Diagrama de uso del administrador de red departamental

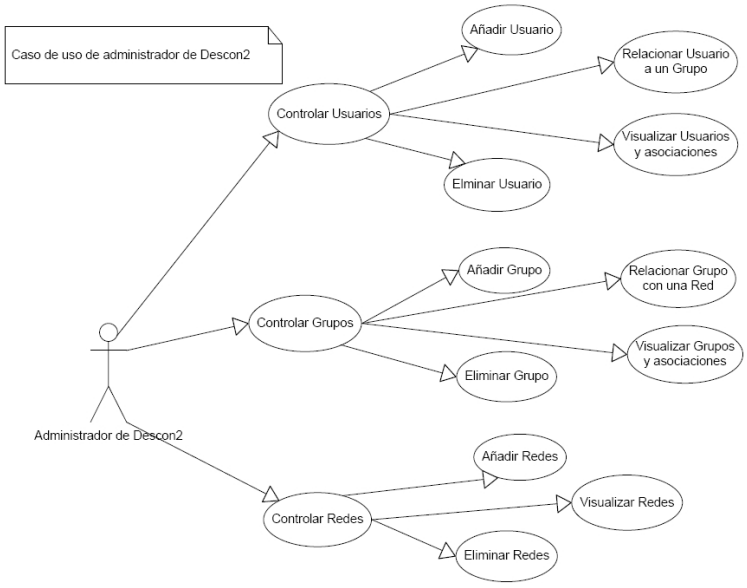


Figura 3.5: Diagrama de uso del administrador de Descon2

Diagramas de estados

En la figura 3.6 se muestran los seis estados observables externamentes en la aplicación:

- validar usuario
- comprobación de permisos de usuario
- localizar equipos
- ver alertas en tiempo real

- ver alertas BD
- administrar sistema

Como se ve además en el esquema se muestras las acciones que desplazan a un usuario de un estado a otro.

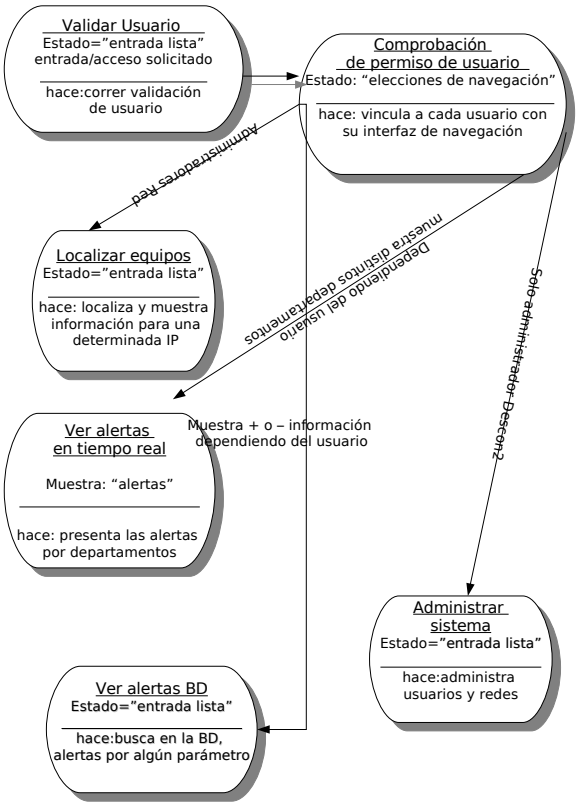


Figura 3.6: Diagrama de estados de la WebApp

Capítulo 4

Diseño e implementación del Descon2-Web

Para la realización del diseño de Descon2-Web se estableció una organización modular en la que los distintos módulos debían de trabajar de manera independiente, aportando distintas funcionalidades al sistema. Esto permite que el sistema pueda crecer en el futuro de manera sencilla y escalable.

Los módulos principales con los que cuenta el sistema son cuatro, **Módulo de Localización de equipos**, responsable de determinar el punto de conexión de un equipo a la red corporativa, **Sistema Colector de alarmas**,que agrega la información de seguridad que proporcionan los sensores en forma de alarma, **Almacenado de las alarmas**,que permite obtener la información histórica sobre las alarmas recibidas ,de manera que sea fácil y rápidamente recuperable, y por ultimo **Portal Web** que se utilizara como interfaz entre el usuario y el sistema.

En la figura 4.1 observamos los distintos bloques en los que se divide Descon2.

4.1. Diseño

Para el modelado de Descon2 se ha utilizado el Lenguaje de Modelado Unificado (UML, *Unified Modeling Language[19]*). UML es un conjunto de herramientas, que per-

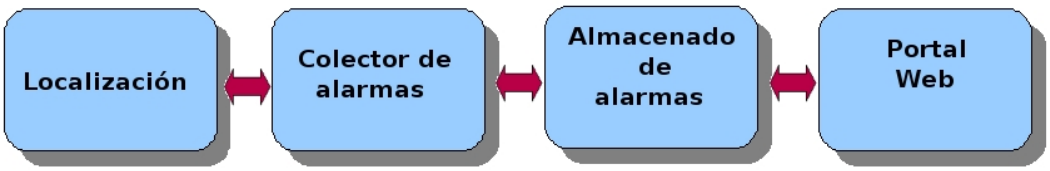


Figura 4.1: Diagrama de bloques de descon2

mite modelar (analizar y diseñar) sistemas orientados a objetos. Es importante resaltar que UML es un "lenguaje"para especificar y no para describir métodos o procesos. Se utiliza para definir cualquier tipo de sistema, sea informático o no, mediante diagramas. Para ello UML define 13 tipos de diagramas que permiten describir la arquitectura y funcionalidades del sistema a desarrollar.

Estos diagramas son:

Diagramas de Estructura, enfatizan en los elementos que deben existir en el sistema modelado:

- Diagrama de clases
- Diagrama de componentes
- Diagrama de objetos
- Diagrama de estructura compuesta (UML 2.0)
- Diagrama de despliegue
- Diagrama de paquetes

Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades
- Diagrama de casos de uso

- Diagrama de estados

Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia
- Diagrama de comunicación, que es una versión simplificada del Diagrama de colaboración (UML 1.x)
- Diagrama de tiempos (UML 2.0)
- Diagrama de vista de interacción (UML 2.0)

Para el desarrollo de este proyecto se han utilizado los **diagramas de clase**, y los **diagramas de secuencia** para el modelado **estático** y **dinámico** del sistema.

Estos diagramas se usaran para describir el sistema, puede obtener más información sobre el lenguaje y la sintaxis utilizadas en los diagramas se recomienda visitar la pagina web de Object Management Group (*www.uml.org*).

Lenguaje de programación utilizado en Descon2

Para la implementación del diseño y atendiendo a los requisitos establecidos en el capitulo anterior se utilizo JAVA que es un lenguaje de programación orientado a objetos y desarrollado por Sun Microsystems.

El motivo de porque se eligió JAVA frente a otros lenguajes de programación fueron los siguientes:

- es un lenguaje independiente de la plataforma y del sistema operativo.
- posibilidad de realizar paginas web de contenido dinámico, de forma sencilla incluyendo («embebiendo » o «empotrando») el código en las paginas HTML.
- elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

El lugar donde trabajara la aplicación JAVA desarrollada, sera en un servidor TOMCAT como indican los requisitos,Tomcat es un contenedor de servlets [1].

4.2. Diseño del Sistema de localización de equipos

Atendiendo a los requerimientos expuestos anteriormente, de que el sistema fuera escalable y portable. Se diseño el sistema siguiendo los siguiente diagramas.

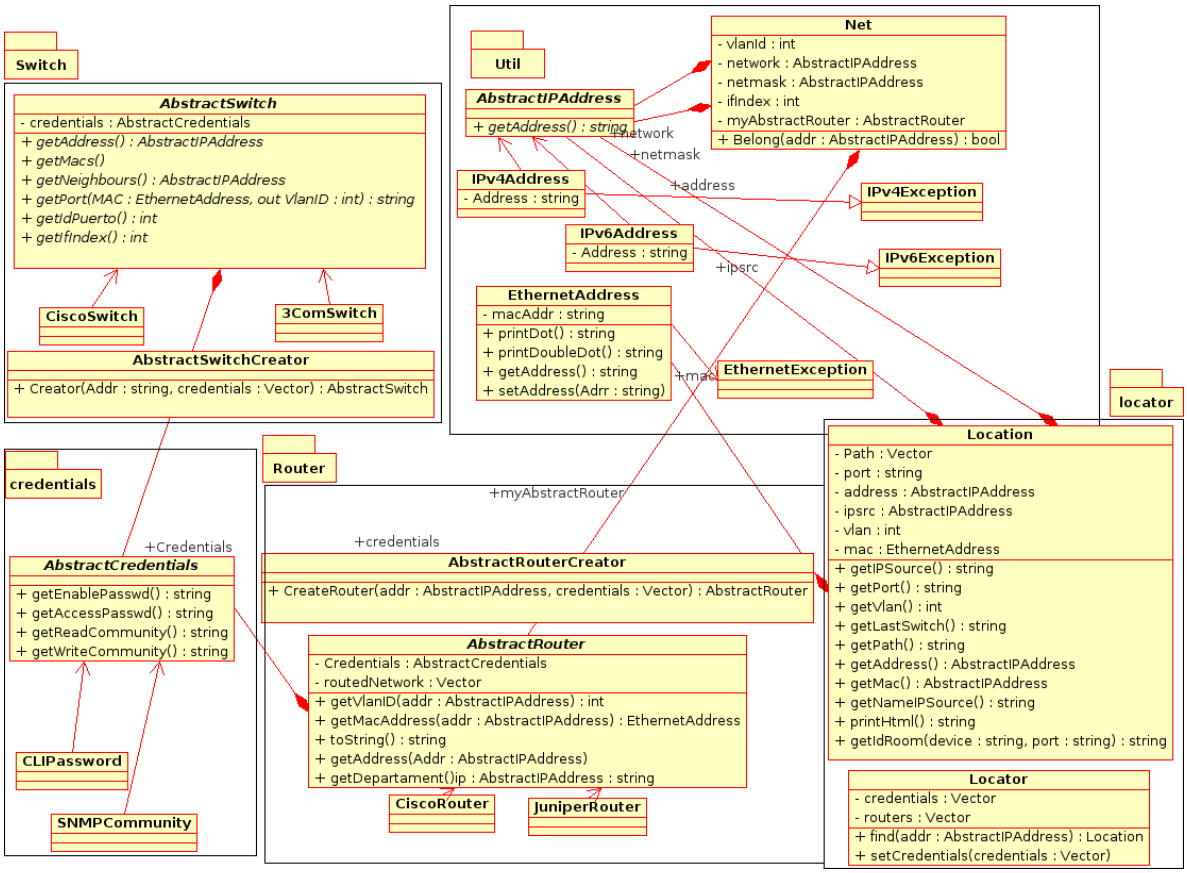


Figura 4.2: Diagrama de clase del Sistema de localización

En el diagrama de clase 4.2 se puede observar los cincos paquetes en que se distribuye el diagrama.

Estos paquetes son:

Util. Paquete que se encarga de todas los estructuras de la red como las direcciones de red, direcciones físicas, o valores de una red como la mascaras de red, VLAN's,

[1] Son programas que se ejecutan en un servidor Web y construyen páginas Web

dirección de red, etc.

Credentials. Soporte a los mecanismos de control de acceso y escritura de los equipos existentes en la red.

Location. Bloque encargado de realizar el proceso de localización y la muestra de los resultados.

Switch . Paquete que representa las funcionalidades de los conmutadores de la red para el sistema.

Router. Paquete que representa las funcionalidades de los enrutadores de la red para el sistema.

4.2.1. Estructura del paquete Util.

Este paquete se compone como se puede ver en la figura 4.3 de cinco clases.

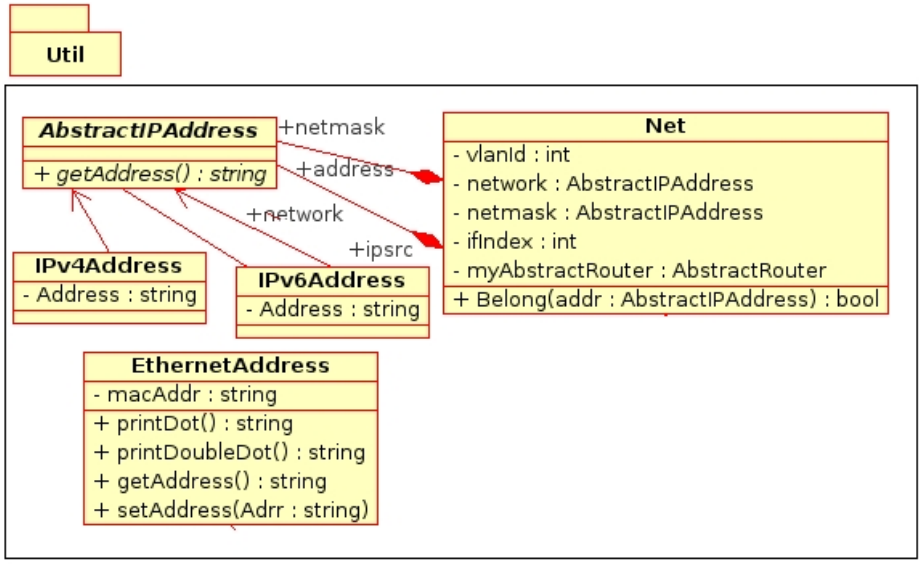


Figura 4.3: Paquete Util

AbstractIPAddress Es una clase abstracta que representa un jerarquía de clases en el que se extraen las características que deben poseer las direcciones IP.

getAddress() : String : Devuelve un objeto String con la dirección IP que se representan.

IPv4Address, IPv6Address Permite gestionar direcciones IP para las versiones 4 y 6 del protocolo.

EthernetAddress Clase que representa una dirección del tipo Ethernet consta de cuatro métodos:

printDot(): String : Establece que la dirección Ethernet se muestre de la siguiente forma aaaa.bbbb.cccc

printDoubleDot(): String : Establece que la dirección Ethernet se muestre de la siguiente forma aa:aa:bb:bb:cc:cc.

getAddress() :String Devuelve la dirección Ethernet tal y como fuera almacenada.

setAddress(Addr:String) Modifica el valor de la dirección Ethernet después de que el objeto ya este creado.

Net Esta clase agrupa las características que determinan una red, estas características son la vlan, la dirección de red, la mascara de red, y la representación del router que la encamina.

Belong(addr : AbstractIPAddress) : Boolean Este método determina si una dirección IP pertenece a la red que representa el objeto Net.

IPv4AddressException,IPv6AddressException Lanza una excepción cuando se construye una dirección IP de versión 4 o de versión 6 de forma incorrecta.

EthernetException Excepción utilizada para indicar que la dirección de Ethernet que se intenta crear es incorrecta.

4.2.2. Estructura del paquete credentials.

Credentials Paquete que ofrece funcionalidades de almacenamiento de contraseñas de lectura y escritura de los equipos de red (Figura 4.4).

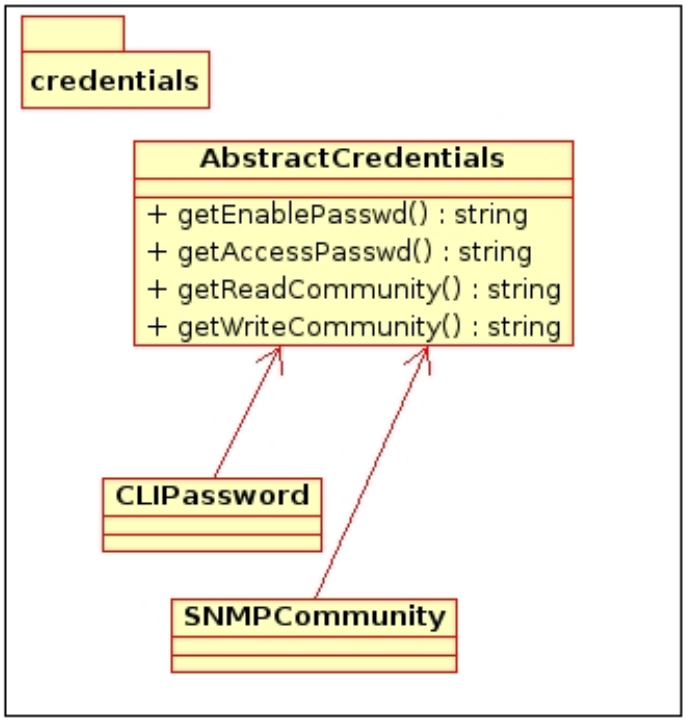


Figura 4.4: Paquete Credentials

AbstractCredentials Clase abstracta, que tiene una serie de funciones que representan las contraseñas para los distintos tipos de acciones en función del método de acceso y mecanismo de control de acceso Telnet o SSH,etc.. para el protocolo SNMP.

Estas credenciales son para acceder a los equipos de comunicaciones de la siguiente forma:

Para acceder via Telenet o SSH.

getEnablePasswd().

getAccessPasswd().

Via el campo community de Snmp

getReadCommunity().

getWriteCommunity().

CLIPasswords, SNMPCommunity Diferentes implementaciones de AbstractCredentials para el acceso Telnet ó SSH y SNMP, respectivamente.

4.2.3. Estructura del paquete switch.

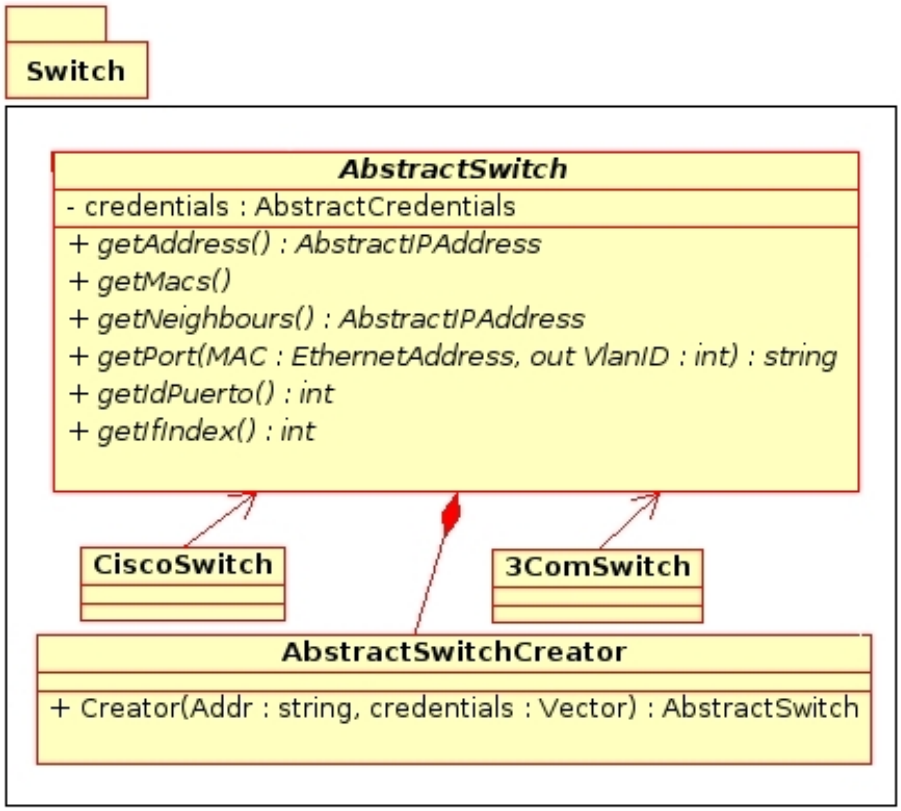


Figura 4.5: Paquete Switch

Switch Este paquete, que podemos ver en la figura 4.5, modela los conmutadores de red extrayendo su información e interaccionando con ellos pueden obtener la información en tiempo real.

AbstractSwitch Clase abstracta que representa un switch genérico, de la que se se extraen las características que de los objetos del tipo switch.

getAddress(): AbstractIPAddress. Devuelve un objeto AbstractIPAddress con la dirección IP del conmutador.

getPort(Mac EthernetAddress,int VlanID):String. Indica el puerto correspondiente en un conmutador, para una dirección de Ethernet y dentro de una

VLAN determinada.

getMacs(int vlanID,int ifIndex):Vector. Da como resultado un vector de objetos EthernetAddress, correspondientes a un puerto del conmutador en el que se encuentre la dirección Ethernet que se quiere localizar.

getNeighbours(String port): AbstractIPAddress. Devuelve un objeto AbstractIPAddress con la dirección IP del conmutador vecino en el puerto en el que se encuentra la dirección Ethernet a encontrar.

getIdPuerto():String y getIfIndex():String. Estos métodos obtienen valores intrínsecos de los puertos del conmutador necesarios para el proceso de localización.

AbstractSwitchCreator Crea objetos de uno de los tipos de AbstractSwitch

CreateSwitch(addr : AbstractIPAddress,credentials : AbstractCredentials) : AbstractSwitch. Método que crea un objeto AbstractSwitch, este objeto podrá ser uno de los múltiples tipo de conmutadores implementados.

CiscoSwitch, 3ComSwitch En el caso de este proyecto solo esta implementado para equipos Cisco, el caso de 3Com esta a modo de ejemplo para indicar la posibilidad de adaptación del sistema a conmutadores de distintos fabricantes.

4.2.4. Estructura del paquete router.

Este paquete, figura 4.6 se compone de:

AbstractRouter Clase abstracta que representa un router genérico, del que se se extraen las características que deben poseer los objetos del tipo Router.

getAddress() : AbstractIPAddress : Devuelve un objeto AbstractIPAddress con la dirección IP del router.

toString(): String Muestra la dirección IP del router como un objeto del tipo String.

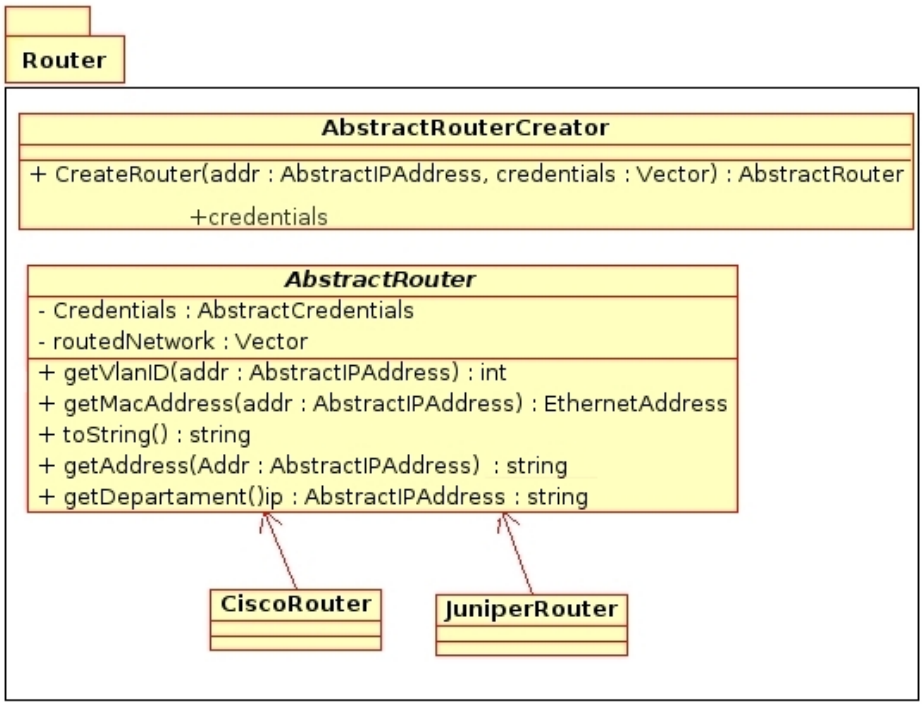


Figura 4.6: Paquete router

getDepartament(AbstractIPAddress ip): String Retorna la descripción de la VLAN correspondiente a la IP que se le pasa como parámetro, que en el caso de la UC3M corresponde al departamento que usa esa VLAN.

getVlanID(AbstractIPAddress) : int Identificador de Vlan para una determinada dirección IP

getMacAddress(AbstractIPAddress) :EthernetAddress : Devuelve un objeto EthernetAddress correspondiente a la dirección IP que se le pase como parámetro, en el caso de que el router encamine esa red.

AbstractRouterCreator Clase que crea objetos AbstractRouter.

CreateRouter(addr : AbstractIPAddress,credentials : AbstractCredentials) : AbstractRouter Método que crea un objeto AbstractRouter. Este objeto podrá ser uno de los múltiples tipo de router implementados.

CiscoRouter y JuniperRouter Implementaciones de ejemplo para una serie de routers de distintos fabricantes. Como en el caso de los Objetos Switch en este

proyecto solo esta implementado para equipos Cisco.

4.2.5. Proceso de localización

La localización de un equipo se realiza mediante la dirección IP. Esta dirección deberá ser del rango asignado a la institución en la que se utilice Descon2.

En este proceso habría que tener en cuenta la red de la institución.

La figura 4.7 muestra de manera gráfica el proceso de localización.

Figura 4.7: Método de localización de DESCON 2

Si la dirección no pertenece al rango, o no es controlada por el servicio de informática no podrá ser localizada, ya que no está conectado a la red mediante nivel de enlace.

El proceso de localización consta de tres pasos:

- Consulta a los distintos routers de las subredes, para obtener la dirección MAC que corresponde a la IP buscada. Si no se recibe respuesta, se deberá a que los enrutadores no tienen su dirección de Ethernet en memoria, asumiendo que el equipo ahora mismo no está conectado a la red.

- Cuando se recibe la respuesta del router de la subred, con la dirección MAC se envían dos consultas al conmutador vecino:
  - En qué puerto está conectado esa dirección MAC.
  - Se consulta al conmutador si en el puerto, en el que esta asociada la dirección MAC, tiene conectado otro conmutador del servicio informática. Si es así repite la pregunta, para ver en que puerto esta asociada la dirección MAC y se pregunta si por ese puerto tiene algún conmutador. Esto se repite recursivamente hasta que se localiza un conmutador en el que el puerto buscado no tenga ningún conmutador vecino.
- Cuando lleguemos al último conmutador nos quedaremos con el puerto al que esta conectado, el nombre del último conmutador, la dirección IP de éste, la cadena de conmutadores por la que se pasa hasta su localización y la lista de MAC en ese puerto.

  La lista de MAC es necesaria, ya que puede que en el último conmutador controlado por el servicio de informática haya conectado otro conmutador de otro departamento. Esto provoca que en el puerto del conmutador controlado, se puedan ver varias direcciones MAC además de la buscada. Esto se puede ver de manera gráfica en la figura 4.8.

  Esta situación provoca, que el procedimiento de actuación sea diferente si se encuentra un equipo comprometido ya que al bloquear el trafico en nuestro conmutador, los demás equipos pasarían a la red de cuarentena.

En el caso que el ultimo conmutador sea un punto de acceso inalámbrico también nos quedaremos con el nombre de este.

Si se quiere obtener mas información concreta sobre el protocolo SNMP o las consultas empleadas en este proyecto vea el anexo

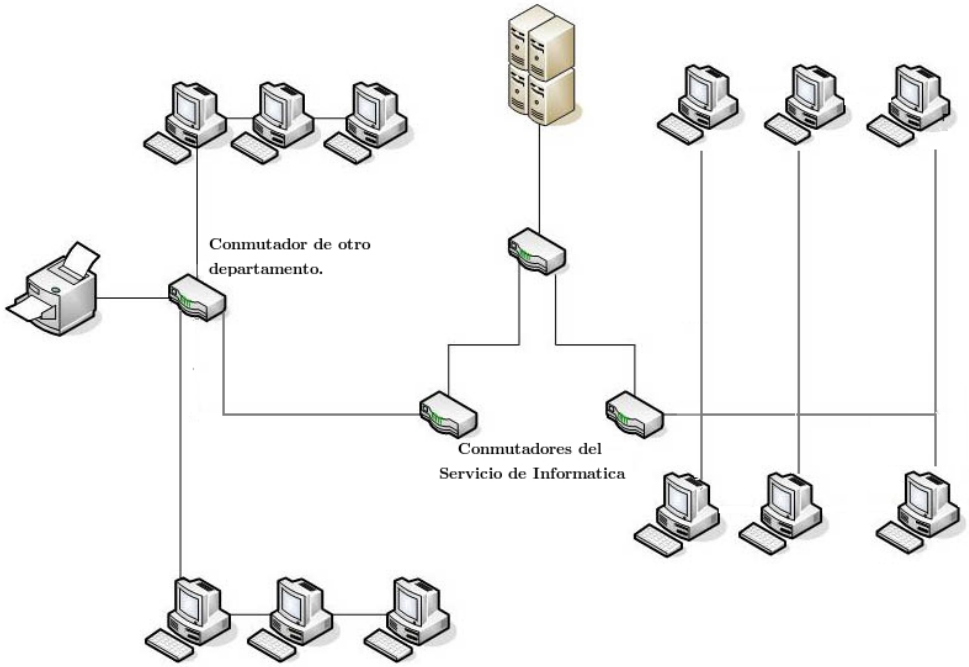


Figura 4.8: Ejemplo de arquitectura de red.

4.3. Diseño del colector de alarmas

Para este bloque se establecieron los mismos principios de escalabilidad y portabilidad que para el bloque de localización. Para ello se dividió este bloque en dos paquetes **collector** y **alarm**.

4.3.1. Estructura del paquete collector

Collector: agrupa todas las funcionalidades de recoleccion de alarmas, comprobación de el coste asociado a cada dirección IP y almacenaje de dicha información en memoria. Las alarmas provienen de distintos sensores colocados en puntos de la red aportando información de los eventos que suceden y el riesgo los mismos suponen.

En la figura 4.9 se puede observar como se obtienen las alarmas desde distintos sensores.

El paquete 4.10 se compone de la siguiente estructura de clases:

- **AbstractCollector**: Es una clase abstracta y permite crear otros tipos de colec-

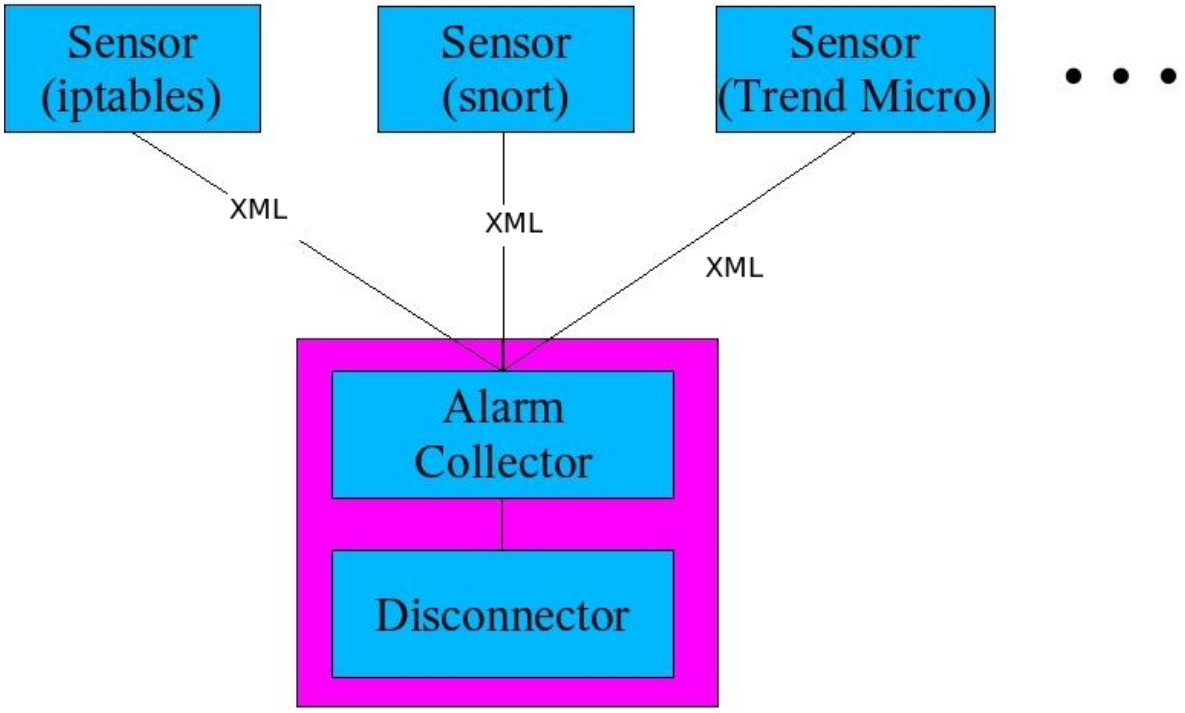


Figura 4.9: Estructura del colector de alarmas.

tores, patrón Strategy [2]obtenido del libro Gamma Design Patterns [20].

Atributos

***list:HostList*:** Objeto HostList que almacena la lista de Host que a su vez componen de una lista de alarmas correspondientes a dicho host.

Métodos

***addAlarm(alarm: AbstractAlarm)*:** Este método asocia la alarma al Host correspondiente.

***getIterator():Enumeration* :** Proporciona un iterador, que permite recorrer la lista de Host de los que se tiene alarmas.

***removeIpWithoutAlarms(IPKey: String)*:** Elimina todos aquellos Host que no tienen alarmas en vigor para reducir el consumo de memoria.

***disconnect(ip: AbstractIPAddress*:** Marca una determinada IP con la etiqueta de desconectada, de modo que se notifique ha superado el coste permitido.

[2]El patrón Strategy permite mantener un conjunto de algoritmos de los que el objeto cliente puede elegir aquel que le conviene e intercambiarlo según sus necesidades.

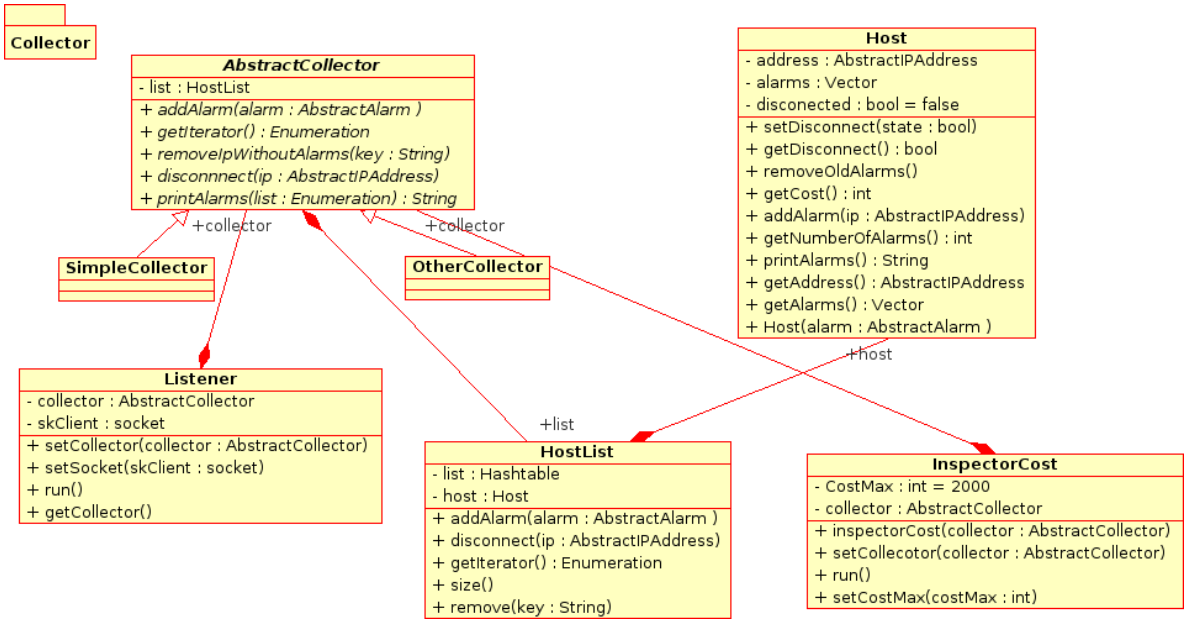


Figura 4.10: Paquete collector

***printAlarms(list: Enumeration) :String*:** Muestra el listado de las direcciones IP que poseen alarmas.

- **SimpleCollector y OtherCollector** Implementaciones de AbstractCollector.
- **Listener**: Escucha de un puerto las alarmas crea los objetos del tipo AbstractAlarm y los almacena en un collector.

Atributos

***collector: AbstractCollector* :** Atributo que asocia al objeto listener un collector para almacenar las alarmas.

***skCliente: socket* :** Socket por el que se escuchan las alarmas. Utiliza el puerto 1123 de Tcp.

Métodos

setCollector(collector : AbstractCollector Establece un collector.

setSocket(skCliente: socket) Establece un socket.

run() Crea un hilo de ejecución, en el que escucha las alarmas de un socket.

getCollector() Devuelve el collector donde se almacenan las alarmas.

- **HostList**: Crea una lista de objetos Host almacenados en una tabla del tipo Hash Table.

  Atributos

  list:HashTable Objeto HashTable, Representa una colección de pares de clave y valor organizados en función del código hash de la clave, que hacen que resulte fácilmente accesibles, en el caso de este proyecto se ha utilizado para almacenar los objetos Host usando como clave el objeto AbstractIPAddress.

  Métodos

  ***addAlarm(alarm: AbstractAlarm)*:** Añade alarma a la lista de Host, comprueba a quien corresponde la alarma viendo la dirección IP y la almacena en el Host correspondiente dentro de la HashTable.

  ***getIterator():Enumeration* :** Método que devuelve un objeto del tipo Enumeration, con la lista de Host que tiene alarmas.

  ***disconnect(ip: AbstractIPAddress*:** Marca una determinada IP con la etiqueta de desconectada.

  size(): int Devuelve el número de Host almacenado en la HashTable.

- **Host**: Representación del equipo o sistema que tiene alarmas asociadas en la red.

  Atributos

  address: AbstractIPAddress Dirección IP del Host.

  alarms: Vector Vector que contiene todas las alarmas asociadas al objeto Host

  disconnected: boolean=false Se utiliza para indicar que un equipo ha sido desconectado de la red y por lo tanto las alertas que se reciban una vez desconectado tendrán un tratamiento diferente, en la actualidad esas alertas se ignoran.

  Métodos

setDisconnect(state: boolean) Establece un valor a la variable disconnected que representa la desconexión de un equipo.

getDisconnect():boolean: Devuelve el valor actual de el Host, si ha sido desconectado o no.

removeOldAlarms(): Borra aquellas alarmas cuyo tiempo de vida haya expirado.

getCost:int: Devuelve el coste total de todas las alarmas del Host.

addAlarm(ip: AbstractAlarm): Añade alarmas al Host.

getNumberOfAlarms(): int: Número de alarmas.

printAlarmsHtml():String: String con todas las alarmas del objeto en formato html.

getAddress():AbstractIPAddress: Dirección IP del objeto.

getAlarms():Vector: Vector con todos las objetos AbstractAlarms correspondientes.

- **CostInspector**: Clase que contabiliza el coste global de las alarmas en memoria, teniendo en cuenta que si se supera el coste se pondrá el sistema en desconexión y enviará un email de notificación del motivo de la desconexión. También controla las fechas de expiración de las alarmas, borrándolas de memoria en el caso de que caduquen.

Atributos

CostMax: Coste máximo o umbral que pueden tener los equipos antes de ser desconectados. El valor por defecto son 2000.

collector: AbstractCollector: Colector donde están almacenadas las alarmas.

Métodos

setCollector(collector: AbstractCollector:) Establece el colector a usar.

run() Crea el hilo de ejecución que comprueba los costes y realiza las acciones asociadas.

setCostMax(costMax :int) Ajusta el valor del umbral o del coste.

4.4. Diseño del Almacenado de las alarmas.

Las alarmas son la unidad básica de información del sistema. Estas se transmiten desde los distintos sensores distribuidos por la red hasta el colector de alertas siendo por este motivo una pieza fundamental del sistema por ello en este bloque se detallará, como son las alarmas empleadas en Descon2, así como su tratamiento y almacenamiento para que esta información pueda ser recuperada rápida y eficientemente.

Esta sección se compone de tres bloques, como se pueden observar en la figura 4.11 Cada bloque tiene una relación con el bloque que le precede a la derecha.

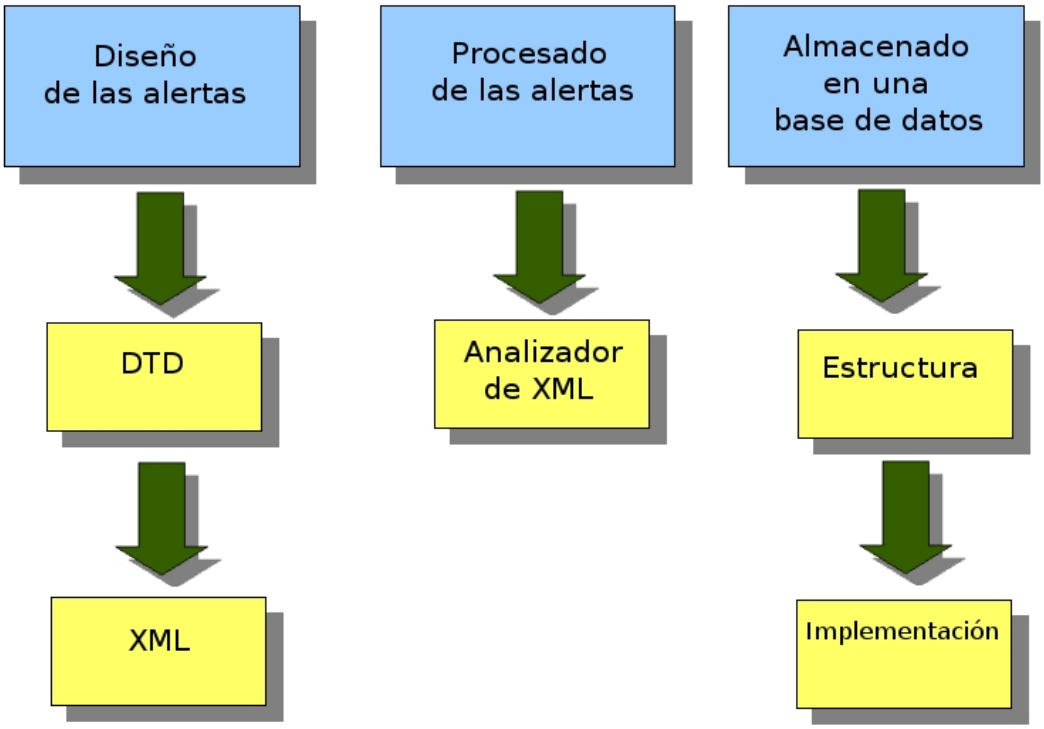


Figura 4.11: Diagrama de bloques del almacenado de alarmas

- El primer bloque se encarga de definir las unidades de información del sistema.
- El segundo bloque mostrará como el sistema obtiene una de estas unidades y procesa la información para adaptarla al tipo de información que el colector o la base de datos usada reconoce.
- El tercer bloque se mostrará el tipo de base de datos usada, su estructura y su relación con el lenguaje de programación usado en este proyecto.

4.4.1. Diseño de las alertas

Para el diseño inicial de las alarmas utilizadas en Descon2 se definió el uso de un DTD (Document Type Definition) para el intercambio de alertas entre los detectores y el colector de alertas.

La definición de tipo de documento (DTD) es una descripción de estructura y sintaxis de un documento XML o SGML. Su función básica es la descripción del formato de datos, para usar un formato común y mantener la consistencia entre todos los documentos que utilicen la misma DTD. De esta forma, dichos documentos, pueden ser validados, conocen la estructura de los elementos y la descripción de los datos que trae consigo cada documento, y pueden además compartir la misma descripción y forma de validación dentro de un grupo de trabajo que usa el mismo tipo de información.

En Descon2 se decidió la utilización de XML, debido a su versatilidad, XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y ofreciendo unas prestaciones mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Entre las múltiples ventajas de utilizar XML, para Descon2 nos resultaron útiles las siguientes:

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda modificar las características de la información procesada sin problemas excesivos.

- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan errores y se acelera el desarrollo de aplicaciones.

- Es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones.

Para la definición de de la DTD, se pensó utilizar el estándar IDMEF que es un formato común para alertas IDS. Este es una especificación basada en XML para un formato de alertas de intrusión. Este estándar es demasiado completo y añadía un nivel de complejidad innecesario para la versión inicial de Descon2, por lo cual se desarrollo una versión reducida de este con las siguientes niveles:

Una clase de nivel superior llamada **Alarm**, doce subclases:

- **ipsrc:** Dirección IP de la máquina que ha producido la alerta.

- **alarmid:** Identificación de la alarma en el sistema.

- **cost:** Peso o coste que supone la alerta al sistema.

- **alarmtime:** Hora a la que se ha producido la alarma en el sistema.

- **description:** Descripción de la alerta producida.

- **expiretime:** Tiempo que la alarma puede estar en el sistema siendo válida.

- **ipdst:** Dirección IP de destino de la alarma.

- **portdst:** Puerto de destino detectado en la alarma.

- **portsrc:** Puerto origen detectado en la alarma.

- **sensorid:** Identificador del sensor que ha creado la alarma.

- **url:** Dirección con información adicional sobre el tipo de alerta producida.

Estas subclases, pueden modificarse, añadiéndolas o eliminándolas dependiendo de las necesidades del sistema.

En la figura 4.12 siguiente vemos un ejemplo de una alerta en formato XML.

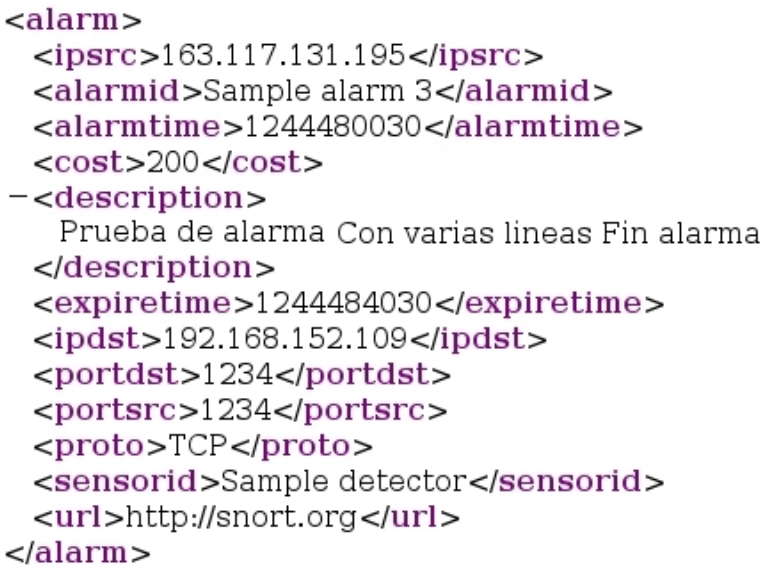


Figura 4.12: Ejemplo de una alerta en XML

4.4.2. Obtención y procesado de las alertas.

Para la obtención de las alertas se decidió usar TCP por ser un protocolo fiable y orientado a conexión ya que la información que circula es sensible a perdidas. TCP usa el concepto de número de puerto para identificar a las aplicaciones emisoras y receptor, para Descon2 se decidió utilizar el puerto 1123 en el receptor (colector de alarmas). La elección del puerto 1123, también llamado Murray, es debido a que esta asignado como protocolo de información y aviso.

En Descon2 se ha realizado una aplicación " que escucha " en el puerto 1123 y la pasa al modulo de procesado. Esta aplicación es multihilo y tiene el siguiente funcionamiento (figura 4.13):

- Establece un hilo de ejecución cada vez que se establece una conexión.

- Cuando se captura el programa separa del flujo de datos las alarmas y la convierte a un formato " entendible " por el colector. Este formato "entendible" es un objeto JAVA (AbstractAlarm) del que se hablara con mas profundidad mas adelante.
- Se añade la alarma al colector.

Figura 4.13: Esquema de funcionamiento del "escuchador".

La aplicación " escuchador", se ha denominado listener y tiene la siguiente estructura que podemos observar en la figura 4.14.

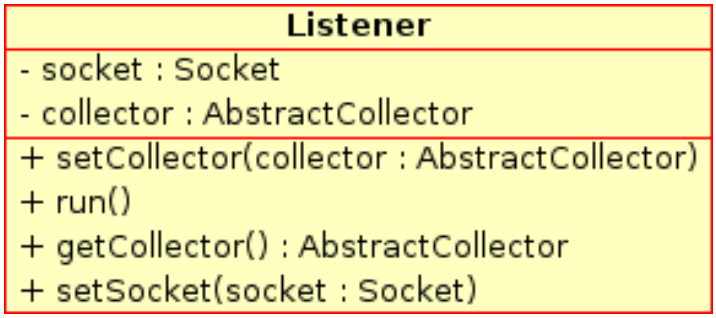


Figura 4.14: Estructura de la clase listener.

4.4.3. Procesado de las alarmas

Para el procesado de ficheros XML hemos utilizado el paquete que proporciona javax.xml.parsers que dispone de métodos para procesar documentos xml. Por este motivo se ha desarrollado un paquete denominado alarm que agrupa todas las funcionalidades de procesado de la información obtenida en formato XML. El paquete tiene la siguiente estructura de clases:

- **AbstractAlarm**: Clase abstracta que representa las características que debe poseer un objeto Alarm.

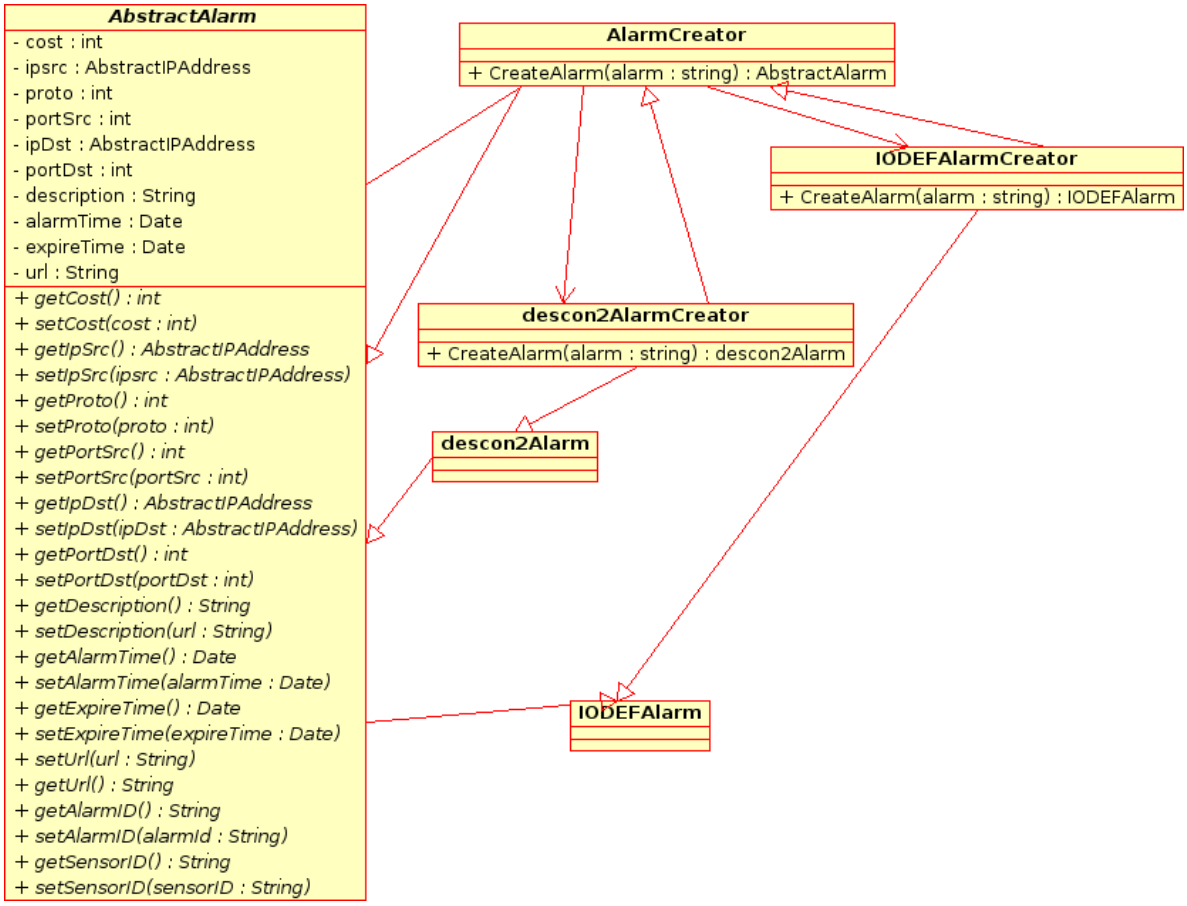


Figura 4.15: Paquete alarm

- **descon2AlarmCreator y IODEFAlarmCreator**: Son distintas implementaciones de AbstractAlarm.

- **AbstractAlarmCreator**: Es otra clase Abstracta que establece las condiciones necesarias para implementar un creador de alarmas de los distintos tipos.

  Métodos

  createAlarm(alarm: string) AbstractAlarm : Crea distintos tipos de objetos dependiendo del tipo de alarma a crear.

- **descon2AlarmCreator y IODEFAlarmCreator** : Implementaciones de AbstractAlarmCreator.

4.4.4. Base de datos

Debido a la gran información que se genera en el sistema, es necesario poder almacenarla de una manera eficiente y que se tenga un acceso sencillo a ella. Permitiendo hacer consultas sobre cualquier atributo de las alertas o las desconexiones. Además, se generaran informes periódicamente almacenados en formato XML en la base de datos, para su posterior consulta. Estos informes pueden ser enviados por correo electrónico o utilizados para detectar tendencias o posibles patrones de ataques.

También se empleara una base de datos para mantener los usuarios del sistema y los grupos que gestionan, implementando el control de accesos.

Selección de la base de datos.

De las diferentes opciones que existen en el mercado se decidió usar MySQL de Sun MicroSystems, MySql es un sistema de gestión de base de datos relacional, multihilo y multiusuario que además se distribuye como software libre. Además existe una API que permite acceder a las bases de datos MySQL siendo la implementación del driver de conexión nativo en JAVA.

Para simplificar la estructura, el sistema se dividió en dos bases de datos, una denominada descon2 encargada de las alertas y desconexiones,y otra llamada UserGroups encargada del control de accesos.

4.4.5. Control de las alertas y las desconexiones

Para el control de alertas se ha diseñado una tabla que almacena directamente la información de las alertas como muestra la figura 4.16.

Para el control de los métodos de desconexión se establecen cinco tablas correspondientes a los métodos de conexión de los usuarios, por lo cual se corresponden a los posibles métodos de desconexión:

- Método 0. Empleado para sistemas externos a la red UC3M, consistirá en bloquear el acceso a la red UC3M empleando un cortafuego perimetral. Además, utilizando la información contenida en la base de datos, se puede notificarse al IRIS-CERT

los sistemas pertenecientes a la red académica española que hemos bloqueado por estar presuntamente comprometidos.

- Método 1. Empleado para sistemas conectados a la red UC3M a través de un concentrador o conmutador no gestionado por el Área de Seguridad y Comunicaciones. En este caso se bloquearán el trafico del equipo comprometido en el conmutador mas cercano al mismo, gestionado por AsyC.

- Método 2. Empleado para sistemas conectados a la red UC3M a través un conmutador gestionado por AsyC. En este caso, el sistema comprometido es el único al que se accede a la red a través de del puerto en el que esta conectado y podremos cambiar su VLAN a la de cuarentena. Este método es el método preferido ya que ofrece todas las funcionalidades.

- Método 3. Empleado para sistemas conectados a UC3M y que no se han podido bloquear empleando los métodos 1 y 2 , consiste en bloquear en el conmutador central el trafico del sistema comprometido.

- Método 4. Empleado para sistemas conectados a través de la red WiFi, en la que no es posible aplicar cuarentena, se ha desligado del método 3 por en un futuro es posible aplicar cuarentena.

- Método 5. Empleado para las conexiones eduroam [3] y VPN. En este caso, la desconexión consiste en bloquear el acceso en el servidor Radius que permite la conexión a eduroam o a los túneles VPN.

[3]eduroam: Es un mecanismo de conexión a redes WIFI para instituciones de investigación. Mas información en www.eduroam.es

Estructura de la base de datos Descon2.

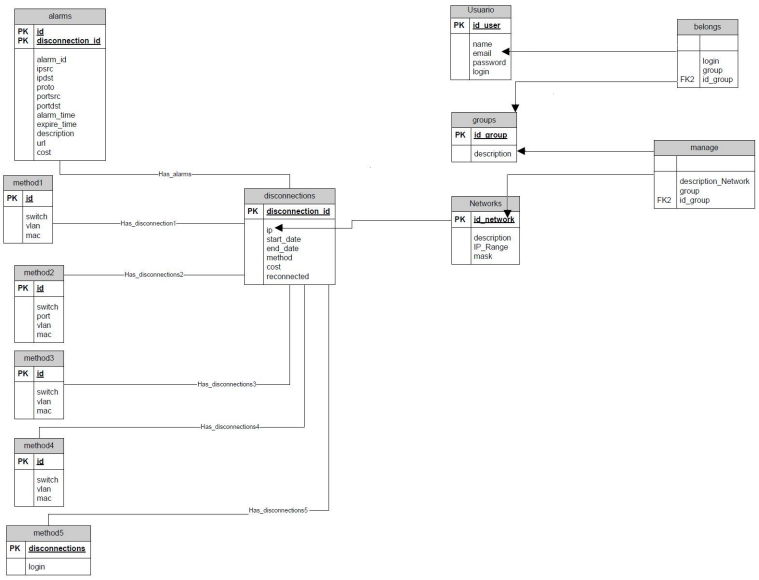


Figura 4.16: Desconexiones

4.4.6. Control de accesos

Como se ha comentado anteriormente se ha desarrollado una base de datos encargada del control de accesos. Esto permitirá poder controlar que usuarios, pueden acceder a los distintos servicios y a la información almacenada en Descon2.

Para ello se ha definido el uso de una tabla correspondiente a los usuarios, otra correspondiente a los grupos de usuarios y una ultima correspondiente a las redes existentes en la organización.

Para establecer las relaciones entre las distintas tablas, se han utilizado dos tablas adicionales que relacionan los usuarios con los grupos y los grupos con las redes que pueden controlar.

Así cuando un usuario se valide en la aplicación se comprobará sus permisos y se le mostrara la información que le corresponda.

Tablas de la base de datos de usuarios

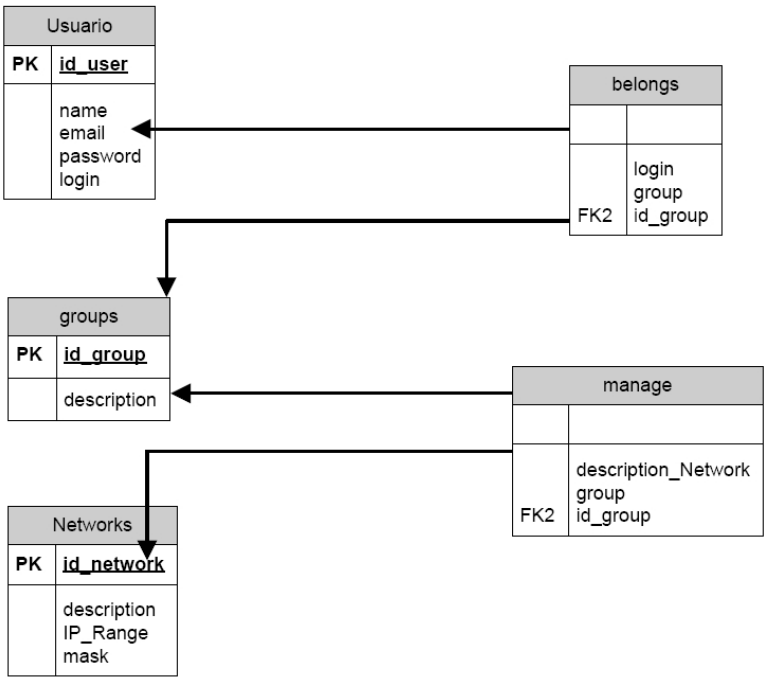


Figura 4.17: Estructura de userGroups

4.4.7. Control de accesos, el caso de la UC3M

En el caso de la UC3M, además de la tabla de usuarios, se utiliza además el login del usuario de directorio de la universidad LDAP [4]. Para lo cual se establece una conexión con el servidor de LDAP de la UC3M y se comprueba que el usuario existe y su contraseña coincide con la almacenada en el directorio.

[4]LDAP (Lightweight Directory Access Protocol), (Protocolo Ligero de Acceso a Directorios) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

4.5. Aplicación Web

En los primeros días de la World Wide Web, los "sitios Web" consistían en poco mas de un conjunto de archivos de hipertexto ligados que presentaban información mediante textos y gráficos limitados. Conforme el tiempo pasó, el HTML aumento al desarrollar herramientas (por ejemplo, JAVA) que permitieron ofrecer capacidades de calculo junto con información. Nacieron los sistemas y aplicaciones basados en Web (WebApps).

Una WebApps es un conjunto de páginas Web estáticas y dinámicas. Una página Web estática es aquélla que no cambia cuando un usuario la solicita: el servidor Web envía la página al navegador Web solicitante sin modificarla. Por el contrario, el servidor modifica las páginas Web dinámicas antes de enviarlas al navegador solicitante. La naturaleza cambiante de este tipo de páginas es la que le da el nombre de dinámica.

En una WebApp se encuentran una serie de atributos que hay que tener en cuenta a la hora de diseñar.

- **Concurrencia.** Un gran número de usuarios puede tener accesos a la WebApp al mismo tiempo.
- **Desempeño.** Si un usuario de WebApp debe esperar demasiado o le es tedioso trabajar con la aplicación puede decidir dejar de usarla, entonces el trabajo realizado no ha servido para nada.
- **Evolución Continua.** A diferencia del software tradicional, que evolucionan a lo largo de una serie de planeadas liberaciones espaciadas cronológicamente, las aplicaciones Web evolucionan de manera continua.
- **Seguridad.** Puesto que las WebApps están disponibles mediante el acceso a la red, es difícil, limitar la población de usuarios finales que pueden tener acceso a la aplicación. Por ello se deben establecer métodos que protejan el contenido confidencial y ofrecer modos seguros de transmisión de datos, se deben implementar fuertes medidas de seguridad a lo largo de la infraestructura que sustenta una aplicación Web.

4.5.1. Herramientas y tecnología

Para el desarrollo de la aplicación se ha utilizado el uso de servlets, que son objetos que se ejecutan en un servidor o contenedor JEE, especialmente diseñado para ofrecer contenido dinámico desde un servidor web, generalmente HTML. Las ventajas de esta tecnología con respecto a otras como CGI, son las que proporciona el lenguaje JAVA en cuanto a portabilidad, y seguridad , ya que un servlet es una clase JAVA igual que otra y por eso en ese sentido tiene toda las características del lenguaje. Otra ventaja de los servlet son el rendimiento ya que son cargados la primera vez que son llamados en el servidor y quedan cargados en memoria hasta que el programa que los controla los desactiva. De esta manera se minimiza el tiempo de respuesta.

Dentro de las características que presenta la plataforma de desarrollo de servlets podemos numerar:

1. Es independiente de la plataforma en la que se este ejecutando.

2. Ejecución multihilo. Cada una de las peticiones sobre el servlet creará una instancia que se ejecutará de manera independiente. A no ser de que le indiquemos lo contrario. El servlet permanece cargado en memoria por lo que atiende rápidamente las peticiones.

3. Un servlet puede llamar a otro servlet, incluso a métodos de otros servlets. Esto nos permite que un servlet realize balanceado de carga entre diferentes servlets. Además, desde un servlet, podemos redirigir una petición sobre otro servlet (en la misma máquina o en una máquina remota).

4. El servlet puede obtener información acerca de la maquina que ha realizado la petición (IP, puerto, tipo de método de envió: get o post,...).

5. Uno de los problemas del protocolo HTTP es que es un protocolo sin estado. No existe una relación entre las diferentes peticiones HTTP realizadas por un usuario sobre un servidor, sino que tiene que ser el propio servidor el que mantenga esta sesión. Por ejemplo, por si queremos mantener algún tipo de información del

usuario (su identificación). En los servlets podemos utilizar las sesiones y cookies para poder llevar a acabo esto. La única diferencia es que en las sesiones la información del usuario se almacena en el servidor, mientras que con las cookies la información del usuario se almacena en su propia máquina.

6. Conexión a Bases de Datos. A través de los servlets podemos establecer conexiones a diferentes tipos de bases de datos. Esta característica acopla perfecta a los servlets dentro de una arquitectura cliente/servidor (figura 4.18) en 3 capas (cliente - servidor - datos).

7. Generación dinámica de código. Esta es una de las características más utilizadas en los servlets, la generación dinámica de HTML. Esto nos permite que una misma página tenga múltiples salidas o representaciones en cuanto a estructura y contenido atendiendo a las evaluaciones que tome el servlet: ip del usuario, información de una base de datos, fecha del sistema,....

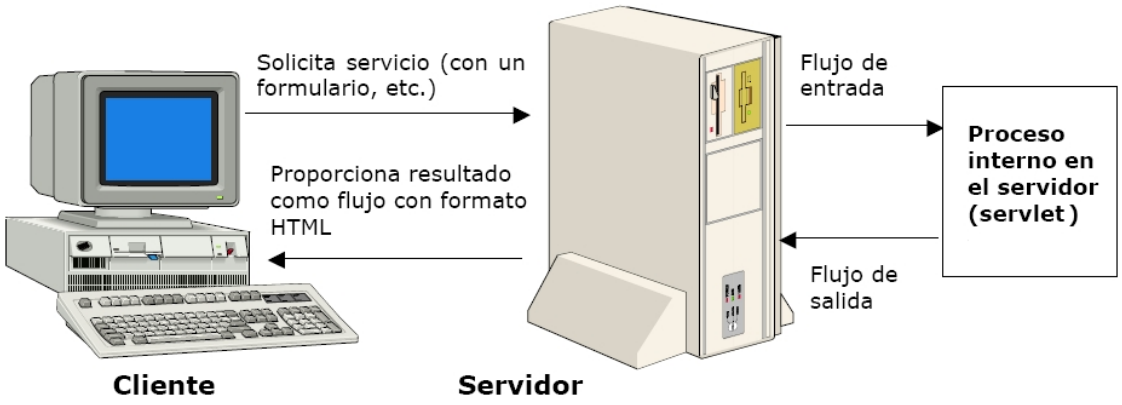


Figura 4.18: Arquitectura Cliente - Servidor interactiva para la Web

4.5.2. Arquitectura de la aplicación Web

Para la arquitectura de una aplicación Web se suelen usar unos patrones de diseo que pueden facilitar un diseño apropiado.

Uno de los patrones que ha demostrado ser fundamental a la hora de diseñar aplicaciones web es el Modelo-Vista-Control (MVC). Este patrón propone la separación en distintos componentes de la interfaz de usuario (vistas) y el modelo de control (Figura 4.19).

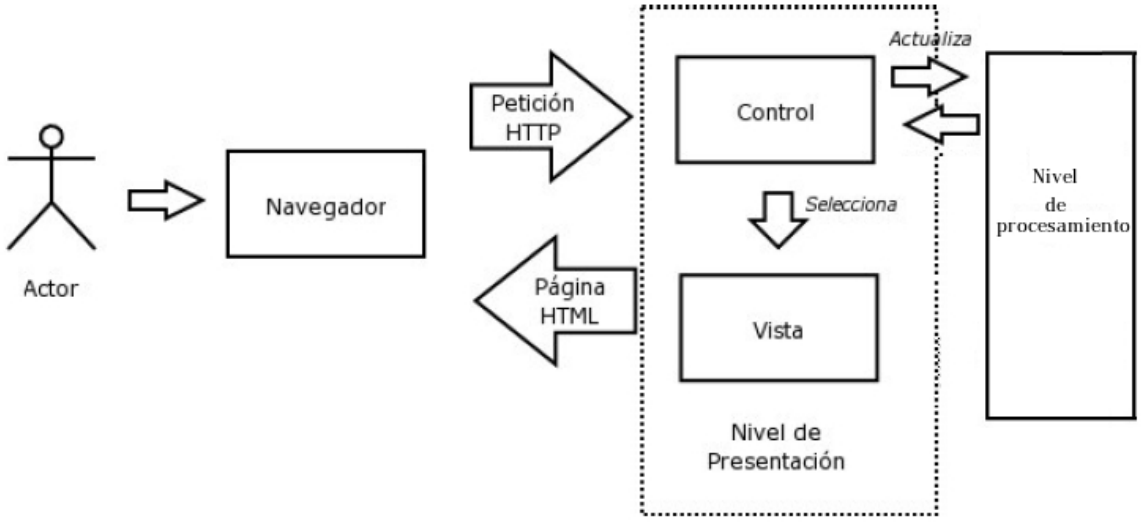


Figura 4.19: Esquema del modelo MVC

Una vista es una "fotografía" del modelo (o una parte del mismo) en un determinado momento. El modelo de control recibe un evento disparado por el usuario a través de la interfaz, accede al modelo procesamiento de manera adecuada a la acción realizada, y presenta una nueva vista con el resultado de dicha acción. Por su parte, este modelo de control consiste en el conjunto de objetos que modelan los procesos que se realizan a través del sistema.

En Descon2, las vistas serían las páginas HTML que el usuario visualiza en el navegador. A través de estas páginas el usuario interactúa con la aplicación, enviando eventos al servidor a través de peticiones HTTP. En el servidor se encuentra el código de control para estos eventos, que en función del evento concreto actúa sobre el modelo convenientemente. Los resultados de la acción se devuelven al usuario en forma de página HTML mediante la respuesta HTTP. La clave está en la separación entre vista y modelo.

El modelo suele ser más estable a lo largo del tiempo y menos sujeto a variaciones mientras que las vistas puede cambiar con frecuencia, ya sea por cambio del medio de presentación (por ejemplo HTML a WAP o a PDF) o por necesidades de usabilidad de la interfaz o simple renovación de la estética de la aplicación. De esta manera se cumplen los requisitos que se establecieron en el capitulo anterior.

Si tomamos como referencia a la plataforma J2SE utilizada en Descon2, las vistas serán JSPs los controladores serían servlets y el modelado de los procesos serían objetos Java normales que trabajan en combinación con los servlets.

La utilización del MVC en aplicaciones web recomienda utilizar un único servlet como controlador para toda la aplicación y así es como se ha hecho en Descon2. Este control gestiona todas las peticiones, incluyendo invocaciones a servicios de seguridad, gestión de excepciones, selección de la siguiente vista, etc. Esto también se conoce como el patrón Front Controller (controlador frontal o fachada).

El poder centralizar en un solo punto servicios como la gestión de conexiones a base de datos, comprobaciones de seguridad o gestión de errores favorecen que la aplicación sea mucho más robusta y aisla de todos estos aspectos al resto de componentes.

Para simplificar el desarrollo de la aplicación Web, se desarrolló una unica aplicacion Web con tres usos principales:

- La aplicación, control de alertas y localización.
- Control y visualización de la información de la base de datos.
- Administración de usuarios.

Cada una de las tres aplicaciones se compone de los tres modelos comentados anteriormente (vista, control, procesado).

- **Control de alertas y localización.**
  - **El modelo de vista** se compone de una serie de paginas HTML , JSP's y hojas de estilos CSS que dan forma a la interfaz del sistema.

    En las siguentes capturas se muestran:

En la figura 4.20 se muestra la pagina de entrada a la aplicación, al ser un desarrollo para la UC3M, de ha decidido que siga el estilo corporativo de la institución.

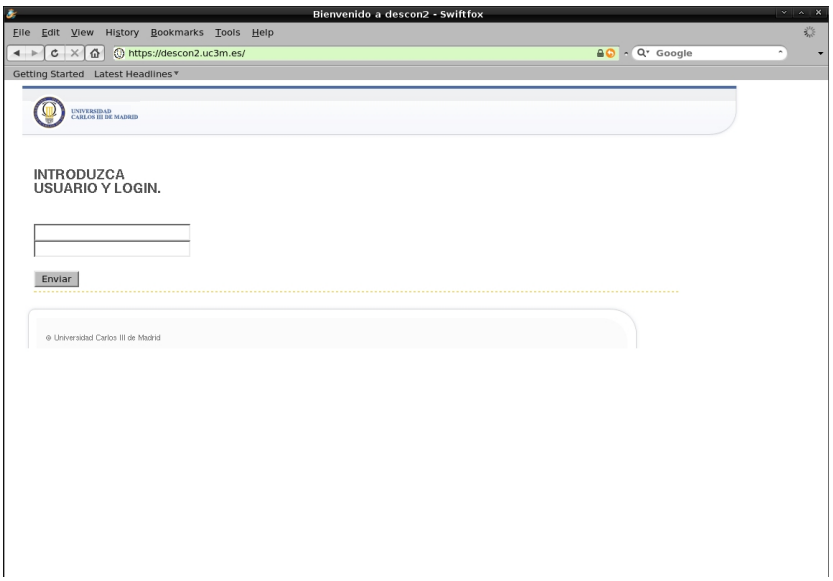


Figura 4.20: Pantalla de entrada a Descon2

Esta siguiente captura 4.21 muestra las distintas funciones de la aplicación, como se puede observar en el marco superior de la ventana:

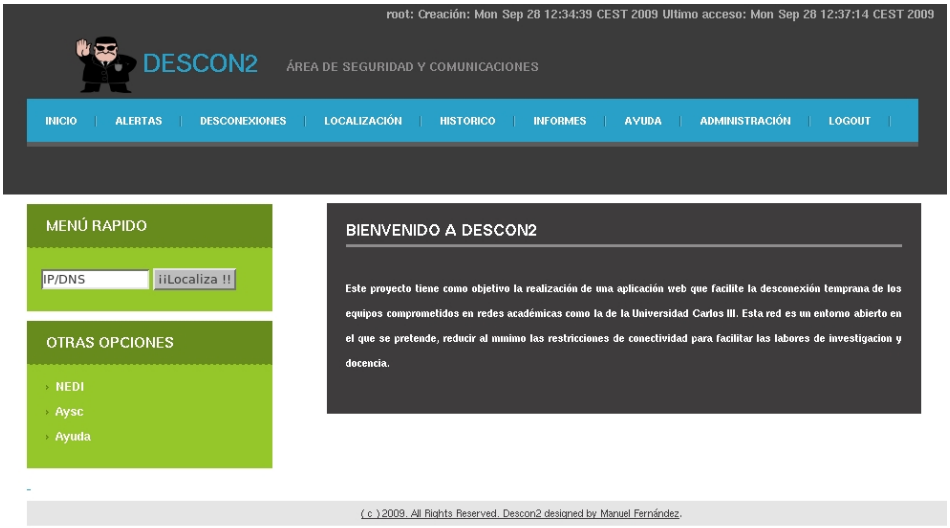


Figura 4.21: Menú de Descon2

La figura 4.22 muestra en el marco izquierdo un ejemplo de como se muestran las alertas en la aplicación.

Figura 4.22: Ejemplo de vista de las alertas

La figura 4.23 muestra el ejemplo de una alerta en el sistema.

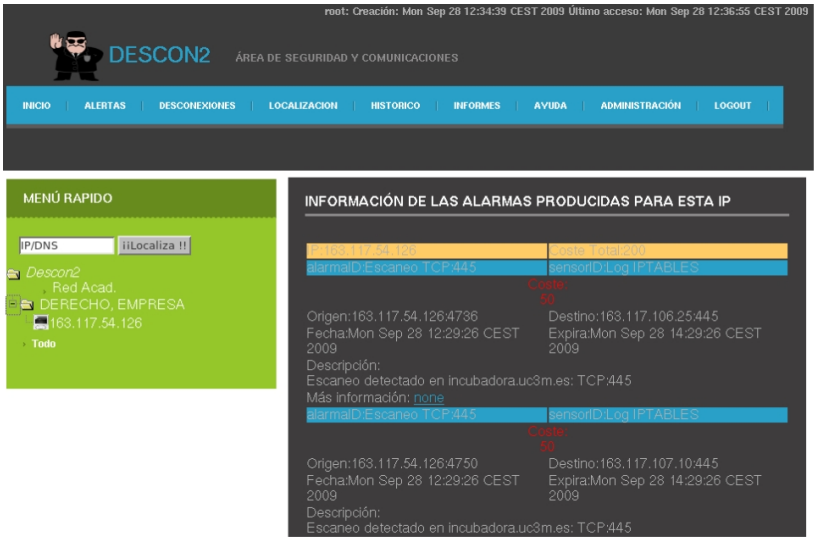


Figura 4.23: Ejemplo de alertas

La figura 4.24 muestra el menú de acceso al aplicación de muestra de información de la base de datos.

En las siguientes figuras se muestran ejemplos de distintas localizaciones, tanto de usuarios conectados a la red cableada (figura:4.25 como para usuarios conectados a Eduroam(figura: 4.26).

Figura 4.24: Vista de la Base de datos

Figura 4.25: Ejemplo del resultado de una localización

- **El modelo de control** para este bloque se han diseñado un servlet que maneja todas las peticiones hechas por el navegador cuando el usuario teclea una URL en la línea de direcciones, sigue un enlace desde una página Web, o rellena un formulario que no especifica un METHOD. El Servlet también maneja peticiones POST, que son generadas cuando se crea un formulario HTML que especifica METHOD=”POST”.

  Este servlet desde ahora Descon2Servlet se comunica con las distintas clases JAVA (ListenerBean, Locate y UserDB) que gestionan los distintos hilos de

Departamento: RED WIFI PARA CLIENTES 802.1X (EDUROAM)	Despacho: Roseta:
Usuario: 2009062310 mat@math.uc3m.es	
DNS: wifi-81-121.uc3m.es	IP: 163.117.81.121
MAC: 0022.41b7.6426	VLAN: 343
CONMUTADOR: B01014Z.uc3m.es	PUERTO: Po285
CADENA DE CONMUTADORES: B01014Z.uc3m.es(163.117.49.4)	LISTA DE MACS: Número de MACS:46 0012.f061.b9ea 0013.ce28.2c89 0013.e8df.94b7 0014.517c.a78b 0015.afb6.4a5f 0016.44dc.7be2 0016.6f57.7c21 0016.6fb2.e683 0016.ea9a.98fa

Figura 4.26: Ejemplo del resultado de una localización para el caso de Eduroam

ejecución y con los modulo de procesado.

ListenerBean, se encarga conectar con los módulos de procesado referentes a la escucha de las alarmas, contabilización del coste y obtención de la información de la base de datos.

Locate, activa y recoge los resultados de los módulos de localización.

UserDB, enlaza la información de los usuarios con el sistema.

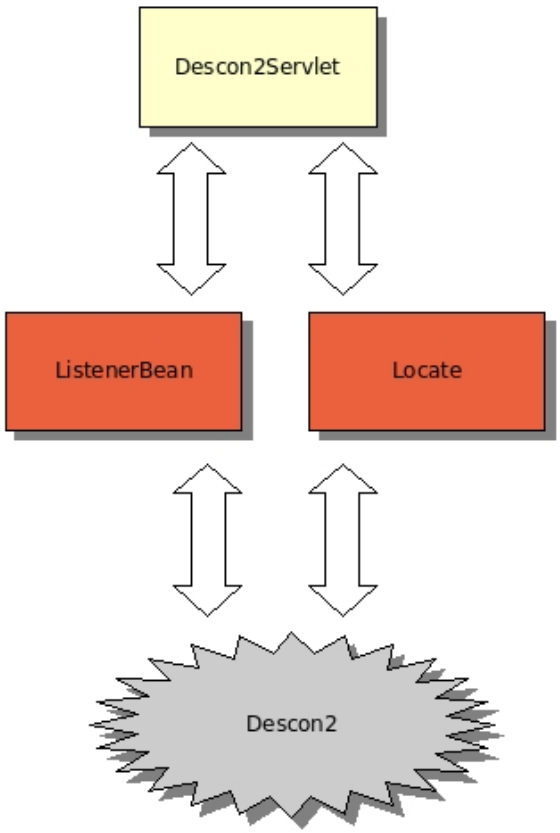


Figura 4.27: Estructura Descon2Servlet

El funcionamiento de Descon2Servlet consta de los siguientes pasos:

1. Un método init, que crea objetos del tipo Listener Bean y Locate, para así conseguir que la instanciación de estos objetos se produzca la primera vez que el servlet es cargado en memoria por parte del servidor de aplicaciones. Y así conseguir que con solo arrancar el servidor de aplicaciones Descon2 ya este preparado para recibir y procesar alertas. Este método también crea un hilo de ejecución que permite encargarse de la escucha y contabilización de coste.

2. Unos métodos doGet o doPost, que controla, las peticiones de la aplicación y obtienen las respuestas. La estructura de estos métodos es la siguiente: Una serie de condiciones que discriminan entre las distintas peticiones que pueden ocurrir en la aplicación. Las distintas peticiones 4.28 están

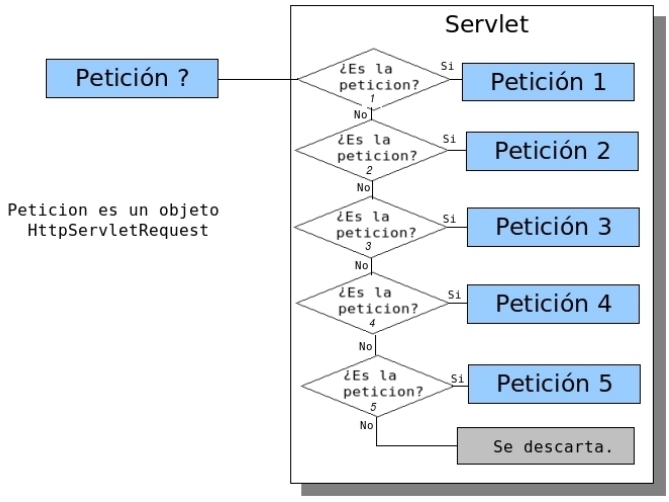


Figura 4.28: Flujograma de Descon2Servlet

representadas por el objeto HttpServletRequest, que permite obtener los parámetros adjuntos a la petición.

4.5.3. Usabilidad de la pagina Web

Para empezar este apartado se intentara introducir el concepto de usabilidad (del inglés usability).

La usabilidad es la característica de un sistema que pretende ser utilizado por:

- El tipo o tipos específicos de usuario/s,
- La tarea o tareas que para las cuales el sistema se ha hecho, y
- El contexto en el que se da la interacción.

El "grado de usabilidad"de un sistema es, por su parte, una medida empírica y relativa de la usabilidad del mismo.

- Empírica porque no se basa en opiniones o sensaciones sino en pruebas (del inglés tests) de usabilidad, realizadas en laboratorio u observadas mediante trabajo de campo.
- Relativa porque el resultado no es ni bueno ni malo, sino que depende de las metas planteadas (por lo menos el 80 % de los usuarios de un determinado grupo o tipo definido deben poder instalar con éxito el producto X en N minutos sin más ayuda que la guía rápida) o de una comparación con otros sistemas similares.

El concepto de usabilidad se refiere a una aplicación (informática) de (software) o un aparato (hardware), aunque también puede aplicarse a cualquier sistema hecho con algún objetivo particular.

El modelo conceptual de la usabilidad, proveniente del diseño centrado en el usuario, no está completo sin la idea utilidad. En inglés, utilidad + usabilidad es lo que se conoce como usefulness.

Jackob Nielsen [21]una de las personas mas respetadas en cuanto a usabilidad web definió Usabilidad como el atributo de calidad que mide lo fáciles que son de usar las interfaces Web

Los principales beneficios son:

- Reducción de los costes de aprendizaje.
- Disminución de los costes de asistencia y ayuda al usuario.
- Optimización de los costes de diseño, rediseño y mantenimiento.

- Mejora la imagen y el prestigio.
- Mejora la calidad de vida de los usuarios, ya que reduce su estrés, incrementa la satisfacción y la productividad

Evaluación:

Para la evaluación de nuestra aplicación Web se ha utilizado dos tipos de de evaluación, la heurística y la de los usuarios.

Para la evaluación heurística se ha utilizado los 10 heurísticos de Nielsen, enumerados aquí :

- Visibilidad del estado del sistema
- Emparejamiento entre el sistema y el mundo real
- Control y libertad del usuario
- Consistencia y estándares
- Prevención de errores
- Reconocimiento sobre recuerdo
- Flexibilidad y eficiencia de uso
- Estética y diseño minimalista
- Ayudar a reconocer, diagnosticar y solucionar errores
- Ayuda y documentación

Para la evaluación con usuarios:

Se deberá tomar una muestra de cinco personas y se les pide que realicen varias tareas, en ello se registran siempre dos aspectos: el rendimiento y la opinión de los sujetos. En el rendimiento se observa la consecución de tareas. En cuanto a la opinión se usara el uso de un cuestionario. Este proyecto ha realizado el cuestionario pero el asunto de la usabilidad no estaba en los requisito principales de este proyecto, ya que

el objetivo era obtener un prototipo funcional. Por ese motivo no se ha podido realizar la evaluación, de la que se hablara con mas detalle en las lineas futuras.

Capítulo 5

Evaluación y pruebas

A lo largo de todo el proyecto hemos explicado en profundidad cada uno de los pasos que se llevará a cabo a la hora de realizar cada una de las posibles funciones del sistema.

A continuación, podremos ver todos los procesos descritos mediante las pruebas.

El objetivo que se pretende conseguir en este apartado es demostrar que las funcionalidades se realizan correctamente y funcionan tal y como se especifica en los requisitos. Para ello se definirán varios casos de pruebas que comprenderán toda la funcionalidad descrita y se mostrarán los resultados obtenidos.

5.1. Proceso de evaluación

En la siguiente tabla se almacenan los datos de los casos de pruebas que hemos determinado necesarios para demostrar el cumplimiento de los requisitos. Veremos únicamente el resumen de los casos de pruebas. A continuación se mostraran las tablas con los resultados de los casos de pruebas concretos por cada módulo de pruebas, de esta manera comprobaremos el correcto funcionamiento del programa.

Identificador del Caso de Prueba	CP1	CP2	CP3
Módulo a probar	Localización	Alertas	Administración
Descripción del caso	Localizar un equipo	Tratamiento de alertas	Administración
Resultado esperado	Muestra del resultado	Muestra del resultado	Modificación de la configuración
Resultado obtenido	Ok	Ok	Ok

Prueba CP1

En este caso la prueba describe la funcionalidad necesaria para localizar un equipo en la red.

CP1: Localización de un equipo	
Prerrequesitos	El rol de usuario debe ser de administrador de red y el equipo debe estar en la red
Datos de entrada	Se le pasa como parámetro la IP o nombre de DNS
Datos de salida	Devuelve una tabla con la información de red del equipo buscado
Conclusiones	Localización efectuada correctamente.

Prueba CP2

Este módulo se compone de tres pruebas, ver alertas en tiempo real, ver desconexiones producidas y ver la información de la base de datos.

En el siguiente caso, la prueba 2.1, la podemos ver en la siguiente tabla:

CP2.1: Ver alertas actuales	
Prerrequesitos	El rol de usuario observará las alertas correspondientes a las redes que controle
Datos de entrada	No hay datos de entrada
Datos de salida	Devuelve un desplegable organizado por carpetas con los equipos con alertas.
Conclusiones	Muestra los equipos que han producido alertas, y sus alertas satisfactoriamente.

La siguiente prueba 2.2 será el visionado de los equipos que deberían estar en la red de cuarentena.

CP2.2: Ver desconexiones	
Prerrequesitos	El rol de usuario observará las alertas correspondientes a las redes que controle
Datos de entrada	No hay datos de entrada
Datos de salida	Devuelve una tabla con la IP y el departamento al que pertenece.
Conclusiones	Muestra los equipos que han superado el coste, correctamente.

Y la última prueba del segundo bloque se muestra en la siguiente tabla:

CP2.3: Ver información de la Base de Datos	
Prerrequesitos	El rol de usuario observará las alertas correspondientes a las redes que controle
Datos de entrada	No hay datos de entrada
Datos de salida	Devuelve un desplegable organizado por carpetas con los equipos con alertas de la BD.
Conclusiones	Muestra las alertas de la BD como se pedía.

Prueba CP3

Ahora mostraremos las distintas pruebas de Administración del sistema, para ello realizaremos solo la prueba de administrar un usuario. Para administrar una red o un grupo son esencia la misma prueba, por eso se tratara todo como una prueba única que veremos en la siguientes tablas:

CP3.1: Alta de usuario	
Prerrequesitos	El rol de usuario debe ser administrador
Datos de entrada	Login, Nombre, Email, Contraseña (opcional) y el tipo de usuario
Datos de salida	Devuelve un mensaje del resultado de la operación.
Conclusiones	Hace correctamente la función.

CP3.1: Eliminación de un usuario	
Prerrequesitos	El rol de usuario debe ser administrador
Datos de entrada	Se pide el login del usuario a borrar
Datos de salida	Devuelve un mensaje del resultado de la operación.
Conclusiones	Hace correctamente la función.

CP3.1: Modificación de usuario	
Prerrequesitos	El rol de usuario debe ser administrador
Datos de entrada	Login, Nombre, Email, Contraseña (opcional) y el tipo de usuario.
Datos de salida	Devuelve un mensaje del resultado de la operación.
Conclusiones	Hace correctamente la función.

CP3.1: Visualización de un usuario	
Prerrequesitos	El rol de usuario debe ser administrador
Datos de entrada	No hay datos de entrada
Datos de salida	Muestra una tabla con la información del usuario
Conclusiones	Hace correctamente la función.

5.2. Análisis de los resultados

Una vez realizadas las pruebas anteriores, podemos comprobar que las distintas tareas disponibles en la aplicación se realizan correctamente. De los casos de pruebas realizados en el apartado anterior (5.1 Proceso de evaluación), se han podido extraer las siguientes conclusiones.

Es muy importante tener en cuenta el nivel de permisos de usuario que disponemos para que el sistema nos devuelva el resultado que esperamos, ya que si tenemos un nivel inferior al que el sistema demanda no se podrá hacer la consulta. Por lo tanto sería interesante que en el futuro se indique en la aplicación el nivel del usuario al lado del login.

CASO DE PRUEBA	CONCLUSIÓN
CP1: LOCALIZACIÓN	
PRUEBA 1	OK
CP2: TRATAMIENTO ALERTAS	
PRUEBA 2.1	OK
PRUEBA 2.2	OK
PRUEBA 2.3	OK
CP3: ADMINISTRACIÓN	
PRUEBA 3.1	OK

Capítulo 6

Costes y duración del proyecto.

El objetivo de este capitulo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado así como el coste del proyecto.

Para el análisis de duración se utilizara, un diagrama de Gantt que indica, la posición de cada tarea a lo largo del tiempo para así poder identificar las relaciones e independencias entre las tareas. Para la determinación del coste, solo se tendrán en cuenta el coste de los recursos humanos. El coste sera asociado a cada tarea.

6.1. Análisis de duración.

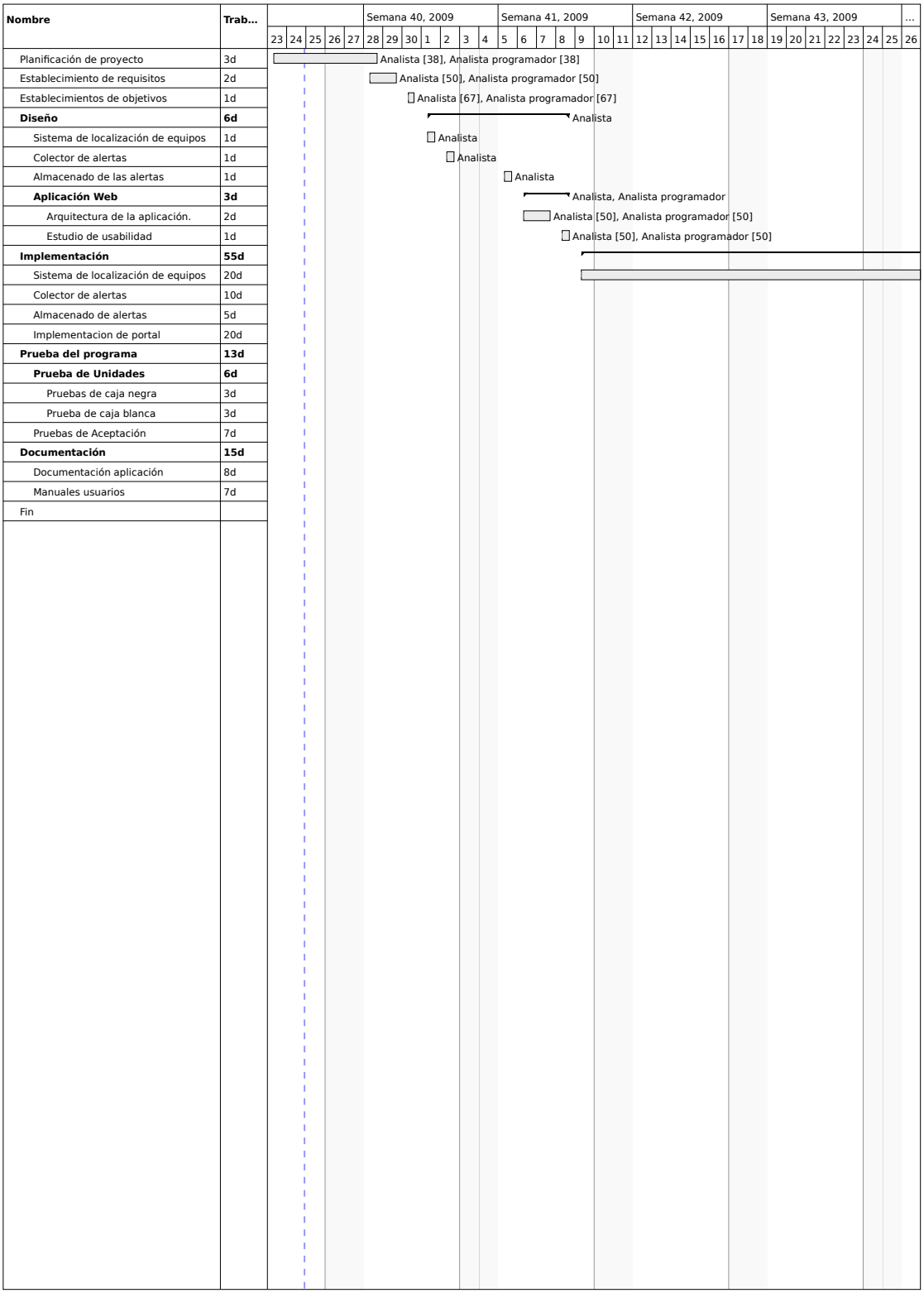
En las figuras siguientes se muestra el diagrama de gantt de la duración del proyecto.

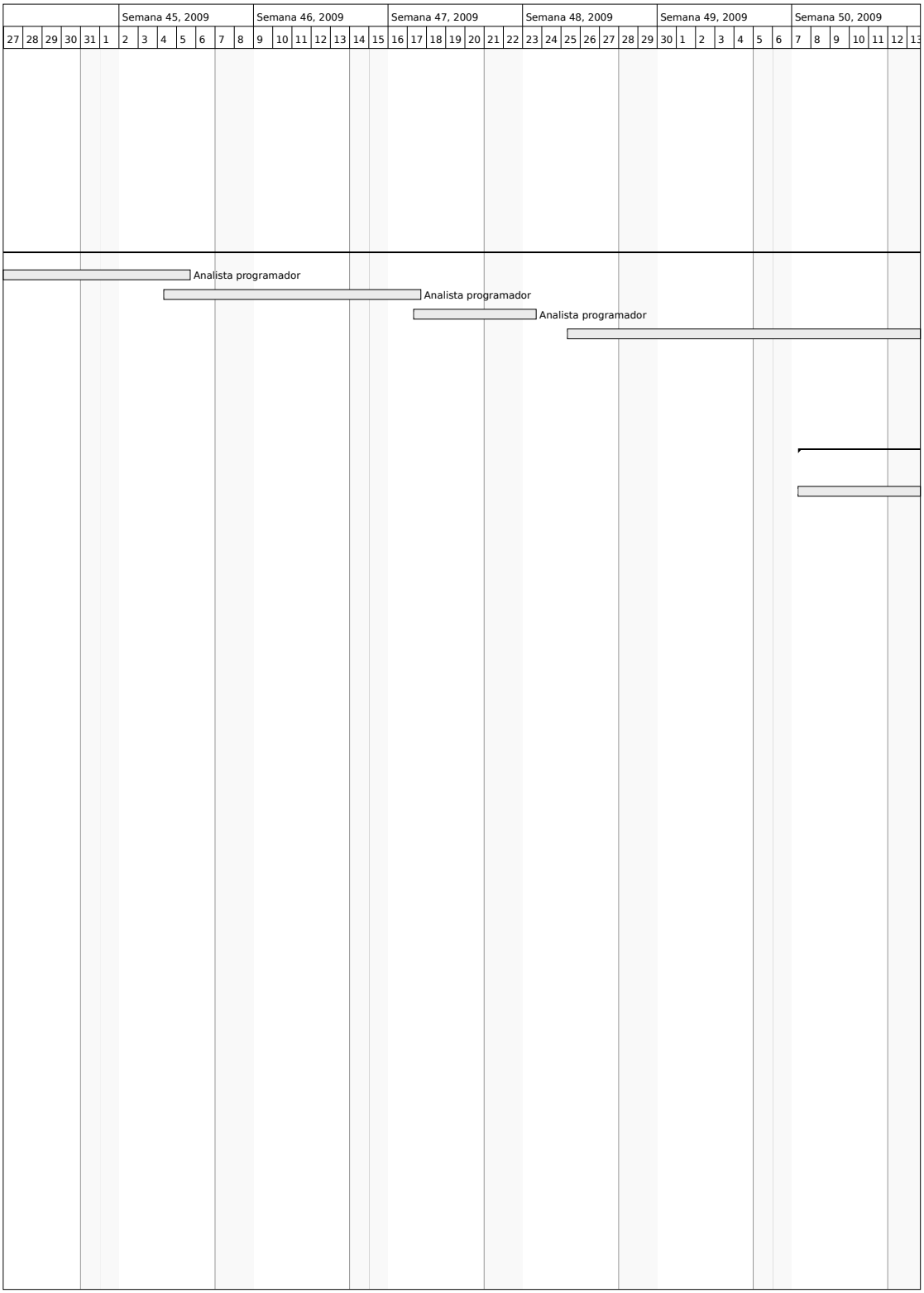
6.2. Análisis de costes.

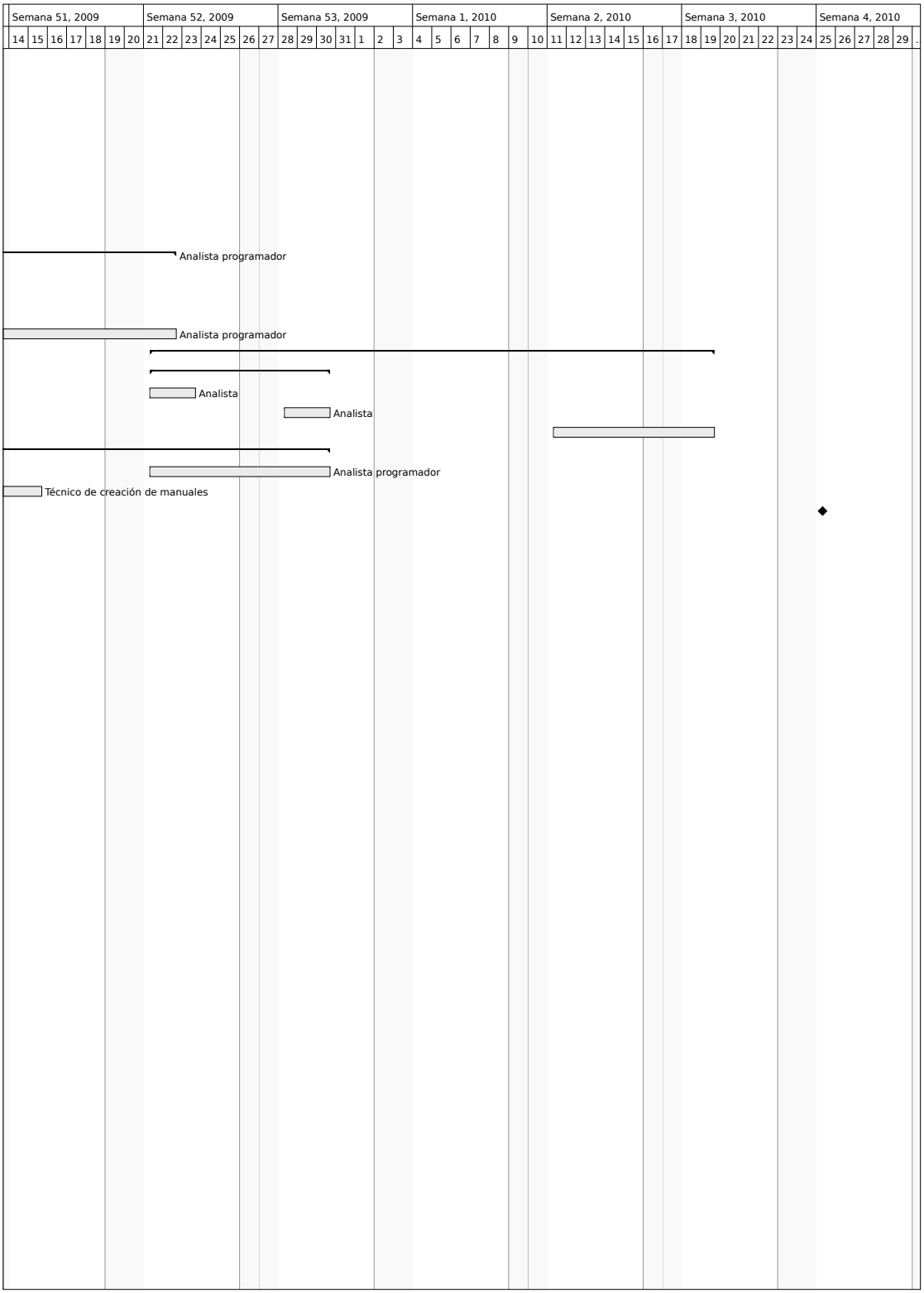
En la siguiente tabla 6.1 se observan los recursos humanos necesarios y el coste por hora.

En la figura 6.2 se muestra el coste asociado a las diferentes tareas del proyecto.

El coste total del proyecto será sin incluir gastos externos de material y equipos de desarrollo de *diez mil ochocientos veintiseis euros.*


Nombre
Trab...
Semana 40, 2009
Semana 41, 2009
Semana 42, 2009
Semana 43, 2009
...
23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
Planificación de proyecto 3d
Establecimiento de requisitos 2d
Establecimientos de objetivos 1d
Diseño 6d
Sistema de localización de equipos 1d
Colector de alertas 1d
Almacenado de las alertas 1d
Aplicación Web 3d
Arquitectura de la aplicación. 2d
Estudio de usabilidad 1d
Implementación 55d
Sistema de localización de equipos 20d
Colector de alertas 10d
Almacenado de alertas 5d
Implementacion de portal 20d
Prueba del programa 13d
Prueba de Unidades 6d
Pruebas de caja negra 3d
Prueba de caja blanca 3d
Pruebas de Aceptación 7d
Documentación 15d
Documentación aplicación 8d
Manuales usuarios 7d
Fin
Analista [38], Analista programador [38]
Analista [50], Analista programador [50]
Analista [67], Analista programador [67]
Analista
Analista
Analista
Analista
Analista, Analista programador
Analista [50], Analista programador [50]
Analista [50], Analista programador [50]


Semana 45, 2009
Semana 46, 2009
Semana 47, 2009
Semana 48, 2009
Semana 49, 2009
Semana 50, 2009
27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13
Analista programador
Analista programador
Analista programador


Semana 51, 2009
Semana 52, 2009
Semana 53, 2009
Semana 1, 2010
Semana 2, 2010
Semana 3, 2010
Semana 4, 2010
14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
Analista programador
Analista programador
Analista
Analista
Analista programador
Técnico de creación de manuales

Nombre	Nombre corto	Tipo	Grupo	Correo-e	Coste
Analista programador		Trabajo			15
Analista		Trabajo			20
Técnico de creación de manuales		Trabajo			10

Figura 6.1: Tabla de recursos y costo por hora

WBS	Nombre	Inicio	Fin	Trabajo	Duración	Coste	Asignado a	% Completado
1	Planificación de proyecto	sep 23	sep 28	3d	4d	425,6	Analista, Analista programador	0
2	Establecimiento de requisitos	sep 28	sep 29	2d	2d	280	Analista, Analista programador	0
3	Establecimientos de objetivos	sep 30	sep 30	1d	6h	140,7	Analista, Analista programador	0
4	**Diseño**	**oct 1**	**oct 8**	**6d**	**6d**	**900**	Analista	0
4.1	Sistema de localización de equipos	oct 1	oct 1	1d	1d	160	Analista	0
4.2	Colector de alertas	oct 2	oct 2	1d	1d	160	Analista	0
4.3	Almacenado de las alertas	oct 5	oct 5	1d	1d	160	Analista	0
4.4	**Aplicación Web**	**oct 6**	**oct 8**	**3d**	**3d**	**420**	Analista, Analista programador	0
4.4.1	Arquitectura de la aplicación.	oct 6	oct 7	2d	2d	280	Analista, Analista programador	0
4.4.2	Estudio de usabilidad	oct 8	oct 8	1d	1d	140	Analista, Analista programador	0
5	**Implementación**	**oct 9**	**dic 22**	**55d**	**53d**	**6600**	Analista programador	0
5.1	Sistema de localización de equipos	oct 9	nov 5	20d	20d	2400	Analista programador	0
5.2	Colector de alertas	nov 4	nov 17	10d	10d	1200	Analista programador	0
5.3	Almacenado de alertas	nov 17	nov 23	5d	5d	600	Analista programador	0
5.4	Implementacion de portal	nov 25	dic 22	20d	20d	2400	Analista programador	0
6	**Prueba del programa**	**dic 21**	**ene 19**	**13d**	**22d**	**960**		0
6.1	**Prueba de Unidades**	**dic 21**	**dic 30**	**6d**	**8d**	**960**		0
6.1.1	Pruebas de caja negra	dic 21	dic 23	3d	3d	480	Analista	0
6.1.2	Prueba de caja blanca	dic 28	dic 30	3d	3d	480	Analista	0
6.2	Pruebas de Aceptación	ene 11	ene 19	7d	7d	0		0
7	**Documentación**	**dic 7**	**dic 30**	**15d**	**18d**	**1520**		0
7.1	Documentación aplicación	dic 21	dic 30	8d	8d	960	Analista programador	0
7.2	Manuales usuarios	dic 7	dic 15	7d	7d	560	Técnico de creación de manuales	0
8	Fin	ene 25	ene 25	N/D	N/D	0		0

Figura 6.2: Tabla de costes y tareas

Capítulo 7

Manual de utilizacion del sistema

7.1. Guía de instalación

Este capitulo mostrara el proceso de instalación del sistema en un equipo de gestión. Los configuración de los permisos de gestión que deba tener el equipo dependerá de lo administradores de la red de la institución en la que se implemente.

7.1.1. Requisitos técnicos del sistema

- Un PC con un procesador como mínimo de Pentium 4 o equivalente (Recomendable un procesador de doble núcleo)
- Memoria: 1 Gb de ram. (Recomendable 2 Gb)
- Disco Duro: Tamaño, depende de las necesidades de almacenamiento de las alertas de la institución.
- Sistema operativo, cualquiera con posibilidad de instalación JRE 6.
- Tomcat 5.5
- Base de datos de MySql 5.

Las variaciones de rendimiento variaran dependiendo de las características del equipo. Descon2 ha sido probado en un equipo con las siguientes características:

- Procesador Amd Athlon X2 4200.
- 2 Gb de ram.
- Sistema operativo Debian Etch

7.1.2. Preparación del sistema para Descon2

Para la instalación de Descon2 es necesario previamente a tener instalado el JDK (J2SE), es la base para operar cualquier producto que utiliza "Java.[es] el "JDK"de la plataforma correspondiente, puede encontrar instrucciones y descarga para plataformas Linux así como Windows, en la siguiente dirección:

- JDK para todas las plataformas: *http://java.sun.com/javase/downloads/index.jsp*

También es necesario MySql, para instalarlo Sun, ofrece estos dos enlaces con una guía de instalación y de descarga.

- Instalación: *http://dev.mysql.com/doc/refman/5.0/es/installing-source.html.*
- Descarga: *http://dev.mysql.com/downloads/mysql/5.1.html.*

El siguiente paso seria la instalación de Tomcat.

Lo primero que hay que hacer es bajarse la versión de Tomcat que se desee (Descon2 funciona apartir de la 5.5)

La web de descargas de tomcat es la siguiente:

- Tomcat Descargas: *http://tomcat.apache.org/*

En el momento de redacción de este manual, la versión más nueva de Tomcat 6 es la 6.0.20, cuyo binario se puede descargar de aquí:

- Tomcat 6.0.20 (UNIX, tar.gz) http://apache.mirrors.tds.net/tomcat/tomcat-6/v6.0.20/bin/apach tomcat-6.0.20.tar.gz
- Tomcat 6.0.20 (Windows, .exe) http://apache.mirrors.tds.net/tomcat/tomcat-6/v6.0.20/bin/apac tomcat-6.0.20.exe

El archivo .tar.gz contiene los binarios de tomcat que funcionan en cualquier plataforma, pueden descomprimirse por ejemplo, en /usr/share/tomcat6.0.20.

El archivo .exe es un instalador para Windows.

Si se desea instalar Tomcat en Debian, existe la posibilidad de descargarlo e instalarlo a través de los repositorios oficiales.

Una vez instalado Tomcat, si se quiere configurar Tomcat con SSL, obtendrá más información en el Anexo2.

Para la configuración y creación de la base de Datos necesaria para Descon2, se utilizara el comando de mysqldump, que crea una nueva base de datos, con la estructura almacenada en un fichero. Los ficheros de estructura, están en la carpeta DataBase_maker del paquete de instalación.

7.1.3. Instalación de Descon2 en Tomcat

El instalador de Descon2, es un fichero comprimido tar.gz que se compone de dos carpetas y un fichero war en su interior.

La carpeta Library contiene ocho ficheros de librerías Java.

- AdventNetLogging.jar
- descon2.jar
- mail.jar
- AdventNetSnmp.jar
- dnsjava-2.0.3.jar
- mysql-connector-java-5.1.6-bin.jar
- crimson.jar
- jaxp.jar
- servlet.jar

La carpeta DataBase_maker contiene los ficheros de la estructura de la base de datos.

- descon2.sql
- userGroups.sql

El fichero Descon2.war, es la aplicación WEB.

La aplicación se instala como cualquier aplicación Web en Tomcat, de las que se mostraran un ejemplo, con capturas de pantallas a continuación:

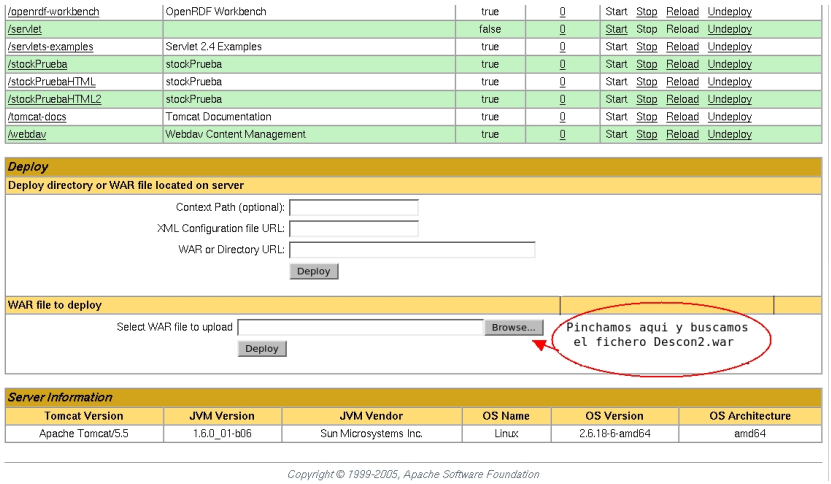


Figura 7.1: Paso 1 de la instalación del sistema en Tomcat

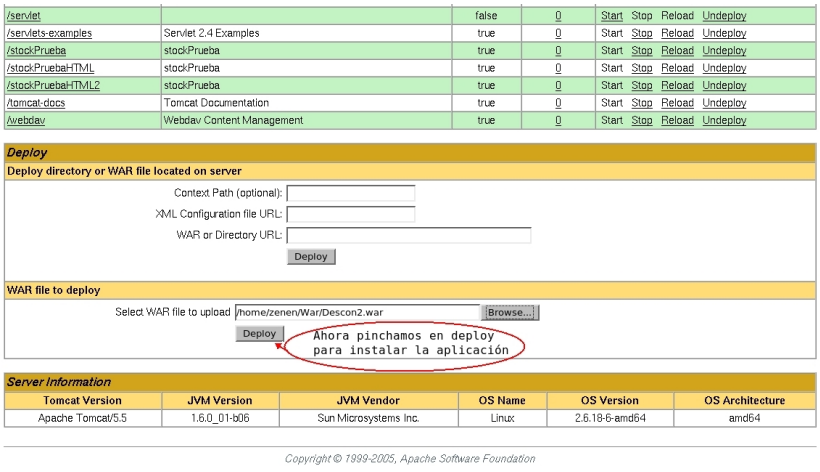


Figura 7.2: Paso 2 de la instalación del sistema en Tomcat

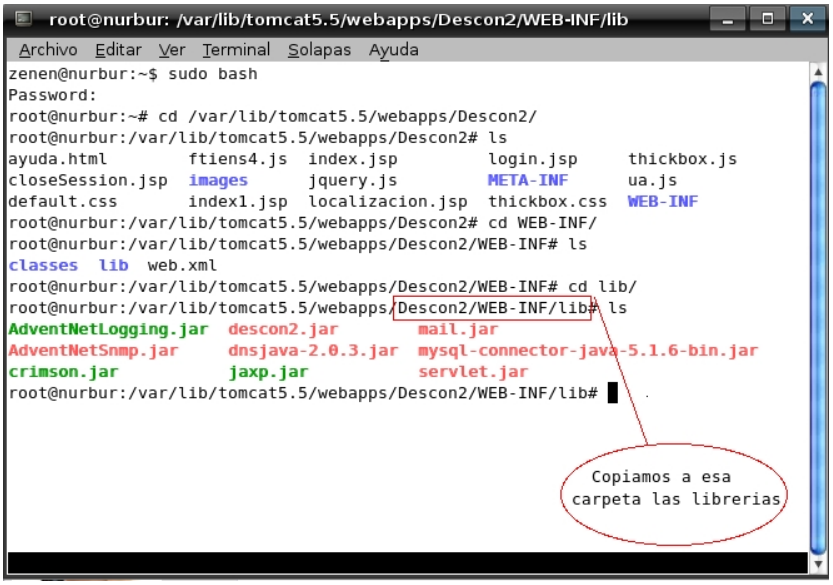


Figura 7.3: Paso 3 de la instalación del sistema en Tomcat

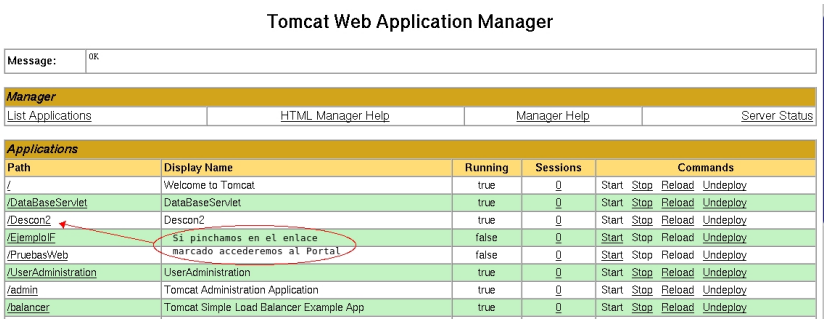


Figura 7.4: Paso 4 de la instalación del sistema en Tomcat

7.2. Manual administración de usuarios, redes y grupos

Esta sección esta dedicada a usuarios que tengan los permisos necesarios para dar de alta a usuarios, redes o grupos en el sistema. La administración se compone de cuatro posibles acciones:

- Añadir
- Modificar
- Eliminar
- Visualizar

7.2.1. Añadir

En el sistema se puede añadir usuarios, redes, grupos y asociaciones como por ejemplo Usuario-Grupo ó Grupo-Red.

Para realizar se debe pinchar en la pestaña correspondiente(a usuario, grupo, etc..) , y saldrá un desplegable con la opción añadir. Al pinchar en la opción añadir saldrá una nueva ventana para que introduzcamos la información de lo que queremos añadir.

7.2.2. Modificar

El campo modificar es muy similar a la opción añadir comentada anteriormente, la diferencia es que en este caso el campo "login", o "descripción"de la red debe existir en el sistema. La opción añadir solo esta disponible para usuarios y redes. Para acceder a ella se pincha en la pestaña a usuario o red y después en la opción modificar.

7.2.3. Eliminar

Este acción se encarga de borrar del sistema, un usuario, grupo o red así como una asociación. Para acceder a ella se selecciona la pestaña correspondiente y se clickea en la opción eliminar.

7.2.4. Visualizar

La opción visualizar muestra la información de todo lo almacenado en el sistema así como su relación entre los distintos campos. La información se muestra en forma de tablas.

Importante: Para que el usuario, pueda ver las redes que tenga asignadas es muy importante que se añada a un grupo y a este grupo se le asigne una red, en la que el campo descripción es el nombre de la red tal cual viene en los equipos de la entidad. En el caso de que se desee ver todas la alertas de la UC3M sera necesario asignar al usuarioa la red 163.117.0.0 con campo descripción Uc3m (El sistema es sensible a minusculas o mayusculas).

7.3. Manual de usuario

7.3.1. Localizar un equipo conectado a la red.

La localización de un equipo puede realizarse desde los cuadros de textos distribuidos en la interfaz (Menú rapido), o bien en la pestaña " Localización ". La localización se hace mediante dirección IP o nombre de DNS. El resultado de la localización es una tabla como se muestra en la figura.

7.3.2. Ver alertas en tiempo real.

Para visualizar la alertas en tiempo real de la redes manejadas por el usuario, se debe pinchar en la pestaña "alertas", y en el lateral izquierdo de la pagina aparecera un arbol con carpetas que represeantan las distintas redes. Dentro de las carpetas se encuentran ordenados por dirección IP los equipos de los que se tiene alerta.

7.3.3. Ver las desconexiones de los equipos.

Para ver los equipos que ya han sido "desconectados" de la red, se debe pinchar en desconexiones, apareceran en carpetas correspondientes a las redes que tienen equipos que han superado las politicas de seguridad de la entidad.

7.3.4. Buscar alertas en la base de datos.

El procedimiento de busqueda es similar al ver las alertas en tiempo real, solo que en este caso se debera pinchar en la pestaña de historico.

7.3.5. Ayuda

Muestra la información de uso del sistema, al pinchar en cualquiera de los enlaces de ayuda distribuidos por la pagina, se abrira una ventana con la información.

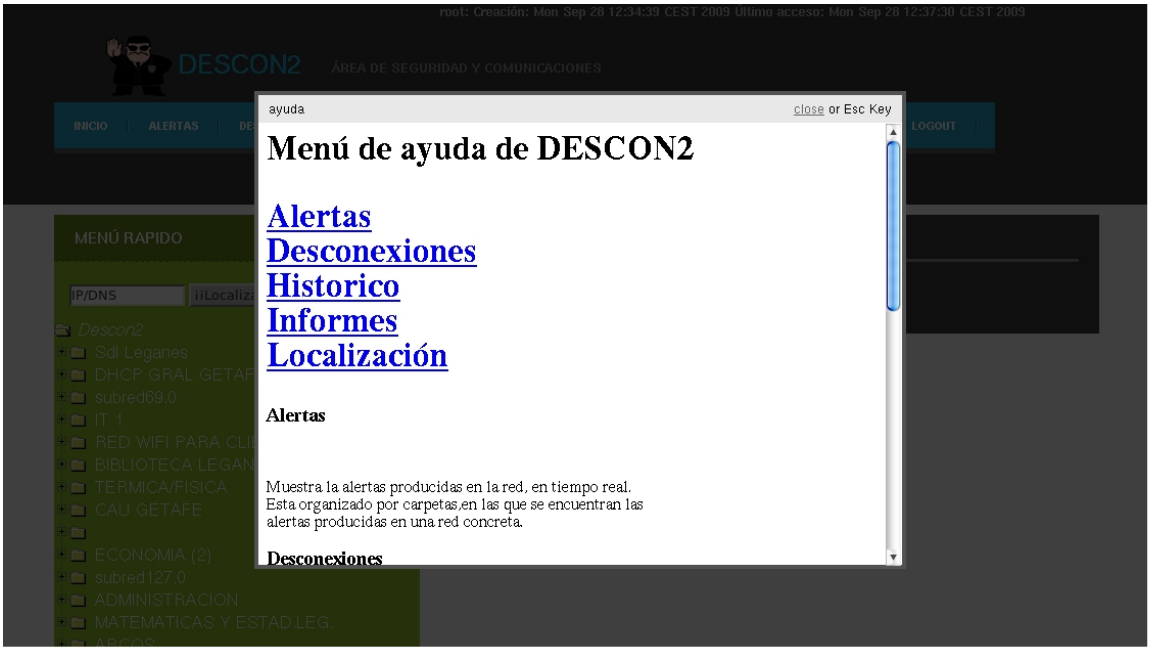


Figura 7.5: Menú de ayuda

Capítulo 8

Conclusiones

En este capítulo se presentan las conclusiones obtenidas del desarrollo de este proyecto y se ofrecen algunas sugerencias para una posible ampliación futura.

Esta sección se dividirá en cuatro apartados. El primero, **Aportaciones realizadas**, se referirá a lo que hemos aportado con la realización del proyecto Descon2. En el segundo apartado , **Trabajos futuros**, nos centraremos en las mejoras que podemos incluir para que su funcionamiento sea mas eficaz. En el tercer punto , **Problemas encontrados**, se hablará sobre los problemas que se han encontrado a la hora de realizar este proyecto. Y por último, veremos las **Opiniones personales** donde se comentará de forma particular las impresiones que se han obtenido con la realización del proyecto.

8.1. Aportaciones realizadas

Descon2 nació por la necesidad de solucionar las carencias de seguridad en redes empresariales de los sistemas vigentes en el mercado. Por ello se estableció una serie de objetivos y que este proyecto ha cumplido. Ahora mostraremos como se ha afrontado el cumplimiento de cada uno de los objetivos.

1. **Agrupar toda la información de seguridad**.

   Para el cumplimiento de este objetivo, Descon2 ha realizado un sistema que re-

copila toda la información existen mediante el **Colector de alertas** desarrollado. También el uso de una **interfaz Web** facilita a los administradores de la red el conocimiento, en tiempo real y pasado, de la información de seguridad en un único punto.

2. **Analizar toda la información de seguridad**.

   El Colector de alertas y su sistema de evaluación del coste de las alertas ha permitido establecer el estado de seguridad de los equipos que se encuentran en la red sin necesidad de la instalación de algún tipo de software de seguridad.

3. **Facilitar la puesta en redes de cuarentena**, además de la determinación del compromiso de seguridad de un equipo, que la realiza el Colector de alerta, el sistema obtiene la información de red de ese equipo aportando el punto exacto por el cual accede a la red de la institución. Esto permite saber a los responsables de la red donde se debe modificar su enrutamiento en la red para que en caso de compromiso el equipo pase a la red de cuarentena. Todo esto se ha podido realizar gracias al sistema de **Localización** desarrollado en Descon2.

4. **Posibilidad a los administradores de las subredes existentes de conocer el estado de compromiso de los equipos de su red**.

   El uso de la interfaz Web y el manejo de los usuarios en el sistema ha permitido que cada administrador de subred acceda a la información correspondiente de su subred pudiendo atajar los problemas de seguridad en la institución de manera mas rápida.

5. **Facilitar a los usuarios la corrección del compromiso de seguridad de su sistema**, al facilitar a la puesta en cuarentena de los equipos comprometidos, los usuarios en la red de cuarentena podrán acceder a los medios necesarios (guías actualizaciones, etc) para que solucionen su problema de seguridad en el menor tiempo posible.

8.2. Trabajos Futuros

Con la finalización de todo el proyecto quedan establecidas una serie de cuestiones para un mayor desarrollo del sistema, así como una visión de las posibles líneas de evolución en el futuro.

- Una mejora importante del sistema sería la **automatización del paso de los sistemas comprometidos a la red de cuarentena**.Esto permitiría mejorar el tiempo de respuesta del sistema ante infecciones masivas. Para la realización de esta mejora, solo sería necesario realizar el mecanismo que cambie la configuración de los equipos de red, ya que disponemos de toda la información de red del equipo comprometido.
- El uso de IDMEF, como unidad de información, permite ampliar la información de seguridad y mejorar la integración con los otros sistemas de detección de intrusiones existentes en la red.
- Otra posible mejora podría ser la ampliación del uso de red de cuarentena para los equipos que acceden a la red WiFi o VPN y que no pueden ponerse en cuarentena con el sistema actual. Para ello se deben valorar las posibilidades de las extensiones de Radius, para configurar los equipos de comunicaciones e implementar una cuarentena similar a la existente para la red cableada.
- Este proyecto ha realizado un prototipo de interfaz Web atendiendo a los requisitos, pero estos no indicaban nada sobre la usabilidad de la aplicación, aun así en el diseño de la interfaz se ha intentado utilizar todos los conocimientos que se disponía sobre usabilidad Web, por lo cual se elaboró un cuestionario de mejora que está en el anexo C. En el futuro sería recomendable validar este prototipo interfaz aplicando el cuestionario y modificando lo que sea necesario para mejorar la experiencia de usuario.
- Mejorar el sistema de correlación de alertas de seguridad. El sistema actual es muy eficiente al detectar ciertos patrones en un espacio de tiempo, aunque la

aparición de falsos positivos hace que sus resultados puedan mejorarse empleando técnicas de aprendizaje supervisado (p.e. basándose en redes bayesianas). Por ello se pueden utilizar herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, como WEKA, que permitirán mejorar la clasificación del compromiso de los equipos en el sistema.

8.3. Problemas encontrados

En todo este tiempo se han encontrado ciertas complicaciones a la hora de realizar el proyecto. En el inicio de el proyecto surgieron algunas complicaciones aunque yo había programado con el lenguaje orientado a objetos JAVA, no había usado la tecnología Java Server Pages y Servlets, que permite la creación de paginas Web dinámicas. Esto implicó la necesidad de documentarme sobre esta tecnología y aprender a la vez que me enfrentaba al proyecto.

Otro inconveniente que debí superar fue el aprendizaje del uso del protocolo de gestión de red SNMP, para lo cual me resultó de mucha ayuda la documentación del anexo 1. También he tenido que aprender a utilizar bases de datos, y más concretamente la base de datos Mysql.

8.4. Opiniones personales

Este proyecto me ha servido para aprender a abstraerme a la hora de afrontar nuevos problemas desconocidos para mí, como el aprendizaje de tecnologías de programación Web, el uso del modelo vista-controlador, muy utilizado en las estructuras cliente servidor,o como la mejora de mi conocimientos sobre los protocolos de red más utilizados.

Además este proyecto me ha proporcionado mi primera experiencia laboral, me ha dado una visión distinta de la carrera cursada, que durante los años aprendí multitud de conceptos, pero que la mayoría se quedaban en la teoría. Hoy después de todo este tiempo he puesto en practica todo aquello que nos explicaron sobre la gestión de

proyectos y lo importante que era un buen estudio y planificación de las tareas que queríamos desarrollar. Sin embargo, he de decir, que hubo momento en que el proyecto fue difícil, empleaba muchas horas y los avances eran pocos, pero a lo largo del tiempo me he dado cuenta de que todo ese esfuerzo ha merecido la pena. En este momento, una vez finalizado el proyecto se observan algunas conclusiones que hoy se ven obvias, pero al inicio y con todo el trabajo por hacer no se veían claramente. Desde el primer diseño hasta el diseño definitivo han pasado muchas etapas, y puedo decir que las fases de análisis y diseño han sido las más importantes. El principal aporte, a nivel personal, de este proyecto ha sido tener una verdadera conciencia de como funcionan realmente los procesos de desarrollo software, sobre todo cuando se deben ir incluyendo nuevas funcionalidades. También me ha permitido mejorar mi conocimiento sobre el tema de la seguridad informática haciéndome interesarme sobre este campo de estudio.

Debo agradecer a mi tutor, Rafael Calzada Pradas, su apoyo y transmisión de su conocimiento, para permitir avanzar en este proyecto. También debo agradecer a mis compañeros de servicio de informática, por contestar a mis preguntas y dudas.

Apéndice A

Protocolo SNMP

A.0.1. Beneficios de usar SNMP

Para la realización de este proceso comentado anteriormente es necesario usar protocolo de gestión de red para poder realizar las consultas comentadas anteriormente, en nuestro caso hemos utilizado SNMP [1] es un protocolo estándar de Internet. Su estado actual recomendado y la especificación actual se puede encontrar en el RFC 1157 [13]. Descon2 usa más concretamente la versión SNMP v2 debido a que tiene un mejor sistema de seguridad que SNMP v1, evitando con problemas con intrusos que observen el estado o la condición de los dispositivos administrados. Tanto la encriptación como la autenticación están soportadas por SNMP v2. Existe una versión más moderna de SNMP la versión 3 pero está menos aceptada por la industria. Por lo cual se ha usado la versión 2c para evitar problemas de incompatibilidades con los diferentes equipos de la red.

A.0.2. Modelo de gestion basado en en SNMP

Para empezar explicaremos como se organiza la información de este protocolo. SNMP se puede implementar usando comunicaciones UDP o TCP, pero por norma general,

[1] El Protocolo Simple de Administración de Red o SNMP es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red

se suelen usar comunicaciones UDP en la mayoría de los casos. Con UDP, el protocolo SNMP se implementa utilizando los puertos 161 y 162. SNMP tiene el siguiente elemento de información la MIB que 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. La MIB de cada equipo nos permite acceder a la información que deseemos del equipo de manera fácil y ordenada en una consulta.

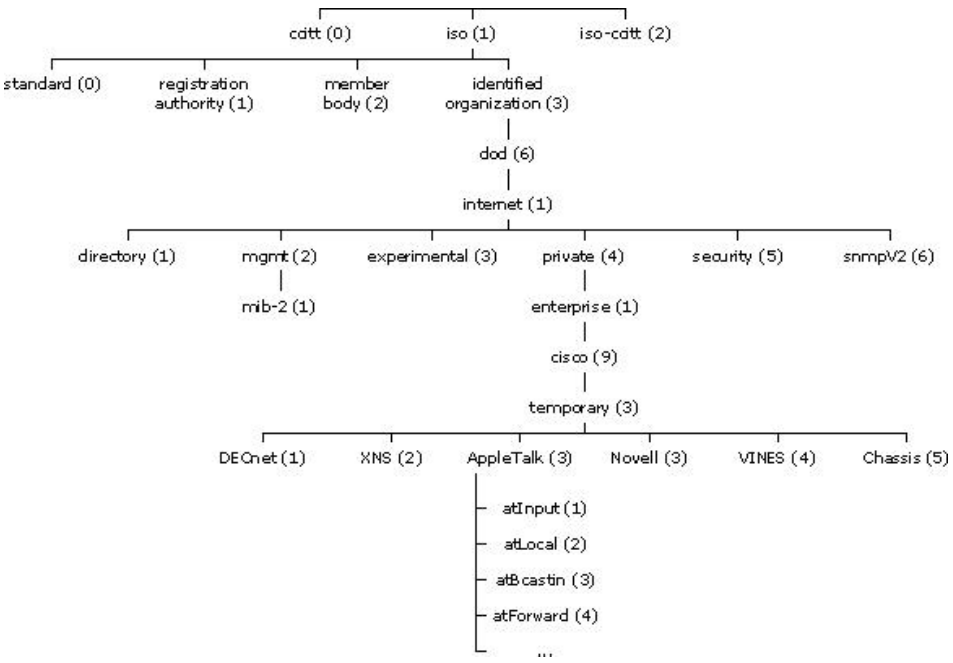


Figura A.1: Estructura de árbol de una MIB

Elementos de la arquitectura SNMP

- Nodos administrados que ejecutan agentes SNMP (los equipos de red de la universidad) y estación administradora (la aplicación Descon2).

- La base de datos MIB correspondiente a cada nodo administrado con formato SMI [2]

[2]SMI es una estructura de manejo de información con unas características determinadas.

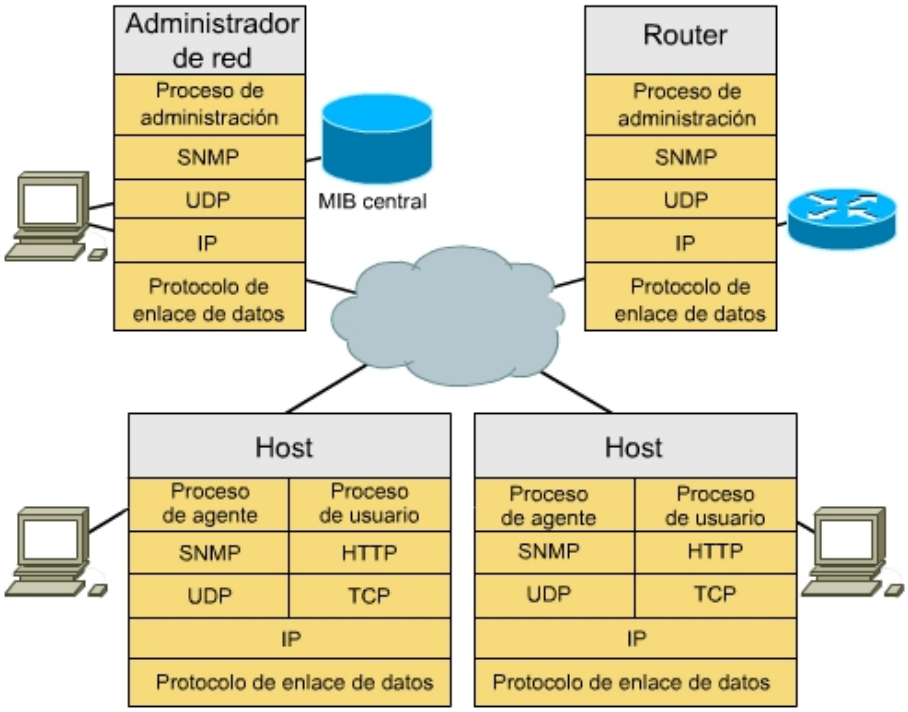


Figura A.2: Ejemplo de la arquitectura de SNMP

A.0.3. Mecanismos de administración

Snmp dispone de una serie de comandos básicos para comunicarse con los nodos estos son :

- OBTENER (GET), que implica que la consola de administración recupera datos del agente

- COLOCAR (PUT), que implica que la consola de administración establece los valores de los objetos en el agente

- CAPTURAR (TRAP), que implica que el agente notifica a la consola de administración acerca de los sucesos de importancia por interrupción

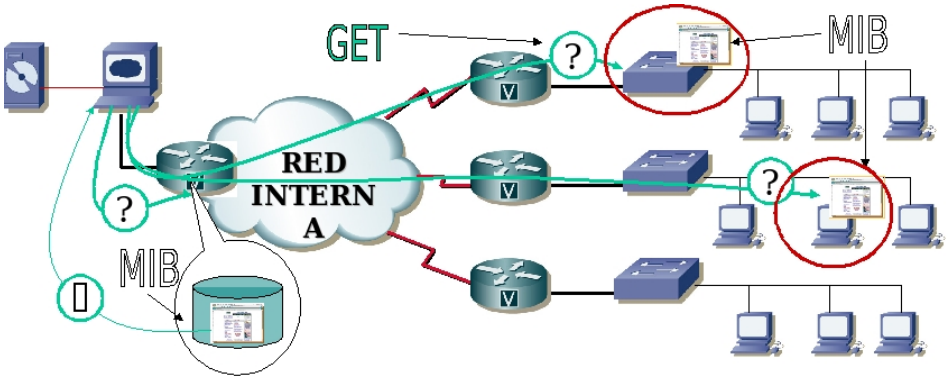


Figura A.3: Funcionamiento consultas SNMP

A.0.4. Tipos de mensajes

- **GetRequest**

  A través de este mensaje el NMS[3] solicita al agente retornar el valor de un objeto de interés mediante su nombre. En respuesta el agente envía una respuesta indicando el éxito o fracaso de la petición. Si la petición fue correcta, el mensaje resultante también contendrá el valor del objeto solicitado. Este mensaje puede ser usado para recoger un valor de un objeto, o varios valores de varios objetos, a través del uso de listas.

- **GetNextRequest**

  Este mensaje es usado para recorrer una tabla de objetos. Una vez que se ha usado un mensaje GetRequest para recoger el valor de un objeto, puede ser utilizado el mensaje GetNextRequest para repetir la operación con el siguiente objeto de la tabla. Siempre el resultado de la operación anterior será utilizado para la nueva consulta. De esta forma un NMS puede recorrer una tabla de longitud variable hasta que haya extraído toda la información para cada fila existente.

- **SetRequest**

  Este tipo de mensaje es utilizado por el NMS para solicitar a un agente modificar valores de objetos. Para realizar esta operación el NMS envía al agente una lista de nombres de objetos con sus correspondientes valores.

[3]Nodo administrado

- **GetResponse**

  Este mensaje es usado por el agente para responder un mensaje GetRequest, GetNextRequest, o SetRequest. En el campo "Identificador de Request "lleva el mismo identificador que el request al que está respondiendo.

- **GetBulkRequest**

  Este mensaje es usado por un NMS que utiliza la versión 2 del protocolo SNMP típicamente cuando es requerida una larga transmisión de datos, tal como la recuperación de largas tablas. En este sentido es similar al mensaje GetNextRequest usado en la versión 1 del protocolo, sin embargo, GetBulkRequest es un mensaje que implica un método mucho más rápido y eficiente, ya que a través de un solo mensaje es posible solicitar la totalidad de la tabla.

- **Trap**

  Una trap es generado por el agente para reportar ciertas condiciones y cambios de estado a un proceso de administración. Una trap es un mensaje espontáneo enviado por el Agente al Administrador, al detectar una condición predeterminada, como es la conexión/desconexión de una estación o una alarma.

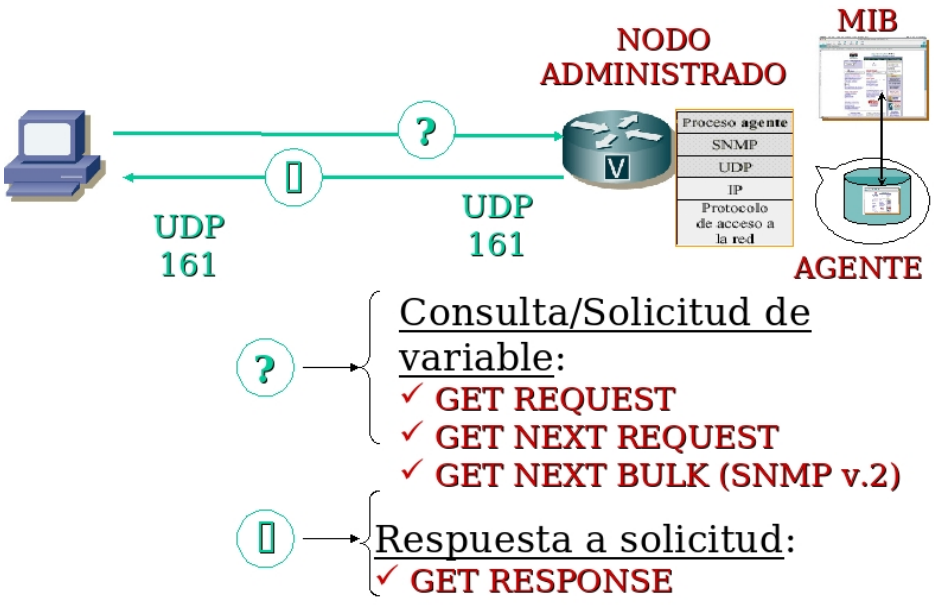


Figura A.4: Obtención de información de SNMP

A.0.5. ¿Como es una consulta de SNMP?

En esta sección vamos a mostrar un ejemplo practico de como es una consulta SNMP mediante comandos en un terminal Linux, para realizar las consultas de forma manual usamos el software UCD-SNMP que implementa el protocolo SNMP para todas las distribuciones de Linux, para los otros sistemas operativos habrá otras aplicaciones con uso similar.UCD-SNMP incluye el agente snmpd y las herramientas de gestión snmpget, snmpgetnext, snmpset, snmpwalk, snmpnetstat, snmptrapd y snmptest.

Configuración Básica

Instalado el software el fichero de configuración del agente snmpd está en /etc/snmp/snmpd.conf aquí se definen las comunities, que son un par de claves que se utilizan para acceder al agente SNMP, normalmente está configurado con public para lectura y private para escritura de forma predeterminada, por razones de seguridad obvias es conveniente cambiarlas.

Veamos un ejemplo para snmpget

snmpget -c public -v 2c localhost oid

- **-c** Nos indica la community en el caso de este ejemplo public que sirve para lectura.
- **-Localhost** Es el nombre del equipo de el que queremos obtener la información.
- **-oid** Nos indica el identificador de objeto que queremos que nos devuelva.

Para los diferentes comandos se seguiría el mismo formato expuesto para snmpget.

Para navegar por el el árbol de la mib de objetos de Cisco y saber como acceder a una consulta, Cisco ofrece una pagina web en la que introduciendo el OID te muestra el árbol y el significado del objeto que se esta buscando.

http://tools.cisco.com/Support/SNMP/do/BrowseOID.do (06/10/2008)

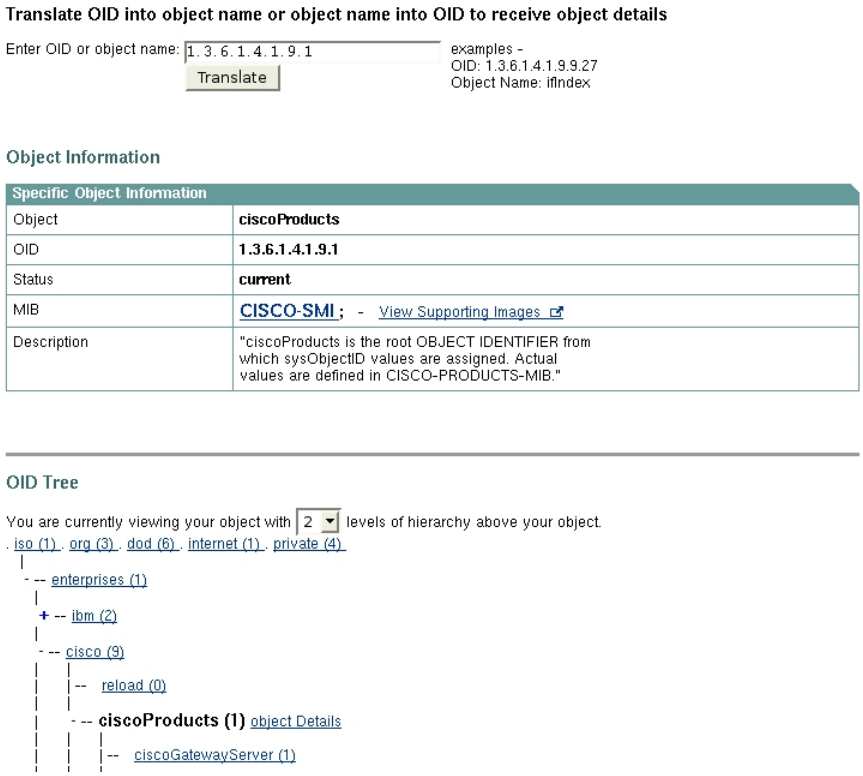


Figura A.5: Ejemplo de una búsqueda de un OID con Cisco Navigator

A.0.6. Implementación especificas de consultas de SNMP para la localización de equipos

Actualmente (Octubre 2009) todos los equipos de la red del servicio de informática son la empresa Cisco, por lo cual las implementaciones específicas de consultas de SNMP han sido implementadas para equipos de esta empresa.

A continuación se mostrara un ejemplo práctico de como realizan las consultas SNMP vamos a utilizar un terminal Linux y el software UCD-SNMP, que implementa el protocolo SNMP para todas las distribuciones de Linux. Para otros sistemas operativos hay otras aplicaciones con uso similar.

UCD-SNMP incluye el agente snmpd y las herramientas de gestión snmpget, snmpgetnext, snmpset, snmpwalk, snmpnetstat, snmptrapd y snmptest.

Ver redes encaminadas en router.

Este comando muestra las direcciones de las interfaces del router que encaminan redes a las que se encuentra conectado el router.

snmpwalk -c public -v2c nombre del router .1.3.6.1.2.1.4.20.1.1

Ejemplo:

snmpwalk -c public -v2c b01014z.uc3m.es .1.3.6.1.2.1.4.20.1.1

IP-MIB::ipAdEntAddr.10.117.30.100 = IpAddress: 10.117.30.100

IP-MIB::ipAdEntAddr.10.118.0.2 = IpAddress: 10.118.0.2

IP-MIB::ipAdEntAddr.10.119.0.2 = IpAddress: 10.119.0.2

IP-MIB::ipAdEntAddr.127.0.0.51 = IpAddress: 127.0.0.51

IP-MIB::ipAdEntAddr.163.117.15.2 = IpAddress: 163.117.15.2

IP-MIB::ipAdEntAddr.163.117.30.100 = IpAddress: 163.117.30.100

IP-MIB::ipAdEntAddr.163.117.31.2 = IpAddress: 163.117.31.2

.....

Ver mascara de red

Este comando devuelve la mascara de red, para ello es necesario conocer la dirección IP del interfaz del router.

snmpwalk -c public -v2c nombre del router .1.3.6.1.2.1.4.20.1.3.dirección ip del interfaz de la red en el router

Resultado:

snmpwalk -c public -v2c b01014z.uc3m.es .1.3.6.1.2.1.4.20.1.3.163.117.131.2

IP-MIB::ipAdEntNetMask.163.117.131.2 = IpAddress: 255.255.255.0

Obtener el índice de interfaz para una red

Con este comando obtenemos el IfIndex que es necesario, para obtener otros parámetros como ifDescr o ifName así como la VLAN correspondiente a una interfaz. Para obtener el IfIndex es necesario conocer la IP del interfaz del router.

snmpget -c public -v2c nombre del router .1.3.6.1.2.1.4.20.1.2.dirección ip del interfaz de la red en el router

Nos devuelve el ifindex [4]

Ejemplo:

snmpget -c public -v2c b01014z.uc3m.es .1.3.6.1.2.1.4.20.1.2.163.117.131.2

IP-MIB::ipAdEntIfIndex.163.117.131.2 = INTEGER: 108

Obtener Vlan

Para obtener el valor de una VLAN, el proceso es algo diferente a los casos anteriores, ya que lo primero que se debe obtener es el índice de interfaz de la red que queremos obtener la VLAN. Con el comando siguiente vemos todos los índices de interfaz, y en el correspondiente al IfIndex buscado en el oid de la respuesta SNMP, esta el valor de la VLAN asociado a ese IfIndex.

snmpwalk -c public -v2c nombre del router .1.3.6.1.4.1.9.9.128.1.1.1.1.3

Ejemplo:

snmpwalk -c public -v2c b01014z.uc3m.es .1.3.6.1.4.1.9.9.128.1.1.1.1.3

SNMPv2-SMI::enterprises.9.9.128.1.1.1.1.3.1.0 = INTEGER: 105

SNMPv2-SMI::enterprises.9.9.128.1.1.1.1.3.**2**(Vlan).0 = INTEGER: **108** (ifIndex)

SNMPv2-SMI::enterprises.9.9.128.1.1.1.1.3.3.0 = INTEGER: 109

SNMPv2-SMI::enterprises.9.9.128.1.1.1.1.3.4.0 = INTEGER: 110

SNMPv2-SMI::enterprises.9.9.128.1.1.1.1.3.5.0 = INTEGER: 111

Obtener MAC

Para obtener la dirección de Ethernet, correspondiente a una dirección IP, es nece-

[4] Ifindex es el índice de interfaz en un router o un conmutador

sario la dirección IP y el ifIndex.

snmpwalk -c public -v2c nombre del router .1.3.6.1.2.1.3.1.1.2."+ifIndex+".1."+dirección IP

Ejemplo:

snmpwalk -c public -v2c b01014z.uc3m.es .1.3.6.1.2.1.3.1.1.2.**108**.1.**163.117.131.194**

RFC1213-MIB::atPhysAddress.108.1.163.117.131.194 = Hex-STRING: 00 1A 92 33 3B 0D

Obtener puerto en un conmutador para una dirección MAC

Para obtener el puerto en el que se encuentra un conmutador es necesario tres pasos:

1. Obtenemos el identificador del puerto:
   snmpwalk -c public -v2c nombre del conmutador .1.3.6.1.2.1.17.4.3.1.2.+dirección MAC en decimal.

2. El if Index:
   snmpwalk -c public -v2c nombre del conmutador .1.3.6.1.2.1.17.1.4.1.2.+idPuerto.

3. Y se obtiene el puerto:
   snmpwalk -c public -v2c nombre del conmutador 1.3.6.1.2.1.31.1.1.1.1."+ifIndexString.

Ejemplo:

snmpwalk -c public -v2c b01014z.uc3m.es .1.3.6.1.2.1.17.4.3.1.2.0.1.236.207.131.156 SNMPv2-SMI::mib-2.17.4.3.1.2.0.1.236.207.131.156 = INTEGER: 19

snmpwalk -c public -v2c b01014z.uc3m.es .1.3.6.1.2.1.17.1.4.1.2.19
SNMPv2-SMI::mib-2.17.1.4.1.2.19 = INTEGER: 19

snmpwalk -c public -v2c b01014z.uc3m.es 1.3.6.1.2.1.31.1.1.1.1.19

IF-MIB::ifName.19 = **STRING: Gi1/19**

Obtener conmutadores vecinos de un puerto

Para la obtención de los conmutadores vecino en un puerto se utiliza CDP (Cisco Discovery Protocol), que es un protocolo propietario de red de nivel 2 desarrollado por Cisco que sirve para el descubrimiento de equipos directamente conectados, y que esta implementado en la mayoría de los equipos Cisco.

snmpwalk -c public -v2c nombre del router .1.3.6.1.4.1.9.9.23.1.2.1.1.4."+this.ifIndex

Devuelve la dirección ip en hexadecimal del siguiente conmutador por el puerto que indica el ifIndex.
Ejemplo:
snmpwalk -c public -v2c b01014z.uc3m.es .1.3.6.1.4.1.9.9.23.1.2.1.1.4.19

NMPv2-SMI::enterprises.9.9.23.1.2.1.1.4.19.121=

Hex-STRING:A3 75 1E A1

A.0.7. Como implementar las consultas SNMP en JAVA

Para la implementación de las consultas en SNMP se han buscado librerías externas que implementaran el protocolo SNMP. Otro requisito es que fueran gratuitas y que su uso fuera sencillo. Para ello encontramos dos soluciones: SNMP4j y SNMP Adventnet.

SNMP4j

Esta primera es software libre y fue la primera en utilizarse debido a este motivo. Podemos conseguir esta librería desde la pagina oficial [14], es gratuita y está desarrollada por Frank Fock and Jochen Katz. Es una librería muy completa, pero tiene el inconveniente de ser un poco complejo su uso.

SNMP Advent

Esta librería pertenece a la empresa AdventNet [15] y ofrece dos librerías para SNMP,

una versión gratuita, pero limitada en opciones y otra más completa, siendo esta última una versión de pago. Para las funcionalidades utilizadas en este proyecto observamos que la versión gratuita ofrecía suficientes funcionalidades. La pagina de descarga de las versiones:

http://snmp.adventnet.com/download.html (15/10/2008)

Para este proyecto se decidió utilizar SNMP Advent debido a su sencillez de uso. Podemos ver un ejemplo A.6 de código para crear una sesión y hacer una consulta con SNMP AdventNet

```
SnmpAPI api = new SnmpAPI();
SnmpSession session = new SnmpSession(api);
session.open();
SnmpPDU pdu = new SnmpPDU();

pdu.setProtocolOptions(new UDPProtocolOptions("localhost"));
pdu.setCommand(SnmpAPI.GET_REQ_MSG);
pdu.addNull(new SnmpOID(".1.3.6.1.2.1.1.1.0"));
SnmpPDU response_pdu = session.syncSend(pdu);

if(response_pdu == null)
{
     System.out.println("The Request has timed out.");
}
else
{
     System.out.println(response_pdu.printVarBinds());
}
```

```
        La salida del programa sera:
        Object ID: .1.3.6.1.2.1.1.1.0
STRING: Linux localhost  2.6.18-6-amd64
```

Figura A.6: Código de ejemplo de como crear una sesión SNMP

Para realizar cualquier tipo de las consultas se toma como base el ejemplo que muestra la figura, y modificando los diferentes parámetros para obtener la respuesta que se necesite para cada consulta.

Apéndice B

SSL

B.1. Configuración de SSL en Tomcat

Antes de comenzar con el proceso de configurar Tomcat con SSL, se dará una breve explicación de este protocolo y en que consiste un certificado digital.

B.1.1. Secure Sockets Layer - Protocolo de Capa de Conexión Segura (SSL)

SSL es un protocolo que proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere un despliegue de infraestructura de claves públicas (o PKI) para los clientes. Los protocolos permiten a las aplicaciones cliente-servidor comunicarse de una forma diseñada para prevenir escuchas (eavesdropping), la falsificación de la identidad del remitente (phishing) y mantener la integridad del mensaje.

SSL implica una serie de fases básicas:

- Negociar entre las partes el algoritmo que se usará en la comunicación
- Intercambio de claves públicas y autenticación basada en certificados digitales
- Cifrado del tráfico basado en cifrado simétrico

B.1.2. Autoridades de certificación y certificados digitales

Una **autoridades de certificación (CA)** su objetivo es autenticar la propiedad y las características de una clave pública, de manera que resulte digna de confianza, y expedir certificados.

Las funciones de una Autoridad de Certificación deben ser, entre otras, las siguientes:

- Generación y Registro de claves.
- Identificación de Peticionarios de Certificados.
- Emisión de certificado.
- Almacenamiento en la AC de su clave privada.
- Mantenimiento de las claves vigentes y revocadas.
- Servicios de directorio

B.1.3. Modo de funcionamiento

Solicitud de un certificado

El procedimiento habitual para la solicitud de un certificado de servidor web a una CA consiste en que la entidad solicitante, utilizando ciertas funciones del software de servidor web, incluye un conjunto datos identificativos (entre los que se incluye el localizador URL del servidor) y genera una pareja de claves pública/privada. Con esa información el software de servidor compone un fichero que contiene una petición CSR (Certificate Signing Request) en formato PKCS10 que contiene la clave pública y que se hace llegar a la CA elegida. Esta, tras verificar por sí o mediante los servicios de una RA (Registration Authority, Autoridad de Registro) la información de identificación aportada y la realización del pago, envía el certificado firmado al solicitante, que lo instala en el servidor web con la misma herramienta con la que generó la petición CSR.

En este contexto, PKCS corresponde a un conjunto de especificaciones que son estándares de facto denominadas Public-Key Cryptography Standards.

La Jerarquía de Certificación

Las CA disponen de sus propios certificados públicos, cuyas claves privadas asociadas son empleadas por las CA para firmar los certificados que emiten. Un certificado de CA puede estar auto-firmado cuando no hay ninguna CA de rango superior que lo firme. Este es el caso de los certificados de CA raíz, el elemento inicial de cualquier jerarquía de certificación. Una jerarquía de certificación consiste en una estructura jerárquica de CAs en la que se parte de una CA auto-firmada, y en cada nivel, existe una o más CAs que pueden firmar certificados de entidad final (titular de certificado: servidor web, persona, aplicación de software) o bien certificados de otras CA subordinadas plenamente identificadas y cuya Política de Certificación sea compatible con las CAs de rango superior.

B.1.4. Instalación de del certificado en Tomcat

La instalación se ha realizado en un equipo con las siguientes características:

- Apache Tomcat 5.5
- Java JDK 6, Standard Edition
- Debian GNU/Linux 5.0 ” Lenny ”

Después de rellenar la solicitud para la obtención del certificado la entidad nos devuelve dos ficheros con extensión pem, SCSCAs (La lista de CA’s) , descon21039564158 (El certificado privado). El primer paso es copiar nuestra clave privada, el certificado para nuestra clave y la lista de CA desde la raíz.

Estos certificados están el formato Base64, y debería guardarse todos en un fichero con extensión pem. Para ello creamos un fichero de texto vació con un editor de texto y copiamos en primer lugar la clave privada que deberá ser de este tipo:
Empieza con ” BEGIN PRIVATE KEY ” finaliza con ” END PRIVATE KEY ”.

En nuestro caso es el fichero descon2.key.

Ahora abriremos el fichero SCSCAs y copiamos todas las partes que empiezan " BEGIN CERTIFICATE " y acaben como " END CERTIFICATE ".

En nuestro caso concreto tenemos dos CA por lo cual tendremos dos bloques. CA de la autoridad raíz.

Certificate Data:

- Versión: 1 (0x0)
- Serial Number: 421 (0x1a5)
- Signature Algorithm: md5WithRSAEncryption
- Issuer: C=US, O=GTE Corporation, OU=GTE CyberTrust Solutions, Inc., CN=GTE CyberTrust Global Root
- Validity
- Not Before: Aug 13 00:29:00 1998 GMT
- Not After : Aug 13 23:59:00 2018 GMT
- Subject: C=US, O=GTE Corporation, OU=GTE CyberTrust Solutions, Inc., CN=GTE CyberTrust Global Root

CA que firma nuestro certificado.

Certificate Data:

- Version: 3 (0x2)
- Serial Number: 67109883 (0x40003fb)
- Signature Algorithm: sha1WithRSAEncryption
- Issuer: C=US, O=GTE Corporation, OU=GTE CyberTrust Solutions, Inc., CN=GTE CyberTrust Global Root
- Validity
- Not Before: Mar 14 20:30:00 2006 GMT

- Not After : Mar 14 23:59:00 2013 GMT
- Subject: C=BE, O=Cybertrust, OU=Educational CA, CN=Cybertrust Educational CA

Ahora abriremos el fichero descon21039564158 y copiamos la partes que empieza por " BEGIN CERTIFICATE" y acabe como " END CERTIFICATE ". Con todo este ya tenemos un fichero .pem con toda la información necesaria.

El segundo paso es configurar SSL en nuestro servidor. Sin embargo necesitamos convertir el fichero . perm que hemos creado un formato que sea soportado por Tomcat. Para esto se puede utilizar un herramienta de software libre llamada OpenSSL. Cuando tengamos instalada esta aplicación ejecutaremos el siguiente comando con el que convertiremos nuestro fichero .pem:

Al fichero que hemos combinado, le llamare a partir de ahora descon2-final.pem pero se puede elegir el nombre que se desee. openssl pkcs12 -export -in descon2-final.pem -out descon2.p12

Este comando nos pedirá una clave para cifrar la información del fichero, después devolverá un fichero con el nombre descon2.p12 que es un formato ya conocido por Tomcat.

Ahora abrimos con un editor de texto el fichero server.xml, típicamente este fichero en Linux esta en la siguiente carpeta /etc/tomcat5.5/server.xml.

Añadimos servicio https en puerto 443.

Connector port ="443" protocol="HTTP/1.1" maxThreads="1500"

keystoreFile="**/etc/tomcat5.5/descon2.p12**"

keystorePass=" ***** " keystoreType="PKCS12"

SSLEnabled="true"scheme="https" secure="true"

clientAuth=”false” sslProtocol=”TLS”

KeystorePass, es la contraseña que se eligio al crear el fichero descon2.p12

Apéndice C

Cuestionario de usabilidad

- **Datos del usuario**

1.¿ Cual es su nombre?

Respuesta:

2. ¿A qué se dedica?

Respuesta:

3. ¿Qué experiencia tiene con herramientas de seguridad?

Respuesta:

4. ¿Que herramientas usa habitualmente?

Respuesta:

5. ¿Usa herramientas con interfaz web?

Respuesta:

6. ¿Prefiere una aplicación tradicional, o una aplicación Web?

Respuesta:

7. Utiliza alguna aplicación Web, del estilo de Google Docs o Office Online, ¿Si las utiliza le resultan utiles? ¿Cuales son ? ¿Si no las utiliza cual es el motivo de no usarlas?

Respuesta:

- **Preguntas acerca de la identidad de la web.**

1. ¿Con la información que se ofrece en pantalla, es posible saber a qué a que tipo de aplicación corresponde la aplicación? ¿Cómo lo sabe?

Respuesta:

2.¿Hay algún elemento gráfico o de texto que le haya ayudado a entender más claramente a qué tipo de aplicación pertenece el sitio?

Respuesta:

3. ¿Resultan los colores predominantes en el sitio web comodos para un trabajo diario con ellos? ¿Le da la impresión de ser un producto atractivo a la vista?

Respuesta:

4. ¿De los elementos que muestra esta pantalla, hay algo que usted crea que está fuera de lugar en una aplicación de estas caracteristicas?

Respuesta:

5. ¿Hacia qué tipo de audiencia cree usted que esta dirigido este sitio? ¿Por qué?

Respuesta:

6. Si tuviera que tomar contacto telefónico o enviar una carta tradicional a la institución o empresa propietaria de la web, ¿se ofrece información de números o direcciones? ¿Son útiles como para hacer esa tarea? ¿Le costó encontrar esa información?

Respuesta:

- **Preguntas acerca del contenido de la web.**

1. ¿Le parece adecuada la selección de contenidos destacados en la portada o usted echó de menos otras áreas de información que le habría gustado ver destacadas?

Respuesta:

2. ¿Al ver la portada del sitio, pudo distinguir de una sola mirada cuál era el con-

tenido más relevante que se ofrecía? ¿Cómo logró hacer esa distinción?

Respuesta:

3. ¿Los textos usados en los contenidos de los enlaces son suficientemente descriptivos de lo que se ofrece en las páginas hacia las cuales se accede a través de ellos?

Respuesta:

4. En caso de haber información relacionada con la que estaba viendo, ¿se le ofreció de manera simple? ¿O tuvo que volver a navegar para encontrarla?

Respuesta:

- **Preguntas acerca de la navegación por la web.**

1. ¿Puede ver en la portada y las demás páginas, la forma en que se navega por el sitio? ¿Se distingue fácilmente?

Respuesta:

2. ¿Existen elementos dentro de las páginas, que le permitan saber exactamente dónde se encuentra dentro de este sitio y cómo volver atrás sin usar los botones del programa navegador?

Respuesta:

3. ¿Cómo vuelve desde cualquier página del sitio a la página de inicio? ¿Ve alguna forma de hacerlo, que no sea presionando el botón del buscador? ¿Le parece claro?

Respuesta:

4. ¿Habitualmente, cómo logra acceder directamente a los contenidos sin tener que navegar? ¿Usa el buscador? ¿Usa el Mapa del Sitio? ¿Los puede ver en este sitio? ¿Echa de menos alguno? ¿Le ayuda esa diferencia?

Respuesta:

5. ¿Logra distinguir gráficamente los enlaces visitados de aquellos que no ha visitado aún? ¿Le ayuda esa diferencia?

Respuesta:

6. El sitio tiene varios niveles de navegación y Usted ha ingresado y salido de varios de ellos. ¿La información que se le ofrece en pantalla le parece adecuada para entender dónde está ubicado en cualquier momento? ¿Se ha sentido perdido dentro del sitio? ¿Si lo ha sentido, recuerda en qué área fue? ¿Si no lo ha sentido, qué elemento del sitio cree que le ayudó más a orientarse?

Respuesta:

- **Preguntas acerca de la estructura gráfica de la web.**

1.¿Considera que gráficamente el sitio está equilibrado, muy simple o recargado?

Respuesta:

- **Preguntas acerca de feedback.**

1. ¿Encuentra alguna forma online y offline de ponerse en contacto con la empresa o institución, para hacer sugerencias o comentarios?

Respuesta:

2. ¿Al mandar datos mediante un formulario, el web le avisa si los recibió correctamente o no?

Respuesta:

- **Preguntas acerca de la utilidad de la web.**

1. ¿Tras una primera mirada, le queda claro cuál es el objetivo del sitio? ¿Qué contenidos y servicios ofrece? ¿Los puede enumerar?

Respuesta:

2. ¿Cree que los contenidos y servicios que se ofrecen en este sitio son de utilidad para su caso personal?

Respuesta:

3.¿Qué es lo que más te llamó la atención positivamente o negativamente de la

utilidad que ofrece el sitio web?

Respuesta:

- **Tareas**

1. Localizar un equipo de la institución.

Tarea 1	Resultado	Observaciones
Tiempo empleado		
Dificultad encontrada		
Sensación al acabar		

2. Buscar las alertas actuales del departamento

Tarea 2	Resultado	Observaciones
Tiempo empleado		
Dificultad encontrada		
Sensación al acabar		

3. Desconexiones del equipo

Tarea 3	Resultado	Observaciones
Tiempo empleado		
Dificultad encontrada		
Sensación al acabar		

4. Buscar en la ayuda información sobre desconexiones.

Tarea 4	Resultado	Observaciones
Tiempo empleado		
Dificultad encontrada		
Sensación al acabar		

5. Localizar sin en algún momento ha sido desconectado el siguiente equipo:

Tarea 5	Resultado	Observaciones
Tiempo empleado		
Dificultad encontrada		
Sensación al acabar		

Si es usted administrador conteste la siguientes preguntas

6. Añada un nuevo usuario al sistema.

Tarea 6	Resultado	Observaciones
Tiempo empleado		
Dificultad encontrada		
Sensación al acabar		

7. Asocie el nuevo usuario con una nuevo grupo y este a su vez con una red existente.

Tarea 7	Resultado	Observaciones
Tiempo empleado		
Dificultad encontrada		
Sensación al acabar		

8. Visualice la información del usuario creado anteriormente.

Tarea 8	Resultado	Observaciones
Tiempo empleado		
Dificultad encontrada		
Sensación al acabar		

Apéndice D

Glosario de terminos.

UC3M Universidad Carlos III.

PDA Agenda electronica.

SmartPhone Telefono móvil inteligente, con capacidades avanzadas de conexion a internet.

SSH (Secure SHell, en español: intérprete de órdenes seguro) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red.

Cortafuegos Aplicación de seguridad que controla las conexiones del equipo.

SNMP Protocolo de gestion de red de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red.

SSL Protocolo de Capa de Conexión Segura

Tomcat Servidor de aplicaciones web.

ArpaNet Red precusora del internet actual.

Malware Software malicioso.

Radius Remote Authentication Dial-In User Server, protocolo de autenticación y autorización para aplicaciones de acceso a la red o movilidad IP. Utiliza el puerto

1813 UDP para establecer sus conexiones.

LDAP Lightweight Directory Access Protocol, Protocolo Ligero de Acceso a Directorios es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red

SNORT Sniffer de paquetes y un detector de intrusos basado en red (se monitoriza todo un dominio de colisión).

Iptables Herramientas de cortafuegos que permite no solamente filtrar paquetes, sino también realizar traducción de direcciones de red (NAT) para IPv4 o mantener registros de log.

LINUX Sistema operativo de libre distribución.

Host A una máquina conectada a una red de ordenadores

IDS Sistema detector de intrusiones.

IPS Sistema de prevención de intrusiones.

IPFW Cortafuegos

IDMEF (Intrusion Detection Message Exchange Format) como un formato común para alertas IDS. Este es una especificación basada en XML para un formato de alertas de intrusión.

MAC OS Sistema operativo de la empresa Macintosh

WIFI Protocolo de red local sin cables.

DHCP Protocolo de asignación de direccion IP automatico.

WEKA Software para aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato.

UML Lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

HTML HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.

Servlets Los servlets son objetos que corren dentro del contexto de un contenedor de servlets. El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

VLAN (Virtual LAN, 'red de área local virtual') es un método de crear redes lógicamente independientes dentro de una misma red física.

XML (Extensible Markup Language) lenguaje de marcas extensible, es un metalenguaje extensible de etiquetas

Patrón Strategy conjunto de algoritmos de los que el objeto cliente puede elegir aquel que le conviene e intercambiarlo según sus necesidades.

TCP Protocolo de comunicación orientado a conexión y fiable del nivel de transporte, actualmente documentado por IETF RFC 793

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

ASyC Área de Seguridad y Comunicaciones dela UC3M

VPN Tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo Internet

CGI Importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.

HTTP Protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web

Cookie Fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas.

PDF Fichero de documentos portable.

CSS Mecanismo simple que describe cómo se va a mostrar un documento en la pantalla.

UDP Protocolo del nivel de transporte basado en el intercambio de datagramas.

MIB Tipo de base de datos que contiene información jerárquica, estructurada en forma de árbol, de todos los dispositivos gestionados en una red de comunicaciones.

Bibliografía

[1] http://www-unix.ecs.umass.edu/ gong/papers/earlyDetectionJournal.pdf

[2] http://www.microsoft.com/latam/technet/articulos/tn/2007/abr-17.mspx.

[3] http://www.microsoft.com/windowsserver2008/en/us/nap-product-home.aspx.

[4] http://marketshare.hitslink.com/report.aspx?qprid=8&qptimeframe=Y&qpsp=2009&qpmr=100&

[5] http://www.ieee802.org/1/pages/802.1x.html

[6] http://www.cisco.com/en/US/products/sw/secursw/ps5057/index.html

[7] http://netsquid.tamu.edu/

[8] http://www.snort.org/

[9] http://www.bleedingthreats.net/

[10] http://www.snort.org/docs/

[11] http://www.prelude-ids.org

[12] http://www.ietf.org/rfc/rfc4765.txt

[13] http://www.ietf.org/rfc/rfc1157.txt

[14] http://www.snmp4j.org

[15] http://www.adventnet.com/

[16] [6] Roesch, M.: Snort - lightweight intrusion Rouillard, J.P.: Refereed papers: Real ime log file analysis using the simple event correlator (sec). In: LISA '04: Proceedings

of the 18th USENIX conference on System administration, Berkeley, CA, USA, USENIX Association (2004).

[17] http://www.tsc.uc3m.es

[18] http://www.cs.waikato.ac.nz/ml/weka/

[19] http://www.uml.org

[20] Design Patterns: Elements of Reusable Object-Oriented Software Autores:Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides: (Addison-Wesley Professional Computing Series)

[21] http://www.useit.com/