Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

”Sistema de Administración y Control de IPS en una Red”

Tesis Profesional

Para obtener el Título de
Licenciado en Ciencias de la Computación

Presenta
Oscar Miranda Márquez

Asesor

Dr. Manuel Martín Ortiz

Resumen

En nuestros tiempos los problemas de las redes y su seguridad es de gran importancia ya que nuestra información esta en riesgo cuando existe algún intruso dentro de ella que se dedica a capturar tal información y usarla de forma indebida o para propósitos fraudulentos, por este motivo siempre un administrador de una red busca una herramienta flexible, de fácil uso y en este caso administrable desde cualquier punto del mundo.

Las aplicaciones de red son cada vez mas necesarias debido al crecimiento de la seguridad para las empresas, en este documento se desarrollo una herramienta para prevenir los ataques y para ubicar los equipos físicamente que están en riesgo así además de prevenir el problema se puede corregir, ya que si un equipo es vulnerable a ser manipulado por un hacker puede provocar la saturación del trafico de la red, desencadenando un bloqueo general de servicios disipándose a otros equipos, en el peor de los casos dañando los sistemas además de poder robar información importante o sensible, es decir datos personales como por ejemplo del tipo financiero del usuario.

La herramienta en esta tesis fue desarrollada con la tecnología de servidores y programación en java, JSPs, javabeans, applets para la parte de la administración de usuarios así como el montaje web del sistema usando también la herramienta de mapeo de redes llamada nmap para analizar la red, finalmente toda la información es almacenada con el gestionador de base de datos mysql, aclarando que todo este software es open source o de código abierto, haciendo al sistema una herramienta de bajo costo y de fácil implementación.

Índice

Presentación..... 1

Resumen..... 2

Agradecimientos..... 4

Introducción..... 5

Capítulo I
I.I. Base de Datos..... 9
I.II. Protocolos TCP/IP y Sniffers..... 17
I.III. Modelo y Aplicaciones Cliente-Servidor (tecnología JAVA servlets, JSPs, XML y la Seguridad SSL)..... 37
I.IV. Programación orientada a objetos en multiplataformas..... 61

Capítulo II
II.I. Planteamiento del Problema..... 67
II.II. Análisis del sistema..... 71

Capítulo III
III.I. Diseño del sistema..... 74
III.II. Implementación del Sistema de administración y control de IPs de una red (SACIP)..... 82

Capítulo IV
IV.I. Ejemplos..... 85

Capítulo V
V.I. Pruebas..... 91
V.II. Resultados..... 101
V.III. Conclusiones..... 104
V.IV. Bibliografía y Referencias..... 104
V.V. Glosario y anexos..... 106

Anexo I Notas importantes y legales sobre Nmap 116

Agradecimientos

Le agradezco en primer lugar a Dios, a mis padres, a mi familia en general, a mi novia Norma Angélica que me dio voluntad, esperanza y fe, y a mi asesor, que me soportaron, me ayudaron, apoyaron y me tuvieron paciencia en todo, la fuerza que me dieron por tanto tiempo, gracias también a todos los no mencionados pero que formaron una parte importante en la conclusión de esta tesis, que se la dedico a quien tuvo perseverancia y nunca dejo de creer que podía llegar a la meta con el viento en contra, al final lo importante es vivir y seguir adelante para alcanzar la felicidad en la vida.

Sistema de Administración y Control de IPS en una Red

Introducción.

Las tecnologías que se usan en estos tiempos cada vez avanzan con mayor crecimiento y rapidez por eso las aplicaciones sobre las telecomunicaciones son más exigentes con respecto al tamaño, la consistencia y otros factores que hacen eficientes la comunicación entre las diversas empresas globales, en específico para tener la certeza de saber si están bien instaladas y seguras, las redes internas hace menos propensas a fallar al exterior en redes amplias, también es importante decir que la tecnología que se tiene actualmente y se usa en nuestro país es atrasada a comparación de otros, así que cuando se trate de aplicar la nueva tecnología de hardware, el software va a tener que cambiarse por otro, pero si este software tiene bases que cualquier hardware necesita y puede interpretar, la aplicación no tendrá que cambiar radicalmente, así el estándar de un software se hará más flexible al cambio y de menor costo.

Una aplicación muy común es crear una base de datos para tener registrada, controlada y segura una red interna, sabiendo que esta red se puede conectar a redes externas o conexiones que también se hacen externamente, estas aplicaciones tienen fundamentos teóricos, que han sido probadas a lo largo de muchos años con modelos de tecnología en telecomunicación como son: la telefonía, radio y otros medios de comunicación.

El tema que me interesó sobre el área de computación en específico, de telecomunicaciones consiste en tener en cuenta las aplicaciones en cualquier tipo de red interna para tener un registro actualizado de las estaciones de trabajo, servidores o computadoras que componen internamente una red de una empresa, negocio, escuela, etc. y así tener la seguridad de tener solo las conexiones permitidas y sin problemas de vulnerabilidades, esto se puede traducir en una aplicación ya sea un analizador de vulnerabilidades de red o una relación entre los IPs y las direcciones físicas es decir un verificador de estados de los puertos para evitar intrusiones o problemas dentro la red.

Como se menciona anteriormente la teoría para la creación de aplicaciones similares se basan en tecnologías probadas y usadas actualmente, la mayoría de las redes locales y amplias tienden a desarrollar tecnologías de software más seguras y de menor costo es por eso que necesitan que estas aplicaciones aseguren la información y no dejen que haya infiltraciones externas o ajenas dentro del área de trabajo.

Actualmente existen en la red o empresas que se dedican a hacer muchas aplicaciones para resolver el mismo problema pero estas tecnologías muchas de las veces tienen un costo mayor al que poseen las empresas nuevas o con bajo alcance presupuestal, entonces a veces estar atados a los problemas de tener baja seguridad y de un control de la red costoso hace que estas redes estén en malas condiciones o mal controladas y vulnerables a cualquier intruso externo, aquí el problema no sólo es actualizarse sino mejorar los productos para combatir hackers, por eso una aplicación de este tipo debe estar en constante promoción y actualización, sin olvidar otras tecnologías y plataformas, una aplicación que sea usada remotamente de forma práctica, para mejorar la administración, bajar los costos, ayudando en la seguridad para un mismo propósito y que hace más flexible su precio y mantenimiento.

La aplicación de bases de datos hoy en día es importante para la administración de empresas, negocios, escuelas, dependencias del gobierno y la sociedad en general, la falta de una comunicación segura es consecuencia directa de este dilema, así las telecomunicaciones y las TI (tecnologías de la información) como soporte del sistema le brinda una mayor utilidad y expansión a la aplicación, es por eso que surgen nuevas aplicaciones que ayudan a que sean mínimos los "ataques" fuera de la red, el problema no solo radica en cuando o quien sino, de donde, y como erradicar ese tipo de inconvenientes.

Una base de datos con el registro de IPs, sus propiedades físicas de las conexiones y otros datos importantes para el análisis de los datos de la red, le indican al administrador alguna irregularidad sobre la red así como un control de la misma, ya que no se puede asegurar al 100% la red pero si se puede evitar ataques además de tener una experiencia a partir de un historial guardado

con las direcciones físicas y su respectivo IP que fue cambiando hasta donde la red haya fallado, o encontrar la raíz de un ataque sabiendo su ubicación dentro de la red y cuando suceda el evento obtener la mayor información que se pueda para su solución y posterior análisis.

Cabe mencionar que al actualizar los datos con regularidad es primordial para un buen desempeño de la aplicación ya que la base de datos por si sola es información y una administración-reconocimiento de las entradas y salidas le da a este sistema un amplio manejo de calidad de servicio.

Por lo tanto nos podemos hacer la pregunta:

¿Cómo apoyar para que una red de servicios en la seguridad y monitoreo funcione mejor?

No es posible que dos máquinas compartan el mismo IP, cuando a dos maquinas se les asigna el mismo IP la segunda no puede enviar paquetes a través de la Red y aparece un problema de negación de servicio o DoS (deny of service). Al establecer manualmente dichas direcciones es probable que se presenten este tipo de “colisiones”. Por otro lado, en últimas fechas el servicio de correo electrónico de la Facultad de Ciencias de la Computación recibe constantemente ataques de SPAM e incluso algunos de sus equipos fueron generadores de SPAM. La forma de identificar una computadora de otra es a través de una única dirección física, su ubicación y el responsable de la misma. Cuando enviamos un paquete IP entre estas dos máquinas sólo indicamos la dirección IP. Por lo tanto es necesario tener un mecanismo que nos proporcione la correspondencia entre la dirección IP y la dirección física. El problema de mapear las direcciones de alto nivel en direcciones físicas recibe el nombre de “*Address resolution*”.

Por los problemas antes mencionados particularmente es de suma importancia poder monitorear la red de la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla y poder ubicar los equipos que causen conflictos en la red para tomar las medidas pertinentes.

La respuesta es crear un sistema que se apoye en una base de datos que servirá como registro o historial de los IPs que deben estar dentro de la red solamente, además de sus características físicas de cada uno para poder administrarlos como parte de toda la red y asegurarse de que están correctamente instalados y configurados.

Con lo previsto sería aún más difícil poder administrar el sistema con un gigantesco software que tal vez no sea necesario, así entonces una simple aplicación como apoyo y control correcto de este sistema será más fácil y concreto de usar, eso no implica que sea de menor calidad o inseguro.

Se propone una asignación de los IPs de acuerdo a su ubicación dando un orden secuencial a edificios que a su vez se segmentan en módulos, salones, oficinas, laboratorios y cubículos de profesores, este orden y la comparación de los resultados obtenidos de la asignación de cada equipo a su respectiva ubicación con los datos recopilados de cual equipo esta en que departamento, piso y edificio es importante para el desempeño de la aplicación.

Este sistema permite monitorear dinámicamente el equipo conectado a Red de la Facultad de Ciencias de la Computación en la BUAP, con el fin de poder ubicar los equipos conflictivos de manera oportuna, así como detectar violaciones en la asignación manual de los IPs de cada área. De esta forma se logra un control local sobre los segmentos de la red y dicha información es disponible para su consulta vía Web.

Capitulo I

I.I. Base de Datos

I.I.I. Introducción.

Las bases de datos día a día empiezan a mejorar, facilitar y ayudar a la vida cotidiana en los negocios de pequeñas, medianas y grandes empresas; la interacción con las bases de datos ha sido prescindible desde que se creo la necesidad que los datos fueran recuperados para su posible modificación, consulta o eliminación como forma de administración y control de datos, en cualquier parte que tengamos que hacer una transacción virtual o que tenga que ver con los movimientos electrónicos y teniendo como intermediario a un sistema por computadora o instrumentos electrónicos, seguramente nos encontraremos que toda la información esta guardada en una base de datos, además de otros usos para la organización de datos científicos, estadísticos, imágenes digitales o multimedia, bibliográficos, médicos, geográficos, legislativos, medios educativos, etc.

Al cabo de 5 décadas de investigación (tal vez más), 30 años de desarrollo y pruebas de tecnologías en bases de datos se han usado diferentes tipos para las diferentes aplicaciones, ya que no existe una única forma de crear las bases de datos, esto da una generalidad al desarrollo de tecnología y amplía la capacidad de solventar problemas de organización y aplicación de datos; todo esto se basa en la experiencia que se adquiere en la solicitud de los requerimientos o las necesidades de la creación de una base de datos y en fundamentos teórico para generar todo lo anterior.

Históricamente la necesidad de organizar los datos, encontrarlos más rápido y con mayor eficiencia llegó al punto que fue demasiada la información y mucho el trabajo para que todas las tareas las pudieran desempeñar unas cuantas personas, aunque en la actualidad no toda la información esta al alcance con una base de datos, poco a poco se ira dando el cambio, esto significa que la

tendencia deberá cubrir todas las necesidades de la obtención de la información a través de un medio más eficaz y sin pérdida de la misma.

Así entonces la definición de una base de datos de una forma muy general es una colección de datos relacionados. Por dato entendamos como hechos conocidos que pueden registrarse y que tienen un significado implícito. Como por ejemplo: los nombres, números de teléfono y direcciones de personas. Normalmente esto se anotaría en una agenda o libreta de direcciones ordenados alfabéticamente o indexados por el apellido o por nombre de la persona, o también podría almacenar la información para el caso de mayor flexibilidad en un disquete en un archivo de texto de forma muy sencilla, o en software que pueda ayudar para este propósito. Por lo tanto esto es una colección de datos relacionados con un significado implícito, es decir, una base de datos.

El uso habitual del término base de datos suele ser más restrictivo. Una base de datos tiene las siguientes propiedades implícitas, esto es que debe cumplir estrictamente con lo siguiente:

- Una base de datos representa algunos aspectos de mundo real.
- Una base de datos es una colección coherente de datos con significados inherentes. Un conjunto de datos aleatorios no puede considerarse como una base de datos.
- Una base de datos se diseña, se construye y se llena con datos para un propósito específico. Esta destinada a un grupo de usuarios en concreto y tiene algunas aplicaciones preconcebidas en las cuales están interesados dichos usuarios. [1]

Los puntos anteriores son de gran importancia en la creación de las bases de datos ya que son el inicio de la coherencia que deben llevar antes de comenzar el diseño y la implementación de las mismas.

[1] Definición de Base de datos, “Fundamentos de sistemas de bases de datos”, 1.1 Introducción, Pág.4, P.2, 3 y 4, Aut. Elmasri-Navate, EDT. Addison Wesley, 3ª ED.

El tamaño de las bases de datos puede incrementarse dependiendo de la cantidad de datos que contenga y se quiera anexar, la complejidad puede variar dependiendo también de los datos que se quieran manejar y la forma en que deban relacionarse.

Las bases de datos nunca estarán estáticas en el momento que estén en uso ni en el tiempo de vida en el que se usen, esto quiere decir que el mantenimiento de las bases de datos es de gran relevancia ya que la forma en que se construyen no es la única y si es prescindible hacerle cambios para actualizarlas, extenderlas e incluso cambiar completamente su implementación por nueva tecnología, es necesario saber hasta cuando usarla y por cual otra cambiarla. Normalmente esto casi no sucede ya que un buen diseño no requiere grandes cambios, solo si los cambios suceden en la base de las necesidades del negocio.

La preocupación por el mantenimiento de las bases de datos se ha solucionado con un sistema de gestión o administración de base de datos (SGBD, o bien DBMS por las siglas en inglés *database management system)* que es una colección de programas que permiten a los usuarios crear y mantener una base de datos. Estos tienen el propósito de generalizar y facilitar los procesos como son: la definición, construcción y manipulación de bases de datos para distintas aplicaciones, así solo hay que preocuparse por decidir cual o que DBMS usar y se acomoda mejor a nuestras necesidades. Entonces la tecnología la debe desarrollar cada DBMS y no la persona o empresa que hace el diseño o la estructura de los datos.

La definición de una base de datos consiste en especificar los tipos de datos, la estructura y restricciones para los datos que se van a almacenar en dicha base. La construcción de la base de datos es el proceso de almacenar los datos concretos sobre algún medio de almacenamiento controlado por el SGBD. La manipulación de la base de datos incluye funciones tales como consultar la base de datos para recuperar unos datos específicos, actualizar la base de datos

para reflejar los cambios ocurridos en contexto de nuestro mundo real y generar informes a partir de los datos.[2]

I.I.II. Enfoque de las bases de los datos.

Las tecnologías que se han ido desarrollando a lo largo de varios años, mejorando la forma en que es almacenada y organizada la información ha cambiado el sentido de guardar los datos como tales, sin embargo lo aprendido en lo anterior sirve para investigar y buscar nuevas formas de hacer lo mismo, desde entonces se han generado una serie enfoques hacia las bases de datos, como las usadas para guardar en sistemas de ficheros, que aun ahora son usados por tener algunas "ventajas" sobre los SGBD como es la simplicidad y el costo, pero actualmente estas "ventajas" no son justificables ya que con la información importante se debe ofrecer garantías de seguridad, contener datos actualizados sin redundancia y sin errores que no dependan de los humanos.

La separación de los programas de los datos y su abstracción de los mismos es prescindible para poder construir una buena base de datos, ya que no debe existir mas relación que la de manejar los datos y control de los mismos a través del programa, es decir que los usuarios que tienen acceso a la base de datos solo deben operar sobre los datos a través de operaciones que les permitan ejecutar este tipo de acciones. Esta abstracción de los datos o separación, se utiliza para representar conceptualmente un modelo de datos. Este modelo puede ser entendido mejor por sus conceptos lógicos, y sus relaciones que por conceptos de como guarda la información una computadora. Por lo tanto se ocultan todos los detalles de implementación y almacenamiento que no importan para el desarrollo de las aplicaciones y para los usuarios de la base de datos.

También el problema que se quiere resolver con las bases de datos es la distribución y concurrencia de los datos entre los usuarios ya que estos deben estar actualizados, mostrar lo que realmente contiene la base de datos además de ofrecer los datos en tiempo real su modificación si así sucede y la veracidad de

[2] Sistema de gestión de base de datos, "Fundamentos de sistemas de bases de datos", 1.1 Introducción, Pág.5, P.3, Aut. Elmasri-Navate, EDT. Addison Wesley, 3ª ED.

ellos es decir asegurar el control del acceso de los usuarios a la base de datos y los datos que devuelva el sistema administrador de base de datos, para solucionar esto se genera el soporte de varias vistas de los datos para los usuarios activos dentro de la base de datos. Así la perspectiva de cada usuario puede ser diferente ya que cada uno de ellos desearía tener el resultado de una consulta o cualquier acción sobre la base de datos al mismo tiempo que los otros usuarios que también quisieran hacer esto, el problema de la concurrencia lo resuelve internamente el manejador del lenguaje, incluyendo la vista de los datos para los usuarios que se definen como un subconjunto de la base de datos o datos virtuales que se derivan de los ficheros de la base de datos pero no están explícitamente almacenados, es decir que no importa si los datos están almacenados o derivados.[3]

Para construir una buena base de datos se necesita, un administrador de la base de datos, un diseñador, analistas de sistemas y programadores de aplicaciones, además de los usuarios que usaran la misma. Cada uno desempeña un papel importante ya que en conjunto crean el buen funcionamiento del sistema, estos personajes dentro de este sistema se encargan de diferentes tareas en primer lugar el administrador de autorizar el acceso a las base de datos coordinar y vigilar la utilización de los recursos necesarios para mantenerlas funcionando correctamente y de que la seguridad no sea violada.

Otro personaje dentro del sistema es el diseñador de base de datos y como lo dice se encarga de verificar, de ver como se van a almacenar y encajar los datos con la estructura para representarlos, es decir ver cuales son las necesidades de los usuarios que utilizaran el sistema, entonces se presenta un diseño que cumpla con esto pero como los grupos de usuarios no tienen las mismas necesidades se toma un estándar y se incluyen varias peticiones para satisfacer las necesidades de todos.

[3] Múltiples vistas o Datos virtuales y procesamiento de transacciones multiusuarios, “Fundamentos de sistemas de bases de datos”, 1.3.3. , Soporte de múltiples vistas de los datos Pág.10, 11, Aut. Elmasri-Navate, EDT. Addison Wesley, 3ª ED.

Los que se encargan de hacer real o posible la interacción entre usuario-base de datos, además de depurar, documentar y mantener las transacciones programadas, esto se hace conociendo bien el sistema de administración de base de datos.

Los usuarios representan el reflejo del esfuerzo de los personajes anteriores, por lo general son los que consultan y realizan tareas comunes como dar de alta, baja y modifican los datos, todo los tipos de usuarios tienen diferentes intereses en los datos pero siempre necesitaran usar las herramientas que fueron ofrecidos por el administrador, el diseñador, el analista de sistemas y programadores de aplicaciones.

Un punto interesante de los sistemas de administración de las bases de datos es el cumplimiento de lo que debe de hacer como dice que lo hace, es decir, la integridad que deben tener las condiciones con las que esta trabajando o restricciones de integridad, para tal caso se debe garantizar que se debe seguir cumpliendo con lo mismo, por ejemplo un tipo de datos, cuando se declara un valor para un dato de entre 0 y 1 y que la variable que es declarada del mismo tipo no debe pasar la longitud de 15 caracteres.

Dentro de lo que también se ofrece con los puntos anteriores debe existir un método o forma de recuperar y respaldar los datos que están dentro de la base de datos, así si el sistema falla el sistema de administración de base de datos se debe encargar de que todos los datos que existían hasta antes del fallo queden sin modificaciones y coherentes, además de crear un registro completo del fallo para saber que hacer en otra ocasión que exista el mismo problema.

A las bases de datos se les puede ver como un conjunto de relaciones, esto significa que cada fila estaría representando de las relaciones como una tabla de valores, al igual que en el modelado es expresada como una entidad del mundo real, además de filas, al título de la columna de la tabla se le llama atributo y a toda la tabla se le nombra relación. Al tipo de dato que describe el valor que puede haber en cada columna se le dice dominio, esta descripción necesita de un formato para cada tipo de dato.

Las operaciones dentro de las bases de datos han sido estandarizadas debido a que son las que normalmente podemos encontrar en las aplicaciones que involucren movimientos de datos. Estas operaciones dentro del modelo relacional se pueden ver como de actualización y de recuperación; las de operación pueden ser *alta, baja, modificación, consultas, respaldo y restauración de la base de datos*. El movimiento de alta sirve para anexar uno o más registros nuevos dentro de una relación. El movimiento de baja sirve para eliminar o borrar registros, y las modificaciones como su nombre lo dice es para corregir o actualizar los registros dentro de la base de datos. El movimiento de consultas realiza verificaciones de datos y visualiza la información del contenido de la base de datos dependiendo de las necesidades del usuario. El movimiento de respaldo de la base de datos se utiliza para poder guardar el estado de los datos para poder recuperar o restaurar después si ocurre alguna alteración en la integridad de los datos, o de la misma base de datos, así queda en el estado antes de ese error, pero es responsabilidad del administrador del sistema hacer respaldos de la base de datos regularmente para tener un respaldo de los últimos datos obtenidos dentro de la base de datos.

I.I.III. Lenguaje del sistema administrador de base de datos (el estándar SQL y su relación con MySQL).

El lenguaje SQL ha sido el mejor avance en cuanto a las bases de datos relacionales se refiere, ya que se convirtió en un estándar. El álgebra relacional se ha enfocado a desarrollar y sacarle provecho a este lenguaje durante todos los años de búsqueda de una herramienta que pudiera englobar las necesidades de todos los usuarios que querían migrar aplicaciones de base de datos de un sistema a otro. Así, si se tienen 2 o más sistemas bajo un mismo estándar pueden trabajar y migrar datos de uno a otro sin problemas de compatibilidad y la simplicidad será mayor, entonces el usuario debe especificar como o en que orden se ejecutan las operaciones de consulta. La interfaz declarativa del lenguaje SQL proporciona al usuario esta ventaja de recuperar cual es resultado especificándolo, SQL se encarga de la optimización y la mejor forma de ejecutar la consulta. El significado de SQL son las siglas en inglés Structured Query Language, implementado por IBM Research como una interfaz para un sistema

experimental de base de datos relacional, después con ayuda del *Instituto Americano Nacional de Estándares (ANSI)* y la *Organización Internacional para la Estandarización (ISO)* dio lugar a un estándar SQL1 en su versión de 1986 y después al siguiente estándar del año 1992 llamado SQL92, y en los siguientes años se estará desarrollando la versión SQL3 que extenderá otros conceptos que son específicos para nuevas aplicaciones en base de datos.

En esta tesis solo basta con mencionar la relación del estándar SQL con su versión gratuita MySQL desarrollada por MySQL AB para el sistema de administración y control de IPS que es el interés principal para la implementación del mismo.

La definición de MySQL como herramienta de desarrollo de base de datos relacionales empieza por ser un sistema de gestión de bases de datos relacional, que tiene licencia por la *Licencia Pública General o GPL (General Public License)* de la *GNU* es decir acrónimo recursivo que significa *"GNU No es Unix"*, y la *Iniciativa de Código Abierto u Open Source Initiative (OSI).* El diseño que esta basado en multihilo permite soportar una gran cantidad de carga de forma muy eficiente. MySQL fue creada como una opción mas para SQL y que mantiene el copyright del código fuente del servidor SQL, la empresa de origen sueco MySQL AB.

Como MySQL surgió de la necesidad de un grupo de personas sobre un gestor de bases de datos rápido, sus desarrolladores fueron implementando únicamente lo fundamental, intentando hacerlo funcionar de forma óptima. Es por ello que, aunque MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas características. Aunque básicamente MySQL fue diseñada con estas características, debido a que lo que buscaban era un gestor de bases de datos con una gran rapidez de respuesta. Pero ha sido con la distribución de MySQL por Internet, cuando más y más aplicaciones requieren mayores funcionalidades, por lo que serán incluidas en futuras versiones del gestor. Para el sistema de administración y control de IPs de una red se implementó bajo la versión que se ha descrito anteriormente y es de suma importancia saber cuales son sus ventajas, desventajas y riesgos para el desarro-

llador, ya que esto indica una buena aplicación del sistema. Se puede encontrar más información sobre este gestor en el manual dentro de la documentación contenida en la distribución en la siguiente dirección www.mysql.com que esta sujeta a actualizaciones y cambios por obvias razones del paso del tiempo.

I.II. Protocolos TCP/IP y Sniffers.

I.II.I. El modelo OSI de 7 capas.

Para poder empezar una explicación más extensa sobre la introducción al análisis de protocolos normalmente se debe dar una revisión al modelo OSI.

Cualquier sistema previo al modelo OSI, la red era generalmente era "monolítico". En otras palabras, la aplicación que mostraba los datos en la pantalla fue también responsable para que el hardware moviera los bits a través de la red. No se podía cambiar ninguno de los dos (software o hardware) para actualizar el sistema entero. Este problema se traduce a tener que comprar una nueva computadora solamente para actualizar el software.

El concepto detrás del modelo OSI es separar la funcionalidad en diferentes módulos conceptuales. Como una práctica introducción, se considera el siguiente modelo de 3-capas en la tabla I.II.I.I. :

Aplicación	Web browser, e-mail, RealAudio
Transferencia	TCP/IP
Conexión	Módem de marcación por tono, módem de cable, DSL, Ethernet

Tabla I.II.I.I Descripción del modelo OSI de 3 capas.

Con esta representación conceptual, donde la aplicación "Web Browser" o navegador del usuario trata de visualizar una página web o “web-page” localizada en un "Web Site" o sitio web que puede estar en la red en cualquier parte. El navegador pasa por la pila del protocolo "TCP/IP", el cual lo manda a través de

la "NIC" o tarjeta de red (Network Interface Card) por la conexión local a la entrada del "Router" o ruteador más cercano. En este punto, el cliente no sabe que le esta pasando a los datos. En realidad, los datos pasan de un ruteador-a-ruteador a través de Internet hasta que encuentra el Servidor destino que alberga el sitio web.

Lo importante que se debe aprender de todo esto es el concepto de abstracción. Cada componente del diagrama sabe lo de los otros componentes. Por ejemplo, un cartero que entrega el correo (físicamente). El cartero no sabe del contenido de las cartas. Simplemente mueve la correspondencia entre la oficina postal local y la pone en el buzón. De la misma forma la capa IP dentro de una computadora no tiene conocimiento del contenido de los paquetes. Su única responsabilidad es aceptar paquetes para la capa TCP, y mandarlos hacia la NIC al ruteador local. La capa IP es incluso difusa en lo detalles de como la NIC transporta los paquetes al ruteador local, y es completamente incierto como todo lo que le pasa a los paquetes en ese punto. En otras palabras, cada capa tiene un simple trabajo que hacer, y no sabe nada acerca de lo que pasa en las otras capas.

El grupo de trabajo de la OSI (OSI = Open Systems Interconnect o Interconexión de Sistemas Abiertos) como parte de la ISO (International Organization for Standardization u Organización Internacional para la Estandarización) fue creado para estandarizar protocolos de red. En teoría, si todos acordaban estandarizar, entonces los consumidores que compraban productos de diferentes vendedores lo harían a menor precio y ahorrando más dinero.

El modelo OSI fue el plano de diseño para el conjunto completo de protocolos que querían implementar capas individuales. Finalmente se tuvo éxito en generar un estándar, pero este nunca llegó a ser de uso popular y a la larga tuvo que ser sustituido por TCP/IP. Mientras varias organizaciones (gubernamentales e industriales), la mayoría de Europa ha intentado usarlo, este por mucho se convirtió en ancla de barco y se estancó.

Así las capas se pueden resumir en una serie de definiciones de conceptos como sigue:

- La capa física (1) envía bits sobre la red.
- La capa de enlace de datos (2/Ethernet/PPP) envía frames lo más próximo al “next hop” o siguiente salto de ruteador.
- La capa de red (3/IP) envía paquetes lo más próximo a la máquina destino a través de Internet.
- La capa de transporte (4/TCP) crea conexiones para el programa en la máquina de destino.
- La capa de aplicación (7/HTTP/SMTP/POP/IMAP) comunica la información recibida (por ejemplo archivos) al usuario.
- La capa de sesión (5) y la capa de presentación (6) como se describió anteriormente no tienen utilidad para fines prácticos.

I.II.II. Paquete.

Todo lo que se transfiere bajo Internet empaquetado individualmente en unidades llamadas "packets" o paquetes. A estos les toma entre 230 y 250 paquetes para transmitir este documento a alguna computadora, Por ejemplo cada paquete es etiquetado con una dirección que especifica su destino.

El truco es que todo lo que es enviado por la computadora necesita hacerse por estos paquetes. Otro ejemplo es cuando se escucha radio por Internet a un flujo de difusión, este aparece como un flujo continuo, pero en realidad el transmisor esta dividiendo los datos en paquetes individuales, después la computadora los reensambla en un flujo completo.

El esfuerzo de los sniffers consiste en buscar paquetes individuales, reensamblar los datos, o recuperar información (como passwords o contraseñas).

La manera que se conecta en tres-pasos el protocolo TCP. TCP es un protocolo "orientado a la conexión". Esto significa que antes de que se envíen datos a través de la red, se debe establecer la conexión. En términos de 3 pasos se puede definir lo que TCP realiza para hacer esto:

- Me gustaría comunicarme contigo
- Claro, vamos a hablar

- Gracias

Los protocolos similares a TCP usan 2, 3, o 4 packet “apretones de manos” o “handshakes” para poder establecer la conexión. Una gran cantidad de trabajo se realizó tratando de optimizar este intercambio de 3 paquetes.

Hablando de TCP, este intercambio es el siguiente:

- SYN
- SYN-ACK
- ACK

Donde "SYN" es una bandera en la cabecera TCP que significa "vamos a empezar a comunicarnos ", y el cual solo ocurre en los 2 primeros paquetes en el intercambio. El campo "ACK" significa que el campo "reconocimiento" es válido.

Otra cosa importante que recordar es cuando una conexión TCP sucede desde un puerto en una máquina al puerto de la otra. Por ejemplo, un servidor web típicamente usa el puerto 80, y el puerto del cliente es alojado empezando en el puerto 1024. Por lo tanto, para poder entender mejor la comunicación observe la tabla I.II.II.I:

Banderas	Fuente	Destino	Secuencia	Reconocimiento
SYN	1037	80	102723769	0
SYN-ACK	80	1037	1527857206	102723770
ACK	1037	80	102723770	1527857207

Tabla I.II.II.I Tabla de ejemplo conexión TCP entre dos equipos por el puerto 80.

El estándar que genera el conjunto de tecnologías que permiten interconectar redes muy distintas entre sí es llamado Internet este no es un nuevo tipo de red física. Este no depende de la computadora ni el sistema operativo que utilice para poder transmitir la información. Así esta comunicación puede ser posible entre diferentes tipos de conexiones y desde diferentes lugares ya sea un ser-

vidor Unix con una computadora que utilice Windows. O entre plataformas completamente distintas como Macintosh, AMD o Alpha, entonces la interacción entre una computadora y otra o entre redes y que en la realidad existirán generalmente distintas tales como: Ethernet, Token Ring u otro tipo de conexión por ejemplo por satélite. No se puede estandarizar ningún protocolo que dependa de arquitecturas cerradas, esto es que no sean portables o compatibles con las demás, ya sea la plataforma, sistema operativo o tipo de red. La estandarización permitió que los protocolos que se eligieron para Internet fueran para que se creara una red mundial y uno de estos es el protocolo TCP/IP.

La red esta definida por medio de las direcciones IP para poder ser configurado en cada computadora y no con el cableado. Así, si instalamos varias redes con el cableado solamente las computadoras que pertenezcan a una misma red podrán comunicarse entre sí. Si el objetivo es comunicar computadoras de una red a otra deben existir los llamados ruteadores. Estos ruteadores generalmente son computadoras con distintas direcciones IP, una por cada red, y se dedica a transferir el tráfico de paquetes por ellas.

Sobre la capa de red se fragmentan cada uno de los mensajes enviados a través de la red en *datagramas* además de enviarlos de forma independiente a través de la red de redes. En cada uno de los datagramas IP se incluye un espacio asignado para la dirección IP a donde se dirige o IP destino. La información de este último campo se utiliza para agilizar el tráfico y enrutar los datagramas pasando sobre las redes que sean necesarias hasta llegar a su destino.

I.II.III. Protocolo IP.

IP es el principal protocolo de la capa de red. Este protocolo define la unidad básica de transferencia de datos entre el origen y el destino, atravesando toda la red de redes. Además, el software IP es el encargado de elegir la ruta más adecuada por la que los datos serán enviados. Se trata de un sistema de entrega de paquetes (llamados datagramas IP).

Para fines teóricos para entender más sobre el siguiente capítulo que es el de sniffers y tomando como referencia marcos intercambiables de una red física los datagramas contienen un encabezado y un área de datos. El protocolo IP específicamente no muestra el contenido de los datos, por eso el protocolo de transporte utilizará los datos arbitrariamente.

En la trama física se tiene un campo de datos donde se transportan los datagramas IP, pero este debe tener una longitud definida ya que está limitado por el diseño de la red. La unidad de datos para medir la cantidad de datos que puede transportar su trama física se le llama MTU (*Maximum Transfer Unit*) o Unidad Máxima de Transferencia, que para las redes Ethernet es 1500 bytes y el de las redes Token-Ring, 8192 bytes. Así sobre una red Ethernet solo podrá transportar un datagrama de a lo máximo 1500 bytes sin fragmentarlo. Un ruteador fragmenta un datagrama en varios si es necesario es decir si el siguiente tramo de la red por el que tiene va a viajar el datagrama tiene un MTU menor que la longitud del datagrama.

I.II.IV. Dirección MAC o MACAddress.

Todas las computadoras de una misma red comparten el mismo medio, por lo que debe de existir un identificador físico único para cada equipo, o mejor dicho para cada tarjeta de red. Esto no sucede en una conexión telefónica mediante módem, ya que se supone que cualquier dato que se envía está destinado al equipo que se encuentra al otro lado de la línea. Pero cuando se envían datos en una red local, hay que especificar claramente a quien van dirigidos. Una dirección MAC es de la forma 00:C0:4F:68:BA:50. Esto se consigue mediante la dirección MAC, un número compuesto por 12 dígitos hexadecimales o 6 octetos que identifica de forma única a cada dispositivo ethernet, dicha dirección se encuentra grabada de fábrica en la tarjeta NIC ("Network Interface Card"), donde los primeros tres octetos (00:C0:4F) pertenecen al vendedor de la tarjetas NIC (asignado por IEEE) y los otros tres (68:BA:50) son una serie exclusiva asignada por el vendedor, es decir se compone de 48 bits, así los 24 primeros bits identifican al fabricante del hardware, y los 24 bits restantes corresponden al número de serie asignado por el fabricante, lo que garantiza que dos tarjetas

no puedan tener la misma dirección MAC. Si existieran direcciones MAC duplicadas causarían problemas en la red. Para comunicarnos en una red ethernet con otro equipo de la misma es necesario conocer su dirección MAC, la cual como se mencionó anteriormente es única, para saber las direcciones MAC de los equipos de nuestra red, y cada vez que se requiere hacer comunicación con otra computadora de la red, se debe conocer su dirección MAC.

I.II.V. Direcciones IP.

Como ya se ha hablado antes de lo que es una dirección IP, se definió como el identificador de cada computadora concentrada en su red de redes. Cada host o computadora conectado a cualquier red tiene una dirección IP asignada por disposición del ruteador, además debe ser distinta a todas las otras direcciones que estén asignadas previamente a otras computadoras en ese momento. Para el caso de pertenecer a la red de redes internacional o Internet, no debe existir dos computadoras con dos direcciones IP iguales (públicas). Pero de forma general se puede tener dos computadoras con una misma dirección IP con la condición que ambas sean de redes independientes.

Entonces las direcciones IP se pueden dividir en varias clases como son:

Públicas. Estas direcciones IP son generales en Internet es decir tienen acceso desde todo el mundo o desde donde se pueda acceder a internet. Es visible para todas las computadoras y para acceder a ellas se debe tener Internet y una dirección IP pública.

Privadas o reservadas. Su acceso solo es permitida por otros host o computadoras de cualquier red privada con los respectivos privilegios ya sea externa o desde otras redes privadas interconectadas por ruteadores. Este tipo de redes son usadas para comunicar solamente al host dentro de una empresa en puestos de trabajo y que deban ser accesadas solo por personal autorizado, este tipo de redes además tiene otra característica los host con direcciones IP privadas tiene acceso a Internet a través de un ruteador o un servidor Proxy

que tenga una IP pública. Por otra parte, cualquier otro tipo de red que no sea de tipo privada que este dentro de Internet no puede acceder a computadoras con direcciones IP privadas.

Dentro de la red los hosts pueden tener diferentes tipos de direcciones IP dependiendo de las necesidades de la red pueden ser:

Estáticas o fijas. Una computadora dentro de una red que tenga un IP de este tipo siempre tendrá el mismo IP cada vez que se conecte a la red. Un servidor de aplicaciones Web, de ftp o de otro tipo de servicio tiene una dirección IP pública estática así cuando se les mande a llamar desde otro host siempre puedan ser encontradas con la misma dirección por usuarios de Internet.

Dinámicas. Cada vez que cualquier computadora que se conecte por medio de dirección IP dinámica, se conectará con una IP diferente siempre. Este tipo de IP dinámica son asignados automáticamente a una computadora por un servidor que realiza el protocolo de configuración de host dinámico o DHCP (Dynamic Host Configuration Protocol). Esto es para asignarle un IP a los host que no tienen que configurarse para darse de alta y obtener acceso a la red.

Una dirección IP esta descrita por 4 cifras de 1 bytes cada uno es decir 32 bits, separados por un punto cada uno es decir de la forma byte1.byte2.byte3.byte4 cada byte esta dentro del rango de 0 y el 255.

Así un IP 148.228.20.34 que esta de manera decimal se traduce de forma binaria de la siguiente forma: 10010100.11100100.00010100.00100010.

Las necesidades de cada red definen las clases primarias de direcciones de red que existen en internet, estas pueden ser direcciones de clase tipo A, B y C. Por definición las direcciones de clase tipo D identifican un grupo de host.

Por otra parte, el gran crecimiento de Internet en los últimos años ha creado también dificultades para encaminar el tráfico entre el número cada vez mayor de redes que la componen. Esto ha creado un crecimiento exponencial del ta-

maño de las tablas de encaminamiento que se hacen cada vez más difíciles de sostener.

Los problemas comentados se han solucionado en parte hasta la fecha introduciendo progresivos niveles de jerarquía en el espacio de direcciones IP. No obstante, la solución a largo plazo de estos problemas obligó a desarrollar la próxima generación del protocolo IP (IPng - IP "next generation" o IPv6).

I.II.VI. Protocolo ARP.

En una red interna las computadoras se comunican a través de las tramas físicas. Estas tramas son de tipo Ethernet y contienen campos para almacenar las direcciones físicas del origen y del destino cada una contiene 6 bytes, en la tabla I.II.VI.I. se muestra la estructura de la trama Ethernet con sus respectivos campos.

8 bytes	6 bytes	6 bytes	2 bytes	64-1500 bytes	4 bytes
Preámbulo	Dirección física destino	Dirección física origen	Tipo de trama	Datos de la trama	CRC

Tabla I.IIVI.I. Campos de trama del protocolo ARP.

Cuando la computadora manda información a una red, necesita una forma de encontrar su destinatario, esto se logra en varias capaz dependiendo de cuan lejos debe viajar. En la primera capa, se encuentra la dirección MAC. La dirección MAC solo se usa para comunicar dentro de segmentos de la red local o subredes. Una vez que pasa por el ruteador o switch y pasa a otra subred entra a la segunda capa. En esta capa se tiene que transformar el MAC porque la base de datos seria demasiado grande para poder procesar rápidamente ya que la dirección de MAC es un hexadecimal de 48bits. Para esto existen diferentes servicios como DNS (Domain Name Service), WINS (Windows Internet

Naming Service), y la IP (Internet Protocol) que se encargan del manejo de datos.

El protocolo ARP (Address Resolution Protocol, protocolo de resolución de direcciones), es el encargado de “traducir” las direcciones IP de 32 bits a las correspondientes direcciones de hardware. En ethernet & Token Ring estas direcciones suelen tener 48 bits. La traducción inversa la hace el protocolo RARP o Reverse ARP. ARP se encarga de manejar la relación entre el identificador MAC y la IP.

Cuando cualquier computadora dentro de la red necesita resolver una dirección IP a una MAC, lo que hace es efectuar una petición arp (Arp request) a la broadcast de dicho segmento de red, es decir se difunde la petición por todo el segmento, FF:FF:FF:FF:FF:FF, solicitando que el equipo con dicha IP responda con su dirección ethernet (MAC).

Esquemáticamente el proceso se observa en la figura I.II.VI.II.

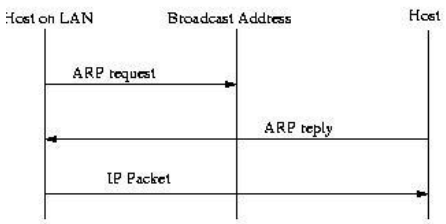


Figura I.II.VI.II. Broadcast o difusión de petición del protocolo ARP.

Con el fin de reducir el tráfico en la red, cada arp-reply que llega a la tarjeta de red es almacenado en la caché o memoria de la tabla de resolución que se almacena temporalmente, incluso si la petición no la realizamos nosotros. Es decir, todo arp-reply que nos llega es almacenado en la caché. Este factor es el que se utiliza para realizar arp-spoofing esta técnica se describirá más a fondo en la sección de sniffers.

Cada ordenador almacena una tabla de direcciones IP y direcciones físicas. Cada vez que formula una pregunta ARP y le responden, inserta una nueva

entrada en su tabla. Para evitar incongruencias en la red debido a posibles cambios de direcciones IP o adaptadores de red, se asigna un tiempo de vida de cierto número de segundos a cada entrada de la tabla. Cuando se agote el tiempo de vida de una entrada, ésta será eliminada de la tabla. Las tablas ARP reducen el tráfico de la red al evitar preguntas ARP innecesarias.

I.II.VII. Redes Ethernet.

La ethernet fue concebida en torno a una idea principal: todas las máquinas de una misma red local comparten el mismo medio (el cable). Todas las máquinas son capaces de "ver" todo el tráfico de la red. Debido a esto, las tarjetas ethernet incorporan un filtro que ignora todo el tráfico que no está destinado a él. Esto se consigue ignorando aquellos paquetes cuya dirección MAC (Media Access Control) no coincide con la suya. Un sniffer elimina este filtro de la tarjeta de red y la coloca en modo promiscuo. De esta forma la tarjeta es capaz de "ver" todo el tráfico que pasa por la red, o con solo utilizar una aplicación que trabaje a nivel bajo de las conexiones para obtener los resultados deseados. Solo es cuestión de colocar los filtros adecuados y comenzar a capturar los paquetes que más nos interesen o recuperar los datos a través de una auditoria de la red (obtener estadísticas del tráfico de la red y de las conexiones por cada puerto, verificar que computadoras están activas, su IP y MacAddress, consultar su estado de conexión, verificar que computadoras trabajan como servidores o que sistemas operativos tienen dentro de una red, etc.).

I.II.VIII. Administración de una Red.

Lo que puede o no puede hacer una red está generalmente determinado por los protocolos que dicha red soporta y la calidad de sus implementaciones, más que por la tecnología concreta de red usada, como Ethernet, Token Ring, etc. Además, en la práctica, la elección de la tecnología de red está basada en decisiones puramente pragmáticas: qué tipo de red soporta el tipo de ordenadores que queremos conectar, las distancias entre los equipos, las características del cableado, etc. Por regla general, se suele usar Ethernet para sistemas de media escala, Ethernet o una red basada en el cableado de par trenzado para pequeñas redes, o redes de alta velocidad. En específico dentro la Facultad de

Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla (BUAP) se amplió el servicio de internet instalando redes inalámbricas como parte del servicio al estudiante y docente.

La administración de redes abarca un amplio número de asuntos. En general, se suelen tratar con muchos datos estadísticos e información sobre el estado de distintas partes de la red, y se realizan las acciones necesarias para ocuparse de fallos y otros cambios.

La técnica más primitiva para la monitorización de una red es hacer "pinging" a los hosts críticos; el "pinging" se basa en un datagrama de "echo" (eco), que es un tipo de datagrama que produce una réplica inmediata cuando llega al destino. La mayoría de las implementaciones TCP/IP incluyen un programa (generalmente, llamado "ping") que envía un echo a un host en concreto. Si recibimos réplica, sabremos que host se encuentra activo, y que la red que los conecta funciona; en caso contrario, sabremos que hay algún error. Mediante "pinging" a un razonable número de ciertos hosts, podremos normalmente conocer qué ocurre en la red. Si los ping a todos los hosts de una red no dan respuesta, es lógico concluir que la conexión a dicha red, o la propia red, no funciona. Si sólo uno de los hosts no da respuesta, pero los demás de la misma red responden, es razonable concluir que dicho host no funciona.

Técnicas más sofisticadas de monitorización necesitan conocer información estadística y el estado de varios dispositivos de la red, desde hardware o software. Para ello necesitará llevar la cuenta de varias clases de datagramas, así como de errores de varios tipos. Desde hardware este tipo de información será más detallada en los ruteadores, puesto que el ruteador clasifica los datagramas según protocolos e, incluso, él mismo responde a ciertos tipos de datagramas. Es posible recopilar toda esta información en un punto de monitorización central por medio de un software o sistema que tenga un soporte de este tipo de opciones y no necesariamente obtener los datos desde un ruteador, aunque si seria ideal.

También hay un enfoque oficial TCP/IP para llevar a cabo la monitorización. En la primera fase, se puede usar un conjunto de protocolos ARP, TCP/IP, ICMP y

UDP entre otros, que forman la opción de hacerlo desde software o los llamados sniffers.

En términos generales, todos estos protocolos persiguen el mismo objetivo que es permitir recoger información crítica de una forma estandarizada. Se ordena la emisión de datagramas TCP/IP o UDP desde un programa de administración de redes que se encuentra ejecutando en alguno de los hosts de red. Generalmente, la interacción es bastante simple, con el intercambio de un par de datagramas: una orden y una respuesta. También existen mecanismos de seguridad más elaborados, basados en la criptografía por ejemplo los certificados y firmas usados por SSL (Secure Socket Layer) y TLS (Transport Layer Security) que se comentará de forma más amplia como funcionan en el capítulo I.III. de este documento.

También existen varias herramientas que ahorran tiempo, esfuerzo y dinero en la implementación de esto como por ejemplo los sistemas donde se puede visualizar la descripción de los equipos o un mapa de la red, donde los objetos cambian de color cuando cambian de estado, además de mostrar cuadros de estadísticas sobre los datagramas y otros objetos, por decir algunas ideas para llevar a cabo la administración de la red gráficamente o de forma más fácil.

Otro tipo de monitoreo deseable es recolectar información para hacer informes periódicos del porcentaje de uso de la red y prestaciones. Para ello, necesitamos analizar cada dispositivo de conmutación y quedarnos con los datos de interés.

Sería posible que cualquier tipo de conmutador pudiese usar cualquier tipo de técnica de monitorización. Sin embargo, generalmente los repetidores no proporcionan ningún tipo de estadística.

Por otro lado, es posible usar un software de administración de redes con repetidores con buffer, puentes (bridges) y ruteadores (gateways). Los ruteadores, en la mayoría de los casos, incluyen un avanzado software de administración de redes. La mayoría de los ruteadores pueden manejar IP y los protocolos de monitorización anteriormente mencionados. Puesto que los puentes no están

dirigidos a ningún protocolo en particular, la mayoría de ellos no tienen el software necesario para implementar los protocolos TCP/IP de administración de redes.

SACIP mantiene un equilibrio entre la interfase gráfica y la funcionalidad de análisis de cualquier red mediante la utilización de una aplicación de auditoria nativa multiplataforma llamada Nmap que se describe en la sección I.II.X, la utilización de esta herramienta separa la parte de la implementación de su acceso remoto por internet para poder enfocarse en su objetivo que es el análisis de los datos de resultado.

I.II.IX. Sniffers.

El termino sniffer o sniffing se refiere al vocablo en inglés que significa rastrear para investigar claro que para nuestro caso seria dando un sentido dirigido a las computadoras y las redes a nivel de enlace, de este modo estas aplicaciones son usadas por los administradores de una red para monitorear y validar el trafico de la red pero no leer toda la información que circule por el tramo de red en el que se encuentre a través de varios niveles de la capa OSI.

Así como la mayoría de las herramientas de seguridad los sniffers también pueden ser usados para hacer buenas cosas o para propósitos destructivos.

En el lado de la luz de la administración de la red, los sniffers ayudan rápidamente a encontrar problemas tales como cuellos de botella y filtros maliciosos, y hay que recordar que Ethernet fue diseñado para que las computadoras compartieran la misma línea.

En el lado oscuro, los sniffers pueden ser usados para generar gran cantidad de caos con solo reunir y capturar claves de acceso, datos que se transmiten, números de secuencia de usuarios legítimos y así las otras computadoras pueden ser rápidamente alteradas. Los usos típicos de tales intervenciones con sniffers incluyen:

- Recuperación de texto-en-claro de contraseñas y nombre de usuarios de la red. Usado por hackers/crackers para irrumpir en los sistemas.
- Conversión de datos para su comprensión humana así las personas pueden observar el tráfico.
- Análisis de fallos para descubrir problemas en la red, tales como cuando una computadora A no puede comunicarse con la computadora B.
- Análisis de funcionamiento para descubrir cuellos de botella en la red.
- Detección de intrusión en la red para descubrir hackers/crackers.
- Registro del tráfico de la red, para crear un registro o logs que los hackers no puedan irrumpir o borrar.
- Buscar equipos atacantes o que no pertenecen a la red.

En teoría, es imposible detectar sniffers ya que funcionan de forma pasiva: estos solo recolectan paquetes, y no transmiten nada. Sin embargo, en la práctica estos pueden ser detectados. Es similar a que en teoría es imposible detectar señales de radio/TV, pero en los países Europeos se realiza todo el tiempo para atrapar gente que evita el pago de impuestos de radio/TV.

Un sniffer independiente no transmite paquetes, pero cuando se instala uno no-dependiente en una computadora normal, a menudo el sniffer genera tráfico. Por ejemplo, este puede mandar búsquedas de DNS reversos para encontrar nombres asociados con direcciones IP. Los sniffers no-independiente son los que se quieren detectar. Cuando crackers/hackers invaden computadoras, a menudo instalan sniffers y programas de análisis. De forma predeterminada, las tarjetas de red escuchan y responden solamente a los paquetes que van dirigidos a ellas, ya que el sistema operativo así lo define por default, pero es posible establecer la interfaz de red en un modo de “escucha” de la red por el tramo en el cual este conectada la computadora, este modo es llamado promiscuo.

Por otra parte siempre van a existir métodos para atravesar barreras por muy complicadas que sean, estos ataques son ejemplo de la pericia y destreza que puede llegar a superar un sniffer y sus creadores de estos; algunos ejemplos de ataques realizados en switches son ataques de envenenamiento de caché ARP y los más usados son: ARP spoofing[4], MAC flooding[5], MAC duplicating[6]; existen otros tipos de spoofing dependiendo de la tecnología a la que nos refiramos, como el IP spoofing (quizás el más conocido), DNS spoofing, Web spoofing o e-mail spoofing, aunque en general se puede englobar dentro de spoofing cualquier tecnología de red susceptible de sufrir suplantaciones de identidad, aquí se explica la forma de cómo trabajan estos métodos y otros para tener una solución de como evitar los ataques en menor medida y utilizarlos para administrar correctamente la red. La siguiente explicación de ninguna manera le enseña a una persona a escribir una aplicación de sniffing como tcpdump o ethercap, mas bien es para hacer del conocimiento de quien administra una red, las vulnerabilidades de la misma, la prevención y la solución a probables ataques, además del buen uso de sniffers para administrar correctamente la red.

I.II.X. NMAP.

Nmap ("network mapper" o “mapeador de redes”) es una utilidad libre de código abierto usada para auditoria de seguridad y descubrir vulnerabilidades de seguridad dentro de la red. Se diseñó para analizar rápidamente grandes redes, aunque funciona muy bien en equipos individuales. Nmap utiliza paquetes IP en bruto o paquetes raw IP en diferentes formas para determinar qué equipos se encuentran disponibles en una red o activos, qué servicios (nombre y versión de la aplicación) ofrecen, qué sistemas operativos (y sus versiones)

[4] **Spoofing:** En términos de seguridad informática hace referencia al uso de técnicas de suplantación de identidad.

[5] **Flooding:** Significa que el sniffer hace “inundaciones” MAC a un switch en su Tabla ARP es decir lo llena con direcciones MAC y este queda muy ocupado para poder manejar la localización de cada paquete y actúa como hub, de esta forma es mas fácil redireccionar paquetes y falsificar identidades, aparenta que la máquina espía en realidad tiene todas las MAC de la red. Esto creará una duplicación en las entradas del switch y todo el tráfico acabará en su destino original y en el espía para lograr un “Man in the Middle” o hombre en el medio pasando por un switch.

[6] **MAC Duplicating/cloning*:*** Dado que una MAC debe ser única, existen métodos para clonar o duplicar una dirección física, con el objetivo de que la información que va destinada a una computadora llegue a la computadora atacante causando un DoS (Denial of Service o negación de Servicio) a la victima.

ejecutan, qué tipo de filtros de paquetes o cortafuegos se están utilizando así como docenas de otras características. Muchos administradores de redes y sistemas lo encuentran útil para realizar tareas rutinarias, como puede ser el inventariado de la red, la planificación de actualización de servicios y la monitorización del tiempo que los equipos o servicios se mantiene activos. Nmap es software libre, disponible con código fuente bajo los términos de GNU GPL. Nmap muestra dependiendo de las opciones dadas con una tabla el número de puerto y protocolo, el nombre más común del servicio, su estado, un listado de sistemas operativos posibles que utiliza, secuenciabilidad TCP, nombres de los programas ejecutando de cada puerto, el nombre DNS y algunas otras cosas.

La información puede ser usada para identificar y corregir hoyos de seguridad. La herramienta puede ser usada para escanear redes y desarrollar reconocimientos de los tipos y cantidad de objetivos disponibles y debilidades existentes.

Nmap esta disponible para un amplio rango de plataformas de sistemas operativos. La aplicación estándar viene comprimido en un archivo que contiene la versión UNIX (El cual corre sobre Linux, Solaris, Free/Net/OpenBSD, Mac OS X y la versión de Windows). Nmap soporta muchas técnicas de análisis, como por ejemplo:

- TCP connect () – Este es el método utilizado por defecto.
- Análisis UDP.
- TCP SYN (medio abierto).
- FTP Bounce (ataque de rebote).
- Ident-reverso.
- Detección de hosts no funcionando a través de pings paralelos.
- ICMP (barrido de ping – ECHO_REQUEST).
- FIN.
- Análisis ACK.
- Xmas Tree (literalmente “árbol de navidad”).
- Análisis SYN.
- Null Scan (Análisis Nulo).

- Análisis de protocolo IP.
- Análisis Window.
- Análisis List.
- Detección del Sistema Operativo del host siendo escaneado por huellas TCP/IP.
- Análisis Stealth (indetectable).
- Cálculo de esperas y retransmisión dinámicas.
- Análisis por 'decoys' (hosts intermedios).
- Detección de puertos siendo filtrados (por el host final o uno intermedio).
- Análisis RPC directo (no por mapeador de puertos).
- Análisis de fragmentación.
- Flexible especificación de objetivo[s] y puerto[s]. [7]

El estado de un puerto puede ser open (abierto), filtered (filtrado), closed (cerrado), o unfiltered (no filtrado). Abierto significa que la aplicación en la máquina destino se encuentra esperando conexiones o paquetes en ese puerto. Filtrado indica que un firewall o cortafuegos, filtro, u otro obstáculo en la red está bloqueando el acceso a ese puerto, por lo que Nmap no puede saber si se encuentra abierto o cerrado. Los puertos cerrados no tienen ninguna aplicación escuchando en los mismos, aunque podrían abrirse en cualquier momento. Los clasificados como no filtrados son aquellos que responden a los sondeos de Nmap, pero para los que Nmap no puede determinar si se encuentran abiertos o cerrados. Nmap informa de las combinaciones de estado open **|** filtered (abierto **|** filtrado) y closed **|** filtered (cerrado **|** filtrado) cuando no puede determinar en cual de los dos estados está un puerto. La tabla de puertos también puede incluir detalles de la versión de la aplicación cuando se ha solicitado detección de versiones.

Nmap puede generar la salida en cinco formatos distintos. El formato por omisión es el llamado salida interactiva, y se envía a la salida estándar («stdout»). También está la salida normal, que es similar a la salida interactiva salvo que muestra menos información de ejecución y menos advertencias, ya que se es-

[7] La palabra 'Análisis' se usa en este trabajo de tesis para reemplazar/traducir el término en Ingles "scan" o "escanear".

pera que se analice una vez que el sondeo haya terminado en lugar de ser analizada interactivamente.

La salida XML (Extensible Markup Language o Lenguaje de Marcado Extensible) es uno de los formatos de salida más importantes, ya que puede convertirse a HTML, los programas (como la interfaz de usuario de Nmap) pueden interpretarla fácilmente o puede importarse a una base de datos.

La opción que ofrece N*map* que da salida a un archivo XML es descrita aquí para razones de implementación del parser de los resultados que arroja el análisis de la red en SACIP:

-oX <Nombre del Archivo de Salida > (salida XML)

> Solicita que la salida en XML se redirigida al archivo especificado. Nmap incluye un DTD[8] que pueden utilizar los intérpretes de XML para validar la salida XML. Aunque está dirigida a que la utilicen programas, también puede ayudar a que una persona interprete la salida de Nmap. El DTD define los elementos legales del formato, y generalmente enumera los atributos y valores que pueden tener. La última versión está siempre disponible en http://www.insecure.org/nmap/data/nmap.dtd.
>
> XML ofrece un formato estable que es fácilmente interpretado por cualquier programa. Hay intérpretes libres de XML para los lenguajes de ordenador más importantes, incluyendo C/C++, Perl, Python, y Java. La gente ha escrito bibliotecas para la mayoría de estos lenguajes que manejan específicamente la salida de Nmap. Por ejemplo Nmap::Scanner y Nmap::Parser en el CPAN de Perl y SAX para java. XML es el formato preferente en la mayoría de los casos en que una aplicación no trivial quiere utilizar Nmap.
>
> La salida de XML hace referencia a la hoja de estilo XSL que puede utilizarse para formatear los resultados en HTML. La forma más fácil de utilizarla es simplemente cargar la salida XML en un navegador web como

[8] **DTD** Siglas en inglés de Document Type Definition. La definición de tipo de documento (DTD) es una descripción de estructura y sintaxis de un documento XML o SGML.

> Firefox o IE. Por omisión, esto solo funcionará en el equipo en el que ejecutó Nmap (o uno configurado igual que dicho equipo) ya que la ruta de nmap.xsl se incluye directamente dentro del archivo. Puede utilizar la opción --webxml o --stylesheet para crear un XML portable que pueda mostrarse como HTML en cualquier ordenador conectado a la web. [9]

Uno de los pasos importantes para la obtención de la información la devuelve Nmap con una salida XML, este archivo de salida es especificado por el administrador del sistema (SACIP) para después poderlo analizar con un "*parseador*" y devolver la información necesaria para la base de datos que mostrará los datos actualizados.

Nmap versión 4.22 para Windows basado en bibliotecas de open source WinPcap 4., y Nmap versión 4.22 para Linux basado en librerías nativas de captura de paquete de c (Libpcap), la librería de redes de bajo nivel (Libdnet) y la librería de expresiones regulares compatibles con Perl (PCRE).

LibPCap es una interfase de sistema-independiente para captura de paquetes a nivel usuario. LibPCap provee una plataforma portable para monitoreo de red de bajo nivel.

WinPCap es una herramienta para acceder a la capa de conexión de la red en ambientes Windows. Este permite a aplicaciones capturar y transmitir paquetes de red "entunelando" la pila del protocolo, y tiene una característica de uso adicional. WinPcap es usado por el paquete de clases Nmap para analizar paquetes en sistemas operativos Windows.

Nmap es suficiente para obtener todos los datos relacionados con las computadoras de una red, se le puede pedir específicamente que información hace falta de la computadora en una red, como es la MACAddress, y su relación con el IP dentro de la red, para esto no se necesita estar monitoreando toda la red sino ejecutar un archivo de Java que devuelve la relación MAC Address - direc-

[9] *Referencia a las opciones de salida de nmap* - http://www.insecure.org/nmap/man/es/man-output.html

ción IP comparándose dentro de la base de datos y su estado actual con el objetivo de mantener la seguridad de lógica entre tales direcciones y su ubicación.

I.III. Modelo y Aplicaciones Cliente-Servidor (tecnología JAVA servlets, JSPs, XML y la Seguridad SSL).

Existen diversos puntos de vista sobre la manera en que debería efectuarse el procesamiento de datos, aunque la mayoría que opina, coincide en que nos encontramos en medio de un proceso de evolución que se prolongará todavía por algunos años y que cambiará la forma en que obtenemos y utilizamos la información almacenada electrónicamente.

El principal motivo detrás de esta evolución es la necesidad que tienen las organizaciones (empresas o instituciones públicas o privadas), de realizar sus operaciones más ágil y eficientemente, debido a la creciente presión competitiva a la que están sometidas, lo cual se traduce en la necesidad de que su personal sea mas productivo, que se reduzcan los costos y gastos de operación, al mismo tiempo que se generan productos y servicios más rápidamente y con mejor calidad.

El modelo Cliente-Servidor reúne las características necesarias para proveer esta infraestructura, independientemente del tamaño y complejidad de las operaciones de las organizaciones públicas o privadas, consecuentemente desempeña un papel importante en este proceso de evolución.

Esto se traduce a que desde una computadora local el usuario establece conexiones con otras computadoras, denominados remotas, a los que solicita algún servicio. Estas computadoras remotas que ofrecen servicios reciben también el nombre de servidores o hosts.

Por un lado, el usuario, quien ejecuta una aplicación en el ordenador local: el denominado programa cliente. Este programa cliente se encarga de ponerse en contacto con el ordenador remoto para solicitar el servicio deseado. El ordenador remoto por su parte responderá a lo solicitado por el programa cliente mediante otro programa, denominado programa servidor. Los términos cliente y

servidor se utilizan tanto para referirse a los programas que cumplen estas funciones, como a los ordenadores donde son ejecutados esos programas.

El programa o los programas cliente que el usuario utiliza para acceder a los servicios de Internet realizan dos funciones distintas. Por una parte, se encargan de gestionar la comunicación con el ordenador servidor, de solicitar un servicio concreto y de recibir los datos enviados por éste; y por otra, es la herramienta que presenta al usuario los datos en pantalla y que le ofrece los comandos necesarios para utilizar las prestaciones que ofrece el servidor.

Los usuarios invocan la parte cliente de la aplicación, que construye una solicitud para ese servicio y se la envía al servidor de la aplicación que usa TCP/IP como transporte.

El servidor es un programa que recibe una solicitud, realiza el servicio requerido y devuelve los resultados en forma de una respuesta. Generalmente un servidor puede tratar múltiples peticiones (múltiples clientes) al mismo tiempo.

Algunos servidores esperan las solicitudes en puertos bien conocidos de modo que sus clientes saben donde deben dirigir sus peticiones. El cliente emplea un puerto arbitrario para comunicarse. Los clientes que se quieren comunicar con un servidor que no usa un puerto bien conocido tienen otro mecanismo para saber a qué puerto dirigirse.

El diálogo cliente/servidor es casi siempre bidireccional. Por un lado, el cliente envía información al servidor (el tipo de servicio solicitado mas los parámetros); por otro, el servidor devuelve información al cliente (los resultados del servicio, códigos de error en caso de producirse, etc.)

También es importante hacer notar que las funciones Cliente/Servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red. Su capacidad para permitir integrar los equipos ya existentes en una organización, dentro de una arquitectura informática descentralizada y heterogénea. Designa un modelo de construcción de sistemas informáticos de carácter distribuido.

Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización, al tiempo que puede acceder a los recursos de este host central y otros sistemas de la organización ponen a su servicio.

El modelo Cliente-Servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias, funcionando todos al mismo tiempo. Por lo tanto, su implantación involucra diferentes tipos de estándares: APPC, TCP/IP, OSI, NFS, DRDA corriendo sobre DOS, OS/2, Windows o PC UNIX, Linux, Macintosh, en Token Ring, Ethernet, FDDI, sólo por mencionar algunas de las posibilidades.

Servidores Web, éstos se usan como una forma inteligente para comunicación entre empresas a través de Internet. Este servidor permite transacciones con el acondicionamiento de un navegador específico.

I.III.I. Las aplicaciones Cliente-Servidor.

En un entorno cliente-servidor, cada servidor ofrece una serie de servicios de usuario compartidos a los clientes. El tipo más común de servidor es el servidor de base de datos que permite el acceso a los clientes y el uso de un sistema de computación para gestionar la base de datos.

Tanto en el cliente como en el servidor el software básico es un sistema operativo que se ejecuta en la plataforma del hardware. Las plataformas y los sistemas operativos del cliente y el servidor pueden ser diferentes. En tanto un cliente particular y un servidor compartan los mismos protocolos de comunicación, y soporten las mismas aplicaciones.

El software de comunicaciones es el que permite interoperar con el cliente y el servidor. El ejemplo principal es el TCP/IP. El objeto de todo este software de soporte (comunicaciones y sistema operativo) es proporcionar una base para las aplicaciones distribuidas.

En la mayoría de los sistemas cliente-servidor, se da prioridad para ofrecer una interfaz de usuario gráfico que sea fácil de utilizar y de aprender, pero potente y flexible. Así pues, se puede pensar en un módulo de servicios de presentación en el puesto de trabajo del cliente, responsable de ofrecer una interfaz fácil de usar a las aplicaciones distribuidas disponibles en el entorno.

En tanto que un cliente particular y un servidor compartan los mismos protocolos de comunicación y soporten las mismas aplicaciones las diferencias entre plataformas y sistemas operativos no son relevantes. En la figura I.III.I.I se muestra el flujo de la información entre el cliente y el servidor a nivel lógico:

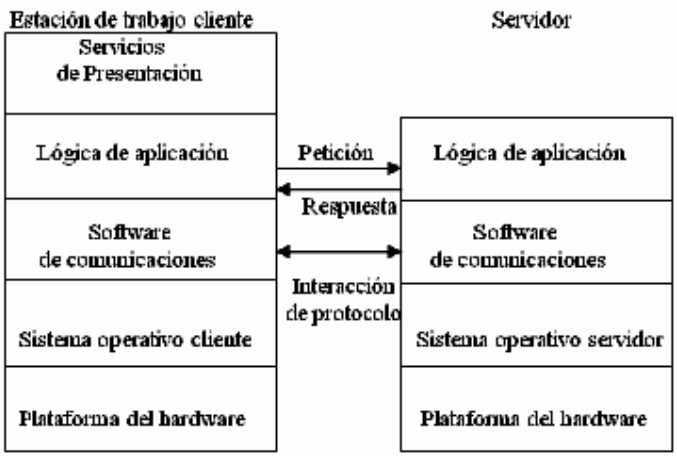


Figura I.III.I.I. Flujo de información entre el Servidor y el cliente.

I.III.II. Apache Tomcat.

Los proyectos de Tomcat empezaron en Sun Microsystems como la referencia de la implementación de las especificaciones de Java Servlet y Java Server Pages. El sitio de la Comunidad de Procesos de Java o Java Community Process (JCP) site[10] contiene todos los detalles de estas especificaciones. El código base de Tomcat fue donado por Sun a la fundación de software Apache en 1999 cuando desarrollaban un "Servidor de Páginas" o "Web Container" que utilizara "Java", y la primera versión de Apache que fue liberada fue la versión 3.0. Desde entonces, múltiples voluntarios de Sun y otras numerosas organizaciones han contribuido al producto. Ha habido múltiples liberaciones mejoradas

[10] **Java Community Process (JCP) site** - *http://jcp.org/en/home/*

y el producto ha disfrutado de adopciones considerables en la industria así como también en la comunidad activa, este es conocido como "Servlet Engines". A pesar del nombre Apache-Tomcat; Tomcat no requiere de Apache para su funcionamiento (solo requiere de un JDK ("Java Development Kit"), es open-source. En el 2005, Tomcat llegó a tener su propio nivel superior en el proyecto de Apache, dejando Jakarta como su asociado y por haber tenido un descuido en la estructura para manejarlo por si mismo.

Apache Tomcat es un contenedor que es usado en la referencia de la implementación oficial para las tecnologías de Java Servlet y JavaServer Pages. Las especificaciones de Java Servlet y JavaServer Pages son desarrollados por Sun bajo la comunidad de procesos de Java. Este también es desarrollado en un ambiente abierto y participativo y se libera bajo la Licencia de Apache (véase Anexo de Licencias). Apache Tomcat intenta ser una colaboración de desarrolladores de lo mejor de su clase de todo el mundo. Se invita libremente a participar en el proyecto para desarrollar. También le da poder a numerosas aplicaciones web de gran-escala, de misión-critica web a través de diversos planos de la industria y organizaciones. Las especificaciones de este contenedor y su correspondencia con las versiones que se han liberado por Apache Tomcat se ilustran en la figura I.III.II.I. hasta la fecha (07 de noviembre de 2007) y con la que se implementó el sistema, siendo este el que lo contiene y da servicio de forma cliente-servidor, las versiones hasta la fecha que han salido de servlet/JSP se muestran en la Tabla I.III.II.I.:

Especificaron de Servlet/JSP	Versión de Apache Tomcat
2.5/2.1	6.0.14
2.4/2.0	5.5.25
2.3/1.2	4.1.31
2.2/1.1	3.3.2

Tabla I.III.II.I. Versionado de apache-tomcat.

El "Servlet Engine" ofrece un "Ambiente" donde habitan los JSP y Servlets, es ahí donde se contemplan una gran cantidad de funcionalidades como: threa-

ding o hilación, manutención de sesiones, conectividad con el "Servidor de Páginas", es por esto al "Servlet Engine" también se le denomina "Web-Container".

Como se observa en el diagrama también se requiere de un JDK ("Java Development Kit") , el cual llevará acabo la ejecución de los programas ("Servlets" y "JSP's") en Java; como toda otra implementación existen diversas versiones de JDK's, esto se debe a que cada JDK debe ser diseñado alrededor de un Sistema Operativo (para ser más exactos es el JVM "Java Virtual Machine" el que debe ser diseñado alrededor del Sistema Operativo), algunos JDK's son: J2SE's (Java 2 Standard Edition) de Sun y JDK's de IBM.

En el sistema implementado (SACIP) se utiliza este contenedor de servlets y JSPs ya que todo el sistema esta basado en estas especificaciones además que la parte del servidor y el cliente están implementados de esta forma para interactuar remotamente entre estos con la única diferencia que los procesos para obtener la información de la computadoras se manda a llamar un programa nativo que en este caso es Nmap, y con el procedimiento que se explicará mas adelante como la “obtención de información”.

I.III.III. Log4j.

Insertar líneas de registros en el código es un método de baja tecnología para hacer un debug u optimizado del código fuente. Este podría también ser la única forma porque los debuggers no siempre están disponibles o son aplicables.

Por otro lado, hay personas que están de acuerdo que las líneas de código que registran o logs llenan de basura el código fuente y decrementan legibilidad. En el lenguaje Java donde un preprocesador no esta disponible, las sentencias de registro o logs incrementan el tamaño del código y reduce su velocidad, incluso cuando el logging esta apagado. Dando eso un tamaño razonablemente a la aplicación podría contener miles de sentencias de registro, la velocidad es de particular importancia.

Con log4j es posible activar en tiempo de ejecución el registro o logging sin modificar la aplicación. El paquete log4j esta diseñado para que estas sentencias puedan permanecer en el código transportado sin incurrir en costos de rendimiento pesado. El comportamiento del registrador o logging puede ser controlado editando un archivo de configuración, sin tocar la aplicación.

Los registros o Logging equipa al desarrollador con contextos detallados de los fallos de las aplicaciones. Por otro lado, las pruebas proveen el aseguramiento de la calidad y confianza en la aplicación. Los registros y pruebas no deben confundirse aunque ambos son complementarios. Cuando se hacen registros es usado sabiamente, este puede probar ser una herramienta esencial.

Un rasgo distintivo de log4j es la noción de herencia en registradores o loggers. Usando la herencia de un logger es posible controlar que sentencias mandar a una salida arbitraria con una fina granularidad pero con gran facilidad. Esto ayuda a reducir el volumen de registros en la salida y minimiza el costo de hacer registros.

Este paquete puede apuntar a un objetivo de salida como puede ser un archive, un flujo del tipo *OutputStream* en java, de otra forma en un *java.io.Writer*, un servidor remoto log4j, un demonio remoto del tipo Unix Syslog, a un correo electrónico o muchos otros objetivos de salida.

En el sistema implementado (SACIP) se usa este logger para mantener informado del estado del sistema y el acceso al mismo. Esta información es importante para seguir los movimientos de un posible atacante, sus huellas que deja sobre el sistema y la forma como se realizo.

I.III.IV. XML.

Por las siglas en inglés de Extensible Markup Language o Lenguaje de Marcado Extensible, este un lenguaje que usa texto basado en etiquetas es más que html ya que este último solo maneja la presentación y no contiene información acerca de los datos mismos. HTML tiene un conjunto de Etiquetas, pero XML

puede ser extendido, es decir, se pueden crear o personalizar etiquetas propias.

Está basado en el anterior estándar SGML (Standard Generalized Markup Language, ISO 8879), que data de 1986, pero que empezó a gestarse desde principios de los años 70, y a su vez basado en el GML creado por IBM en 1969. Esto significa que aunque XML pueda parecer moderno, sus conceptos están más que asentados y aceptados de forma amplia. Está además asociado a la recomendación del W3C DOM (Document Object Model), aprobado también en 1998.

Este no es más que un modelo de objetos (en forma de API) que permite acceder a las diferentes partes que pueden componer un documento XML o HTML. SGML proporciona un modo consistente y preciso de aplicar etiquetas para describir las partes que componen un documento, permitiendo además el intercambio de documentos entre diferentes plataformas. el problema que se atribuye a SGML es su excesiva dificultad. Así que, manteniendo su misma filosofía, de él se derivó XML como subconjunto simplificado, eliminando las partes más engorrosas y menos útiles.

XML es un metalenguaje: es un lenguaje para definir lenguajes. Los elementos que lo componen pueden dar información sobre lo que contienen, no necesariamente sobre su estructura física o presentación, como ocurre en HTML. Usando SGML, por otro lado, se definió precisamente el HTML. En una primera aproximación se puede decir que mediante XML también podríamos definir el HTML, con lo que podríamos considerar los conjuntos de la figura I.III.IV.I.

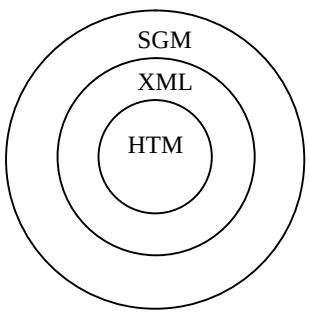


Figura I.III.IV.I. Jerarquías de tamaño de los lenguajes extendidos.

De hecho, HTML es simplemente un lenguaje, mientras que XML como se ha dicho es un metalenguaje, esto es, un lenguaje para definir lenguajes. Y esa es la diferencia fundamental, de la que derivan todas las demás.

Las ventajas de este lenguaje son:

- *Legibilidad*, ya que los documentos son en texto plano y puede ser leídos por el humano; para editar/ver documentos XML es suficiente usar cualquier simple editor de textos.
- *Jerárquico*, los documentos XML tienen una estructura de árbol el cual es suficiente poderoso para expresar datos complejos y lo suficientemente simple para entender.
- *Independiente del lenguaje*, los documentos XML son lenguajes neutrales, esto es, si se escribe un programa en Java que puede crear o leer un archivo XML puede ser “parseado” por un programa escrito en C++ o Perl.
- *Independiente del sistema operativo,* esto es posible gracias a que no existe dependencias de paquetes o la plataforma en que se lea un archivo XML, solamente se debe tener un parseador que pueda entender este tipo de archivos.

Los usos que se le pueden dar a XML pueden ser los siguientes:

- *Metacontenido*, para describir los contenidos de un documento.
- *Mensajería*, donde las aplicaciones u organizaciones intercambian datos entre ellos.
- *Base de datos*, los datos extraídos de una base de datos pueden ser preservados con la información original y pueden ser usados más de una aplicación en diferentes formas. Una aplicación podría solo mostrar los datos y otra aplicación podría desarrollar algún cálculo complejo sobre los datos.

Respecto a su aplicación en Internet, mejorará lo que HTML ha intentado y es establecer un estándar fijo al que todo el mundo pueda atenerse sin necesidad

de depender del Navegador utilizado, del sistema de objetos o de cualquier otra cosa.

Se dejará de lado bastantes aspectos relacionados con la manipulación de documentos XML desde Java, como pueden ser los distintos tipos de parsers y funciones avanzadas, como la validación. Asimismo, se supone un cierto conocimiento de XML, su estructura y su función, ya que no es el objetivo principal de esta tesis.

Uno de los analizadores de documentos XML para Java es XERCES que proporciona documentos XML estándar con el análisis y la generación. Los analizadores (que son válidos) están disponibles para muchos lenguajes de programación (Java, C++, etc.), aplicando los estándares del consorcio W3C como son XML y DOM (nivel 1 y 2), así como la especificación estándar SAX (versión 2).

Parser DOM (Modelo de Objetos del Documento o Document Object Model) este está basado en el API de estructura de árbol, este parser implementa el API DOM y crea un árbol DOM de memoria para un documento XML.

Parser SAX (API simple para XML o Simple API For XML) – API basado en eventos, el parser SAX implementa el API SAX y este es una interface conducido por eventos. Conforme va parseando, este invoca los métodos de retrollamada o callback.

Para saber cuando usar un parser u otro se deben tener en cuenta estos puntos:

- El parser DOM, manipula el documento, recorrer el documento a lo largo de toda la estructura, funciona de forma óptima para archivos XML pequeños. Las desventajas de este parser es que consume mucha memoria.
- El parser SAX, no realiza modificaciones estructurales, sirve para analizar enormes archivos XML.

Validación y No Validación

DOM y SAX pueden ser parser validadores o no validadores. Un parser validador checa el archivo XML en busca de las reglas impuestas por el DTD o el esquema XML. Un parser no validador no valida el archivo en busca de un DTD o un esquema XML. Ambas validaciones de los parsers checan que los documentos XML estén “bien formados”.

I.III.V. Servlets 2.4.

Una aplicación cliente-servidor basada en la tecnología de Java Servlets y JSPs, además de otras tecnologías para las aplicaciones de negocios en donde se muestra el flujo de la información y como están estructuradas estas capas en la figura I.III.V.I.

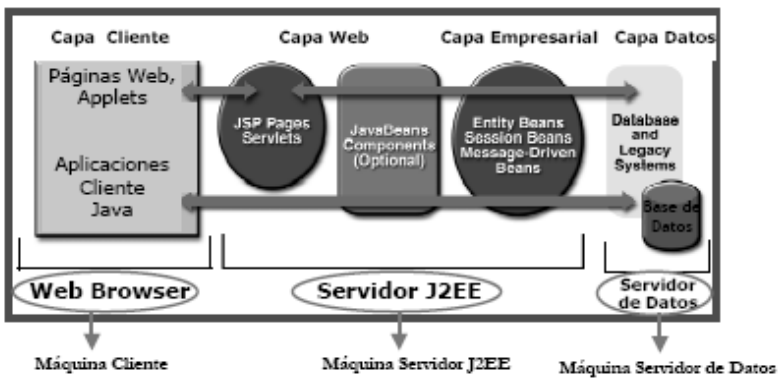


Figura I.III.V.I. Diagrama cliente-servidor de la tecnología java servlets y JSPs.

En la figura I.III.V.I. se observa claramente la parte de aplicación de negocios y la separación evidente del cliente-servidor, la parte de la capa del cliente manipula y ejecuta aplicaciones Java como son Applets por medio del navegador web de su equipo, después este se comunica con la siguiente capa que es la capa web donde esta el servidor de aplicaciones y contiene los JSPs y los servlets que reciben y dan respuesta a las peticiones estos pueden usar la capa empresarial y lógicamente si se trata de una aplicación que usa base de datos existe una capa dedicada a realizar este servicio y proporcionar cualquier información que se necesite almacenada.

Específicamente un servlet es un programa del lado del servidor escrito en el lenguaje Java que interactúa con clientes y que normalmente está unido a un servidor de "HyperText Transfer Protocol" (HTTP). Un uso común para un servlet es ampliar un servidor Web proporcionando contenidos Web dinámicos. Los Servlets tienen la ventaja sobre otras tecnologías ya que están compilados, y tienen capacidad de threading (hilos) interna, y proporcionan un entorno de programación seguro. Incluso las páginas Web que antes no proporcionaban soporte para Servlets, con un módulo Java para el servidor Web Apache es suficiente para dar soporte, por decir un ejemplo.

La tecnología Servlet proporciona las mismas ventajas del lenguaje Java en cuanto a portabilidad (“write once, run anywhere” o “se escribe una vez, se ejecuta en donde sea”) y seguridad, ya que un servlet es una clase de Java igual que cualquier otra, y por tanto tiene en ese sentido todas las características del lenguaje. Esto es algo de lo que carecen los programas CGI, ya que hay que compilarlos para el sistema operativo del servidor y no disponen en muchos casos de técnicas de comprobación dinámica de errores en tiempo de ejecución.

Los Servlets soportan prácticas de programación segura sobre diferente número de niveles. Porque estos están escritos en Java, los servlets heredan el tipo-fuerte de seguridad de tal lenguaje. Además de esto, el API de los Servlets es implementado por ser de tipo-seguro. Los servlets pueden manejar errores con seguridad, negociando un mecanismo para manejo de excepciones de Java. Si un servlet hace una división por cero o efectúa alguna otra operación ilegal este lanza una excepción que puede ser capturada con seguridad por el servidor, el cual puede correctamente mandar a un registro o log el error “disculparse” con el usuario.

Además, los servlets se benefician de la gran capacidad de Java para ejecutar métodos en ordenadores remotos, para conectar con bases de datos, para la seguridad en la información, etc. Se podría decir que las clases estándar de Java ofrecen una solución a muchos problemas que con otros lenguajes tiene que resolver el programador.

El manejo de una petición GET o POST que se realiza desde un cliente al servidor "HyperText Transfer Protocol" (HTTP) por un servlet se describe en la figura I.III.V.II.

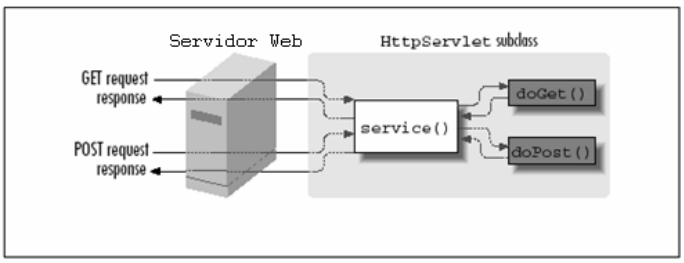


Figura I.III.V.II. Diagrama del manejo de peticiones GET y POST por medio de servlets.

Como se observa en la figura I.III.V.II. un servidor web que maneja las peticiones a través de la subclase HttpServlet de la implementación del API que ofrece Java servlet, el método de la instancia de tal clase service() es el que escucha cual de las dos peticiones se hicieron un GET o un POST, de esta forma se puede elegir la forma de recibir la información.

La función que tiene un servlet es la de recibir peticiones, absorber la información, realizar un proceso con esa información, y devolver una respuesta al cliente, cabe destacar que este tipo de peticiones es de tipo concurrente, es decir, pueden existir *n* peticiones y cada uno es manejado como un proceso aparte en el caso de Java como un hilo independiente y todos estos son despachados por una misma instancia de un servlet, esto se clarifica en la figura I.III.V.III.

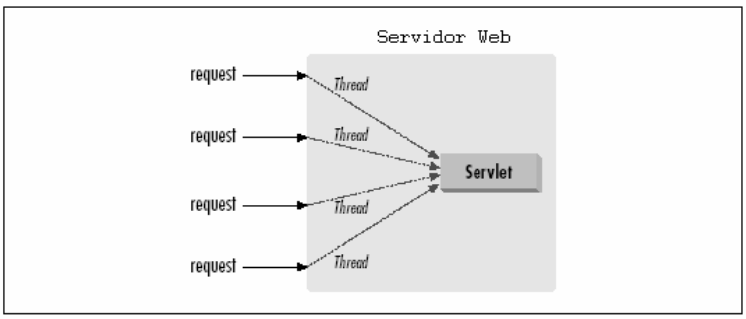


Figura I.III.V.III. Diagrama de varias peticiones a un mismo servlet, es decir un hilo para cada petición de un mismo servlet.

Cuando un servlet es cargado por primera vez, el método init() es llamado por el servidor HTTP.

Este método no será llamado nunca más mientras el servlet se esté ejecutando. Esto permite al servlet efectuar cualquier operación de inicialización potencialmente costosa en términos de CPU, ya que ésta sólo se ejecutará la primera vez. Esto es una ventaja importante frente a los programas CGI, que son cargados en memoria cada vez que hay una petición por parte del cliente. Por ejemplo, si en un día hay 500 consultas a una base de datos, mediante un CGI habría que abrir una conexión con la base de datos 500 veces, frente a una única apertura que sería necesaria con un servlet, pues dicha conexión podría quedar abierta a la espera de recibir nuevas peticiones.

I.III.VI. JSP 2.0 (Java Server Pages).

Java Server Pages (JSP) es una extensión de Java estándar que esta definido por encima de las extensiones de los servlets. El objetivo de los JSPs es simplificar la creación y administración de páginas Web dinámicas. [11]

Los JSPs permiten combinar HTML de una página Web con piezas de código de Java en el mismo documento. El código Java es encerrado por etiquetas

[11] Referencia de la implementación Tomcat de *tomcat.apache.org* que automáticamente soporta JSPs.

especiales que le dice al contenedor de JSPs que este debe usar el código para generar un servlet, o parte de uno. El beneficio de los JSPs es que se puede mantener un único documento que representa ambos la página y el código Java que lo hace posible.

La primera vez que un JSP es cargado por el contenedor de JSP (el cual es típicamente asociado con, o incluso parte de, un servidor Web), el código servlet necesario para rellenar las etiquetas JSP es automáticamente generado, compilado, y cargado dentro del contenedor de servlet. Las partes de la página HTML son producidas mandando cadenas estáticas de tipo cadena u objeto String para hacer un la escritura o write(). Las partes dinámicas son incluidas directamente dentro del servlet.

Desde ese momento, tan pronto la fuente de JSP para la página no es modificado, este se comporta como si este fuera una página estática HTML con servlets asociados (todo el código HTML es realmente generado por el servlet, sin embargo). Si se modifica el código para un JSP, este es automáticamente recompilado y recargado para la siguiente vez que la pagina requerida. Por supuesto, porque todo este dinamismo tendrá una respuesta lenta la primera vez que se accese a un JSP.

La estructura de una página JSP es una combinación entre un servlet y una página HTML. Las etiquetas JSP empiezan y terminan como las etiquetas de HTML, pero estas etiquetas también incluyen signos de porcentaje, así todas las etiquetas JSP son denotadas por:

<% JSP + código Java %>

El primer signo de porcentaje puede ir seguido por otro carácter que determina el tipo preciso de código JSP en la etiqueta, Cada bloque de código servlet es llamado *scriptlet*.

Cuando el cliente crea una petición para la página JSP, el servidor Web debe estar configurado para transportar la petición al contenedor de JSPs, es donde entonces invoca la página. La primera vez que la página es invocada, los com-

ponentes son especificados por la página de las que son generadas y compiladas por el contenedor de JSPs como uno o más servlets. Subsecuentes peticiones incluso podrían ser más rápidas porque el servidor puede reutilizar el servlet creado. La única excepción es cuando el archivo jsp cambia, en tal caso el servidor notifica y recompila un nuevo servlet en segundo plano. Si ocurriera un error en la compilación, se espera que el servidor de alguna forma reporte el problema, usualmente en la misma página retornada al cliente. En el ejemplo anterior, el servlet contendrá el código para configurar el objeto *HttpServletResponse*, produciendo otro objeto *PrintWriter* (el cual casi siempre es llamado out o salida), y entonces cambia el cálculo de tiempo a un objeto de tipo String o cadena el cual es mandado a out o salida. Como se puede ver, todos esto se cumple con un muy concisa declaración de una expresión, el cual un diseñador-programador-Web HTML promedio no tendrá las habilidades para escribir tal código. El rango de cada objeto puede variar significativamente. Por ejemplo, el objeto *application* puede proveer servicios para un grupo de páginas JSP que juntas representan una aplicación Web.

I.III.VII. JavaBeans.

Los JavaBeans no son más que objetos java que siguen cierto patrón bien definido: el bean encapsula sus propiedades declarándolas privadas y provee métodos de acceso públicos (getter/setter) para leer y modificar los valores de estas propiedades.

Antes de poder acceder a un bean dentro de una página JSP, es necesario identificarlo y crear una referencia al mismo, lo cual se hace usando la etiqueta <jsp:useBean ...>.

Para obtener el valor de una propiedad de un bean se utiliza la etiqueta:

<jsp:getProperty name="user" property="name" />

y para modificarla:

<jsp:setProperty name="user" property="name" value="<%=expression %>" />

Aquí tenemos un ejemplo de página JSP que hace uso de un JavaBean para mostrar el contenido de un registro. Le pasamos su 'ID', y obtenemos las propiedades de dicho registro: Nombre del departamento a dar de baja. La implementación del JavaBean es muy sencilla al ser simplemente una clase que, generalmente, hace uso de JDBC para acceder a bases de datos, o que delega la responsabilidad de leer estos datos llamando a otro componente.

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="reporte.MultiUsos, java.sql.*, java.io.*" errorPage="error.jsp"%>
<jsp:useBean id="BD" class="reporte.BajaEPDE" />
<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Bajas de Departamento de SACIP</title>
<LINK REL=STYLESHEET
    HREF="../Principal.css"
    TYPE="text/css">
</head>
<BODY BACKGROUND="../images/mapback.gif">
<% if (esAdministrador == "0") {%>
<BR>
<CENTER><TABLE width="122">
<TH width="60" ALIGN="CENTER">
<INPUT TYPE="BUTTON" VALUE="Atras" onClick="history.back()">
<TH width="60" ALIGN="CENTER"><FORM NAME="MOVES" METHOD="GET" AC-
TION="/sacip/servlet/Logout">
<INPUT TYPE="SUBMIT" NAME="Logout" VALUE="Salir">
</DIV></FORM></TABLE></CENTER>
<BR>
<BR>
<H2>Bajas Principal->Baja de Departamento de SACIP</H2>
<HR ALIGN="CENTER">
<center>
<form method = "POST" name="formulario" action="BajaD2.jsp">
 <center>
  <TABLE>
   <TH ALIGN="CENTER"><H6>(*)</H6>
   <TH ALIGN="CENTER"><H5>Los campos marcados son obligatorios para poder enviar el
formulario</H5>
  </TABLE>
 </Center>
 <CENTER><PRE>Baja de Departamento</PRE></CENTER>
<div align="center">
 <Table width="550" bgcolor="#EEEEEE">
  <TR>
   <TD>
    <center>
     <PRE> Seleccione el nombre del departamento a dar de baja:
          <select name="DEP" size="6">
               <jsp:getProperty name="BD" property="departamentos"/>
          </select>
     </PRE>
    </center>
   </TD>
  </TR>
```

```
 <TR></TR>
 <TD>
    <CENTER>
      <PRE> Escriba la contraseña de administrador:  <input name="PASS" type="password"
size="40" maxlength="40"> </PRE> <TD><H6>*</H6></TD>
    </CENTER>
 </TD>
 <TR></TR>
</Table>
</div>
<INPUT type="image" name="Admin" src="../images/bot_enviar_verde.gif"
                              onClick="javascript:return validar();">
</form>
</CENTER>
<% } else {response.sendRedirect(response.encodeRedirectURL("../error.jsp"));}}%>
<%@ include file="footer.jsp" %>
</div>
</body>
</html>
```

Esto mejora el particionamiento de la aplicación, su reutilización, y su mantenimiento, así como la separación de Roles.

Para poder aumentar el rendimiento, y la potencia sin perder de vista la presentación para el uso de una página web y una aplicación cliente-servidor se pueden usar los siguientes participantes haciendo posible este proceso con una estructura de participación y construcción como es el modelo MVC (Model-View-Controller) y sus respectivos elementos son:

• **JavaBeans**: En el modelo MVC los javabeans se convierten en el modelo o “model” y es la parte que permite el acceso a la información desde y hacia la base de datos además de dar acceso seguro a los datos de objetos para modificar valores.

• **Servlet**: En una petición cualquiera dada que recopile datos necesarios para mostrar un estado dado o que invoque una acción causando una transición fuera de este estado. Esta responsabilidad lo hace el controlador o “controller” en una aplicación basada en el modelo Model-View-Controller (MVC) o Modelo-Vista-Control.

• **JavaServer Pages**: Estos manejan la generación de código HTML para un resultado de una petición, estos son la vista o “view” en el modelo MVC.

• **Página HTML**: En caso de presentar o manejar contenido estático y estados de transición, no se usan tecnologías complejas. Una página HTML manipula los estados “estáticos”.

En la Figura I.III.VII.I. se muestra la interacción y su equivalencia en el modelo MVC de cada elemento involucrado.

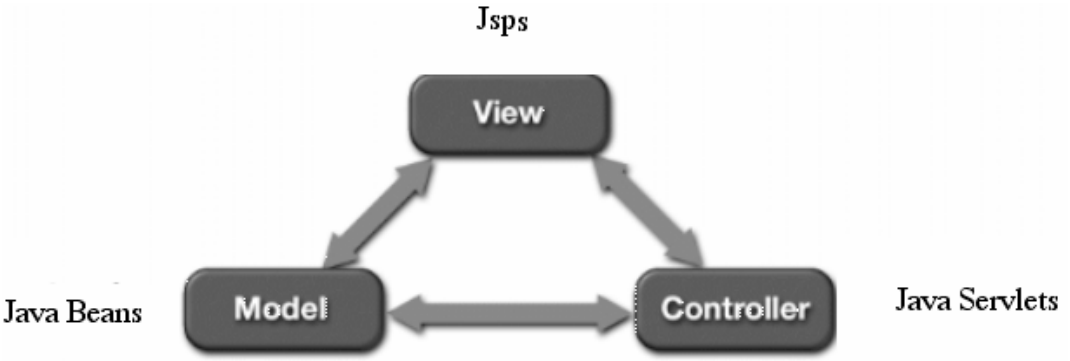


Figura I.III.VII.I Equivalencia MVC(Model-View-Controller) en java.

Los servlets http pueden generar páginas HTML, y ser accesados explícitamente por el nombre, seguido de un link o dirección de hipertexto, o como el resultado de un envío de un formulario. Un servlet HTTP puede también estar embebido o empotrado dentro de una página HTML, donde este funciona como incluido por el lado del servidor. Los servlets pueden ser encadenados para producir efectos complejos, un uso común de esta técnica es para filtrar contenidos. Finalmente, pedazos de código de un servlet puede ser directamente embebido en páginas HTML usando las llamadas JavaServer Pages (JSP) o Páginas de Servidor Java. En la figura I.III.VII.II. se muestra la interacción de las partes involucradas en el sistema:

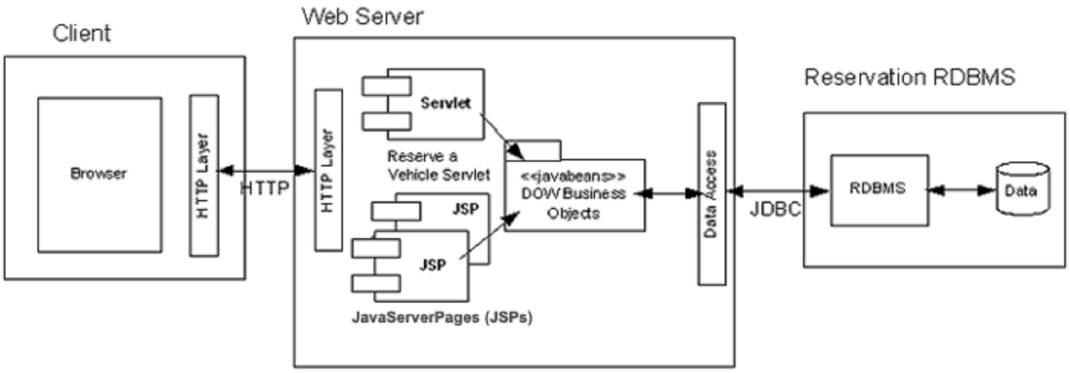


Figura I.III.VII.II Estructura de interacción entre los elementos del MVC del sistema.

Al utilizar MVC como diseño de SACIP ayudó a:

- Clarificar el diseño mediante la separación del modelado de la información con las interacciones del usuario al sistema.
- Permitir que la misma información sea vista de diversas maneras.
- Permitir que la información pueda ser vista por varios usuarios.
- Mejorar el mantenimiento del sistema debido al encapsulado de las diversas funciones.
- Mejorar la reusabilidad de las clases.
- Permitir a la aplicación a ser utilizada de una manera más amplia y más fácil.
- Facilitar la actualización del sistema debido a la partición del sistema, debido a que la información se encuentra separada.
- Facilitar las pruebas mediante la clara distribución de responsabilidades.
- Debido a que si la interfaz de la aplicación falla uno pueda saber que el problema en este caso se encuentra en los JPS dentro del modelo, y así para cada uno de los casos.

I.III.VIII. Encriptación y seguridad dentro de la Red.

La red esta estructurada utilizando Switches y Hubs que están conectados a un sistema CISCO Catalyst, y tiene una configuración lógica de estrella con una pequeña modificación, ya que los puertos del Catalyst no fueron suficientes para conectar a todos los equipos tanto en el edificio 135, 136 y actualmente el nuevo edificio se tuvieron que cascadear varios Hubs y switches. Aunque esto no afecta el rendimiento de la red, ya que el cascadeo es a partir de un Catalyst (Switch de 3ª. Capa) el cual no permite que el ruido de un puerto se propagué por los demás puertos.

En cuanto a la autentificación de los usuarios para entrar al sistema SACIP este abarca la verificación en la base de datos que con un nombre de usuario y una contraseña pueden autenticar a un usuario, una forma más de asegurar que esos datos no sean capturados tan fácil a través de una red se puede usar la autentificación que es el intercambio de certificados digitales para la encriptación de clave pública (SSL usa este método).

Es posible que una persona intercepte la comunicación y cambie sus contenidos. SSL soluciona este problema situando una capa intermedia entre los protocolos HTTP y TCP/IP que encripta los contenidos de estos paquetes. Esto proporciona la seguridad de que los paquetes son intercambiados sin corrupción. Además de utilizar una auditoria (Mantenimiento de logs durante el uso del servidor) permite al administrador hacer un seguimiento de los accesos y reconstruir actividades inusuales como los ataques a la aplicación. Esto también es una medida de seguridad ya que el servidor debe ser protegido de los intrusos y prevenir el crecimiento de este hasta provocar la caída del sistema en una máquina.

Cuando se está usando java en una máquina virtual, los problemas de seguridad son similares a cualquier otra aplicación java. Por ejemplo Tomcat permite definir pólizas de seguridad en un fichero de propiedades que proporciona cobertura a todas las aplicaciones que se ejecutan en ese servidor Tomcat.

La mejor forma de defensa es encriptar los datos así aunque los intrusos puedan capturar los datos no los podrán leer, dentro de esta defensa tenemos las técnicas de encriptación.

I.III.IX. SECURE SOCKET LAYER - Capa de conexión segura (SSL).

"Secure Sockets Layer" o Capa de sockets seguros, SSL esta construido para todos los web browsers o navegadores de red y servidores web. Este permite encriptar la navegación de la web, y es casi siempre usado en e-commerce o comercio electrónico cuando los usuarios envían su información de tarjetas de crédito o datos sensibles.

En la criptografía tradicional, (también llamada criptografía de "clave secreta") tanto el emisor como el receptor poseen una misma clave o password. El emisor utiliza esta clave para cifrar el mensaje obteniéndose el "mensaje cifrado", que es ilegible. El receptor utiliza la misma clave utilizada para el cifrado para descifrar el mensaje y obtener así el mensaje original.

Este tipo de criptografía puede utilizar distintos algoritmos para cifrar los datos. El emisor utiliza una clave para cifrar el mensaje y el receptor utiliza otra clave para descifrarlo. El receptor comunica la clave de cifrado a todo el mundo (De ahí el término "clave pública") que quiera enviarle mensajes cifrados, pero se reserva para sí la clave capaz de descifrar los mensajes. A esta clave que únicamente conoce el receptor se le denomina "clave privada" y es la única capaz de descifrar los mensajes generados por la clave pública. Este tipo de criptografía puede utilizar distintos algoritmos para cifrar los datos.

El concepto de "firma digital" se basa en la criptografía de clave pública, pero el proceso es el contrario. Quien "firma" el mensaje lo cifra mediante su "clave privada", de forma que puede ser descifrado por todo el mundo, siempre que posean la "clave pública" correspondiente a la clave privada utilizada para firmar el mensaje. Si la clave pública es capaz de descifrar un mensaje sólo puede ser por un motivo, que éste mensaje fuera cifrado por el poseedor de la clave privada, lo que autentifica su identidad (además de asegurar que el mensaje

original no ha sido alterado, de lo contrario no podría ser descifrado por la clave pública).

Una Huella digital o "fingerprint" es el proceso de cifrar un mensaje con la clave privada y su desventaja es que puede consumir un tiempo de proceso valioso.

Alguien con pretensiones desconocidas podría suministrar una clave pública haciéndose pasar por otra persona y descifrar con la clave privada los mensajes codificados con esa clave pública. Precisamente para eso, sirve un certificado, para obtener la clave pública de otra persona o entidad.

Para conseguir privacidad y autentificación se debe utilizar el cifrado de los datos con la clave pública del receptor y firmarlos con nuestra clave privada. La autenticidad se consigue mediante el uso de los certificados y firmas digitales.

La integridad se consigue combinando Criptografía, funciones hash y firmas digitales. El no repudio o la seguridad de que una vez enviado un mensaje con datos importantes o sensibles el destinatario de los mismos no pueda negar el haberlos recibido. Y en una compra on-line debe garantizarse que una vez finalizada la misma ninguna de las partes que intervienen pueda negar haber participado en ella. Es decir, el problema es estar seguro de que efectivamente la clave pública que es enviada sea de la persona correcta, y no de un suplantador. Este proceso se consigue mediante los certificados y la firma digital.

La solución al problema de falsificaciones de firmas digitales fue la aparición de los Certificados Digitales o Certificados Electrónicos, documentos electrónicos basados en la criptografía de clave pública y en el sistema de firmas digitales. La misión principal de un Certificado Digital es garantizar con toda confianza el vínculo existente entre una persona, entidad o servidor web con una pareja de claves correspondientes a un sistema criptográfico de clave pública.

Un Certificado Digital es un documento electrónico que contiene datos identificativos de una persona o entidad (empresa, servidor web, etc.) y la llave pública de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza, denominada Auto-

ridad Certificadora. El sistema de firma digital liga un documento digital con una clave de cifrado.

El proceso de una conexión SSL (*handshake*) es el que se muestra en la figura I.III.IX.I.

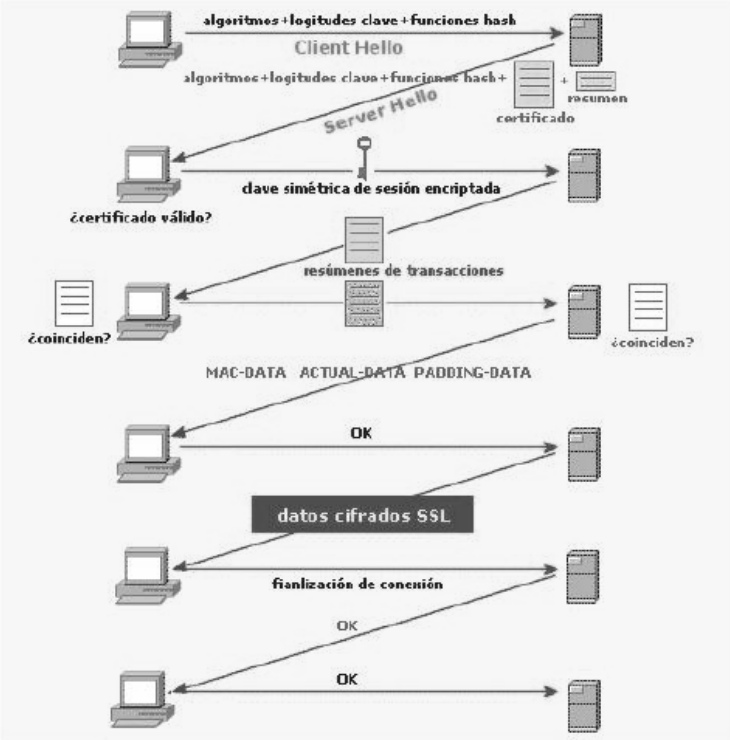


Figura I.III.IX.I. *Handshake* de SSL.

La identidad del servidor web seguro (y a veces también del usuario cliente) se consigue mediante el Certificado Digital correspondiente, del que se comprueba su validez antes de iniciar el intercambio de datos sensibles (Autentificación), mientras que de la seguridad de Integridad de los datos intercambiados se encarga la Firma Digital mediante funciones hash y la comprobación de resúmenes de todos los datos enviados y recibidos.

La base de un buen sistema vía web es la seguridad que se ofrece al intercambiar datos sensibles y la certeza que tales llegaron a su destino, así el Sistema de Administración y control de IPs (SACIP) utiliza como comunicación segura de las transacciones y envío de formularios a través del protocolo http con SSL

o https, por lo tanto la información esta protegida para el buen funcionamiento y ofrecer un servicio de calidad a los usuarios finales de este sistema.

I.IV. Programación orientada a objetos en multiplataformas.

Implementaciones especializadas, basadas en sistemas multiplataformas, ejecutables en red y adaptables a diferentes entornos, con utilización de estándares que permiten la comunicación y el acceso a la información de manera universal.

Linux es un sistema operativo para computadoras Intel y algunas otras plataformas como Mac, Alpha, Sun y Mips. El sistema lo han diseñado cientos de programadores dispersos por todo el mundo. El objetivo ha sido crear un clon de Unix que cualquier persona puede usar.

Linux es el único sistema operativo actual disponible gratuitamente que proporciona capacidades de *multitareas y multiprocesamiento* a numerosos usuarios en plataformas de hardware compatible con las PCs de IBM. Son muchas las aplicaciones de Linux que están a nuestro alcance en forma gratuita en Internet, así como está disponible en Internet, el propio código fuente para Linux.

Esta característica también es desventaja potencial para los usuarios de Linux, ya que como ningún vendedor comercial le da soporte, cuando se necesita ayuda no basta con una llamada telefónica. No obstante, Linux es relativamente estable en numerosos sistemas y, en consecuencia, le proporciona una económica oportunidad de aprender y usar uno de los sistemas operativos más populares del mundo actual: Unix.

Cuando se escribe un programa, en la mayoría de los lenguajes de programación, es necesario decidir el procesador y sistema operativo en los que se va a ejecutar, porque estos lenguajes incluyen llamadas a funciones específicas de una biblioteca asociada al sistema operativo de la plataforma destino. Cuando se está preparado para probar el programa, se envía el código fuente a un compilador que lo transforma en un conjunto de instrucciones propias de la plataforma destino. Por ejemplo, Windows se ejecuta generalmente en un proce-

sador Intel, como un Pentium, mientras que los Macintosh utilizan procesadores Motorola 68000 o PowerPC.

Cuando se escribe en Java, no necesitamos pensar en llamadas a Windows, Mac OS, u otras bibliotecas del sistema operativo. Java tiene sus propias bibliotecas, llamadas paquetes, que son independientes de la plataforma. Por ello no es necesario preocuparse si la aplicación se va a ejecutar en una plataforma Intel, una PowerPC o una SPARC. El compilador de Java no genera instrucciones nativas, en su lugar genera los llamados "código de byte" (byte code) para la Máquina Virtual Java (Java Virtual Machine o JVM), que es una máquina que no existe físicamente. Este código (byte code) está diseñado para ejecutarse en una máquina hipotética que es implementada por un sistema run-time, que sí es dependiente de la máquina y del sistema operativo, que interpreta dinámicamente el byte code y añade el 20% de instrucciones que faltaban para su ejecución.

Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancias. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria.

El código fuente escrito con cualquier editor se compila generando el byte code. Este código intermedio es de muy bajo nivel, pero sin alcanzar las instrucciones máquina propias de cada plataforma. El byte code corresponde al 80% de las instrucciones de la aplicación. Ese mismo código es el que se puede ejecutar sobre cualquier plataforma. Con este sistema es fácil crear aplicaciones multiplataforma, pero para ejecutarlas es necesario que exista el run time correspondiente al sistema operativo utilizado. Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (run-time) puede ejecutar

ese código objeto, sin importar en modo alguno la máquina en que ha sido generado.

Sun Microsystems (creadores de Java) y otras empresas han desarrollado versiones software de la JVM para una gran parte de las plataformas existentes en el mercado, es decir cada plataforma tiene su propia máquina virtual de Java. En Java podemos distinguir dos tipos de programas, las aplicaciones autosuficientes que son conocidas como aplicaciones y los que se ejecutan con la ayuda de otro programa (un navegador Web), que se conocen como applets.

Java se presenta como ambos, como lenguaje y como sistema de tiempo de ejecución (runtime), siendo posible integrar hilos dentro de ambos. El resultado final es que se pueden usar hilos Java como estándar, en cualquier plataforma.

I.IV.I. Concurrencia en JAVA (Threads y Daemons).

Cada *thread* (hilo, llamado así por el flujo de control del programa) representa un proceso individual ejecutándose en un sistema. A veces se les llama procesos ligeros o contextos de ejecución. En Java, los hilos comparten el mismo espacio de memoria. Incluso comparten gran parte del entorno de ejecución, de modo que la creación de nuevos hilos es mucho más rápida que la creación de nuevos procesos. La ventaja que proporcionan los hilos es la capacidad de tener más de un camino de ejecución en un mismo programa.

Típicamente, cada *thread* controla un único aspecto dentro de un programa, como puede ser supervisar la entrada en un determinado periférico o controlar toda la entrada/salida del disco. Todos los threads comparten los mismos recursos, al contrario que los procesos en donde cada uno tiene su propia copia de código y datos (separados unos de otros).

Considerando el entorno de SACIP, gráficamente los threads funcionan por ejemplo, como se muestra en la figura I.IV.I.I:

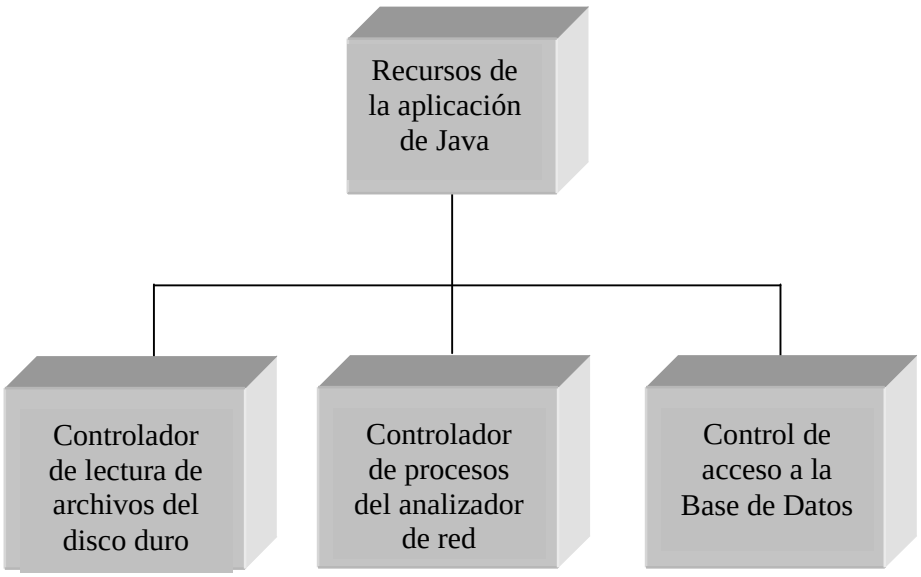


Figura I.IV.I.I. Funcionamiento de un thread.

Hay dos modos de conseguir threads en Java. Una es implementando la interface *Runnable*, la otra es extender la clase *Thread*. La única diferencia entre los dos métodos es que este último es mucho más flexible. La mayoría de las clases creadas que necesiten ejecutarse como un *thread*, implementarán la interface *Runnable*, ya que probablemente extenderán alguna de su funcionalidad a otras clases.

Java no soporta herencia múltiple de forma directa, es decir, no se puede derivar una clase de varias clases padre. Esto plantea la duda sobre cómo se puede añadir la funcionalidad de hilo a una clase que deriva de otra clase, siendo ésta distinta de *Thread*. Para lograr esto se utiliza la interfaz- *Runnable*. La interfaz *Runnable* proporciona la capacidad de añadir la funcionalidad de un hilo a una clase simplemente implementando la interface, en lugar de derivándola de la clase *Thread*.

Y esto es todo lo que hay sobre la interface *Runnable*. Como se ve, una interface sólo proporciona un diseño para las clases que vayan a ser implementadas. En el caso de *Runnable*, fuerza a la definición del método *run*(), por lo tanto, la mayor parte del trabajo se hace en la clase *Thread*. De forma general se

puede apreciar esto en la definición de la clase *Thread* que nos da idea de lo que realmente está pasando:

```
public class Thread implements Runnable {
     ...
     public void run() {
         if( tarea != null )
             tarea.run() ;
             }
         }
     ...
     }
```

De este trocito de código se desprende que la clase *Thread* también implemente la interface *Runnable*. *tarea.run*() se asegura de que la clase con que trabaja (la clase que va a ejecutarse como un thread) no sea nula y ejecuta el método *run*() de esa clase. Cuando esto suceda, el método *run*() de la clase hará que corra como un thread.

Las clases que implementan la interfaz Runnable proporcionan un método run que es ejecutado por un objeto hilo asociado que es creado aparte. Esta es una herramienta muy útil y a menudo es la única salida que se tiene para incorporar multihilo dentro de las clases.

I.IV.II. Sincronización y comunicación entre hilos o threads.

Afortunadamente Java se encarga de todo el proceso de sincronización, ejecuta internamente todo el mecanismo de acceso, manipula el estado del cerrojo, o sea, su bloqueo y correcto desbloqueo, por lo que las aplicaciones sólo tienen que indicar los recursos que se van a bloquear.

Por regla general se suele necesitar sincronización entre *threads* cuando queremos acceder ordenadamente a un recurso compartido, normalmente un objeto. En otras palabras, la sincronización se encarga de asegurar que un *thread* en un determinado momento pueda manipular un objeto. Se puede utilizar la palabra clave *synchronized* que se utiliza para definir qué parte de código (en realidad un método completo) puede ser accedido por un único thread, por lo que sé encarga de gestionar el mecanismo de cerrojos.

Además, la cláusula *synchronized* permite ejecutar dos métodos diferentes sobre el mismo objeto o el mismo método sobre dos objetos diferentes, pero impide que más de un *thread* ejecute la misma porción de código al mismo tiempo sobre el mismo objeto. La cláusula *synchronized* permite sincronizar el acceso a un determinado método.

I.IV.III. El thread "Daemon" o demonio.

Un *thread* Daemon (demonio) es un thread de baja prioridad que consta de un bucle infinito y se suele utilizar para prestar servicios cuando se necesite, por ejemplo, refresco de memoria, mostrar imágenes, etc. Java utiliza un *thread daemon*, el recolector de basura para realizar todas las gestiones sobre la creación y destrucción de los objetos en memoria. También se llaman servicios proporcionan un servicio básico a un programa o programas cuando la actividad de la máquina es reducida.

Un *thread* demonio se declara *Thread.setDaemon();* y se puede comprobar si un thread es tipo "demonio" mediante *isDaemon();*

La diferencia entre los *threads* normales y los demonios es que Java no espera la muerte de los demonios para detener su ejecución. Los *threads* normales impiden al núcleo de Java parar el programa principal hasta que estos no mueren bien por la llamada a *stop()* o por la terminación normal del *thread*. En el caso de los demonios, Java los elimina cuando comprueba que únicamente quedan *threads* del tipo demonios. Su ventaja consiste en que no es necesario controlar su finalización, de tal forma que se evitan los clásicos procedimientos de *pararAnalizar()*, *detenerRespaldo()*, etc.

Otra característica consiste en que cualquier thread que se crea desde otro que es daemon también es un thread demonio.

Capitulo II

II.I. Planteamiento del Problema

El Sistema de Información Universitaria (SIU) utiliza un DHCP[12] para asignar dinámicamente un IP a cada equipo de la universidad, sin embargo en la Facultad de Ciencias de la Computación (BUAP) muchas veces se asigna de manera manual dicha dirección, ya que es necesario que la máquina mantenga su IP (muchas son usadas como servidores). La forma en que se administra la red, específicamente en la Facultad de Ciencias de la Computación de la BUAP es externamente por el SIU este organismo interno de la universidad se encarga de concentrar y controlar los ruteadores principales para darle salida a toda la red al mundo exterior e informar a cada facultad de problemas que tienen dentro de las subredes que existen en cada una de las mismas dentro de su área respectivamente, de esta forma los encargados de mantener este sistema, esto es posible gracias a que tiene ruteadores que administran la red interna, la solución para cada subred dentro de la antes mencionada es que pueda haber un sistema que informe a tiempo y de forma precisa para obtener la información de la computadora o grupo de computadoras que causan problemas.

Tanto el SIU como la facultad están trabajando conforme a Cisco System y en cuanto a su software trabajan sobre el IOS[13] que es el sistema operativo de ruteo, esto es todo a cerca del software.

La red de la facultad llega desde el SIU a través de fibra óptica monomodo a través de un switch de tercera capa que ruteo (Cisco System 4908GL3) y sale otro tipo de fibra para repartir a la facultad. A semiconductores y a química.

---

[12] **DHCP** son las siglas en inglés de Protocolo de configuración dinámica de servidores (Dynamic Host Configuration Protocol). Es un protocolo de red en el que un servidor provee los parámetros de configuración a las computadoras conectadas a la red informática que los requieran (máscara, puerta de enlace y otros) y también incluye un mecanismo de asignación de direcciones de IP; *Colaboradores de Wikipedia. DHCP [en línea]. Wikipedia, La enciclopedia libre, 2006 [fecha de consulta: 8 de julio del 2006]. Disponible en <http://es.wikipedia.org/w/index.php?title=DHCP&oldid=3830780>.*

[13] **IOS** son las siglas de (Internetwork Operating System, Sistema Operativo de Interconexión de Redes) creado por Cisco Systems para programar y mantener equipos de interconexión de redes informáticas como switches (conmutadores) y routers (ruteadores).

Esta fibra de la facultad entra a un pacht panel (Catalyst 3500XL) y sale con cable UTP, este se distribuye en la facultad de la siguiente manera donde tienen las direcciones ordenadas que se reparten al edificio 136 y 135.

El SIU dio 3 segmentos de direcciones de IP para identificar a la facultad las cuales son: 20,21y 22. Al edificio 136 se le quedan la 20 y 21 que se encuentra en el Catalyst, de aquí se supone que pasan directamente a las Computadoras, pero como son mayores al número del servidor se tiene que hacer un cascadeo mínimo casi sin ruido con concentradores (3COM) y dos switchs; para el otro edificio 135 se manda la dirección 22, se pasa un cable del Catalyst a dicho edificio donde se coloca otro servidor que reparte la red a esos cubículos y salones.

La velocidad a la que se transmite en la facultad es de 10 a 100 MB, pero nunca se usan más de 100 MB por que se pueden quemar las tarjetas, los cables, etc.; además de que nunca llegan a ocupar los 100 MB porque las tarjetas no tienen tanta capacidad. Se usan normas para audio, video y datos, la red de la facultad es de tipología estrella.

Para conectarse con el exterior la información viaja hacia el SIU y sale al exterior, y viceversa.

Antes la red salía vía MODEM al Carolino y de ahí se repartía hacia fuera y se trabajaba con la compañía telefónica Telmex, hoy en día contamos con un enlace C1s con la compañía Avantel, que es mucho más rápida y eficiente.

El SIU transmite por medio de cable UTP a categoría (Gigabits). La planificación de los enlaces de fibra óptica monomodo utilizan el backbone. Utiliza un switch de 12, 24 y 48 puertos. El software es de monitoreo.

No es posible que dos máquinas compartan el mismo IP, cuando a dos máquinas se les asigna el mismo IP la segunda no puede enviar paquetes a través de la Red. Al establecer manualmente dichas direcciones es probable que se presenten este tipo de “colisiones”. Por otro lado, en últimas fechas el servicio

de correo electrónico de la FCC recibe constantemente ataques de SPAM e incluso algunos de sus equipos son generadores de SPAM.

Por los problemas antes mencionados es de suma importancia poder monitorear la red de la FCC para poder ubicar los equipos que causen conflictos en la red y tomar las medidas pertinentes.

Solución Propuesta

Se propone una asignación de los IPS de acuerdo a su ubicación dando un orden secuencial a edificios que a su vez se segmentan en módulos, salones, oficinas, laboratorios y cubículos de profesores, esto se puede realizar virtualmente con la administración de los IPs conforme a su subred.

Este sistema proporciona un sistema que permite monitorear dinámicamente el equipo conectado a Red de la FCC, con el fin de poder ubicar los equipos conflictivos de manera oportuna, así como detectar violaciones en la asignación manual de los IPS. De esta forma se logra un control local sobre los segmentos de la red y dicha información es disponible para su consulta vía Web.

Objetivo General

Crear la primera de una serie de herramientas que permita la automatización en la administración de la Red, mejorando significativamente los servicios prestados a la comunidad de la FCC.

Objetivos Específicos

- El sistema debe ser accesible vía Web.
- Dado un IP, obtener un historial de todas las direcciones físicas que lo han utilizado.
- Dado una dirección física obtener un historial de todos los IPS que le han sido asignados.

Alcances y Limitaciones

- El sistema monitoreará sólo los segmentos de la facultad (20, 21, 22) virtualmente.
- La información referente a los responsables, ubicación y rosetas se captura manualmente, por lo que el sistema necesita ser actualizado por lo menos cada mes.
- En la primera versión no se considera el monitoreo de trafico en la red ni estadísticas sobre el uso de la Red.
- Al monitorear la asignación de IPS es posible implementar políticas y normatividades sobre el uso de los mismos, así como las respectivas sanciones.
- En las redes privadas el sistema sólo reconocerá al equipo conectado físicamente a la roseta (WireLess y/o Switch).

Trabajo a futuro

- Extender la administración a cualquier área de trabajo o segmentos de red agregados a la FCC.
- Vender el sistema o reutilizar las clases que integran el sistema para otros sistemas.
- Actualizar o mejorar la presentación-vista del sistema, conforme avance la tecnología, por ejemplo la visualización 3D de las ubicaciones.
- Implementar el mapa de la red para apreciar de una mejor forma y gráficamente todas las ubicaciones, como mejora agregada al punto anterior.
- Mantener el sistema actualizado con versiones mejoradas de las partes que componen el sistema.
- Extender el trabajo para poder monitorear toda la red de la BUAP.
- Agregar funcionalidades de monitoreo de tráfico en la red, con un paquete de java llamado jpcap.
- Ampliar el sistema al protocolo IPV6.
- Generar estadísticas sobre la red de su uso, equipos, IPs, direcciones físicas, los puertos más usados, etc.

- Complementar el sistema con servicio de E-mail y servicio de mensajes instantáneos para usuarios dentro de la facultad o para toda Ciudad Universitaria, por decir un ejemplo existen varias aplicaciones para realizar este trabajo e implementarlo sin tanta complicación por ejemplo el API google web toolkit (GWT).[14]
- Mantener una auditoria física y lógica obligatoria de la relación entre la dirección física y el número de roseta.
- Implementar una función para acceso vía celular o por medio de un producto multifuncional WiFi.

II.II. Análisis del sistema.

II.II.I. Lista de Requerimientos.

En la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla, todos los usuarios tienen acceso a laboratorios, por lo regular existe un registro de cada usuario en los laboratorios en otra base de datos, que administra otro sistema. Los profesores tienen a su cargo laboratorios y computadoras dentro de la facultad, estos laboratorios también son administrados por alumnos de la misma facultad o que prestan servicio social, básicamente para poder dar servicio de autoacceso a los demás alumnos que cursan materias en la carrera. Es de vital importancia que se mantenga una supervisión-administración de los equipos en la facultad ya que una herramienta de supervisión y auditoria facilitaría la forma de hacer estas actividades que normalmente se hacen a mano y tardan para realizarse. Aun más tener un registro de todos los días o de movimientos realizados por cambios de direcciones lógicas o IPs de las computadoras con las que se da servicio o actualización de algunas de ellas.

[14] Más información en**: http://code.google.com/webtoolkit/**

Usuarios y concurrencia

1. El sistema debe ser accesible vía Web, y debe permitir concurrencia.
2. El sistema tiene dos tipos de usuario, el administrador del sistema y los responsables de departamentos.
3. Cuando un usuario entra o sale del sistema, el sistema busca usuarios de la misma característica y si los encuentra se les notifica a los demás el acceso o la salida.
4. La sesión de cualquier usuario caduca si no tiene actividad en 10 minutos.

Monitoreo

1. Cuando el sistema detecta una MACADDRESS que no se encuentra registrada en la BD se considera como un *equipo no registrado*.
2. El sistema hace un monitoreo para detectar que los rangos de IPS y MACADDRESS de los equipos se mantengan en su departamento, y cualquier violación de espacio es reportada por el sistema.
3. El sistema propone las posibles vulnerabilidades de la conexión de cada equipo al administrador y avisa que puerto esta abierto para tal MACADDRESS.

Búsquedas

1. Se pueden buscar equipos por los siguientes criterios:
    a. Dirección IP; Se consulta en la BD todas las máquinas que han usado ese IP
    b. MACADDRESS; Se consulta en la DB todas las máquinas que han usado esa MACADDRESS.
2. Se pueden ejecutar búsquedas remotas sobre cualquier tabla de la base de datos del tipo: general condicionadas y anidadas.

Requerimientos Administradores del Sistema

1. Siempre que se necesite ejecutar el monitor se realiza una búsqueda de MACADDRESS e IPS, una vez recorridos todos los sectores se detiene

el monitoreo, pero puede ser reiniciado y detenido cuando el administrador lo considere pertinente.

2. El administrador puede dar alta, eliminar o modificar: edificios, pisos y departamentos.
3. Al dar de alta un departamento se le debe asignar un rango de IPS, dando una recomendación de acuerdo a su IP.
4. El sistema permite al administrador asignar equipos a departamentos.
5. El sistema proporciona una opción para respaldar la BD en cualquier momento de manera segura.
6. El sistema puede mostrar el historial y la relación IP-MACADDRESS de cualquier computadora.
7. El administrador puede dar de alta, baja o modificar los responsables de departamento.
8. El administrador puede realizar búsquedas (ver requerimiento 12 donde se trata las Búsquedas) sobre todos los sectores.
9. El administrador debe ver un listado con los IPS y MACADDRESS que se hayan monitoreado en una sesión anterior, al dar un clic en un equipo se debe mostrar toda la información que se tenga registrada en la base de datos.

Requerimientos Responsables de Departamentos

1. El responsable de departamento debe ver un listado con los IPS y MACADDRESS registrada en la base de datos únicamente de los departamentos de los que esta encargado (consulta).
2. El responsable puede anexar nuevos equipos a su departamento.
3. El responsable puede visualizar únicamente la información de las computadoras de las que es encargado.
4. El responsable visualiza la ubicación y la información de los equipos a su cargo.
5. El responsable puede modificar sus datos personales cuando lo requiera.

Capitulo III

III.I. Diseño del sistema.

El diseño de SACIP fue pensado para ser extensible, de fácil manejo, y actualizable, por eso para cumplir con un buen esquema de la aplicación y estándar del diseño, el sistema fue basado en la especificación UML ya que el sistema debe ser orientada a objetos sobre la plataforma Java además, es la mejor forma de mantener un sistema el ciclo de vida asociado, haciéndolo fácil de modificar, mejorar e incrementar la reusabilidad del código. La ventaja más importante de todas es la estandarización de diseño del sistema y que sea entendido a nivel mundial y orientado a un lenguaje Orientado a Objeto. El diseño en UML establece normas específicas, pero a la vez, permite libremente modificar algunas y complementarlas.

El siguiente diseño mediante diagramas UML es sobre el modelo del Sistema de Administración y Control de Ips (SACIP) para entender mejor su interacción, su implementación, los pasos que pretenden explicar la estructura del sistema y su comportamiento, mediante casos de usos con sus respectivos diagramas, los diagramas de clases y los diagramas de secuencia.

Así, el ciclo de desarrollo del sistema se representa en la figura III.I.I.

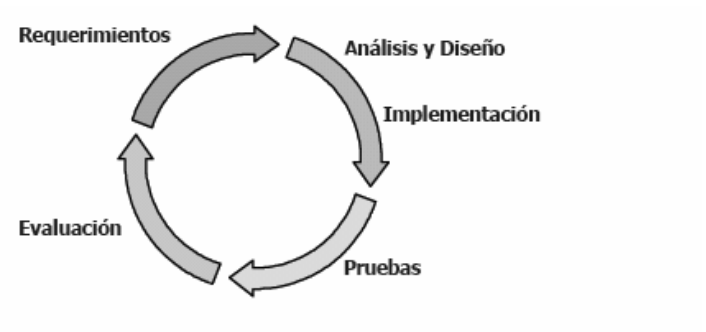


Figura III.I.I. Ciclo de desarrollo del sistema.

En cada ciclo se aborda un conjunto relativamente pequeño de requerimientos, pasando por el análisis, diseño, construcción y pruebas y el sistema va creciendo en cada ciclo que concluye.

De acuerdo con los requerimientos identificados los siguientes casos de uso se modelan para ayudar a obtener los resultados deseados, en donde cada caso de uso representa las posibles acciones que un usuario puede realizar con el sistema.

III.I.I. Metodología de desarrollo.

Internet ha traído consigo necesidades nuevas, que ha hecho que las necesidades de desarrollo también deban cambiar. Si por ahora hemos asistido a un cambio en los lenguajes de desarrollo utilizados, debemos adaptar todavía los demás puntos que forman parte del desarrollo de un proyecto, como pueden ser gestión y análisis. Así se deben adaptar nuestras metodologías a las necesidades de estos proyectos.

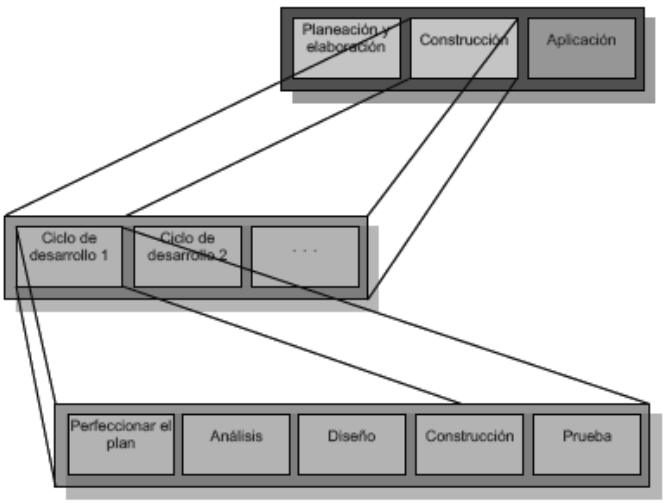


Figura III.I.I. Ciclo iterativo de desarrollo (Larman, 99).

Las ventajas que ofrece este modelo de desarrollo son las siguientes:

- Reducción de los riesgos basándose en una retroalimentación temprana.

- Mayor flexibilidad para manejar cambios nuevos o modificaciones a los mismos.
- La complejidad nunca resulta abrumadora.

III.I.II. Funciones del sistema.

Las actividades y funciones de forma general que debe realizar el sistema son:

Gestionar los datos de la red y de sus responsables de departamentos o módulos ya sea que puedan ser docentes o encargados de cada área y que a través de administradores del sistema sean capaces de analizar su respectiva red de computadoras para poder administrar y controlar sus direcciones IP así como de mantener un control actualizado de la información y de las vulnerabilidades que puedan tener los puertos abiertos avisando oportunamente alguna irregularidad, de esta forma se tiene una visión general de la red de la facultad, ubicando los puntos de conflicto.

En este caso cada una de estas funciones se engloban en los siguientes puntos:

- Ingreso al sistema
- Administración de usuarios
- Administración de ubicaciones
- Administración de equipos
- Consultas
- Análisis de la red
- Respaldo y recuperación de la base de datos

III.I.IV. Tabla de Anexos, referencias de códigos fuentes, diagramas y manuales de SACIP.

Referencia	Nombre del archivo externo	No. de pagina	Descripción
1. Organización de paquetes principales del sistema.	**Diagramas generales UML de SACIP.doc**	1	Diagramas de paquetes principales representados en UML.
2. Diagrama general de Casos de Uso.	**Diagramas generales UML de SACIP.doc**	1	Diagrama general de los casos de uso de SACIP en UML.
3. Diagrama de clases principal (paquete sacip) parte A.	**Diagramas generales UML de SACIP.doc**	2	Parte A del diagrama de clases paquete (sacip) estas clases sirven para validar y presentar los equipos y departamentos de cada usuarios del sistema en SACIP en UML.
4. Diagrama de clases principal (paquete sacip) parte B.	**Diagramas generales UML de SACIP.doc**	3	Parte B diagrama de clases paquete (sacip) estas clases sirven para recuperar y respaldar la base de datos completa de SACIP en UML.
5. Diagrama de clases principal (paquete sacip) parte C.	**Diagramas generales UML de SACIP.doc**	4	Parte C diagrama de clases paquete (sacip) estas clases sirven para el análisis de red, y el parseo del archivo de análisis de la red de SACIP en UML.
6. Diagrama de clases principal (paquete sacip) parte D.	**Diagramas generales UML de SACIP.doc**	5	Parte D diagrama de clases paquete (sacip) estas clases sirven para mostrar el applet de consulta libres en la base de datos, la salida del sistema y la consulta de registros de entrada a SACIP en UML.
7. Organización de paquetes dentro del paquete sacip.	**Diagramas generales UML de SACIP.doc**	6	Diagrama de paquetes dentro del paquete (sacip) de SACIP en UML.
8. Diagrama de clases (paquete reporte) parte A.	**Diagramas generales UML de SACIP.doc**	6	Diagrama de clases (paquete reporte) parte A, estas clases sirven para hacer movimientos en la base de datos desde los JSPs de SACIP en UML.
9.Diagrama de clases (paquete reporte) parte B.	**Diagramas generales UML de SACIP.doc**	7	Diagrama de clases (paquete reporte) parte B, estas clases sirven para hacer movimientos en la base de datos des-

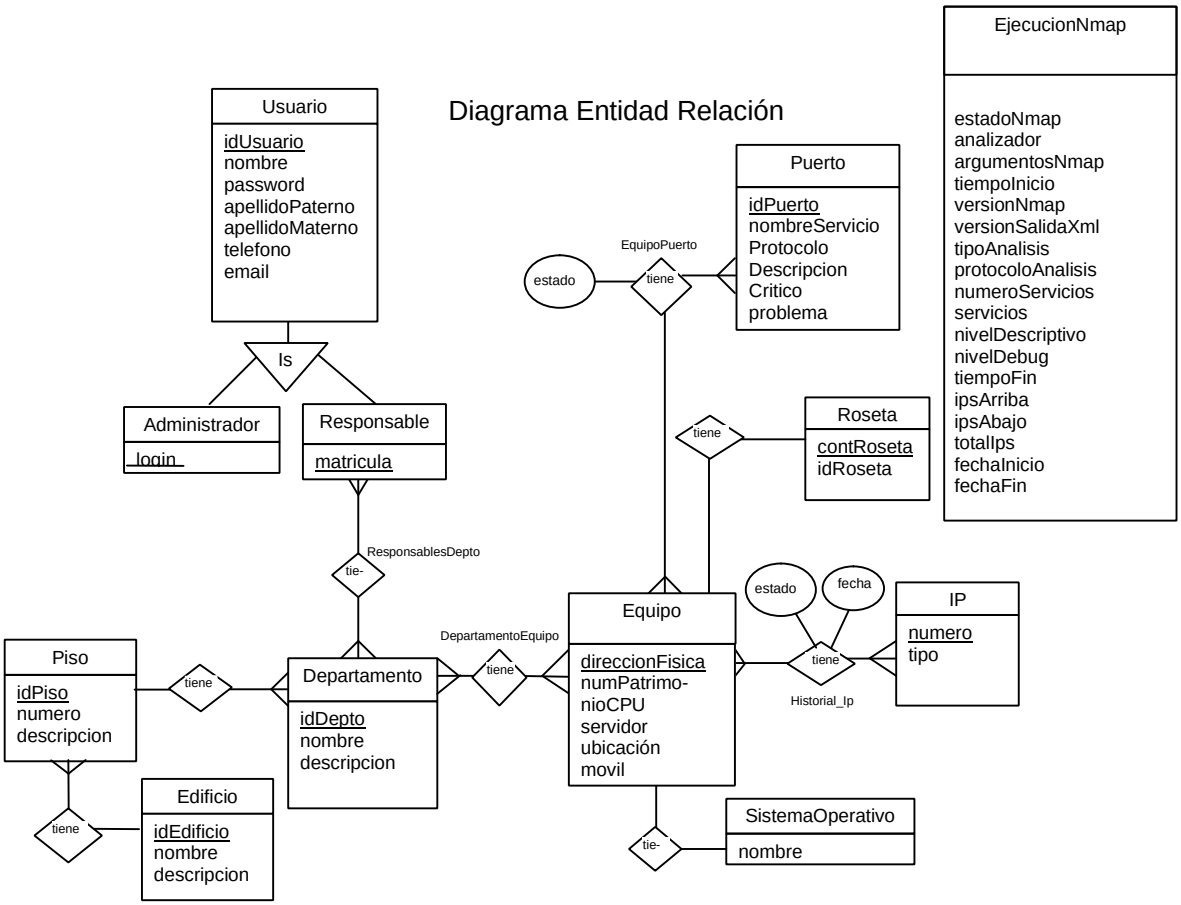
			de los JSPs de SACIP en UML.
10 Diagrama de clases (paquete listeners).	**Diagramas generales UML de SACIP.doc**	8	Diagrama de clases (paquete listeners) estas clases sirven para mostrar el número de usuarios dentro del sistema y saber cuando ha salido o entrado cierto usuario SACIP en UML.
11. Diagrama de clases (paquete conectabd).	**Diagramas generales UML de SACIP.doc**	8	Diagrama de clases (paquete conectabd) estas clases sirven para conectar cualquier clase con la base de datos de SACIP en UML.
12. Diagrama de Clases (applet de consulta).	**Diagramas generales UML de SACIP.doc**	9	Diagrama de clases (applet de consulta) estas clases sirven como interfase para las consultas libres de SACIP en UML.
13. Diagrama de clases JSPs	**Diagramas generales UML de SACIP.doc**	10	Diagrama de clases JSPs, estas clases sirven para representar la vista del sistema y se utilizan en casi todos los procesos de SACIP.
14. Diagrama de secuencia (Ingresar al sistema).	**Diagramas generales UML de SACIP.doc**	11	Diagrama de secuencia de la entrada al sistema que valida los datos del usuario y muestra los usuarios activos en el sistema así como el conteo de los usuarios dentro del sistema.
15. Diagrama de secuencia (Consulta de departamentos).	**Diagramas generales UML de SACIP.doc**	12	Diagrama de secuencia de la consulta de departamentos de los que el usuario activo es responsable dentro de SACIP.
16. Diagrama de secuencia (Consulta de equipos).	**Diagramas generales UML de SACIP.doc**	13	Diagrama de secuencia de la consulta de equipos de los que el usuario activo es responsable dentro de SACIP.
17. Diagrama de secuencia (Alta de administrador del sistema).	**Diagramas generales UML de SACIP.doc**	14	Diagrama de secuencia de la alta de un administrador de SACIP.
18. Diagrama de secuencia (Alta de responsable de departamento).	**Diagramas generales UML de SACIP.doc**	15	Diagrama de secuencia de la alta de un responsable de departamento de SACIP.
19. Diagrama de secuencia (Alta de relación responsable-	**Diagramas generales UML de SACIP.doc**	16	Diagrama de secuencia donde se muestra la alta de la relación del res-

departamento).			ponsable con su respectivo departamento de SACIP.
20. Diagrama de secuencia (Alta de Edificio).	**Diagramas generales UML de SACIP.doc**	17	Diagrama de secuencia donde se muestra como se da de alta un edificio de SACIP.
21. Diagrama de secuencia (Alta de departamento).	**Diagramas generales UML de SACIP.doc**	18	Diagrama de secuencia donde se muestra cual es el proceso de alta de un departamento de un edificio de SACIP.
22. Diagrama de secuencia (Alta de relación departamento-equipo).	**Diagramas generales UML de SACIP.doc**	19	Diagrama de secuencia donde se muestra el proceso de la alta de la relación entre un departamento y un equipo de SACIP.
23. Diagrama de secuencia (Alta de piso).	**Diagramas generales UML de SACIP.doc**	20	Diagrama de secuencia donde se da de alta un piso del edificio de SACIP.
24. Diagrama de secuencia (Baja de administrador del sistema).	**Diagramas generales UML de SACIP.doc**	21	Diagrama de secuencia donde se muestra el proceso de dar de baja un administrador de SACIP.
25. Diagrama de secuencia (Baja de responsable de departamento).	**Diagramas generales UML de SACIP.doc**	22	Diagrama de secuencia donde se muestra el proceso de dar de baja un responsable de departamento de SACIP.
26. Diagrama de secuencia (Baja de edificio).	**Diagramas generales UML de SACIP.doc**	23	Diagrama de secuencia donde se muestra el proceso de dar de baja un edificio de SACIP.
27. Diagrama de secuencia (Baja de piso).	**Diagramas generales UML de SACIP.doc**	24	Diagrama de secuencia donde se muestra la baja de un piso de un edificio de SACIP.
28. Diagrama de secuencia (Baja de departamento).	**Diagramas generales UML de SACIP.doc**	25	Diagrama de secuencia donde se muestra la baja de un departamento de un piso dentro de un edificio de SACIP.
29. Diagrama de secuencia (Baja de relación departamento-equipo).	**Diagramas generales UML de SACIP.doc**	26	Diagrama de secuencia del proceso de baja de la relación de un departamento con un equipo de SACIP.
30. Diagrama de secuencia (Modificación de datos de usuarios del sistema).	**Diagramas generales UML de SACIP.doc**	27	Diagrama de secuencia del proceso de modificación de los datos de un usuario de SACIP.
31. Diagrama de secuencia (Modificación de datos de edificio).	**Diagramas generales UML de SACIP.doc**	28	Diagrama de secuencia del proceso de modificación de los datos de un edificio de SACIP.

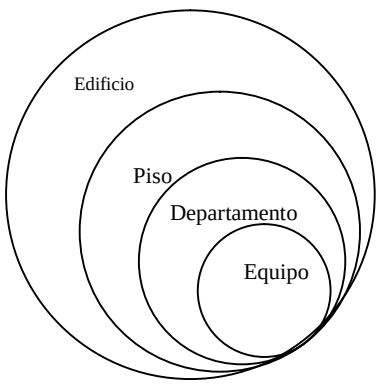
32. Diagrama de secuencia (Modificación de datos de piso).	**Diagramas generales UML de SACIP.doc**	29	Diagrama de secuencia del proceso de modificación de los datos de un piso de SACIP.
33. Diagrama de secuencia (Modificación de datos de departamento).	**Diagramas generales UML de SACIP.doc**	30	Diagrama de secuencia del proceso de la modificación de los datos de un departamento de SACIP.
34. Diagrama de secuencia (Consulta de registro de ingresos al sistema en formato texto plano).	**Diagramas generales UML de SACIP.doc**	31	Diagrama de secuencia del proceso de la consulta del registro de ingresos al sistema en formato de texto plano en SACIP.
35. Diagrama de secuencia (Consulta de registro de ingresos al sistema en formato html).	**Diagramas generales UML de SACIP.doc**	32	Diagrama de secuencia de la consulta del registro de ingresos al sistema en formato html en SACIP.
36. Diagrama de secuencia (Consultas libres).	**Diagramas generales UML de SACIP.doc**	33	Diagrama de secuencia del proceso de consultas libres en el applet en SACIP.
37. Diagrama de secuencia (Consulta de IP específico).	**Diagramas generales UML de SACIP.doc**	34	Diagrama de secuencia del proceso de consulta de un IP en especifico en SACIP.
38, Diagrama de secuencia (Consulta de dirección física específica).	**Diagramas generales UML de SACIP.doc**	35	Diagrama de secuencia del proceso de consulta de una dirección física especifica en SACIP.
39. Diagrama de secuencia (Consulta general de datos departamento:equipo).	**Diagramas generales UML de SACIP.doc**	36	Diagrama de secuencia del proceso de consulta general de los datos de un departamento y sus equipos en SACIP.
40. Diagrama de secuencia (Análisis de la red).	**Diagramas generales UML de SACIP.doc**	37	Diagrama de secuencia del proceso de análisis de la red en SACIP.
41. Diagrama de secuencia (Respaldo de la base de datos).	**Diagramas generales UML de SACIP.doc**	38	Diagrama de secuencia del proceso de respaldo de la base de datos de SACIP.
42. Diagrama de secuencia (Recuperación de la base de datos).	**Diagramas generales UML de SACIP.doc**	39	Diagrama de secuencia del proceso de recuperación de la base de datos de SACIP.
ANEXO MODELADO RATIONAL ROSE.	**MODELADO COMPLETO SACIP JAVA.mdl**	-	Todos los diagramas (casos de uso, clases y de secuencia) creados en Rational Rose 2003.
ANEXO 1	**En este documento**		Nmap, Clasificación de control de exportación de los EEUU.
Manual técnico de instalación de SACIP.	**Manual técnico de instalación de SACIP.doc**	-	Manual de toda la información, procesos e instrucciones necesarias

			para poner en marcha SACIP.
Manual de usuario de SACIP.	**Manual de usuario de SACIP.doc**	-	Manual de la utilización y mantenimiento de SACIP.
Código fuente de SACIP	**SACIP.zip**	-	Archivo empaquetado en formato zip que contiene la carpeta con código fuente para el contenedor de servlets y JSPs APACHE-TOMCAT.
Código fuente de la base de datos SACIP (mysql).	**SACIP.sql**	-	Código fuente para la creación de la base de datos SACIP en mysql.
Código fuente de datos para insertar los números de puertos y descripciones en la base de datos SACIP (mysql)	**PUERTOS.sql**	-	Código fuente para insertar los datos de todos los puertos en la base de datos SACIP.
Código fuente de datos de inicio para ingresar a SACIP (mysql)	**Datos de entrada SACIP.sql**	-	Código fuente para insertar los datos de inicio de SACIP, y arrancar correctamente el sistema.
Lista de puertos vulnerables a troyanos y gusanos.	**puertos vulnerables.txt**	-	Lista de puertos vulnerables a troyanos y gusanos para generar una actualización.
Anexo II Casos de uso de SACIP	**Anexo II - Casos de uso de SACIP.doc**	-	Todos los casos de uso detalladamente de SACIP se encuentran dentro del documento, importante para referencia para desarrolladores.

III.I.V Diagrama Entidad Relación (base de datos SACIP).



III.I.VI. Gráfica de organización de las ubicaciones.



III.II. Implementación del Sistema de administración y control de IPs de una red (SACIP).

III.II.I. Software.

Para la realización del sistema de administración y control de IPs (SACIP) se decidió usar el Software y Hardware siguientes:

- MySQL (Descrito en la sección I.I.III.)
- JDK (Java Development Kit descrito en la sección I.IV.)
- Apache – Tomcat (Descrito en la sección I.III.III.)
- Servlets – JSPs – Beans (Descritos en la sección I.III.VI. y I.III.VII.)
- Log4j (Descrito en la sección I.III.IV.)
- XML (Xerces-J, descrito en la sección I.III.V.)
- Nmap (Descrito en la sección I.II.XI.)

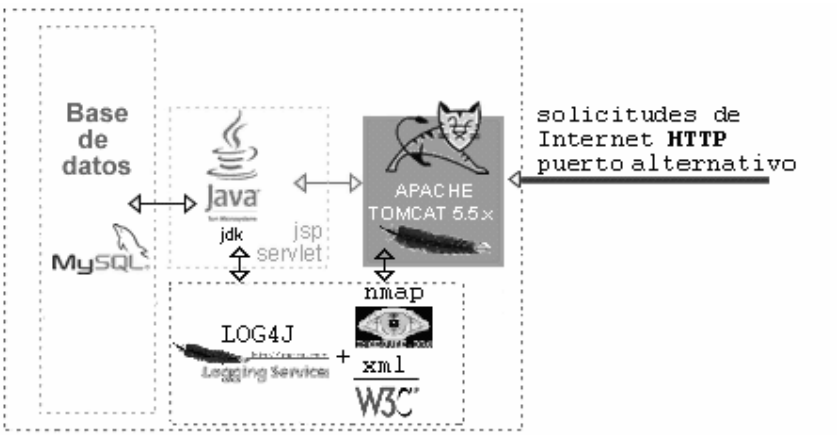


Figura III.II.I. Ilustración de la integración Tomcat Apache-Jsp/Servlet-Mysql y otras herramientas del sistema

Para mayor información del software consultar el manual técnico de instalación de SACIP, donde se explica detalladamente la instalación de todo el software.

III.II.II. Hardware.

- Para el desarrollo de SACIP:
  - CPU Pentium IV a 2.8 Ghz., 768 Mb RAM, sistema operativo Windows Xp.
- Para las pruebas y puesta en marcha de SACIP:
  - CPU Pentium IV a 2.8 Ghz., 256 Mb RAM, sistema operativo Linux Mandrake Versión 10 (Servidor de base de datos bd1.cs.buap.mx), ubicado dentro de la Facultad de Ciencias de la Computación.
  - Red de la Facultad de Ciencias de la Computación, segmento 20 de red.

Capitulo IV

IV.I. Ejemplos.

Los ejemplos de las pantallas del sistema se muestran como sigue:

Figura I.IV.I La pantalla de entrada a SACIP.

Las pantallas de las ligas a las diferentes partes del servidor

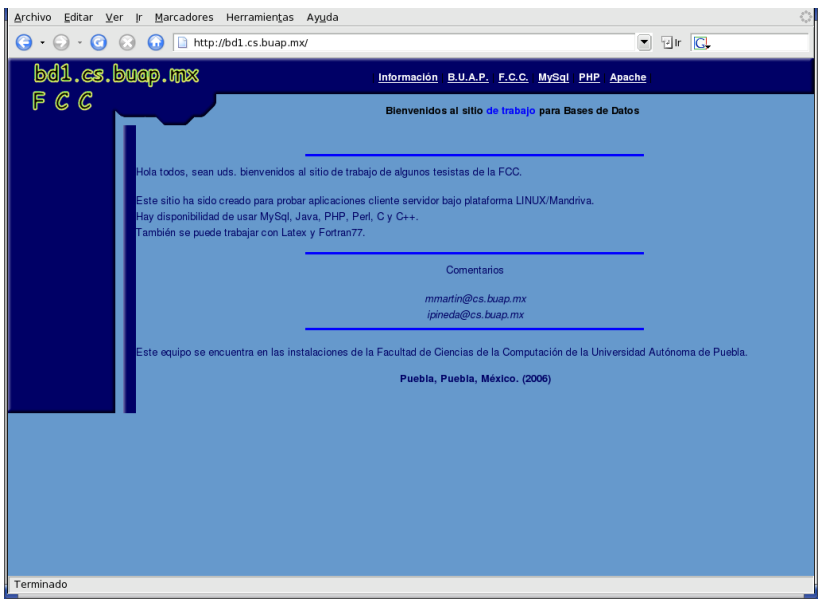


Figura I.IV.II Página principal de desarrollo php

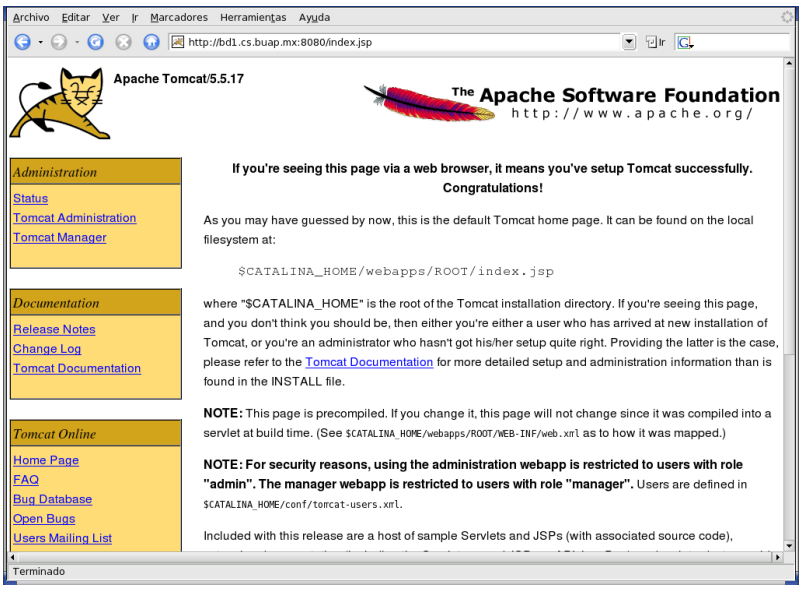


Figura I.IV.III Página principal de administración apache-tomcat

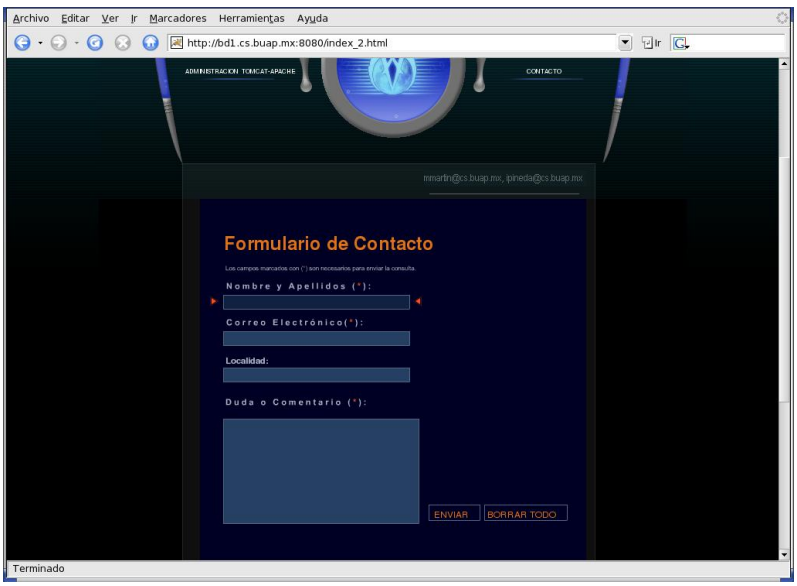


Figura I.IV.IV Página principal de contacto

Figura I.IV.V Página principal de acceso a SACIP

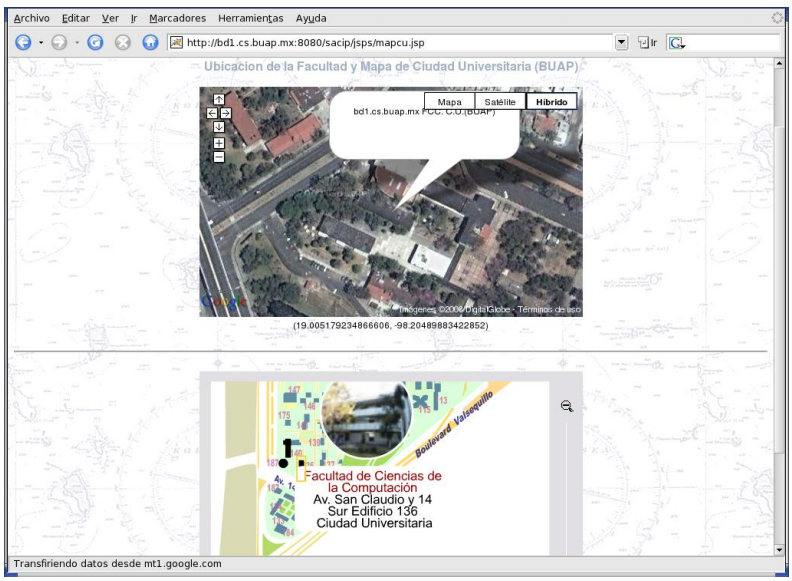


Figura I.IV.VI Página del mapa de CU

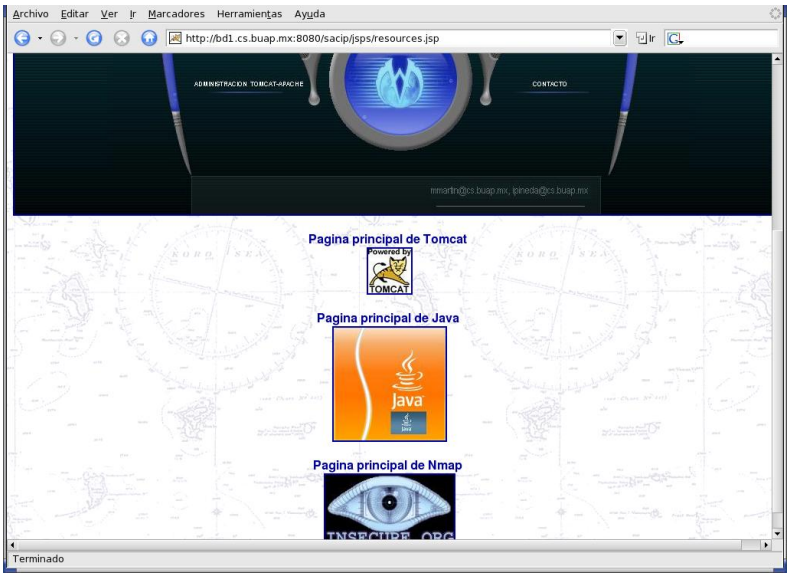


Figura I.IV.VII Página de los recursos del sistema

Figura I.IV.VIII Página principal de bienvenida del administrador del sistema de SACIP

Figura I.IV.IX Página principal de bienvenida del responsable de departamento de SACIP

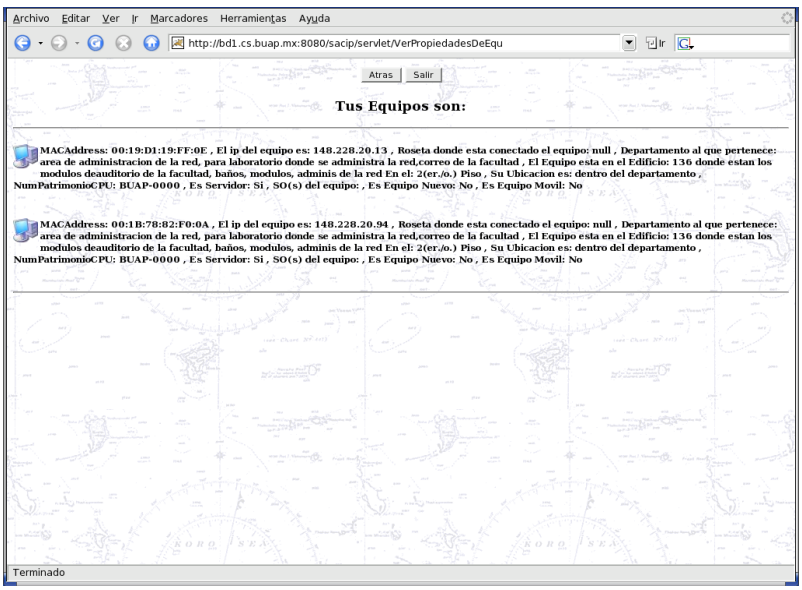


Figura I.IV.X Página de la pantalla de Equipos (Responsable de departamento)

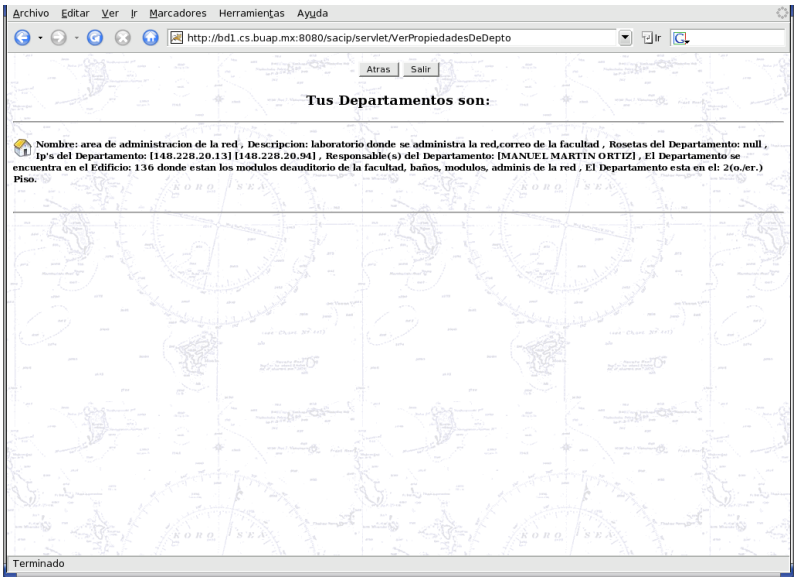


Figura I.IV.XI Página de la pantalla de Departamentos (Responsable de departamento)

Capitulo V

V.I. Pruebas.

Las pruebas que se realizaron fueron principalmente sobre las opciones mas importantes de SACIP para demostrar la funcionalidad y la practicidad con la que se puede trabajar en el sistema, las pruebas hechas se hicieron bajo condiciones normales en un ambiente descrito en la sección III.II. (Implementación del Sistema de administración y control de IPs de una red).

Figura I.V.I Prueba de alta de relación de un responsable de departamento con un departamento.

Figura I.V.II Prueba de alta un piso de un edificio en SACIP.

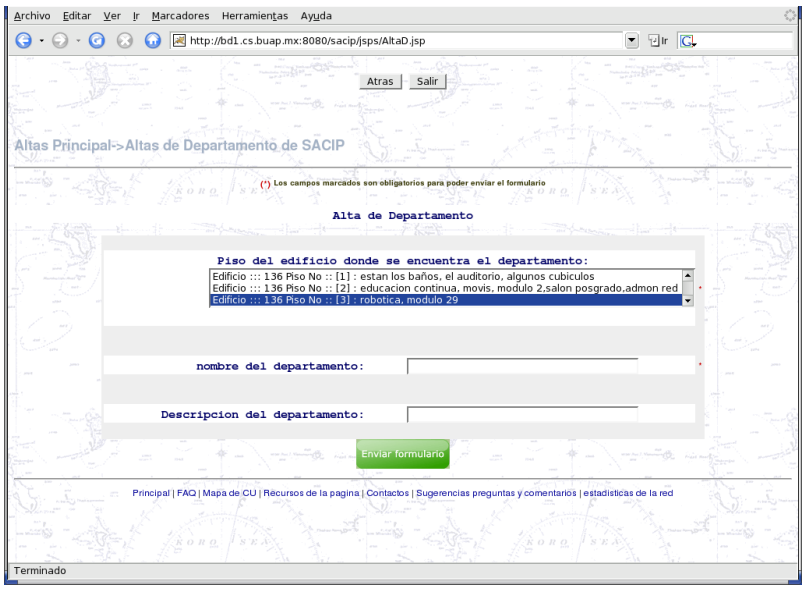


Figura I.V.III Prueba de alta de un departamento en un piso en un edificio en SACIP

Figura I.V.IV Prueba de alta de la relación de un departamento y un equipo en SACIP

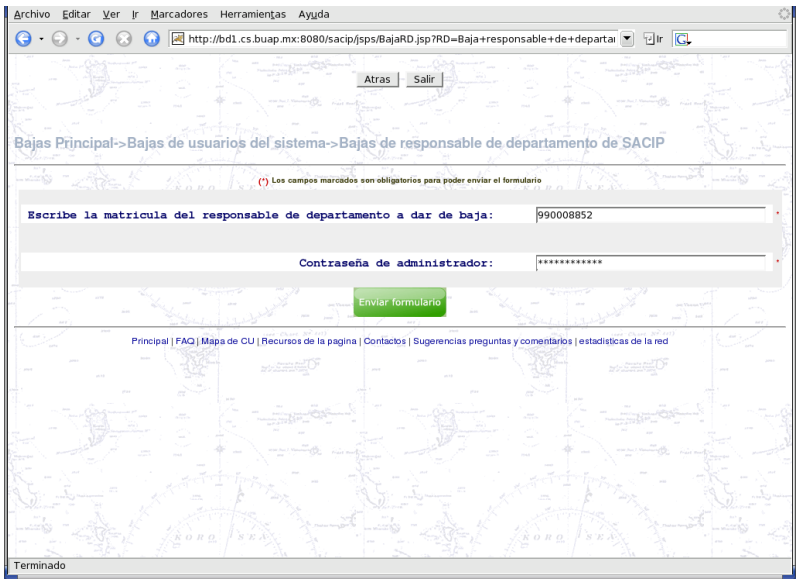


Figura I.V.V Prueba de baja de responsable de departamento de SACIP

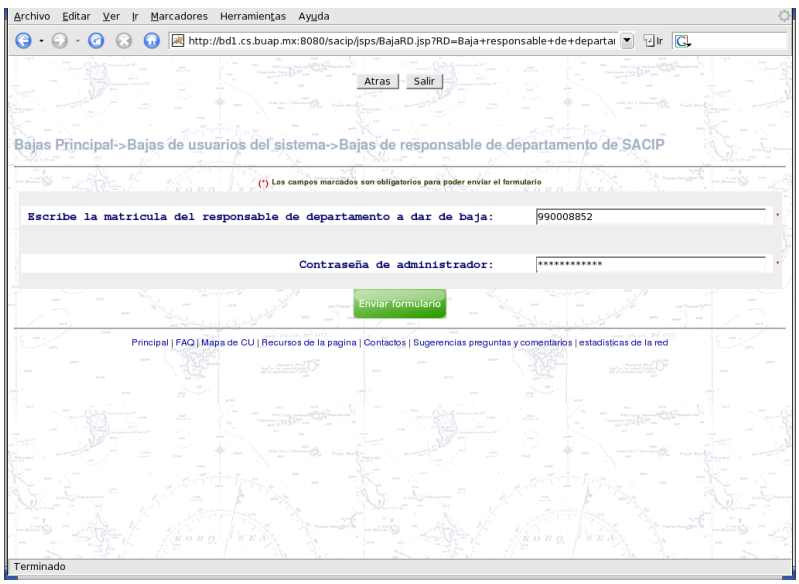


Figura I.V.VI Prueba de baja de responsable de departamento de SACIP

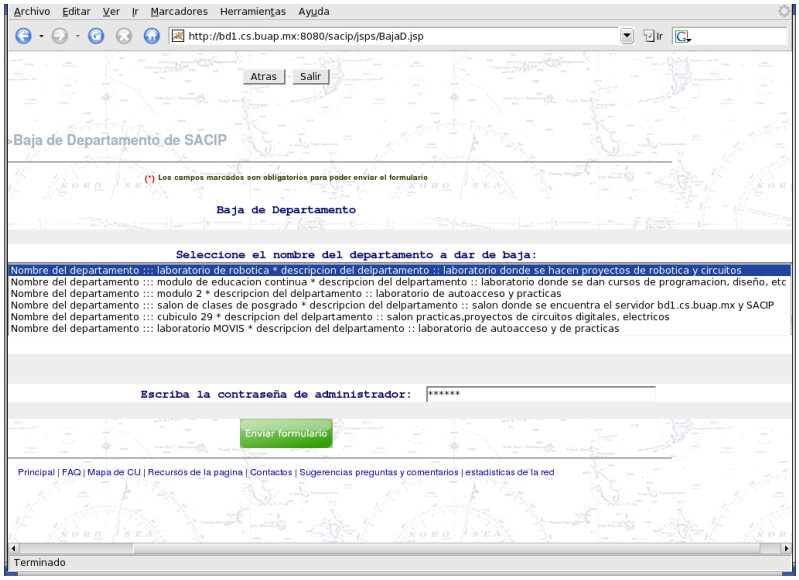


Figura I.V.VII Prueba de baja de departamento de SACIP

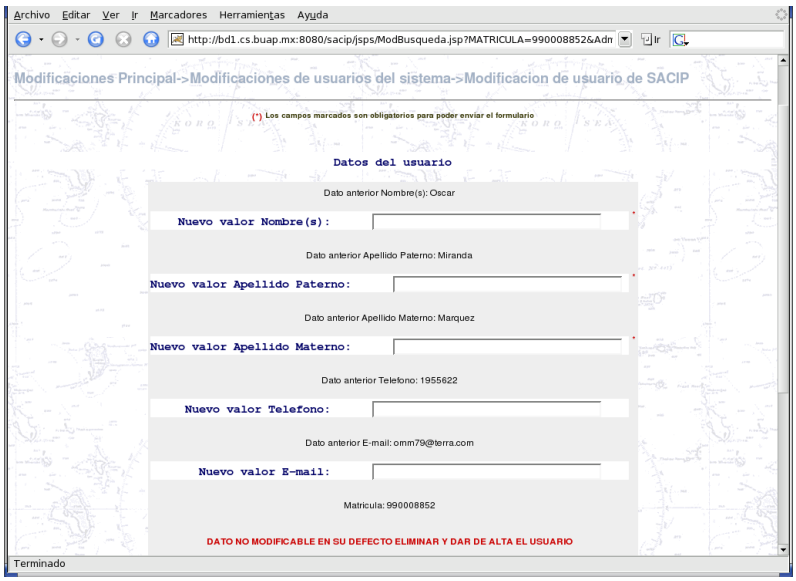


Figura I.V.VIII Prueba de modificación de usuarios de SACIP

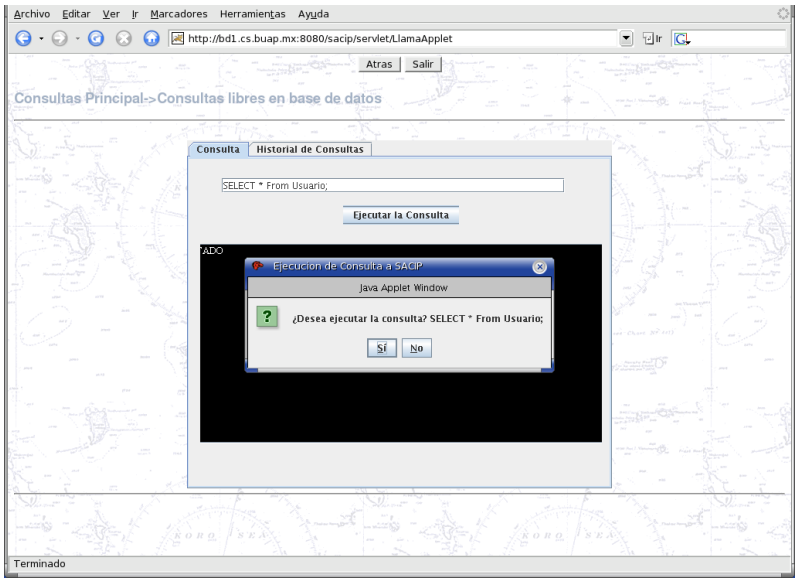


Figura I.V.IX Prueba de consultas libres de SACIP

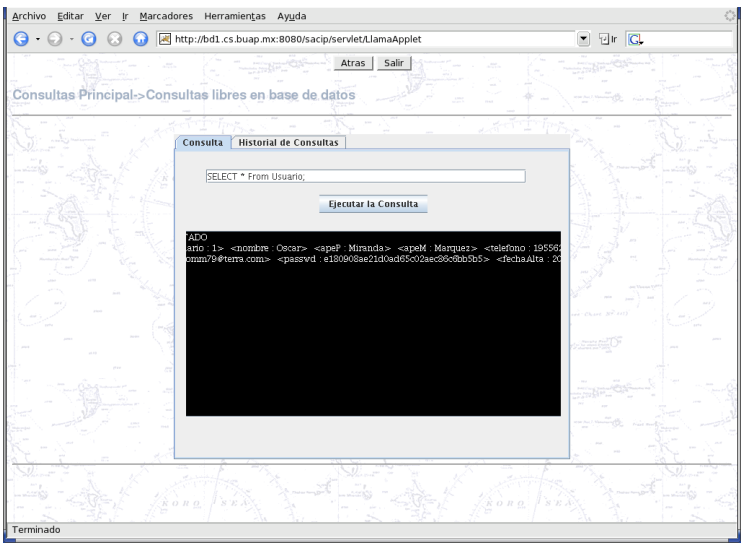


Figura I.V.X Prueba y resultado de consulta libre de de SACIP

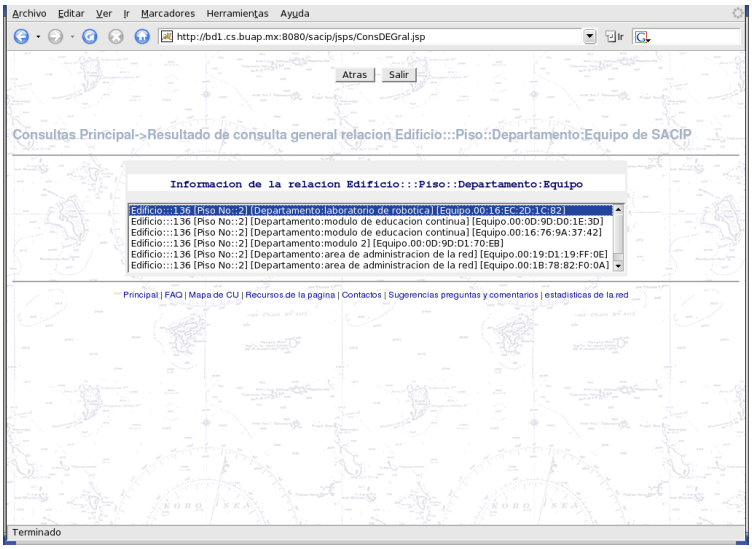
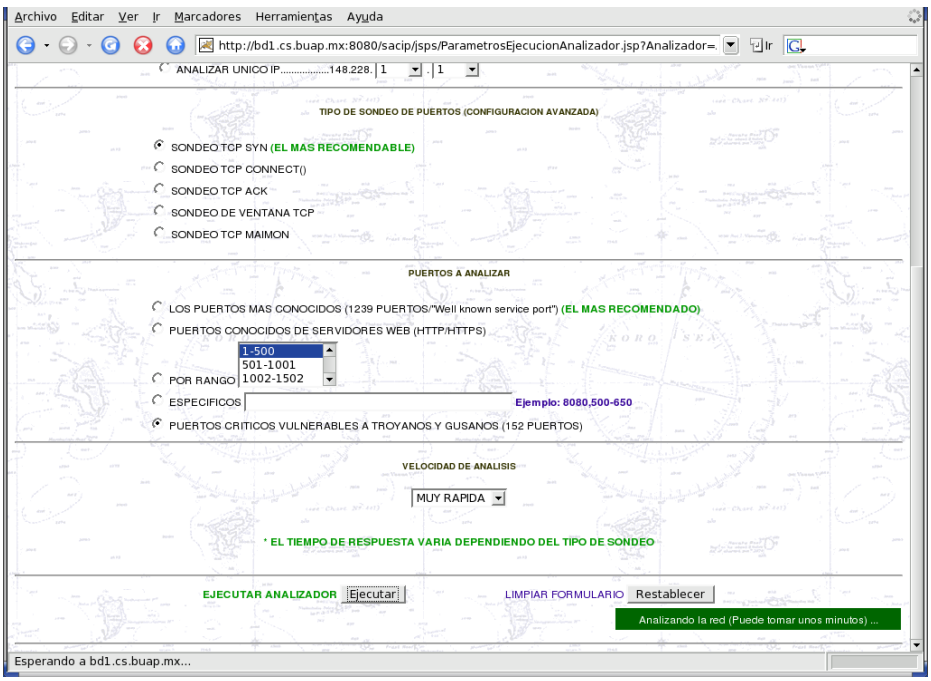
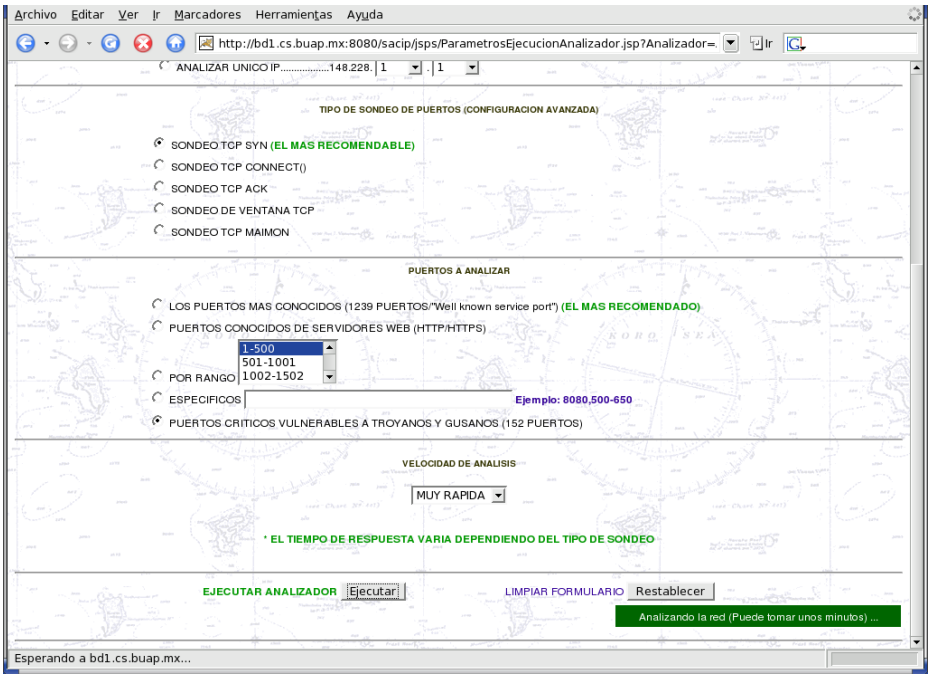
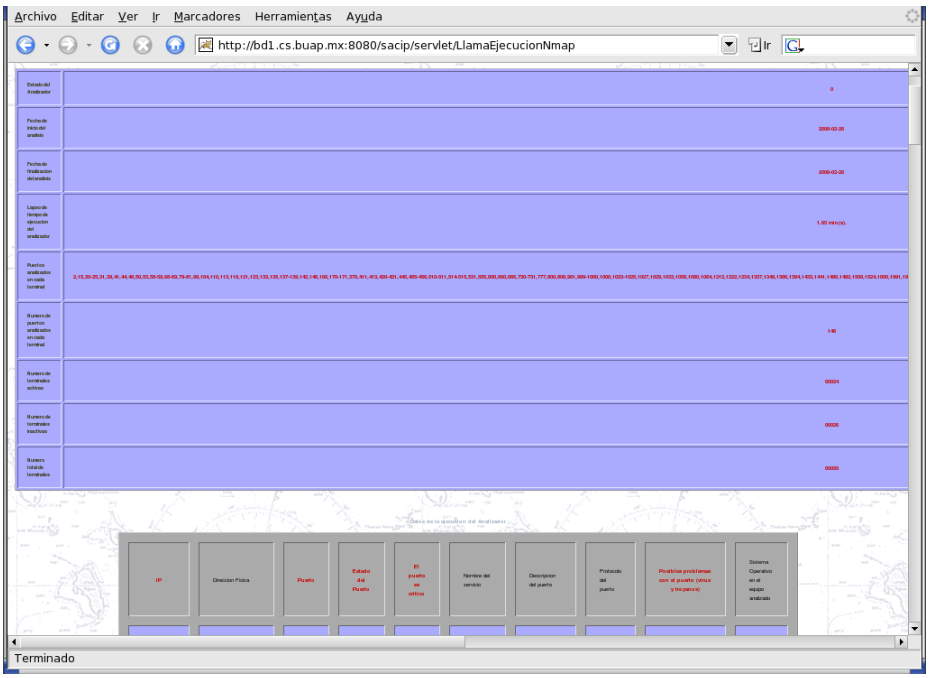


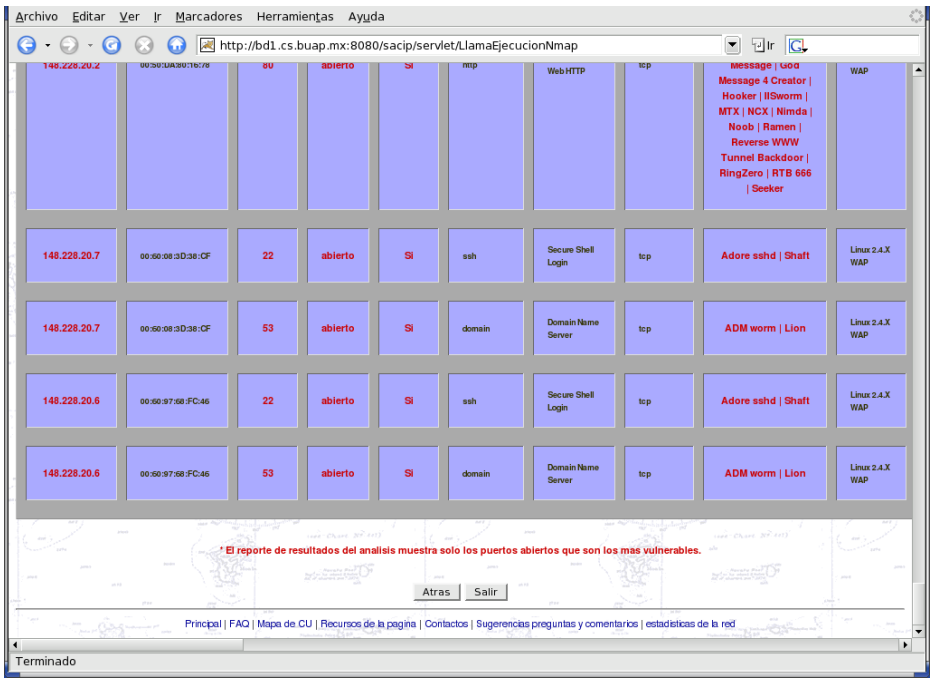
Figura I.V.X Prueba de consulta general de la relación Edificio:::Piso::Departamento:Equipo de SACIP

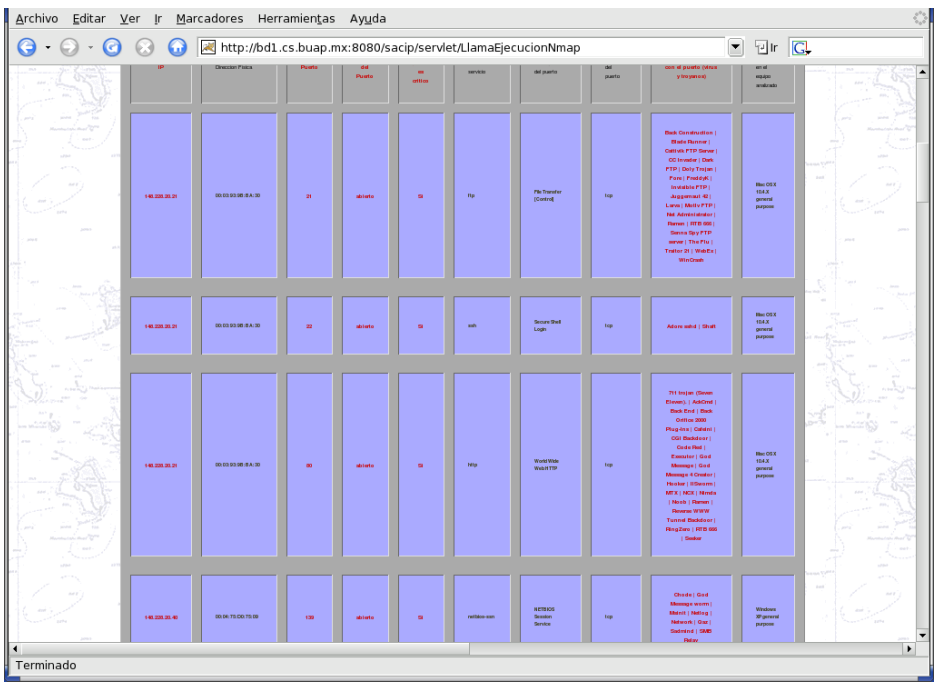
Figuras I.V.XI-XVII Pruebas y resultados del análisis de red.

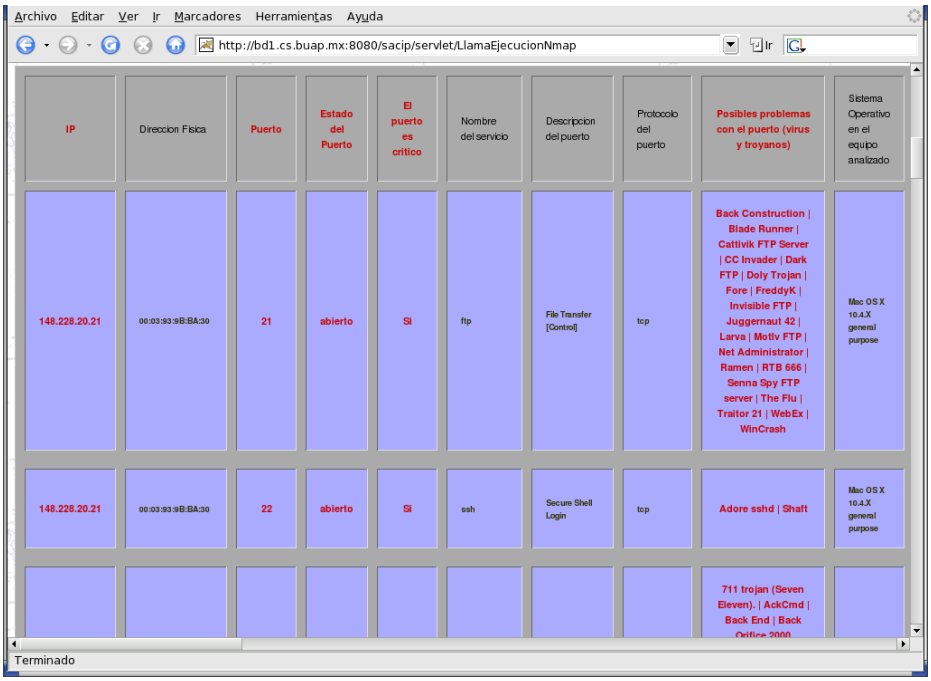


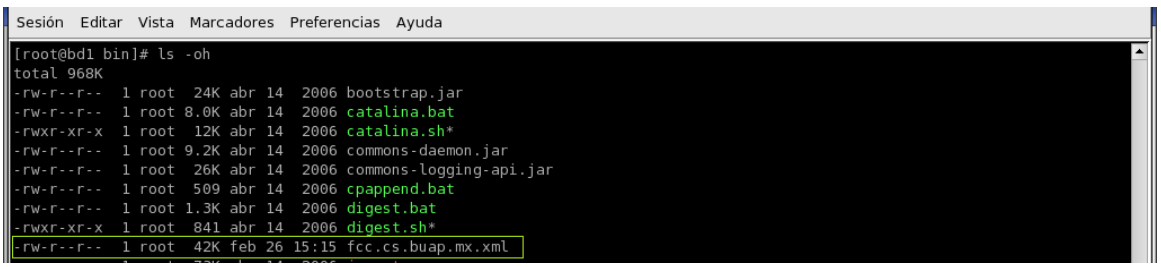

Archivo Editar Ver Ir Marcadores Herramientas Ayuda
http://bd1.cs.buap.mx:8080/sacip/jsps/ParametrosEjecucionAnalizador.jsp?Analizador=
ANALIZAR UNICO IP.....148.228.
TIPO DE SONDEO DE PUERTOS (CONFIGURACION AVANZADA)
SONDEO TCP SYN (EL MAS RECOMENDABLE)
SONDEO TCP CONNECT()
SONDEO TCP ACK
SONDEO DE VENTANA TCP
SONDEO TCP MAIMON
PUERTOS A ANALIZAR
LOS PUERTOS MAS CONOCIDOS (1239 PUERTOS/"Well known service port") (EL MAS RECOMENDADO)
PUERTOS CONOCIDOS DE SERVIDORES WEB (HTTP/HTTPS)
1-500
501-1001
1002-1502
POR RANGO
ESPECIFICOS
Ejemplo: 8080,500-650
PUERTOS CRITICOS VULNERABLES A TROYANOS Y GUSANOS (152 PUERTOS)
VELOCIDAD DE ANALISIS
MUY RAPIDA
* EL TIEMPO DE RESPUESTA VARIA DEPENDIENDO DEL TIPO DE SONDEO
EJECUTAR ANALIZADOR
Ejecutar
LIMPIAR FORMULARIO
Restablecer
Analizando la red (Puede tomar unos minutos) ...
Esperando a bd1.cs.buap.mx...


Archivo Editar Ver Ir Marcadores Herramientas Ayuda
http://bd1.cs.buap.mx:8080/sacip/servlet/LlamaEjecucionNmap
IP
Direccion Fisica
Puerto
Estado del Puerto
El puerto es critico
Nombre del servicio
Descripcion del puerto
Protocolo del puerto
Posibles problemas con el puerto (virus y troyanos)
Sistema Operativo en el equipo analizado
Terminado


Archivo Editar Ver Ir Marcadores Herramientas Ayuda
http://bd1.cs.buap.mx:8080/sacip/servlet/LlamaEjecucionNmap
148.228.20.7 00:60:08:3D:38:CF 22 abierto Si ssh Secure Shell Login tcp Adore sshd | Shaft Linux 2.4.X WAP
148.228.20.7 00:60:08:3D:38:CF 53 abierto Si domain Domain Name Server tcp ADM worm | Lion Linux 2.4.X WAP
148.228.20.6 00:50:97:68:FC:46 22 abierto Si ssh Secure Shell Login tcp Adore sshd | Shaft Linux 2.4.X WAP
148.228.20.6 00:50:97:68:FC:46 53 abierto Si domain Domain Name Server tcp ADM worm | Lion Linux 2.4.X WAP
* El reporte de resultados del analisis muestra solo los puertos abiertos que son los mas vulnerables.
Atras
Salir
Principal | FAQ | Mapa de CU | Recursos de la pagina | Contactos | Sugerencias preguntas y comentarios | estadisticas de la red
Terminado

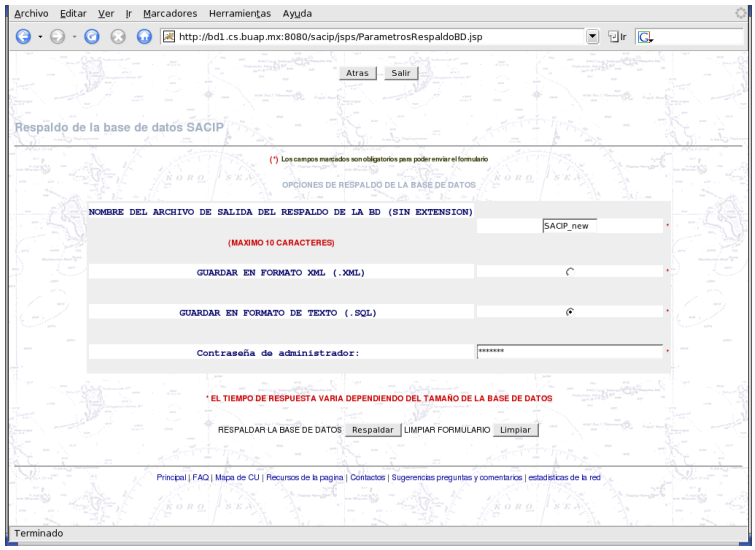

Archivo Editar Ver Ir Marcadores Herramientas Ayuda
http://bd1.cs.buap.mx:8080/sacip/servlet/LlamaEjecucionNmap
Terminado

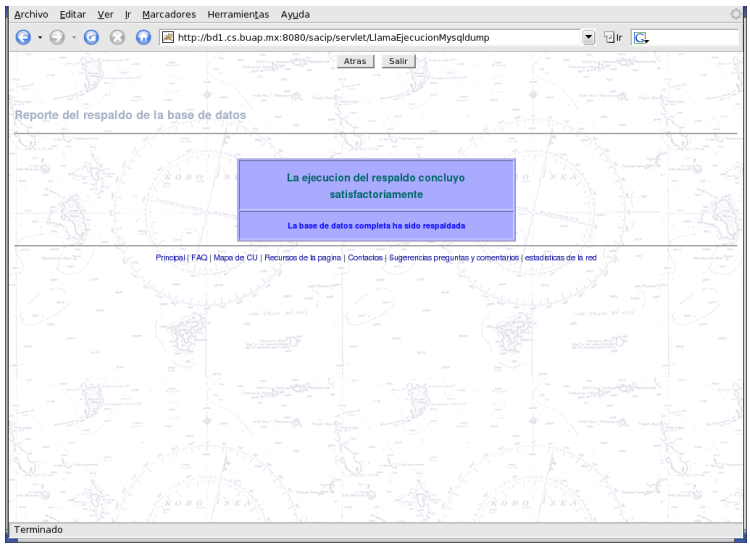


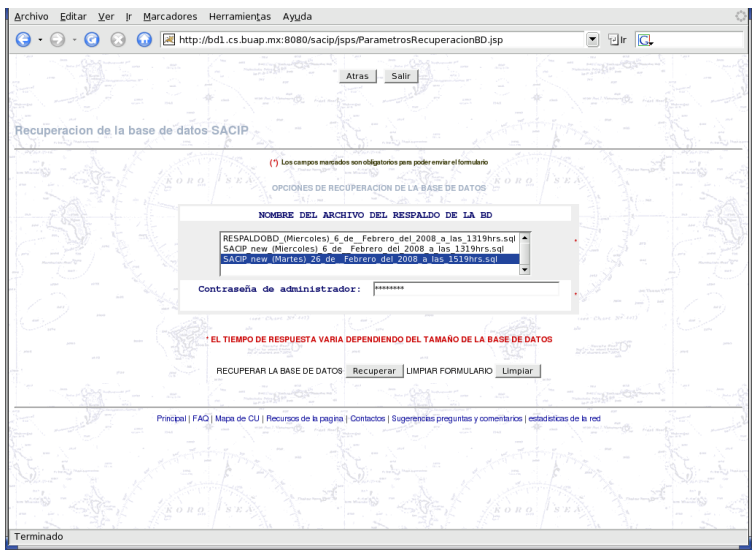


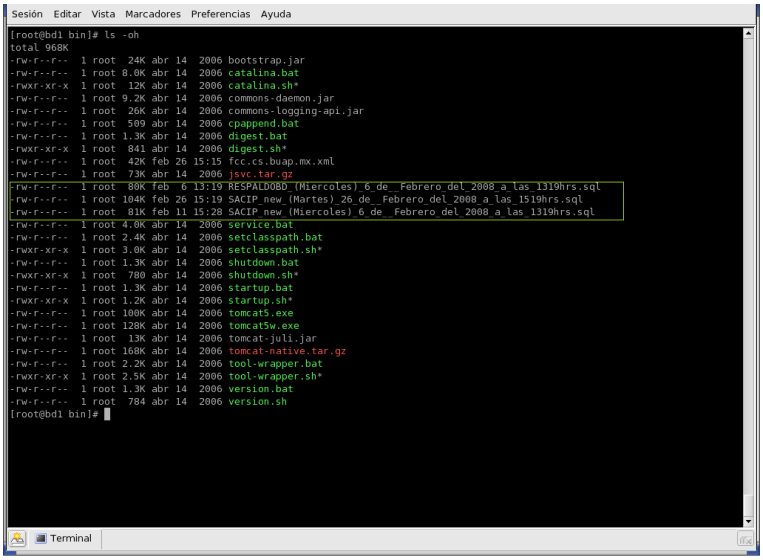
El archivo generado por nmap en formato XML es llamado fcc.cs.buap.mx.xml para su posterior análisis y guardar los datos recuperados en la base de datos.

Figuras I.V.VIII-XI Prueba y resultado del respaldo-recuperación de la base de datos SACIP









Archivos generados de respaldo de la base de datos SACIP para su posterior recuperación, identificados por un nombre, la fecha, la hora.

V.II. Resultados.

Los resultados obtenidos de la implementación del sistema demuestran que el sistema se comporto íntegramente y sin ningún error, sirviendo como apoyo para la solución de problemas de ubicación de equipos, de auditoria de control y la administración de los IPs.

La finalidad de registrar usuarios, edificios, pisos, departamentos, así como la relación de un departamento-equipo es contribuir en la ubicación de los problemas a tiempo con solo analizar los resultados que arroja la red, además tener un control mediante la auditoria continua y consecuentemente una mejor administración de la red.

Al tener un sistema donde la información de la red y sus equipos sean guardados en una base de datos para su posterior análisis es importante cuando existen vulnerabilidades dentro de cada equipo y por consiguiente en la red. Debido a esto el administrador de SACIP debe ser capaz y tener el conocimiento para

decidir cual es la debilidad de su propia red, para realizar esto el administrador puede apoyarse en SACIP, que resulta ser una herramienta potente para quien sabe analizar los datos de resultado.

SACIP se controla a través de administradores que pueden utilizar todas las opciones del sistema como por ejemplo dar de alta a responsables de departamentos los cuales pueden consultar sus IPs o direcciones físicas del segmento donde están ubicados, es decir de sus propios equipos y observar su información general de cada uno, la responsabilidad de cada responsable de departamento recae en que debe saber que vulnerabilidades contienen los equipos que están dentro de su departamento para que este consciente de los problemas que acarrearía dentro de la red; otra opción del administrador es el análisis de la red ya que de esta opción se desprende la mayoría de las actividades que puede desarrollarse dentro del sistema, aparte de las opciones de altas, bajas, modificaciones de usuarios.

Un punto clave para el análisis de la red de un administrador en SACIP es que se pueden actualizar sus definiciones de vulnerabilidades, por medio de la base de datos en la tabla puerto específicamente, ya que esta tabla contiene las vulnerabilidades o puertos críticos de ataques por troyanos y gusanos, en la ultima actualización se obtuvieron 160 puertos que son vulnerables, si se desea actualizar la tabla solo es necesario insertar de forma manual en la base de datos las nuevas definiciones en archivo de texto PUERTOS.sql de la carpeta de instalación de los archivos fuente, añadiendo al final del archivos o insertando directamente en línea de comandos de mysql la siguiente instrucción:

INSERT INTO Puerto VALUES ('monkeycom','9898','tcp',' ',1,'Gusano Dabber');

Donde la descripción de cada campo conforme a la secuencia de inserción anterior es la siguiente:

nombreServicio - Tipo cadena con un máximo de 100 caracteres.
idPuerto - Tipo cadena con un máximo de 100 caracteres.

protocolo - Tipo cadena con un máximo de 10 caracteres.
descripcion - Tipo cadena con un máximo de 300 caracteres.
critico - Tipo numérico con un máximo de 1 carácter,
problema - Tipo cadena con un máximo de 500 caracteres.

Así se asegura que las definiciones de vulnerabilidades de ataques estén actualizadas para informar oportunamente al administrador y al responsable de departamento. La última actualización de puertos vulnerables se puede revisar con la lista completa del archivo “puertos vulnerables.txt” que esta en la carpeta de archivos fuentes, en esta lista las definiciones crecieron considerablemente, para poder evitar problemas con los puertos nuevos se debe realizar un análisis continuo de la red e investigar de nuevos puertos críticos o vulnerables ya que cada cierto tiempo esta lista se ira incrementando dependiendo de la pericia y el pronto descubrimiento de tales vulnerabilidades advirtiendo de los problemas que pueden ocasionar en el equipo y en la red.

El segmento de red administrado por el sistema contiene varios IPs dinámicos es decir se conectan por DHCP y no se mantienen a pesar que se desconecten, así que la variación de cada IP no es conveniente para ser controlados por el sistema ya que la base de datos muestra el resultado del análisis de la red de la relación IP-MACAddress, por lo mismo cualquier cambio puede ser advertido por el sistema y marcarlo como nuevo o como un conflicto de IP o MACAddress, pero no es ningún problema con la ubicación ya que la dirección física o MACAddress es única aunque se cambie de IP, así el cambio de ubicación a otro IP es advertido por el sistema, marcando que hubo un cambio guardándose en el historial de IPs.

SACIP reduce el tiempo para solucionar los problemas que existen dentro de la red optimizando la administración y probando que siempre es necesario tener un control sobre los equipos si se quiere mantener funcionando la red.

V.III. Conclusiones.

El sistema demuestra ser una gran ayuda para ubicar, controlar e informar oportunamente las fallas y vulnerabilidades de la red observadas remotamente, así como un buen funcionamiento en la pruebas realizadas, la administración de usuarios del sistema, el ahorro de tiempo y esfuerzo para el análisis de la red para la pronta solución de los problemas. Además el sistema garantiza a los usuarios un fácil uso debido a su interfaz accesible, intuitiva amigable y seguro ya que el sistema permite el manejo de cuentas de privilegios así los administradores del sistema pueden tener control de los cambios y de los usuarios, la importancia de SACIP reside en que se disminuyen las limitaciones de su funcionalidad, aumenta la eficiencia de su uso y son más accesibles al usuario final.

Los resultados demuestran que la agilización de la solución del problema es una de las mayores ventajas del sistema ya que anteriormente se tenia que buscar donde estaba ubicado el equipo y desconectar o explorar cual era el problema del mismo, así la facilidad de saber el problema y ubicarlo ahorra tiempo y dinero ya que la vulnerabilidad puede acarrear perdida de información, un mal uso de la misma o la caída total de la red.

Esta solución puede ser extendida o ampliada con mejoras y servicios ya que su gran compatibilidad con los diferentes tipos de plataformas lo hace adaptarse a varios usos dentro de la facultad, en toda la BUAP o en una empresa.

V.IV. Bibliografía y Referencias.

V.IV.I. Bibliografía:

[1] "Redes de Computadoras", Tanenbaum, Andrew S. Cuarta Edición. México: Pearson Prentice Hall. 2003.

[2] "Instalación y mantenimiento de servicios de redes locales.", MOLINA, F.J., Julio 2004. E. Rústica.

[3] "Desarrollo web con JSP", Ben Galbraith; Jayson Falkner; Romin Irani, E.Anaya Multimedia,1ª edición, 2002.

[4] "Java Servlet Programming", Jason Hunter with William Crawford. E.O'Reilly, 2001.

[5] "Core Servlets and Java Server Pages (JSP)", Marty Hall, Prentice Hall, 2da edición, 2003.

[6] ”UML para programadores Java”. ROBER C. MARTIN, Pearson Education - Prentice Hall, 2004.

[7] ”MySQL para Windows y Linux”, Pérez, César, E. Ra-Ma, Librería, 2004.

[8] “Fundamentos de Sistemas de Bases de Datos”, ELMASRI-NAVATHE, Pearson Education, Addison Wesley, 3ª. Edición,

1 Definición de Base de datos, “Fundamentos de sistemas de bases de datos”, 1.1 Introducción, Pág.4, P.2, 3 y 4, Aut. Elmasri-Navate, EDT. Addison Wesley, 3ª ED.
2 Definición de Sistema de gestión de base de datos, “Fundamentos de sistemas de bases de datos”, 1.1 Introducción, Pág.5, P.3, Aut. Elmasri-Navate, EDT. Addison Wesley, 3ª ED.
3 Múltiples vistas o Datos virtuales y procesamiento de transacciones multiusuarios, “Fundamentos de sistemas de bases de datos”, 1.3.3. , Soporte de múltiples vistas de los datos Pág.10, 11, Aut. Elmasri-Navate, EDT. Addison Wesley, 3ª ED.

[9] “Designing with JavaScript”. Henli, Nick, EUA, O’Reilly, 2002.

V.IV.II. Referencias:

[1]"Java" (sitio oficial) - http://www.java.sun.com/

[2] APACHE - TOMCAT - http://tomcat.apache.org/
APACHE - http://www.apache.org/

[3] MySQL - http://www.mysql.com/

[4] Libpcap - http://www.tcpdump.org/

[5] Winpcap - http://www.winpcap.org/

[6] Nmap - http://www.insecure.org/nmap/
http://nmap.org

[7] Graham, Robert. Robert Graham, Sniffing (network wiretap, sniffer) FAQ, Versión 0.3.3, publicado el 14 de Septiembre del 2000 mirror - http://gd.tuwien.ac.at/infosys/security/sniffing-faq.html [Accesado el 13 de Marzo de 2007].

[8] Log4j - http://logging.apache.org/

[9] XML - http://xerces.apache.org/xerces2-j/
W3C - http://www.w3.org/TR/REC-xml/

[10]"LGPL Lesser General Public License" (Licencia Pública General Menor) - http://www.gnu.org/licenses/lgpl.html

[11]"Apache License, Version 2.0, January 2004" (Licencia Apache) - http://www.apache.org/licenses/

[12]“El Proceso Unificado de Desarrollo Software”, I. Jacobson, Grady Booch, J. Rumbaugh, Ed. Addison Wesley

[13]“Como proteger nuestro pc”, http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=281

[14]“Virus Informáticos, Caballos de Troya, Gusanos”, http://softwareyprogramas.blogspot.com/2007/06/virus-informticos-caballos-de-troya.html

[15]“Ordenadores convertidos en zombis”, http://www.cybernautas.es/?articulo=virus_seguridad&id=51

IV.V. Glosario y anexos.

Mapear o mapping .- Significa que las redes, IPs o puertos se conocen por medio de una tabla de relaciones entre su ubicación lógica y su dirección física.

DBMS .- Sistema de gestión de bases de datos (DataBase Management System), es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos.

Multihilo .- El paradigma de multihilo ha llegado a ser más popular a medida que los esfuerzos para llevar más adelante el paralelismo a nivel de instrucción se han atascado desde finales de los años 1990. Esto permitió que re-emergiera a una posición destacada el concepto de la computación de rendimiento a partir del más especializado campo del procesamiento transaccional:

Aunque es muy difícil acelerar un solo hilo o un solo programa, la mayoría de los sistemas de computadores son realmente multitarea entre múltiples hilos o programas.

Las técnicas que permitirían acelerar el rendimiento total del procesamiento del sistema en todas las tareas (tasks) darían como resultado un aumento significativo del rendimiento.

Router .- Típicamente una máquina, aunque también puede ser un software, que actúa como puerta para permitir el acceso a los recursos de una red, independientemente de los protocolos o sistemas operativos de los usuarios.

Sniffers .- Literalmente "Husmeador". Pequeño programa que busca una cadena numérica o de caracteres en los paquetes que atraviesan un nodo con objeto de conseguir alguna información. Normalmente su uso es ilegal.

Handshake (SSL) .- Literalmente “apretón de manos” que es el proceso que realiza el protocolo SSL al empezar una conexión con el cliente.

Unix .- Unix (registrado oficialmente como UNIX®) es un sistema operativo portátil, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

Datagramas .- Un datagrama es un fragmento de paquete que es enviado con la suficiente información como para que la red pueda simplemente encaminar el fragmento hacia el equipo terminal de datos receptor, de manera independiente a los fragmentos restantes.

Host .- El término host (*equipo anfitrión*) puede referirse a:

- A una máquina conectada a una red de ordenadores y que tiene un nombre de equipo (en inglés, hostname).
- Es un nombre que se le da a un dispositivo conectado a una red informática. Puede ser un ordenador, un servidor de archivos, un dispositivo de almacenamiento por red, una máquina de fax, impresora, etc.
- Este nombre ayuda al administrador de la red a identificar las máquinas sin tener que memorizar una dirección IP para cada una de ellas.
- Por extensión, a veces también se llama así al dominio del equipo.
- También es el nombre de un fichero denominado fichero host que se encuentra en los ordenadores y resuelve algunos DNS.

Ftp .- Son las siglas de File Transfer Protocol, el nombre del protocolo estándar de transferencia de ficheros. Su misión es permitir a los usuarios recibir y enviar

ficheros de todas las máquinas que sean servidores FTP. El usuario debe disponer del software que permita hacer la transferencia (actualmente todos los navegadores, ya disponen de ese software para recibir ficheros). Los ficheros pueden ser documentos, textos, imágenes, sonidos, programas, etc., es decir, cualquier cosa que se pueda almacenar en un fichero o archivo. En Internet hay miles de ordenadores con centenares de ficheros de todas las clases a los que el público tiene acceso.

DHCP .- Dynamíc Host Confíguration Protocol). Protocolo de configuración dinámica de host. Protocolo que usan las computadoras para obtener información de configuración. El DHCP permite asignar una dirección IP a una computadora sin requerir que un administrador configure la información sobre la computadora en la base de datos de un servidor.

IPng o IPv6 .- Versión 6 del protocolo de internet (IP). Es un protocolo encargado de dirigir los paquetes a través de una red, especialmente Internet. Fue diseñado por Steve Deering de Xerox PARC y Craig Mudge.

IPv6 fue diseñada para sustituir la versión actual (IPv4) que tiene grandes limitaciones, especialmente un limitado número de direcciones de red posibles. IPv6 soporta 340.282.366.920.938.463.463.374.607.431.768.211.456 (2 elevado a 128) de direcciones, mientras que IPv4 sólo 4.294.967.296 (2 elevado a 32).

El uso de IPv6 ha sido frenado temporalmente por el uso de la traducción de direcciones de red (NAT), que alivia parcialmente el problema del faltante de direcciones IP. El problema es que NAT hace difícil o imposible el uso de voz sobre IP (VoIP), los juegos multiusuarios y las aplicaciones P2P.

Se estima que IPv4 seguirá funcionando hasta 2025, por la falta de renovación de dispositivos que sólo funcionan con este protocolo.

Un ejemplo de una dirección IP en versión 6 es:

2001:0db8:85a3:08d3:1319:8a2e:0370:7334

Switch .- El switch (palabra que significa “conmutador”) es un dispositivo que permite la interconexión de redes sólo cuando esta conexión es necesaria. Para entender mejor que es lo que realiza, pensemos que la red está dividida en segmentos por lo que, cuando alguien envía un mensaje desde un segmento hacia otro segmento determinado, el switch se encargará de hacer que ese mensaje llegue única y exclusivamente al segmento requerido.

De esta manera, el switch opera en la capa 2 del modelo OSI, que es el nivel de enlace de datos, y tienen la particularidad de aprender y almacenar las direcciones (los caminos) de dicho nivel, por lo que siempre irán desde el puerto de origen directamente al de llegada, para evitar los bucles (habilitar mas de un camino para llegar a un mismo destino). Asimismo, tiene la capacidad de poder realizar las conexiones con velocidades diferentes en sus ramas, variando entre 10 Mbps y 100 Mbps.

Se puede decir que es una versión mejorada del hub ya que, si bien tienen la misma función, el switch lo hace de manera más eficiente: se encargará de encaminar la conexión hacia el puerto requerido por una única dirección y, de esta manera, produce la reducción del tráfico y la disminución de las coaliciones notablemente, funciones fundamentales por las cuales se originó este dispositivo.

Broadcast .- Un Broadcast es un método de comunicación donde un dispositivo envía un único paquete de datos direccionado a cada uno de los dispositivos en una red, ejemplo, comunicación de uno a todos.

Pinging .- (Packet INternet Groper - Rastreador de Paquetes Internet). Programa que es empleado para verificar si un host o servidor está disponible (conectado, en funcionamiento o activo).

Para comprobarlo envía paquetes de datos, si el servidor remoto responde significa que está activo.

SSL .- (Secure Sockets Layer). Protocolo diseñado por la empresa Netscape para proveer comunicaciones encriptadas en internet.

La empresa VeriSign es la encargada de emitir los certificados digitales RSA para su uso en transmisiones seguras por SSL, especialmente para la protección de sitios con acceso por HTTPS. Por ejemplo, páginas que utilizan tarjetas de créditos.

SSL da privacidad para datos y mensajes, además permite autentificar los datos enviados.

Otro protocolo que se emplea para la transmisión de datos seguros en la WWW es el SHTTP, y puede complementarse con SSL. La principal diferencia con SSL radica en que SSL crea una conexión segura entre el cliente y el servidor web, en esa conexión se pueden enviar cualquier cantidad de datos de forma segura. En tanto, SHTTP está diseñado para transmitir mensajes individuales de forma segura.

TLS .- Transport Layer Security - Seguridad para Capa de Transporte. Versión estándar de la IETF del protocolo SSL que pretende abarcar la capa de transporte del modelo OSI.

Bridges .- Aunque se utiliza también el término puente, es bastante usual encontrar la palabra bridge para designar un dispositivo que conecta dos o más redes físicas que utilizan el mismo protocolo de comunicaciones y encamina paquetes de datos entre ambas.

Gateways .- Puerta de enlace, acceso, pasarela. Nodo en una red informática que sirve de punto de acceso a otra red. Dispositivo dedicado a intercomunicar sistemas con protocolos incompatibles.

Logs .- Registro oficial de eventos durante un período de tiempo en particular. Para los profesionales en seguridad informática un log es usado para registrar datos o información sobre quien, que, cuando, donde y por que (who, what, when, where y why, W5) un evento ocurre para un dispositivo en particular o aplicación.

La mayoría de los logs son almacenados o desplegados en el formato estándar, el cual es un conjunto de caracteres para dispositivos comunes y aplicaciones. De

esta forma cada log generado por un dispositivo en particular puede ser leído y desplegado en otro diferente.

También se le considera como aquel mensaje que genera el programador de un sistema operativo, alguna aplicación o algún proceso, en virtud del cual se muestra un evento del sistema.

Tcpdump .- Es una herramienta de diagnóstico para redes TCP/IP basada en salida textual, que monitoriza los paquetes que entran y salen de una interfaz de red, y los presenta en formato legible, comúnmente denominado sniffer. No es intrínsecamente peligroso, y es de gran utilidad para los administradores de redes que necesiten ver el tráfico para poder buscar problemas en una red. Permite usar filtros mediante una expresión de búsqueda, y solo mostrará los paquetes cuya cabecera coincida con ella. También puede ser usada para la ingeniería inversa de protocolos de red.

Ethercap .- Es una utilidad que nos permite capturar el trafico que circula por una LAN, ya sea en un ambiente switcheado (lo crean o no) o HUBeado. O sea, es un sniffer.

Nos provee de dos modos de funcionamiento: INTERACTIVO y NO-Interactivo. Todo se controla por letras-comando, mas los cursores y el enter, y en cada pantalla del modo interactivo pueden utilizar el comando 'h', para obtener un breve listado de comandos en el área actual del programa. Para salir o volver atrás, pueden utilizar 'q'.

Firewall .- Un cortafuegos (o firewall en inglés), es un elemento de hardware o software utilizado en una red de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas según las políticas de red que haya definido la organización responsable de la red.

Parseador .- Un analizador sintáctico (parser, en inglés) en informática y linguística es un proceso que analiza secuencias de tokens para determinar su estructura gramatical respecto a una gramática formal dada.

Open source .- (Open source). Denominación para aquellas aplicaciones que tienen su código fuente liberado. En general, los programas de código abierto suele ser libres. Aunque existen aplicaciones de código abierto que no son libres.

Open Source es utilizado también para hacer referencia a un nuevo movimiento de software, la Open Source Initative.

Entunelado .- HTTP tunneling encapsulamiento dentro de HTTP (de algún protocolo)

La expresión se refiere al encapsulamiento de algún protocolo (que aquí no se menciona) dentro del Protocolo para Transferencia de Hipertexto, mejor conocido como HTTP.

Interoperar .- La interoperatividad es la condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos. Interoperatividad de la Web como una condición necesaria para que los usuarios (humanos o mecánicos) tengan un acceso completo a la información disponible.

Debugger .- Un depurador (en inglés, debugger),es un programa que permite depurar o limpiar los errores de otro programa informático.

Preprocesador .- Un preprocesador es un programa separado que es invocado por el compilador antes de que comience la traducción real. Un preprocesador de este tipo puede eliminar los comentarios, incluir otros archivos y ejecutar sustituciones de macros.

Logging .- Es el proceso de guardar información acerca de los eventos que ocurren en la red o en un cortafuegos.

Granularidad .- Nivel de modularidad de un sistema. Módulos más pequeños indican una mayor flexibilidad.

Html .- Es el acrónimo inglés de HyperText Markup Language, que se traduce al español como Lenguaje de Etiquetas de Hipertexto . Es un lenguaje de marcado

diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

SGML (Standard Generalized Markup Language) .- SGML son las siglas de "Lenguaje de Marcación Generalizado". Consiste en un sistema para la organización y etiquetado de documentos. La Organización Internacional de Estándares (ISO) ha normalizado este lenguaje en 1986.

Metacontenido .- Un metacontenido es la información relativa al contenido del documento, como su título, autor, amaño del archivo, fecha de creación, historial de cambios, palabras clave, y demás información asociada. Se puede utilizar un metacontenido, por ejemplo, para realizar búsquedas, filtrar información y gestionar el documento.

Callback .- Un Callback es un código ejecutable que es pasado cómo un argumento a otro código. Esto permite al software de la capa de bajo nivel llamar a una subrutina (función) definida en una capa de alto nivel.

Applets .- Pequeña aplicación escrita en Java la cual se difunde a través de la red en orden de ejecutarse en el navegador cliente.

Scriptlet .- Son fragmentos de código Java dentro de una pagina JSP

Encripta .- Cifrado. Proceso para volver ilegible información considera importante. La información una vez encriptada sólo puede leerse aplicándole una clave. Se trata de una medida de seguridad que es usada para almacenar o transferir información delicada que no debería ser accesible a terceros. Pueden ser contraseñas, números de tarjetas de crédito, conversaciones privadas, etc.

Fingerprint .- Huella digital.

On-line .- En línea, conectado a internet.

Funciones hash .- Una herramienta fundamental en la criptografía, son las funciones hash, son usadas principalmente para resolver el problema de la integridad de los mensajes, así como la autenticidad de mensajes y de su origen. Una fun-

ción hash es también ampliamente usada para la firma digital, ya que los documentos a firmar son en general demasiado grandes, la función hash les asocia una cadena de longitud 160 bits que los hace más manejables para el propósito de firma digital.

Encapsulación .- Es el proceso por el cual los datos que se deben enviar a través de una red se deben colocar en paquetes que se puedan administrar y rastrear. El encapsulado consiste pues en ocultar los detalles de implementación de un objeto, pero a la vez se provee una interfaz pública por medio de sus operaciones permitidas. Considerando lo anterior también se define el encapsulado como la propiedad de los objetos de permitir el acceso a su estado únicamente a través de su interfaz o de relaciones preestablecidas con otros objetos.

Run-time .- En tiempo de ejecución.

Patch panel .- Son estructuras metálicas con placas de circuitos que permiten la interconexión entre equipos. Un patch panel posee una determinada cantidad de puertos rj-45, donde cada puerto se asocia a una placa de circuito, la cual a su vez se propaga en pequeños conectores de cerdas.

Fibra óptica monomodo .- Una fibra monomodo esta optimizada en segunda ventana (1.300 nm. de longitud de onda).

Backbone .- Un backbone se puede definir como un enlace de gran caudal o como un sinnúmero de nudos de conexión que forman un eje de conexión principal. Es la columna vertebral de una red.

Actualmente, hay muchos backbones, que llevan grandes cantidades de tráfico entre continentes o a través de los Estados Unidos o Europa, o en un mismo país. La mayoría de las redes en los países en desarrollo tienen conexiones directas por tierra, mar o satélite a un backbone de Internet de los Estados Unidos o Europa.

Autoacceso .- Una de las características principales de los centros de autoacceso es la utilización de equipos de video, audio y cómputo conectados a Internet y al correo electrónico, para realizar cursos a distancia, obtener información actuali-

zada de diversas disciplinas, resultados de los trabajos de investigación y publicaciones recientes.

Rational rose .- Es una de las herramientas de modelado visual para el analisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo.

Zip .- En informática, ZIP o zip es un formato de almacenamiento muy utilizado para la compresión de datos como imágenes, música, programas o documentos. Para este tipo de archivos se utiliza generalmente la extensión ".zip". Muchos programas, tanto comerciales como libres, lo utilizan y permiten su uso más habitual.

Frames .- Cuadros ente los que se divide una página para hacerla más atractiva y ordenada.

Anexo 1.

Notas importantes y legales sobre Nmap:

NMAP

Clasificación de control de exportación de los EEUU

Control de exportación de los EEUU: Insecure.Com LLC cree que Nmap se encuentra dentro del capítulo US ECCN (número de clasificación de control de exportación) 5D992. Esta categoría se denomina “Programas de seguridad de la información no controlados en 5D002”. La única restricción a esta clasificación es AT (anti-terrorismo), que se aplica a casi todos los bienes y deniega la exportación a un número reducido de naciones rebeldes como Irán o Corea del Norte. Así, la exportación de Nmap no requiere de una licencia especial, permiso o cualquier otra autorización del gobierno.

Importante leer también las notas legales de la fuente:

http://nmap.org/man/es/man-legal.html#translation-disclaimer