Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas

INSTALACIÓN Y CONFIGURACIÓN DE UN SERVIDOR DE AUTENTICACIÓN VÍA PORTAL CAUTIVO MEDIANTE EL SISTEMA NOCAT AUTH SOBRE LINUX

Monografía previa a la obtención del título de Ingeniero de Sistemas

Autores:
María Paula Barros Andrade
Joseph Gregorio Cobos Castro

Director: Ing. Rubén Ortega

Cuenca, Ecuador
2006

La responsabilidad por los hechos, ideas y pensamientos expuestos en la presente monografía, corresponden exclusivamente a sus autores.

Joseph Cobos Castro *Ma. Paula Barros A.*

DEDICATORIA:

Dedico este trabajo de graduación: a Dios, que estuvo siempre a mi lado en todo momento, y a quien le debo toda mi vida; a mis padres Mario Cobos y Luisa Castro y a mis hermanos Johann, Fabricio, Mayta, y Lady que pusieron todo su esfuerzo y cariño para llegar a donde estoy; y en agradecimiento a todas sus enseñanzas y sus consejos para ser una persona de bien. Para Uds. padres y hermanos queridos y para ti Dios todopoderoso con todo el amor de mundo.

Joseph Cobos Castro.

Este trabajo lo dedico a mis padres que siempre estuvieron a mi lado, brindándome su apoyo y cariño incondicional. Los dedico por que gracias a su esfuerzo y confianza hicieron posible que alcance mi sueño.

Ma. Paula Barros Andrade.

AGRADECIMIENTO:

Agradezco a todas las personas que de una u otra forma intervinieron y colaboraron en el desarrollo del presente trabajo de grado. A mis profesores, que a lo largo de mi carrera supieron compartir sus conocimientos y brindarme su amistad; en especial al Ing. Pablo Esquivel, Ing. Oswaldo Merchán y a mi director, el Ing. Rubén Ortega.

Agradezco también a la familia Barros Andrade, que supieron brindarme todo su apoyo y confianza para el desarrollo de todo este trabajo.

Joseph Cobos Castro.

Primeramente agradezco a Dios porque siempre esta a mi lado dándome las fuerzas para seguir, a mis padres por ser mis “Mejores Amigos” que con su ejemplo de lucha y constancia me enseñaron a ser perseverante para así alcanzar una meta. Gracias padres por ser tan buenos y haber confiado en mí.

A mis hermanos que de una u otra forma siempre estuvieron a mi lado apoyándome.

Además agradezco a todos los distinguidos Catedráticos de la Facultad, que con su sabiduría me inculcaron y guiaron en el transcurso de esta carrera; en especial al Ing. Pablo Esquivel, Ing. Rubén Ortega e Ing. Oswaldo Merchán.

Ma. Paula Barros Andrade.

INDICE DE CONTENIDOS

INDICE DE ANEXOS

RESUMEN

Debido al fácil acceso hacia el Internet por parte de un usuario de la red, se ha visto la necesidad de contar con una mayor seguridad y de asignar permisos para evitar el mal aprovechamiento de este medio.

Mediante el desarrollo del proyecto se pretende configurar un servidor de autenticación vía portal cautivo que nos permita tener una mayor seguridad en la red, para ello realizaremos la instalación y configuración del software NoCat Auth, y especificaremos los pasos e instrucciones necesarios.

El Portal Cautivo captura las peticiones de usuario a una Web en la que se solicita el login y contraseña, las comprueba ante una base de datos para permitir o denegar el acceso a la red. Todas estas tareas se llevarán a cabo en un Servidor Linux que cuente con los requerimientos previos para el correcto funcionamiento del software NoCat Auth.

NoCat Auth es un software que controla el acceso de los usuarios de una red hacia el Internet, mediante Autenticación Vía Portal Cautivo y basada en SSL (*Secure Sockets Layer*). Para la creación, eliminación y listado de los usuarios en la Base de Datos MySQL, NoCat Auth cuenta con un *script* llamado “admintool”.

Además NoCat Auth permite la configuración y asignación de ancho de banda por los grupos de usuarios *owner, public y member*.

Sus principales archivos de configuración son: /usr/local/nocat/nocat.conf y /usr/local/nocat/gateway/nocat.conf.

ABSTRACT

Due to easy Internet access on the net by the user, it has become more inevitable the need to increase security and to assign security codes to avoid bad use of this media.

Through the development of this project, our goal is to make an authentication server by way of a captive portal which allows us to have more security on the net and to achieve this we will carry out the installation and configuration of the software NoCat Auth and we'll specify the necessary steps and instructions.

The captive portal receives the pleas of the user to a Web where a login and password is requested, and then it is confirmed through a Data Base to be able to permit or reject access to the net. All of these duties are carried out on the server Linux which has the previous requirements for the correct functioning or the software NoCat Auth.

NoCat Auth is software that controls the user's access on a net to the Internet by means of Captive Portal Authentication, which is based on Security Sockets Layer (SSL). NoCat Auth has a script called admintool which creates, eliminates and enlists users on MySQL Data Base.

NoCat Auth also let's us make and assign band width by group users owner", "public" and "member".

The main configuration files are: /usr/local/nocat/nocat.conf and /usr/local/nocat/gateway/nocat.conf.

INTRODUCCIÓN

Debido al gran avance tecnológico vivido en estos últimos años y a la necesidad cada vez más grande de comunicarse entre personas que se encuentran en diferentes lugares físicos, se implementan las telecomunicaciones y con ellas las redes de computadoras. Conforme pasa el tiempo las exigencias de los usuarios de dicha tecnología son más grandes, en la actualidad se encuentran preocupados por la seguridad de sus datos es por ello que se ha empezado a desarrollar sistemas que permitan la autenticación de los usuarios en la red, garantizando mayor confiabilidad y confidencialidad en la información para evitar la pérdida o alteración de la misma, ya que estos son considerados los recursos más importantes de una empresa. Existen varios productos de *software* que permiten la autenticación, nosotros profundizaremos el sistema NoCat Auth debido a los beneficios que ofrece, evitando de esta forma el mal aprovechamiento de los recursos de red, que en muchas ocasiones son asignados a personas que no lo requieren o no los utilizan para beneficio de la empresa, causando grandes pérdidas de dinero.

El desarrollo de este trabajo nos ayudará a reafianzar los conocimientos adquiridos durante todos estos años de estudio, además cumpliremos la meta propuesta desde el inicio de nuestra carrera, logrando el sueño de realizarnos profesionalmente después de entregar todos nuestros esfuerzos para lograrlo.

La metodología que usaremos para la elaboración de esta monografía consiste en dos partes: una parte teórica en la cual trataremos conceptos básicos, y la otra parte consiste en aplicar dichos conceptos mediante la “Instalación y Configuración de un Sistema de Autenticación Vía Portal Cautivo mediante el sistema NoCat Auth sobre Linux”, ya que con la práctica se afianzan los conceptos teóricos y se aclaran muchas dudas existentes. Lo hemos dividido de esta forma ya que creemos que las dos partes son complementarias, e importantes para tener un buen nivel de aprendizaje sobre dicho tema.

PARTE I: AUTENTICACIÓN VÍA PORTAL CAUTIVO PARA UNA RED INALÁMBRICA

CAPITULO 1: SEGURIDAD

Introducción

En el manejo de permisos de acceso al servidor desde las terminales de los clientes se debe tener especial cuidado con el aspecto de las seguridades, determinar las posibles amenazas, vulnerabilidades y ataques que se pueden presentar en nuestro servidor y buscar las correspondientes soluciones mediante mecanismos de autenticación, privacidad e integridad, brindando así una mayor seguridad.

1.1 Amenazas, vulnerabilidades y ataques

1.1.1 Amenazas

Una amenaza, es cualquier situación del entorno del sistema de información, sea esta: persona, máquina, o suceso que podría dar lugar a una violación de la seguridad (confidencialidad, privacidad e integridad). Para la mayoría de las empresas es de gran importancia la protección de su información de los ataques externos, por lo que es prioritario fortalecer la seguridad contra estos, aunque muchos de los peligros también pueden darse dentro de las mismas; esto es considerado como la amenaza de los intrusos internos.

Según el Centro de Coordinación CERT (*Computer Emergency Response Teams* - Equipo de Respuesta para Emergencias Informáticas) de la Universidad Carnegie Mellon, un intruso interno es la persona que pone en riesgo una red, sistema o base de datos, asignado por una persona que tiene o tenía permisos de acceso a la red, sistema o información. Las amenazas de los intrusos internos resultan costosas para la organización debido a que éstos poseen un mayor conocimiento del lugar donde se encuentra la información más importante y confidencial. Las amenazas pueden ser intencionales o no.

Las amenazas internas también pueden ser el acceso indebido al Internet, así como los problemas que pueden producirse porque los empleados envían y revisan materiales ofensivos a través de Internet.

1.1.2 Vulnerabilidades

Las vulnerabilidades son fallas que ponen en riesgo la seguridad de un programa o sistema. Generalmente son producidas por errores en programación y que pueden ser utilizados para ejecutar código con fines maliciosos.

La mayoría de los ataques se aprovechan de las vulnerabilidades existentes y ya conocidas. Sin embargo algunas organizaciones desconocen su existencia por lo tanto cuentan con sistemas vulnerables. Existen una infinidad de vulnerabilidades que pueden afectar a cualquier sistema. Detallamos algunas de las muchas vulnerabilidades, que hemos considerado las más importantes para mantener un sistema seguro, y en lo posible, libre de intrusos y ataques.

- Una vulnerabilidad para un sistema de información puede ser, dejar la mayoría de puertos abiertos ya que por ellos pueden acceder los atacantes al sistema. Sólo deben estar abiertos los puertos que sean necesarios.

- No contar con un filtrado eficiente de direcciones IP, tanto en la entrada como en la salida, es otra vulnerabilidad que se debe tener muy en cuenta, ya que pueden ser suplantadas para infiltrarse en la red. Por esto se debe aplicar mecanismos que impidan la entrada y/o salida en nuestra red de direcciones IP incorrectas.

- No tener un registro de actividades realizadas en el sistema también es otra vulnerabilidad que los administradores de redes suelen descuidar; pero que puede ser de gran utilidad para así detectar los posibles problemas de forma preventiva.

- Recursos de red compartidos no protegidos. Algunos protocolos de red no cuentan con mecanismos de protección adecuados, por lo que un atacante externo puede acceder a la información almacenada en los equipos.

- Obtener información a través de sesiones de usuario anónimo. Al acceder sin nombre de usuario ni contraseña en sistemas operativos que permitan este tipo de sesiones, el atacante puede obtener información del sistema.

- Vulnerabilidades en el programa BIND generalmente utilizado para actuar como servidor de nombres de dominio (DNS). En algunas versiones se puede obtener acceso al sistema con permisos de administrador.

1.1.3 Ataques

Los ataques son la realización de las amenazas. Existen varios tipos de amenazas o ataques, entre las más comunes están: Intercepción, Modificación y Fabricación

1.1.3.1 Intercepción

Es el acceso no autorizado de una persona, un programa o un ordenador a un recurso o servicio atacando así a la confidencialidad. Ejemplo de este ataque son: interceptar una línea para capturar datos que circulan por la red, o la lectura de las cabeceras de paquetes para descubrir la identidad de uno o más de los usuarios implicados en la comunicación.

1.1.3.2 Modificación

Este es un ataque contra la integridad. Además es la manipulación de un recurso por parte de una entidad no autorizada. Ejemplos de este ataque son la alteración de valores en un archivo de datos o mensajes transferidos por la red.

1.1.3.3 Fabricación

Es el insertar objetos falsificados en el sistema atentando contra la autenticidad. Como la inserción de mensajes ilegítimos en una red o añadir registros a un archivo.

1.1.3.4 Ataques pasivos

En los ataques pasivos no existe alteración en la comunicación, sino que únicamente el atacante escucha o monitoriza, para obtener la información que está siendo transmitida. Con la finalidad de interceptar los datos y analizar el tráfico. Este tipo de ataque pasivo es muy difícil de detectar, por lo que no existe alteración en los datos. Sin embargo, es posible proteger la información utilizando mecanismos de encriptación y cifrado.

1.1.3.5 Ataques activos

En estos ataques existe la alteración, modificación del flujo de datos transmitidos o la creación de un falso flujo de datos. Se subdivide en cuatro categorías: Suplantación de identidad, Reactuación, Modificación de mensajes, Degradación fraudulenta del servicio

1.1.3.5.1 Suplantación de identidad

El atacante se hace pasar por una entidad diferente. Por ejemplo, la captura de las claves de autenticación que luego permitan el acceso a recursos privilegiados

1.1.3.5.2 Reactuación

La captura y replicación de mensajes legítimos para realizar acciones inadecuadas como son el ingreso de dinero varias veces a una misma cuenta.

1.1.3.5.3 Modificación de mensajes

Es la alteración de una parte del mensaje original, o del orden en que se envían, produciendo así efectos no autorizados. Por ejemplo, la alteración en el nombre de la cuenta al momento de hacer un ingreso.

1.1.3.5.4 Degradación fraudulenta del servicio

Impide el uso normal o la gestión de recursos informáticos y de comunicaciones. Por ejemplo, el intruso podría interrumpir el funcionamiento de la red inundándola con mensajes falsos.

1.2 Autenticación, Integridad y No-repudio

1.2.1 Autenticación

La autenticación es el mecanismo por el cual se puede asegurar que la identidad indicada es la verdadera. Es el tipo de control de acceso más conocida y utilizada por los usuarios, en la que se les solicita el ingreso de un nombre y una contraseña para así poder asegurar su identidad. Todos los servidores ofrecen esta forma de autenticación, con diferentes mecanismos y niveles de seguridad. El mas común mediante el uso de un nombre de usuario y una contraseña. Existen tres tipos básicos de autenticación mediante nombre de usuario y contraseña:

1.2.1.1 Autenticación básica

Todos los servidores *web*, navegadores y terminales móviles soportan este tipo de autenticación. Al acceder a un recurso del servidor *web* protegido mediante autenticación básica, se realiza el siguiente proceso:

1. El navegador solicita al usuario su nombre y contraseña a través de una ventana de autenticación y con esta información establece una conexión con el servidor.
2. Si la información de autenticación no es correcta, es rechazada por el servidor y nuevamente el navegador solicita al usuario el nombre y contraseña hasta que estos sean válidos o cierre la ventana.
3. Una vez que el servidor *web* confirma la autenticidad de los datos, se establece la conexión de acceso al recurso protegido.

La desventaja de este mecanismo es que la contraseña no viaja encriptada desde el navegador del usuario hasta el servidor, por lo que el atacante puede interceptarla. Al contrario, la ventaja es compatible con todos los navegadores debido a que forma parte del protocolo HTTP 1.0 y sus versiones posteriores. Cada vez que el usuario accede a una nueva página el navegador no presenta nuevamente otra ventana de autenticación

sino que automáticamente envía al servidor el nombre y contraseña ingresados al momento de conectarse.

Para acceder a recursos protegidos en Linux a través de la autenticación básica, no es necesario que los usuarios tengan una cuenta abierta en el sistema, en su reemplazo se cuenta con dominios de autenticación, los cuales incluyen uno o varios directorios.

1.2.1.2 Autenticación mediante resúmenes

Esta autenticación es soportada por todos los servidores y en algunos navegadores. Al momento en que la seguridad es importante, el método de autenticación básica no es conveniente, por lo tanto se plantea una solución: el enviar un resumen criptográfico de la contraseña (un *hash*) en lugar de la propia contraseña. Se lo realiza de la siguiente forma:

1. El navegador recibe información por parte del servidor, dicha información se utilizará en el proceso de autenticación.
2. El navegador añade al nombre de usuario y contraseña la información que recibe del servidor y conjuntamente con otra información adicional crea un resumen. El objetivo de la información adicional es evitar los ataques de reactuación.
3. A través de la red se envía el resumen y la información adicional hacia el servidor.
4. El servidor crea el resumen del conjunto añadiendo la información adicional a una copia en claro de la contraseña del usuario.
5. Luego compara el resumen que creó con el que recibió del navegador
6. En caso que coincidan los números de ambos, se otorga permiso de acceso al usuario.

La autenticación mediante resúmenes se incorporó al estándar HTTP 1.1, pero no todos los navegadores la soportan.

1.2.1.3 Autenticación mediante resúmenes en Linux

Este método de autenticación es igual al de autenticación básica, la diferencia es que en vez de especificar *AuthType Basic*, se indica que es mediante resumen: *AuthType Digest*. Este método utiliza el algoritmo MD5 (algoritmo de encriptación para comprobar integridad en transmisiones de cualquier tipo de datos) para resúmenes criptográficos.

Los archivos de los nombres de usuario y contraseñas se crean con la herramienta *htdigest*, de forma similar a como se hacía con *htaccess*:

```
htdigest -c RaizServidor/pw/digest1.conf DOCUMENTOS joseph
```

El primer argumento es el nombre del archivo donde se encuentran los usuarios, el segundo es el dominio al que se tienen acceso y el tercero es el nombre del usuario. Por cada usuario nuevo que se registra, *htdigest* solicita dos veces la contraseña. También se cambia el directorio que indica la localización de este fichero, que en lugar de *AuthUserFile*, es:

```
AuthDigestFile RaizServidor/pw/digest1.conf
```

Por lo que si utiliza autenticación mediante resúmenes para proteger el directorio confidencial, las directivas requeridas serían:

```
<Directory RaizDocumentos/confidencial>
AuthType Digest
AuthName DOCUMENTOS
AuthDigestFile RaizServidor/pw/digest1.conf
Require valid-user
</Directory>
```

Este método desventajosamente no es soportado por todos los navegadores, pero sí por algunos servidores.

1.2.1.4 Autenticación de Windows integrada

Este tipo de autenticación es exclusivo de Windows NT y es una variante de la autenticación mediante resúmenes criptográficos. Este método de autenticación es seguro ya que no se envía el nombre de usuario ni la contraseña por la red, sino que mediante un intercambio de datos el navegador demuestra al servidor que conoce la clave sin revelarla. Este tipo de autenticación se usa con servidores NT, puesto que no es compatible con la autenticación por resúmenes debido a los detalles de implantación. Funciona de la siguiente manera:

Utiliza el nombre de usuario, contraseña y dominio con la que el usuario inició la sesión en el ordenador motivo por el cual posteriormente no se le presentará una ventana para que ingrese su nombre y contraseña. En caso de no ser correcta la información de sesión, se le solicita el nombre, contraseña y el dominio y este proceso se repetirá hasta que los datos sean los correctos.

Este método presenta dos limitaciones considerables para el Internet: El único navegador que soporta esta autenticación es Microsoft Internet Explorer, en versiones 2.0 y posteriores y en servidores NT y no funciona para conexiones con proxy.

Este tipo de autenticación se utiliza para *Intranets* ya que se puede exigir que todos los usuarios utilicen el Internet Explorer y estos se encuentran detrás del mismo *Proxy*. Todas las cuentas de usuario autenticadas mediante este mecanismo deberían contar con privilegios para acceder a este equipo desde la red.

1.2.1.5 Ventajas e inconvenientes de la autenticación

1.2.1.5.1 Ventajas.

Aumenta la probabilidad de que el usuario que desea acceder mediante su contraseña sea legítimo ya que se autentica al usuario y no al ordenador. Inclusive así compartan la misma dirección IP si cada usuario tiene su contraseña propia puede autenticarse de forma individual.

No interviene el lugar de donde el usuario se autentique siempre y cuando se utilice autenticación básica o mediante resúmenes y la asignación dinámica de direcciones IP no es relevante.

1.2.1.5.2 Inconvenientes.

Los usuarios podrían dar sus contraseñas a otras personas e incluso estas podrían ser publicadas en Internet para acceder a los servicios y recursos que se supone están protegidos, por ello se ve la necesidad de cambiarlas periódicamente.

El acceder a varios servicios en Internet da lugar a que el usuario tenga que recordar un gran número de contraseñas, esto podría ocasionar el olvido, confusión y pérdida de las mismas. Para solucionar este inconveniente se puede recurrir a programas de gestión de contraseñas.

El uso de contraseñas fáciles de descifrar pone en riesgo la confidencialidad de los datos. Esto se pude solucionar obligando al usuario a crear una contraseña segura. En la autenticación básica sin SSL (*Secure Socket Layer* – Nivel de Conector Seguro)los datos se transmiten sin encriptar, por lo que pueden ser interceptados por un atacante.

1.2.2 Integridad.

En este mecanismo la información puede solamente ser modificada por las entidades autorizadas. La modificación consiste en escritura, cambio, eliminación, creación y

reactuación de los mensajes transmitidos. Para asegurar que los datos recibidos no se han modificado, se utiliza el método de integridad de datos, por ejemplo mediante *time-stamps* se garantiza que la secuencia de los bloques o unidades de datos recibidas no ha sido alterada y que no hay unidades repetidas o perdidas.

1.2.3 No repudio.

Mecanismo por el cual un usuario que efectúa una acción no puede en lo posterior negar haberla efectuado. Esto se realiza utilizando evidencias indiscutibles que ayudarán a resolver los problemas que se presenten al momento de asumir una responsabilidad. El no repudio de origen garantiza al usuario que recibe el mensaje de que el emisor niegue haberlo enviado. En cambio el no repudio de recepción protege al que envía el mensaje de que el receptor niegue haberlo recibido. Ejemplo de este mecanismo son las firmas digitales.

1.3 Seguridades en Linux

La seguridad en los sistemas es cada vez más importante y necesaria, debido a que en la actualidad se trabaja con conexiones a Internet que no son seguras y el rápido desarrollo del *software*. El sistema de Linux al ser multiusuario real en el que diferentes usuarios pueden trabajar a la vez desde su propio computador, tiene la obligación de proteger a los usuarios y de protegerse a sí mismo. Al permitir Linux trabajar en red también debe ofrecer seguridades en los servicios que presta.

La seguridad de los sistemas en red debe siempre estar actualizada debido a que continuamente se presentan nuevos mecanismos de ataques para tener accesos a información privada, para ello se recomienda consultar frecuentemente las publicaciones electrónicas de los últimos acontecimientos detectados.

Ningún sistema es completamente seguro, peor aún si se encuentra conectado en red. No todos los usuarios de Linux requieren el mismo nivel de seguridad, por ejemplo los servidores de Internet, bancos, etc. necesitan garantizar mayor seguridad en su

información, es necesario encontrar un equilibrio entre la facilidad de uso del sistema y la seguridad. Se recomienda utilizar una política de seguridad de acuerdo a las diferentes necesidades de seguridad que requiera el sitio y que sistema de comprobación se realiza.

Puesto que las fuentes de Linux son abiertas y que cualquier usuario podría manipularlas, se requiere de niveles altos de seguridad. Linux tiene disponible todos los servicios habituales en una red:

- Bases de datos.
- Servicios de Internet.
- Servicio de ficheros e impresión.
- Utilidades necesarias para mantener el nivel de seguridad requerido.

Cada uno de estos servicios funcionan independientemente, por ejemplo: se podría modificar la dirección IP del equipo, las rutas, añadir, parar o reiniciar un servicio concreto sin afectar al resto de servicios. Lo único que requiere que se detenga el funcionamiento del equipo, es la realización de operaciones con el *hardware*. Esto es lo que le diferencia del resto de sistemas operativos.

Linux cuenta con todas las características de los sistemas Unix: un control de acceso a los usuarios verificando el nombre de usuario y clave; cada archivo y directorio tienen su propietario y permisos. Este sistema operativo trabaja con grupos y usuarios, cada usuario pertenece a uno o varios grupos, y cada recurso pertenece a un usuario y un grupo. Los permisos para un recurso se pueden asignar al propietario, al grupo y al resto de los usuarios. Para tener un sistema seguro y funcional, se deben realizar las combinaciones necesarias entre el propietario y grupo de un recurso con los permisos de los propietarios, grupos y otros.

El administrador debe conocer las necesidades de cada uno de sus usuarios y asignarle los privilegios necesarios para que pueda realizar su trabajo sin resultar un peligro para otros usuarios o el sistema. Para mantener seguro el sistema es suficiente contar con los valores que tienen por defecto las distribuciones de Linux.

El mantener cuentas abiertas que no se utilizan es un problema importante para el sistema ya que estas pueden servir de refugio para un atacante sin percatarse de sus acciones. Un punto muy importante en la seguridad en Linux, son las claves. La seguridad de una sola cuenta puede comprometer la de todo el sistema. Se debe estar seguro de que los usuarios utilizan claves sólidas, para ello existen programas que comprueban que este requisito se cumpla en el sistema.

Además se debe tener en cuenta que las claves cifradas se almacenan en un archivo al cual los usuarios tienen permiso de lectura. Incluso podría accederse a través de navegadores con falencias de seguridad. Para solucionar esta vulnerabilidad, se puede acudir a un mecanismo que consiste en colocar las claves cifradas ubicadas en el archivo habitual en otro archivo, que sólo puede leer el *root*, y dejar el resto de la información en el original. En la mayoría de las veces la destrucción de un sistema ha sido provocada por el administrador (*root*) debido a descuidos, excesiva confianza o falta de conocimientos. Este problema se puede evitar siguiendo las siguientes precauciones:

- Evitar el uso de la cuenta *root*. Es preferible empezar cualquier acción como un usuario normal, de tal forma que cuando se realice alguna acción que podría causar daño no se tendrá permiso de realizarla.
- Antes de ejecutar los comandos se debe estar seguro de la actividad que se va a realizar.
- Utilizar un canal seguro como ssh para transportar la clave del root cifrada por la red.

- No entrar directamente como *root* cuando se realiza una conexión remota, en el caso de que se necesite se debería tener en cuenta la restricción de acceso de ciertos equipos como *root.*

Una regla fundamental de seguridad es otorgar a los usuarios solamente los permisos que lo requieran para poder realizar sus actividades sin que afecte la seguridad en el trabajo de los demás. El sistema podría verse afectado al momento que acceden sin permiso a la información privada, obteniendo privilegios ilegales, utilizando de forma indebida los recursos o alterando la información almacenada en los equipos.

Una forma de mantener un sistema seguro podría ser el contar con distribución correcta del espacio de almacenamiento, para ello se recomienda particionar la unidad de disco. Linux para evitar un ataque al sistema tratando de ocupar todo el espacio del disco duro, da la posibilidad de llevar un control del espacio de almacenamiento por usuario o grupo.

La instalación de muchos sistemas operativos y aplicaciones no toman en cuenta la seguridad como un factor determinante por lo tanto es importante revisar las configuraciones antes de colocar a la máquina en red. Así como también de que todos los usuarios cuenten con contraseñas robustas que no sean fáciles de identificar.

1.4 Conclusión.

Existen varias formas de atentar contra las seguridades de una red, poniendo así en riesgo la integridad, autenticidad, confidencialidad y privacidad de la información que viaja a través de ella; es por ello que se las debe analizar detenidamente para evitar daños irreversibles en los datos.

Es importante recalcar que existen tanto amenazas externas como amenazas internas y por lo general estas segundas no son consideradas y son las más peligrosas y difíciles de detectarlas, por lo que debemos tenerles en cuenta en el momento de configurar los permisos de acceso al servidor. Al momento de realizar el análisis de los posibles

ataques a la red, debemos considerar tanto la parte del *hardware* (puertos) como la parte del *software*.

Conforme el tiempo pasa, se proponen diferentes mecanismos de protección a los datos contra atacantes, ya sean estos pasivos o activos, brindando mayor seguridad a los mismos. Los métodos de encriptación, cifrado, el uso de contraseñas seguras, certificados y firmas digitales para comprobar la identidad del usuario para ver si es quien dice ser y evitar en lo posterior el repudio asumiendo así responsabilidad, son mecanismos de precaución que se toman para evitar así acciones que atenten contra la seguridad de la información.

Es verdad que ningún sistema es completamente seguro y peor si se encuentra en red, pero no por ello podemos dejar de proporcionar seguridad en la información. Se debe tratar de poner todos los esfuerzos para que el margen de inseguridad sea el mínimo, para así contar con información confiable. Se recomienda mantenerse siempre informados y en constante actualización de las nuevas formas de ataques y de las posibles soluciones, estando así preparados para cualquier dificultad que se nos presente.

CAPITULO 2: PORTALES CAUTIVOS

Introducción

Dado la gran importancia de contar con seguridad en una red, se ve la necesidad de implementar mecanismos que nos garanticen una mayor confidencialidad en la información. Una forma de ello es la autenticación vía Portal Cautivo, la que se implementa utilizando una página *Web* en la cual se solicita un nombre y contraseña de usuario para acceder al servidor, en caso de que estas no sean correctas el Portal Cautivo no da los permisos necesarios para poder acceder. Existen varios productos de software que implementan este método de autenticación entre ellos está NoCat Auth y ChilliSpot entre otros.

2.1 ¿Qué es un Portal Cautivo?

Un portal cautivo es una página *web* en la que se presenta una solicitud de inicio de sesión al momento en que el usuario de una red pública y/o privada desea acceder a una conexión a Internet, garantizando así el uso normal y legal de la red. En caso que el usuario no inicie una sesión válida, no se le permite el tráfico de red a través de la puerta de enlace. Estos portales son utilizados principalmente por proveedores de *hot-spots* en aeropuertos, hoteles, cafeterías, cafés Internet, etc.

Al momento en que un usuario se autentica ante una red por primera vez, se presenta una página *Web* en la que se necesita de una serie de acciones antes de acceder como el aceptar las políticas de uso presionando un botón en la página. Un portal cautivo puede ser la puerta de acceso para conectar usuarios, redes u otros servicios. Las funciones principales que realiza un portal cautivo son:

1. Desplegar una página *Web* de inicio de sesión ante el usuario
2. Validar la identificación

3. Notificar a la puerta de enlace quién ingreso correctamente la identificación.
4. Administrar las conexiones locales
5. Gestionar el ancho de banda que se asignado a cada uno de los usuarios
6. Determinar las reglas del "*firewall*" o contrafuegos
7. Cerrar sesiones caducadas, antiguas o finalizadas
8. Utiliza SSL y PGP (*Prety Good Privacy* – Privacidad Bastante Buena) para la comunicación con la puerta de enlace y para las conexiones de los usuarios, esto hace que no se dependa de la seguridad WEP (Wired Equivalent Privacy), ni WPA (Wireless Protected Access – Acceso Protegido a Redes Inalámbricas).

La comunicación entre el portal cautivo y la puerta de enlace se realiza mediante mensajes firmados PGP mediante el sistema de autenticación, con lo que se reduce la posibilidad de que un usuario altere las reglas del "*firewall*" cortafuegos.

2.2 Funcionamiento

El sistema de Portal Cautivo controla el acceso a redes y cuenta con un "*gateway*" o pasarela que encamina las conexiones consultando al Servidor de Autenticación mientras que un Servidor de Autenticación que define la identidad del usuario y qué servicios de la red podrá utilizar; si los datos son correctos el usuario puede pasar a través del "*firewall",* para luego establecer un túnel de cifrado que le permitirá establecer comunicaciones seguras. En caso que el usuario no se ha autenticado correctamente no se permite ningún tipo de tráfico a través del *gateway*.

2.2.1 Software

Existen varios productos de *software* que permiten controlar el acceso de usuarios a los servicios de una red. Los dos mas conocidos son: Nocat Auth y Chillispot.

2.2.2 NoCat Auth

NoCat Auth es un programa escrito en los lenguajes de programación Perl y C y es el encargado de implementar el portal, solicitando al usuario que inicie una sesión mediante la presentación de una página *Web*, posteriormente valida esta identificación y en caso de ser correcta notifica a la puerta de enlace. Además gestiona el ancho de banda asignado a cada uno de los clientes y administra las conexiones. NoCat Auth, puede funcionar en tres modos: *Public*, *Member* y *Owner*.

El proceso de autenticación en NoCat Auth es:

1. El cliente se conecta a la red mediante la configuración TCP/IP, asignándole una IP, puerta de enlace, DNS, etc.

2. Se bloquea el acceso a cualquier lugar más allá del sistema de autenticación, esto se lo realiza de forma predeterminada.

3. El usuario realiza una petición *Web*.

4. A continuación se le redirige al servicio de puerta de enlace.

5. Nuevamente el usuario es redirigido al sistema de autenticación

6. Se le presenta los tres tipos de modos de autenticación, *Member, Public* y *Owner*.

7. En caso que las credenciales sean válidas, se crea un mensaje saliente firmado mediante PGP y se envía nuevamente hacia la puerta de enlace, este proceso se da entre el *gateway*, el sistema de autenticación y el portal cautivo.

8. La puerta de enlace mediante una clave pública proporcionada por el servicio de autenticación, comprueba y verifica la autenticidad del mensaje.

9. Si todo es correcto se modifican las reglas del cortafuegos y se redirige al usuario a la página que deseaba en el punto 3.

Para mantener activa y abierta la conexión, se abre un *pop-up* (JavaScript) al cliente que actualiza cada cierto tiempo el inicio de sesión, es por este motivo que se pueden anular sesiones antiguas o establecer límites de tiempo a las sesiones. En caso que el usuario cierre el *pop-up* o navegador requerirá realizar nuevamente todo el proceso de autenticación.

2.2.3 Chillispot

Chillispot es un *software* de código abierto bajo licencia GPL (*General Public License*), sencillo de configurar y poco exigente en los requisitos de *software* y configuración de red, además permite la instalación de todo el sistema sin problemas en un mismo equipo.

Todas las peticiones http son redirigidas por el *firewall* hacia una *Web* de bienvenida con toda la información que se desee publicar. Una de las principales opciones de este *software* es el acceso a la red.

A este *software* se le puede configurar para restringir el acceso a ciertas direcciones IP o servicios de la red, así como de la misma manera se le puede permitir libre acceso a ciertos servicios y a ciertas direcciones IP. Por ejemplo se puede dar libre acceso al servidor local pero se bloquea el acceso a los demás servicios como al Internet, ftp, etc. El sistema controla en todo momento los accesos indicando el estado del cliente logeado mediante una pantalla.

Los requisitos para su funcionamiento son:

- Conexión a Internet
- El software ChilliSpot
- Servidor Radius
- Servidor Web
- El S.O. Linux

ChilliSpot soporta dos métodos de autenticación: *Universal Access Method* (UAM) y *Wireless Protected Access* (WPA). Con el método UAM el cliente requiere una dirección IP la cual es asignada por el mismo sistema. Al momento en que el usuario comienza a navegar, el sistema Chilli captura la conexión TCP y le redirige al servidor de autenticación. El Servidor *Web* solicita al usuario su nombre y contraseña de usuario, esta contraseña es enviada nuevamente al sistema Chilli pero viaja de forma encriptada. Con WPA la autenticación se realiza mediante un *access point*, la conexión entre el *access point* y el cliente es encriptada.

Para ambos métodos se requiere de un servidor *Radius*. En caso que la autenticación sea correcta el Servidor *Radius* envía un mensaje de aceptación al sistema Chilli, caso contrario si los datos son incorrectos niega el acceso.

2.3 Conclusión

Mediante el uso del mecanismo de autenticación vía Portal Cautivo podemos acceder o no a los recursos de una red, dependiendo de los permisos que tengamos. Luego de haber investigado sobre los diferentes productos de *software* existentes para la implementación de Autenticación vía Portal Cautivo, hemos decidido investigar más a profundidad el *software* NoCat Auth, debido a los diferentes beneficios que nos brinda, puesto que con dicho sistema además de asignar permisos de accesos mediante la creación de usuarios y sus contraseñas, también nos permite distribuir el ancho de banda para cada uno de ellos dependiendo de sus necesidades. Con este *software* tenemos más

opciones de configuración, pero cabe recalcar que no estamos menospreciando al *software* ChilliSpot, este también cuenta con grandes ventajas, pero con menos opciones de configuración, además el más conocido, utilizado y con más documentación es el NoCat Auth a diferencia del ChilliSpot.

CAPÍTULO 3: SOFTWARE NOCAT AUTH

Introducción

En la actualidad se está tratando de utilizar *software* libre, debido a los altos costos que implica la compra de las licencias. Y los Sistemas de Autenticación no se han quedado atrás y también buscan emplear este tipo de *software* para brindar mayores seguridades en la red.

Nocat Auth es una de las muchas aplicaciones que existen para realizar autenticación, encriptación y privacidad sin usar EAP (*Extensible Authentication Protocol*) y mediante *software* libre. Esta aplicación es de gran utilidad cuando de brindar seguridad a una red se desea, pues permite realizar una validación de los usuarios que desean acceder y utilizar los servicios que ofrece la red. Dependiendo del perfil del cliente le permite utilizar el ancho de banda y los servicios que estén asignados para él. Nocat Auth fue desarrollado por la comunidad *wireless* de Sonoma *County-Schuyler Erle-*, California (EE.UU.). Colaborando también *SeattleWireless*, PersonalTelco, BAWUG, *Houston WUG* además de personas y grupos de todo el mundo.

En este capítulo estudiaremos detenidamente sus características, estructura, modos de funcionamiento y la Base de Datos de Autenticación que utiliza para almacenar información acerca de sus clientes.

3.1 Características

- Permite una autenticación segura basada en SSL
- Autoriza mediante usuario y contraseña.
- Informa de la entrada y salida del usuario en la red.
- Añade la implementación de QoS (*Quality of Service* – Calidad de Servicio) por usuarios y grupos.

SSL (*Secure Sockets Layer*) fue diseñado por *Netscape Communications Corporation*. En su versión actual, proporciona cifrado de datos, autenticación de servidores, integridad de mensajes, y opcionalmente, autenticación de cliente para conexiones TCP/IP, para todo esto utiliza un algoritmo de cifrado asimétrico, por lo general RC4(*Radio Control 4th Generation*) o IDEA(*International Data Encryption Algorithm*), y además cifrando la clave de sesión de estos algoritmos mediante un algoritmo de cifrado de clave pública (RSA- *Registered Signature of Authority*). La clave de sesión se la utiliza para cifrar los datos que se transmiten al servidor y viceversa. Por cada sesión se genera una clave diferente, de tal forma que si esta es descifrada no servirá para transacciones futuras.

Para la autorización mediante usuario y contraseña el navegador presenta al usuario la ventana de autenticación, para que introduzca sus credenciales al momento que el usuario pretende acceder a un recurso del servidor. Una vez introducidos los datos solicitados, el navegador intenta establecer una conexión con el servidor utilizando esta información. El navegador le presenta nuevamente la ventana al usuario, si la información fue rechazada; esto seguirá repitiéndose hasta que se introduzca una contraseña válida o hasta que el usuario cierre la ventana. Al momento que el servidor W*eb* verifique con éxito los datos de autenticación, se establece la conexión de acceso.

Una característica muy importante de este software es que se informa la entrada y salida de un usuario en la red, de tal forma que se puede llevar un control robusto de los accesos, así como el tiempo en que un usuario permanece conectado a la red (archivos log).

La implementación de QoS por usuarios y grupos ofrece una gran ayuda al momento de realizar la asignación de ancho de banda a los clientes. Para la gestión de ancho de banda Nocat Auth funciona en 3 modos por defecto:

Owner: hace referencia a usuarios propietarios que cuentan con acceso a todos los recursos de la red sin ninguna restricción.

Member: a este modo pertenecen los usuarios con *login* para iniciar una sesión preestablecida, estos son configurados en la base de datos del servidor de autenticación.

Public: son aquellos que no poseen un *login* para iniciar una sesión, pero deben autenticarse para acceder a los servicios permitidos (a estos les asigna un mínimo de ancho de banda).

3.2 Ventajas del software

Evita el *snifing* en la transmisión de la información de autenticación, ya que esta viaja encriptada mediante SSL (*Secure Socket Layer*).

La autenticación puede ser configurada para realizarse contra un repositorio de datos, pudiendo ser este de varios tipos un archivo propio (MD5), una Base de Datos (MySQL o Postgrest), LDAP (*Lightweight Directory Access Protocol*), Radius, PAM (*Plugglable Authentification Module*), Samba, IMAP (*Internet Message Access Protocol*)

Permite configurar a los usuarios por grupos (*owner, public y member*) con diferentes privilegios.

Captura las direcciones físicas (MAC) de las máquinas que ingresen o intenten ingresar a la red, brindando así información importante al administrador de la red para controlar la seguridad de la misma.

3.3 Modos de funcionamiento

Nocat Auth puede ser configurado en cualquiera de los siguientes modos dependiendo de las necesidades y servicios que preste una red.

Portal Cautivo: Envía a una página *web* todas las peticiones de los usuarios, analiza los credenciales del usuario y máquina y las compara con las contenidas en una base de

datos. Es decir, se realiza un *Login* obligatorio para el usuario, y se mantendrá la sesión mientras este esté logeado.

Portal Pasivo: Realiza la misma función que el portal cautivo con la única diferencia es la utilización de NAT (*Network Address Translation*). Esto además implica la presencia de un *Firewall* antes de llegar al NoCat *Gateway*.

Portal Abierto: Muestra únicamente una página con información de las condiciones de uso, no solicita credenciales.

3.4 Componentes - Estructura y Funcionamiento

Los componentes necesarios para el correcto funcionamiento del *software* son:

- NoCat Auth: Aquí se encuentra el Servicio de autenticación.
- NoCat *Gateway*: Realiza el servicio de redirección y *Forwarding*.
- Auth *Database*: puede ser un archivo propio (MD5), una Base de Datos, Ldap, Radius, PAM, Samba, IMAP.
- *Access Point*: utilizado en redes Inalámbricas

En el proceso de autenticación mediante Nocat Auth se realizan los siguientes pasos:

1. El cliente se asocia al *Gateway* y este le asigna una dirección IP:
2. El *Gateway* redirige a la página de *login* del *Auth Server*
3. El Auth *Server* pide usuario y contraseña al cliente (vía SSL) y la comprueba con la Auth *Database*. Los mensajes de autorización van firmados con PGP/GnuPG, donde el GW utiliza la clave pública del Auth Server. (ver Anexo 1).

4. Si la autenticación ha sido satisfactoria el GW redirige el tráfico a la LAN y/o Internet. (ver Anexo 2).

3.5 Base de Datos

3.5.1 Formas de acceso

Para comprobar si el usuario cuenta con acceso a la red, Nocat Auth puede utilizar como Base de Datos de Autenticación: un archivo propio (MD5), una Base de Datos (MySQL o Postgrest), LDAP (*Lightweight Directory Access Protocol*), Radius, PAM (*Plugglable Authentification Module*), Samba o IMAP (*Internet Message Access Protocol*). Para esto se configura el archivo nocat.conf (este tema se tratará con más detalle en el capítulo 4).

En el presente trabajo utilizaremos como Gestor de Base de Datos a MySQL, y se creará una Base de Datos llamada “nocat” a partir de un archivo esquema que viene incluido en el paquete de instalación de NoCat Auth (etc/nocat.schema).

3.5.2 Administración de los usuarios

Nocat cuenta con un *script* ubicado en /usr/local/nocat/bin/admintool, el cual permite administrar la base de datos de los usuarios. Su forma de uso es:

```
admintool [-c|-p] [Usuario] [Contraseña]
admintool [-a|-d] [Usuario] [Grupo]
admintool -l [Grupo]
```

donde:

-c : Crea un Usuario con su respectiva Contraseña.
-p : Cambia la Contraseña del Usuario por una nueva.
-a : Agrega un Usuario a un Grupo específico. Esta opción es muy importante ya que si un usuario no pertenece a un grupo no podrá acceder al Internet
-d : Elimina un Usuario de un Grupo específico.

-l : Lista todos los Usuarios o los de un grupo específico.

Para la administración de usuarios con una interfaz gráfica, existe un módulo de NoCat para el sistema Webmin (Sistema con interfaz Web para la Administración de los servicios de Linux).(ver Anexo3).

3.6 Conclusión

NoCat es un *software* gratis y de código abierto, que nos da la posibilidad de ampliarlo y aplicar nuevas ideas al código fuente, pudiendo así mejorarlo y ajustarlo a nuestras necesidades y requerimientos. Este *software* es muy útil para controlar los usuarios que ingresan a la red brindando seguridad a la misma.

Como conclusión podemos decir que NoCat es un *software* muy beneficioso, ya que además de ser gratis y de código abierto, nos brinda seguridad en el acceso de usuarios a la red.

PARTE II: EL PROCESO DE INSTALACIÓN Y CONFIGURACIÓN

CAPITULO 4: CONFIGURACIÓN DEL SISTEMA DE AUTENTICACIÓN

Introducción

Luego de haber estudiado y analizado el *software* Nocat Auth, procederemos a configurarlo, determinando primeramente los requerimientos necesarios para que este funcione correctamente. En este capítulo profundizaremos el estudio y detallaremos los pasos que se realizan para instalar y configurar el sistema de autenticación NoCat Auth vía portal cautivo.

4.1 Instalación de los requisitos, previo a la implementación del software.

Requerimientos de Hardware:

COMPONENTE	REQUERIDO	INSTALADO
Procesador	Pentium I o superior.	Pentium IV 2,53 GHz
Tarjeta de red	2 tarjetas Ethernet	2 tarjetas Ethernet
Disco Duro	mínimo 10Gb	40Gb
Memoria RAM	mínimo 256 Mb	1Gb.

Requerimientos de Software:

- LINUX con kernel 2.4.x con Iptables:
- Servidor Apache + mod _ssl
- Perl versión 5.6 o superior con los módulos:
    - Net::Netmask
    - Digest::MD5
- GnuPG
- Un servidor DHCP

- Un Servidor DNS
- MySQL

4.1.1 Sistema Operativo Linux

Linux es un sistema operativo para PC i386. Es *Open Source*, es decir los usuarios pueden tener acceso a los fuentes del S.O., brindando así la opción de modificarlo o ampliarlo como se desee. La mejor cualidad de Linux no es que sea *Open Source*, ni que resulte gratuito, sino que es el S.O. más eficiente que podemos encontrar para las plataformas PC i386 y cuenta con una excelente estabilidad; permitiendo tener buenas bases para el desarrollo de sistemas estables, garantizados y brindar satisfacción a los clientes. La distribución de Linux que se va a utilizar y sobre la cual va a correr nuestro servidor de autenticación es CentOS V4. CentOS es una distribución de Linux orientada al entorno empresarial basada en el código fuente libremente publicado por *RedHat Enterprise* Linux. Centos es una distribución de Linux totalmente basada en los SRPMS (Source RPM) de *RedHat Enterprise Server*.

4.1.2 El kernel

El kernel es el núcleo del sistema operativo. Provee todos los servicios básicos que necesitan las demás partes del sistema operativo. También se encarga de la administración de los procesos, memoria y discos. Debido a que el kernel es independiente de la distribución GNU/LINUX utilizada este podría servir para cualquier caso. La versión de kernel utilizada es la versión por defecto del CentOS.

4.1.3 Los Iptables

Los Iptables son una herramienta que nos permite configurar las reglas del sistema de filtrado (*firewall*) de paquetes del kernel de Linux, se ha desarrollado aun más a partir del kernel 2.4. Las iptables nos permiten crear un *firewall* de acuerdo a nuestros requerimientos, además configurarlo para tener control de quien entra, sale y enrruta a través de nuestra máquina Linux (*Input, Output y Forward*). Iptables esta integrado con el kernel, es parte del sistema operativo, y su funcionamiento es sencillo. Primero a las

Iptables se les proporcionan unas reglas, en las que se especifican las características que debe cumplir un paquete. También se especifica para esa regla una acción o *target* en la que se indica lo que debe hacer el paquete (*Accept, Drop, Reject*). Las reglas tienen un orden, y al momento en que se envía o recibe un paquete, las reglas se recorren en orden hasta que las condiciones que pide una de ellas se cumplen en el paquete, y la regla se activa realizando sobre el paquete la acción que le haya sido especificada. Los Iptables son un requerimiento muy indispensable para NoCat ya que es utilizado por el servidor de autenticación para controlar el acceso a la red de los equipos.

4.1.4 Servidor Apache

El servidor HTTP Apache es un servidor *web* de código abierto, diseñado para varios sistemas operativos de red como las plataformas Unix (BSD, GNU/Linux, Windows, otros), que implementa el protocolo HTTP/1.1 y sitios virtuales. Al comienzo de su desarrollo febrero de 1995 se basó en el código de NCSA (*National Center for Supercomputing Applications*) HTTPd 1.3, pero posteriormente fue reescrito por completo. El nombre Apache se debe a que originalmente consistía solamente en un conjunto de parches a aplicar al servidor de NCSA, que fueron creando varios *webmaster* y que se contactaron vía *e-mail*, y así se creó el grupo Apache. En inglés, *a patchy server* (un servidor parcheado). Permite la creación y publicación de documentos HTML con estabilidad y eficacia comprobadas en la gran cantidad de servidores apache actualmente en uso. Apache es considerado el mejor y más importante servidor Web debido a que es seguro, robusto, confiable y extensible, recomendado para páginas con carga media/alta. La versión de Apache instalada en el servidor es la que viene por defecto en la distribución Centos, la V2.0.52.

4.1.5 El módulo de mod_ssl

Es un módulo de Apache que da soporte al "*Secure Sockets Layer*" (SSL) y "*Transport Layer Security*" (TLS) entre un servidor de *Web* y clientes (*Web browsers*). Apache y mod_ssl dan al sistema seguridad mediante el uso de certificados digitales que permiten conexiones al servidor *web* en forma cifrada y segura. El mod_ssl es utilizado por NoCat Auth por la capacidad criptográfica que el servidor *web* requiere para poder utilizar los

protocolos SSL para el cifrado del tráfico y debido a que al realizarse la autenticación, los datos viajan utilizando el protocolo HTTPS.

4.1.6 Perl

Perl (*Practical Extraction and Report Languaje*) es un lenguaje de *script*, ideal para la manipulación de textos, archivos y procesos. Perl es un lenguaje intermedio entre los *shell scripts* y la programación en C, es decir es una es una mezcla optimizada de un lenguaje de alto nivel y un lenguaje de *script*, siendo así líder en programación de *scripts*. Perl cuenta con una extensa librería de módulos. Su distribución es gratuita. El lenguaje Perl no es precompilado, pero sigue siendo más rápido que la mayoría de lenguajes interpretados. Esto es porque en Perl los programas son analizados, interpretados y compilados por el intérprete Perl antes de su ejecución, haciendo que la depuración y mantenimiento de un programa en Perl sea más sencilla. A pesar de que Perl originalmente se desarrollo en el ambiente UNIX, en la actualidad existen versiones para casi todos los sistemas operativos: DOS, Windows NT, MacOS. Las páginas utilizadas para la autenticación de NoCat están escritas en lenguaje Perl; la versión de Perl instalada es la 5.8.5.

4.1.7 Digest::MD5

Es un módulo de Perl que utiliza MD5. MD5 es un algoritmo de encriptación diseñado para comprobar la integridad de los datos en transmisiones de cualquier tipo. La entrada del algoritmo puede ser un archivo completo. Por ejemplo, el *hash* MD5 de todo este documento es "7f1ac9a83858a8d812437ab87a30e98c". MD5 es imposible de desencriptar debido a que no es básicamente un algoritmo de encriptación (codificación) sino que encuentra un *message digest*. En este *message digest* no existe ninguna información de la cadena original. Es una simple huella digital del mensaje, pero su contenido no se encuentra implícito en los 32 caracteres que componen el *hash*.

La instalación de este módulo se la realizó ejecutando las siguientes líneas en la consola de Linux teniendo previamente descargado el archivo de instalación Digest-MD5-2.27.tar.

```
# tar zxvf Digest-MD5-2.27.tar
# cd Digest-MD5-2.27
# perl Makefile.PL
# make
# make test
# make install
```

4.1.8 Net::Netmask

Es un módulo de Perl que analiza y entiende bloques CIDR(*Classless Inter-Domain Routing*) de IPv4. Se construye con una interfaz orientada a objetos. Casi todas las funciones son los métodos que operan sobre objetos Net::Netmask. Hay los métodos que proporcionan casi todos los *bits* de información sobre un bloque de la red que podría desearse. Tiene también funciones para poner un bloque de red en una tabla y los bloques más últimos de la red de las operaciones de búsqueda por IP ADDRESS en esa tabla. Existen funciones para invertir a un rango de direcciones IP en una lista de bloques de CIDR. Hay funciones para invertir a una lista de bloques de CIDR en una lista de las direcciones IP. Hay una función para clasificar por texto las direcciones IP.

Para la instalación de este módulo se ejecutó las siguientes líneas en la consola de Linux teniendo previamente descargado el archivo de instalación Net-Netmask-1.9004.tar

```
# tar zxvf Net-Netmask-1.9004.tar
# cd Net-Netmask-1.9004.tar
# perl Makefile.PL
# make
# make test
# make install
```

4.1.9 GnuPG

Se utiliza para firma y encriptación de datos. Genera dos claves para el usuario, la clave privada sirve para firmar documentos y se encuentra protegida con una frase de paso "*passphrase*" que debe ser secreta y estar en un lugar seguro. Por otro lado la clave pública de nosotros usarán el resto de usuarios para enviarnos información encriptada la cual desencriptaremos con nuestra clave privada. GnuPG es software libre, puesto que ya no utiliza algoritmos patentados, este es un reemplazo completo de PGP que puede ser utilizado, modificado y distribuido libremente bajo los términos de la *GNU General Public License* (Licencia Pública General de GNU). Brinda una mayor facilidad de implementación de nuevos algoritmos utilizando módulos.

```
# tar zxvf gnupg-1.2.3.tar.bz2
# cd gnupg-1.2.3
# ./configure
# make
# make install
```

4.1.10 Servidor DHCP

DHCP(*Dynamic Host Configuration Protocol*) es un protocolo que permite asignar direcciones IP dinámicas, de forma totalmente automática. Por ello no pierde las prestaciones de BOOTP, su predecesor, sino que las amplía permitiendo nuevas formas de asignación de direcciones y nuevas opciones para poder pasar a los clientes toda la información necesaria. DHCP es un protocolo implementado en los principales sistemas operativos así como otros dispositivos.

DHCP puede usarse cuando el número de direcciones IP es menor que el número de computadores y todos no están conectados a la vez, como en un proveedor de servicio de Internet (ISP-*Internet Servie Provider*).

DHCP está formado por dos partes: un protocolo para el intercambio de los parámetros de red específicos de cada *host* y un mecanismo para la asignación de direcciones de red.

4.1.11 Servidor DNS

DNS (*Domain Name System*) es una base de datos distribuida y jerárquica, con información asociada a los nombres de dominio que sirven para traducir estos nombres que son fáciles de recordar y usar por las personas, en números de protocolo de Internet (IP) que es la forma en la que las máquinas pueden encontrarse en Internet. Esta base de datos está constituida por varios servidores DNS y cada uno de ellos es responsable de una "zona" en Internet. Su función básica es atender a las peticiones de los diferentes programas cliente (ejemplo: el navegador) que acceden a Internet y resolver la dirección IP asociada al dominio consultado. Una vez configurado el servicio DNS, permitirá al cliente autenticado que una *web* y un correo electrónico determinado sean localizados en cualquier lugar del mundo mediante el nombre de dominio.

4.1.12 MySql

MySQL es el sistema de administración de bases de datos relacionales más conocido y desarrollado. Además es muy rápido, multiusuario y multihilo. Uno de los motivos para que MySQL sea el más popular es que es un producto *Open Source* cuyas fuentes pueden ser accedidas o modificadas libremente de acuerdo a las necesidades y sin costo alguno, brindando así un sistema más flexible. Una base de datos relacional es aquella que almacena los datos en tablas separadas pero relacionadas entre ellas agregando velocidad y flexibilidad. SQL significa *Structured Query Languaje* - Lenguaje Estructurado de Consulta.

En la actualidad varias empresas han comenzado a utilizar MySQL como Gestor de Bases de Datos para proyectos *Web*, debido a sus numerosos beneficios, además de que su costo es nulo.

4.2 Instalación del NoCat Auth

Una vez instalados todos los requisitos de software, se procede a realizar la instalación del servidor de autenticación. Los pasos para la instalación del nuestro servidor NoCat Auth se detalla a continuación.

a) Lo primero que se realizó fue descargar el archivo de instalación de la página de NoCat http://nocat.net/download/NoCatAuth/ NoCatAuth-0.82.tar.gz

b) Descomprimir el archivo NoCatAuth-0.82.tar.gz

```
# tar zxvf NoCatAuth-0.82.tar.gz
```

c) Acceder al directorio de compilación

```
# cd NoCatAuth-0.82
```

d) Crear el directorio donde se instalarán los archivos de NoCat

```
# mkdir /usr/local/nocat
```

e) Instalar el *Gateway* NoCat indicando la ruta donde se creará el ejecutable.

```
# make PREFIX=/usr/local/nocat/gateway gateway
```

Si estamos trabajando con una versión de kernel mayor a la 2.4.x como es nuestro caso que estamos trabajando con una versión 2.6 se desplegará en pantalla un error debido a la incompatibilidad con los iptables del kernel, ya que los archivos de instalación de NoCat fueron diseñados para trabajar con versión 2.4.x. La solución a este problema es modificar en el directorio NoCatAuth-0.82/bin/ el archivo detect-fw.sh la línea que dice Linux 2.4 por Linux 2.6., y volver a ejecutar la línea anterior.

f) Crear el servidor de autenticación indicando la ruta donde se instalará el servidor de autenticación NoCat.

```
# make PREFIX=/usr/local/nocat authserv
```

g) Crear las llaves y certificados que utilizará NoCat para su funcionamiento.

```
# make PREFIX=/usr/local/nocat pgpkey
```

La ejecución de esta línea llevará a una serie de preguntas; se puede responder con las respuestas que se indican por defecto.

h) En las últimas líneas de compilación del paso anterior se indica que se ha instalado la clave pública en el archivo usr/local/nocat/trustedkeyd.gpg. Este archivo se lo debe copiar al directorio de las llave públicas del *gateway*.

```
# cp /usr/local/nocat/trustedkeys.gpg
/usr/local/nocat/gateway/pgp
```

i) Dar permisos de lectura para usuario con el que corre apache al directorio de las llaves públicas del *gateway*.

```
# chown -R apache:apache /usr/local/nocat/pgp
```

j) Copiar los archivos authserv.conf y gateway.conf desde el directorio NoCatAuth-0.82

```
# cp authserv.conf /usr/local/nocat/nocat.conf
# cp gateway.conf /usr/local/nocat/gateway/nocat.conf
```

Estos dos archivos llevan el mismo nombre pero cumplirán funciones diferentes. El primero servirá para la configuración del portal y el segundo para la configuración del *gateway*.

k) Copiar el archivo authserv.conf desde el directorio NoCatAuth-0.82/etc/

```
# cp etc/authserv.conf /usr/local/nocat/etc/
```

Este archivo nos servirá para la configuración del SSL del Apache, mas adelante se explicará su uso.

4.3 Configuración del sistema NoCat Auth

Básicamente para la configuración de NoCat se realizó la modificación de algunas líneas en los archivos /usr/local/nocat/nocat.conf, /usr/local/nocat/gateway/nocat.conf, y /etc/httpd/conf.d/ssl.conf adecuándolos a las características de nuestros componentes y a nuestras necesidades.

4.3.1 Configuración del portal /usr/local/nocat/nocat.conf

Las líneas que debe tener este archivo son las siguientes:

a) El directorio donde se encuentran almacenadas las llaves PGP.

```
PGPKeyPath  /usr/local/nocat/pgp
```

b) El directorio donde se encuentran alojadas las páginas que utilizará NoCat.

```
DocumentRoot      /usr/local/nocat/htdocs
```

c) El método de autenticación a utilizarse pudiendo ser este: DBI, Passwd, LDAP, RADIUS, PAM, Samba, IMAP.

```
DataSource  DBI
```

d) Configuración del modo de autenticación utilizado en nuestro caso a través de Base de Datos MySQL (DBI).

```
Database    dbi:mysql:database=nocat
DB_User     nocat
DB_Passwd   nocatauth
```

e) Configuración de las tablas y campos de la base datos a utilizar.

```
UserTable         member
UserIDField       login
UserPasswdField   pass
UserAuthField     status
UserStampField    created
GroupTable        network
GroupIDField      network
GroupAdminField   admin
```

f) El largo mínimo de caracteres para el *password.*

```
MinPasswdLength   6
```

g) La dirección IP de nuestro *gateway* (interfaz interna) así como la de la red a la que pertenece.

```
LocalGateway            192.168.10.1
LocalNetWork      192.168.10.0
```

h) Las páginas que utilizará el servidor para realizar todo el trabajo de autenticación, registro, etc

```
LoginForm               login.html
LoginOKForm             login_ok.html
FatalForm               fatal.html
ExpiredForm             expired.html
RenewForm               renew.html
PassiveRenewForm        renew_pasv.html
RegisterForm            register.html
RegisterOKForm          register_ok.html
RegisterFields          name url description
UpdateForm              update.html
UpdateFields            url description
```

i) Los diferentes mensajes que se presentarán al usuario.

```
LoginGreeting       Bienvenido a la Red.
LoginMissing        Por favor, complete todos los campos
LoginBadUser        El usuario no parece correcto,   inténtelo de
nuevo
LoginBadPass        El usuario y clave escritos no   concuerdan,
escriba nuevamente.
LoginBadStatus      Lo sentimos, no esta registrado en     la Red
RegisterGreeting    Bienvenido!, Introduzca los datos
correspondientes para poder registrarlo.
```

```
RegisterMissing     Nombre, Usuario, y password son  obligatorios
RegisterUserExists  Usuario ya esta registrado.
RegisterBadUser     El usuario no esta correcto, repita
nuevamente
RegisterInvalidPass La contraseña debe ser de 6      caracteres
como mínimo.
RegisterPassNoMatch Las contraseña suministradas     no
concuerda, repita.
RegisterSuccess     Su registro ha sido completado
satisfactoriamente
UpdateGreeting      Introduzca e-mail y contraseña
UpdateBadUser       El mail no parece correcto, repita
nuevamente
UpdateBadPass       El mail y password escritos no   concuerdan,
repita nuevamente.
UpdateInvalidPass   La contraseña debe ser de 6      caracteres
como mínimo.
UpdatePassNoMatch   El mail y password escritos no   concuerdan,
repita nuevamente.
UpdateSuccess       Enhorabuena, su registro se ha   completado
satisfactoriamente.
```

4.3.2 Configuración del Gateway /usr/local/nocat/gateway/nocat.conf

Las líneas que debe tener este archivo son:

a) El nombre del *gateway* que se presenta en el *splash* y las páginas.

```
GatewayName Servidor de Autenticación NoCat
```

b) El modo de operación del *Gateway*. Como se explicó en el Capitulo 3 puede ser: *Captive*, *Passive*, *Open*. Utilizamos el modo *Captive.*

```
GatewayMode Captive
```

c) El archivo .log que nos permitirá controlar todo lo que sucedió en el servidor.

```
GatewayLog  /usr/local/nocat/nocat.log
```

d) El tiempo en Segundos que se le permitirá a un cliente permanecer conectado. Este tiempo puede ser configurado dependiendo de las necesidades.

```
LoginTimeout      86400
```

e) El directorio donde se encuentran alojadas las páginas que utilizará NoCat.

```
DocumentRoot      /usr/local/nocat/gateway/htdocs
```

f) El *splash* de captura del usuario.

```
SplashForm  splash.html
```

g) El *splash* de estado de conexión que deberá permanecer abierto durante la navegación

```
StatusForm  status.html
```

h) La dirección IP del servidor de autenticación

```
AuthServiceAddr   192.168.10.1
```

i) El URL al que será redireccionado el cliente cuando intente acceder al Internet antes de autenticarse.

```
AuthServiceURL  https://$AuthServiceAddr/cgi-bin/login
```

j) El URL al que se redireccionará al cliente después de haber culminado su tiempo de conexión.

```
LogoutURL   https://$AuthServiceAddr/logout.html
```

k) La configuración de las interfaces interna (por donde acceden los clientes) y externa (salida al Internet).

```
ExternalDevice        eth0
InternalDevice        eth1
```

El nombre de las interfaces pueden variar y se debe tener muy en cuenta cuál es cual interfaz ya que si se especifican mal los nombres podrían existir muchos problemas.

l) La dirección IP de la red Interna.

```
LocalNetwork 192.168.10.0/24
```

m) Los puertos TCP a los que se les excluirá el acceso cuando alguien inicie una sesión como un usuario de clase pública.

```
ExcludePorts    25
```

n) El directorio donde se encuentran almacenadas las llaves PGP.

```
PGPKeyPath  /usr/local/nocat/pgp
```

4.3.3 Configuración del archivo /etc/httpd/conf.d/ssl.conf

Lo único que se debe hacer con este archivo es incluir en su contenido la ruta del archivo authserv.conf. La línea que se debe agregar es:

```
Include  /usr/local/nocat/etc/authserv.conf
```

El archivo authserv.conf fue anteriormente copiado de las fuentes del NoCat y lo que hace es indicar a Apache la ruta del alias /cgi-bin/ para que busque en el directorio donde se encuentran los archivos Perl del NoCat y la configuración de ese directorio. Las líneas de este archivo son:

```
ScriptAlias /cgi-bin/ /usr/local/nocat/cgi-bin/
<Directory /usr/local/nocat/cgi-bin>
    SetEnv PERL5LIB /usr/local/nocat/lib
    SetEnv NOCAT    /usr/local/nocat/nocat.conf
</Directory>
```

4.3.4 Configuración del ancho de banda /usr/local/nocat/gateway/bin/throttle.fw

Este archivo sirve para asignar el ancho de banda dependiendo del grupo al que pertenece un usuario. La configuración que tenemos en nuestro archivo es la siguiente:

```
TOTAL_DOWN=3mbit
TOTAL_UP=384kbit
OWNER_DOWN=3mbit
OWNER_UP=384kbit
OWNER_OPTIONS=""
COOP_DOWN=1mbit
COOP_UP=256kbit
COOP_OPTIONS=""
PUBLIC_DOWN=128kbit
PUBLIC_UP=128kbit
PUBLIC_OPTIONS="bounded"
```

4.4 Configuración de los componentes del Sistema de Autenticación

4.4.1 Configuración de las Interfaces

Las interfaces que tiene nuestro sistema de autenticación son 2: la interfaz interna (eth1) que es por donde acceden los usuarios al *gateway*, y la interfaz externa (eth0) que es por donde salen al Internet. Asignamos las direcciones IP, máscara y *gateway* a las dos interfaces. Es conveniente de que cada interfaz pertenezcan a redes diferentes.

	eth0	eth1
Dirección IP	192.188.47.5	192.168.10.1
Máscara de Subred	255.255.255.0	255.255.255.0
Gateway	192.188.47.5	

Además de lo anterior se debe especificar la dirección del DNS que utilizara el servidor.

DNS primario: 127.0.0.1

DNS secundario:192.188.47.2

4.4.2 Configuración del cliente

Lo que hay que configurar en el equipo cliente es que la dirección IP, *gateway* y DNS sea asignada desde el servidor mediante DHCP. Y para conectarse al Internet configurar el *browser* para que no utilice ningún Proxy, sino que se conecte directamente.

4.5 Administración de los usuarios

4.5.1 Creación de la base de datos de los usuarios

Los usuarios estarán registrados en una base de datos MySQL. La base datos como se mostraba en los archivos de configuración tendrá el nombre nocat.

Primeramente iniciamos la base de datos MySQL

```
# /etc/init.d/mysqld start
```

Creamos la base de datos nocat y digitamos una contraseña (la que está en el archivo de configuración es ‘nocatauth’). La contraseña puede ser otra, pero se tendría también que cambiar tanto el archivo de configuración como en algunos de los programas Perl que acceden a la base de datos debido a que ‘nocatauth’ es la que viene por defecto.

```
# mysqladmin create nocat -p
Enter password:
```

Una vez creada la base de datos se procede a la creación de las tablas. Para facilitarnos el trabajo de crear las tablas, Nocat cuenta con un *script* que contiene las rutinas para realizar dicha tarea. Este *script* está en NoCatAuth-0.82/etc/nocat.schema; entonces añadimos la estructura a la base de datos nocat.

```
# mysql nocat < nocat.schema –p
```

Asignamos todos los privilegios de *root* de todas las tablas de la base de datos nocat al usuario nocat.

```
# mysql –u root –pcontrasena
mysql> grant all on nocat.* to nocat@localhost    identified
by ‘nocatauth’;
mysql> flush privileges;
```

4.5.2 Ingreso de los usuarios

La ejecución de la línea que esta a continuación crea el usuario joseph y le asigna la contraseña ‘micontrasena’

```
# /usr/local/nocat/bin/admintool -c joseph contrasena
```

Esta línea lo que hace es insertar un registro en la tabla *members* de la base de datos nocat.

4.5.3 Asignación de privilegios

En el capítulo anterior se habló de las tipos de usuario que tiene NoCat Auth por defecto, *Owner, Members y Public*. Cada uno cuenta con distintos privilegios sobre la red Para la asignación de privilegios simplemente agregamos al usuario a un grupo determinado, en este caso al grupo *members*.

```
# /usr/local/nocat/bin/admintool -a joseph members
```

La línea anterior inserta un registro en la tabla *network*, que es donde están registrados qué usuario pertenece a qué grupo de la base de datos NoCat.

4.6 Conclusión

Luego de haber realizado la configuración completa del sistema de configuración implementando en forma práctica toda la teoría investigada en los capítulos anteriores

hemos comprobado los grandes beneficios que nos ofrece el software NoCat Auth, concluyendo que es muy útil para realizar autenticación de usuarios a una red. Gracias a la implementación práctica realizada en este capítulo reafianzamos los conocimientos teóricos.

CAPITULO 5: PRUEBAS

Introducción

Una vez instalado y configurado completamente el software NoCat Auth, procedemos a realizar las pruebas para comprobar el correcto funcionamiento del mismo. Las pruebas se realizarán en el laboratorio de Internet de la Universidad del Azuay. En caso de existir errores explicaremos el porque y como se solucionarán. El objetivo de este capítulo es garantizar la calidad del sistema de autenticación, ofreciendo a los usuarios de la red la confidencialidad y confiabilidad de la información.

5.1 Prueba de las conexiones físicas.

Para probar si todo esta bien, conectamos una computadora a la interfaz interna del servidor utilizando un cable cruzado, y la interfaz externa esta conectada a la LAN de la Universidad con salida al Internet. (ver Anexo 4).

Probamos la comunicación haciendo un ping desde el servidor a la máquina cliente y viceversa, pero no hay respuesta. Verificamos las posibles causas, que pueden ser:

- Una falla en el cable: para comprobar, cambiamos el cable y volvemos a ejecutar el ping.
- El cable de red puede estar conectado en la interfaz incorrecta (externa): para esto conectamos en la otra interfaz (interna) y volvemos a ejecutar el ping.
- Puede que el servicio dhcpd no este levantado: verificamos el estado del servicio digitando:

```
# service dhcpd status.
```

Si, ese parece ser el problema, el mensaje que apareció fue ‘dhcpd está parado’ iniciamos el servicio y volvemos a ejecutar el ping y hubo respuesta.

```
# service dhcpd start
```

Ahora hacemos un *ping* desde nuestro servidor a una página *web* cualquiera, por ejemplo a www.google.com y si hay respuesta.

5.2 Iniciar Servicios previos

Antes de iniciar nuestro *Gateway* Nocat debemos iniciar los servicios de Linux de los cuales necesita NoCat para su funcionamiento. Estos servicios son:

- httpd: Inicia el servidor web Apache.

  ```
  # service httpd start
  ```

- dhcpd: Inicia el servidor de asignación dinámica de direcciones IP.

  ```
  # service dhcpd start
  ```

  Comprobamos en el computador cliente que se hayan asignado correctamente las direcciones IP.

- named: Inicia el servidor de nombres de dominios.

  ```
  # service named start
  ```

  Probamos si esta resolviendo correctamente los nombres de dominio para eso tratamos de ver si resuelve el nombre de la universidad del azuay como prueba:

  ```
  # Nsloockup uazuay.edu.ec
  ```

- mysqld: Inicia la base de datos Mysql.

  ```
  # /etc/init.d/mysqld start
  ```

Todos estos servicios son muy importantes para el funcionamiento de Nocat. Si uno de ellos no se inicia NoCat Auth no funcionará correctamente.

5.3 Iniciando Nocat Auth

Teniendo configurado todo, y levantado los servicios previos, iniciamos el *gateway* NoCat.

```
# /usr/local/nocat/gateway/bin/gateway start
```

Para comprobar que se levanto correctamente el servicio del *gateway*, abrimos en un *browser* la pagina de estado del mismo http://localhost:5280/status. La pagina mostrada es correcta (ver Anexo 5). En el caso de no estar levantado el servicio no se muestra la página y se presenta un mensaje de error (ver Anexo 6).

5.4 Prueba de la Base de Datos

a. Iniciamos el servicio mysqld.

```
# /etc/init.d/mysqld start
```

b. Nos conectamos a la base de datos

```
# mysql –u nocat –pnocatauth
```

c. Ponemos en uso la base de datos nocat

```
mysql> use nocat;
```

d. Verificamos que estén creadas todas las tablas.

```
mysql> show tables;
```

Las tablas mostradas son: *eventlog, hardware, member, network y node*. Estas son todas las tablas de la estructura de nocat. Las más indispensables son *member* y *network*.

e. Verificamos las tablas principales.

```
mysql> desc member;
mysql> desc network;
```

Todas las tablas están bien creadas (ver Anexo 7)

f. Lo siguiente que hacemos es crear el usuario ‘paula’ con contraseña ‘prueba’ de la forma que se explicó en el capitulo anterior y comprobar que este se haya creado en la base de datos en la tabla *member*, realizando un *select* de la tabla luego de habernos conectado a la base de datos nocat.

```
mysql> select * from member where  login = ‘paula’
```

La sentencia devolvió una fila, eso quiere decir que el usuario se creo correctamente en la base de datos (ver Anexo 8).

g. Ahora asignamos el usuario ‘paula’ al grupo *members* de la forma que se explicó en el capítulo anterior, y comprobamos que se haya creado en la base de datos nocat en la tabla *network* el registro correspondiente. Para eso realizamos un *select* de la tabla luego de habernos conectado a la base de datos nocat.

```
mysql> select * from network where  login = ‘paula’
```

La sentencia devolvió una fila, eso quiere decir que el usuario se asignó correctamente al grupo *members* en la base de datos (ver Anexo 9).

Si se desea agregar un usuario, modificar sus datos, o visualizarlos mediante una interfaz gráfica, existe un módulo de NoCat para Webmin que permite hacer todo esto. Además este módulo permite también realizar las configuraciones de los archivos de NoCat Auth mediante una interfaz gráfica *Web*.(ver Anexo 10)

5.5 Prueba de Conexión de cliente al Gateway NoCat

Para probar tratamos de abrir una página *web* (www.google.com.ec); al tratar de abrir la página el *gateway* nos redirige a la página de *login* para autenticarnos.(ver Anexo 11).

Ahora digitamos el *login* y *password* que creamos de prueba en la base de datos, pero para ver que todo anda bien, nos vamos a equivocar a propósito al digitar la contraseña, y la página presenta un mensaje indicando que la contraseña y el usuario no concuerdan.(ver Anexo 12). Nuevamente digitamos la contraseña pero correctamente y se nos redirige por unos segundos a una pagina de bienvenida (ver Anexo 13), y se abre otra ventana que indica el tiempo de duración de nuestra conexión con un botón para

terminar la misma.(ver Anexo 14). Finalmente nos redirige a la página solicitada en un inicio (ver Anexo 15).

En la página de bienvenida existe un *link* a la página de registro para aquellos usuarios que no poseen de una cuenta (ver Anexo 16). Cabe indicar que para acceder a la red no basta con registrarse; el administrador deberá asignar a este usuario a un grupo de usuarios para que pueda tener el acceso. Para prueba vamos a registrarnos como prueba_reg@uazuay.edu.ec (ver Anexo 17) . Luego en el servidor agregamos al usuario al grupo *members* y realizamos la misma prueba que hicimos anteriormente para autenticarnos pero esta vez como usuario prueba_reg@uazuay.edu.ec y al igual que en la prueba anterior la autenticación resulta exitosa.(ver Anexo 18).

NoCat posee un archivo donde se registra todo lo sucedido en el servidor de autenticación, el archivo se llama nocat.log y se encuentra ubicado en la ruta que se digitó anteriormente en el archivo nocat.conf en la variable GatewayLog.

5.6 Conclusión

Luego de haber revisado el funcionamiento del software NoCat Auth podemos decir que el sistema de autenticación instalado y configurado es confiable y estable, permitiendo el acceso solamente de los usuarios que cuenten con una cuenta en el servidor y cuyos permisos lo permitan. También podemos concluir que el hecho de que hayan existido errores nos ha enriquecido aún más los conocimientos ya que gracias a ellos hemos descubierto y aprendido más sobre ciertos aspectos del sistema.

Es importante resaltar que ningún sistema es completamente perfecto ni seguro, siempre existe un margen de error aceptable y el sistema instalado y configurado en este capítulo se encuentra dentro de dicho margen.

CAPITULO 6: CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Al culminar la monografía nos quedan varias lecciones y conclusiones como son:

- En la actualidad existen diferentes formas de atacar contra las seguridades de las redes, es por esto que los administradores de las mismas están investigando e implementando mecanismos que protejan la integridad de los datos, ya que ningún sistema es totalmente seguro.

- NoCat Auth nos ofrece varias ventajas y opciones de configuración para autenticar las redes, creando cuentas de usuarios y asignándoles permisos de utilización de los diferentes recursos de red como: ancho de banda.

- Algunos productos de *software* para la autenticación en las redes son gratuitos, brindándonos la opción de que nadie se quede sin proteger sus redes, para evitar que los costos y licencias sean un pretexto para prescindir de ellos.

- Hemos aprendido a que el trabajo en grupo es capaz de alcanzar grandes logros como es de ejemplo el grupo Apache.

- Podemos terminar concluyendo que el leer e investigar con la ayuda de buenas fuentes de información da grandes resultados.

6.2 Recomendaciones

- Al momento de revisar las seguridades en una red se debe considerar tanto las amenazas internas como las externas ya que en la mayoría de las veces las primeras son más peligrosas que las segundas puesto que son personas de la misma empresa que cuentan con accesos a las redes y que sus ataques no son fácilmente detectados.

- Debemos contar con las precauciones necesarias para evitar los diferentes tipos de ataques que existen en una red tanto en la parte del *hardware* (puertos) como la parte del *software*.

- Es recomendable que para todo tema que se trate sobre todo en el área de la informática se complemente con la parte práctica, ya que es muy importante para aclarar muchas dudas que se presentan en la parte teórica.

- Se recomienda mantenerse siempre informados y en constante actualización de las nuevas formas de ataques y de las posibles soluciones, estando así preparados para cualquier dificultad que se nos presente.

- Luego de culminar podemos recomendar el uso del sistema de autenticación NoCat Auth debido a sus diferentes ventajas, es sencillo y de código libre que permite una mayor flexibilidad.

CAPITULO 7: BIBLIOGRAFÍA

NOCATNET. NoCatNet [en línea]. [consulta: 03 de diciembre de 2005]. Disponible en World Wide Web:<http://nocat.net>

DIAZ, Toni. NoCatBox HOWTO v1.4 [en línea]. Septiembre 2003 [consulta: 03 de diciembre de 2005]. Disponible en versión PDF en Internet: <http://blyx.com/public/wireless/nocatbox/nocatbox-howto-es.pdf>

PHPBB GROUP. Los Cuadernos de HackxCrack [en línea]. 2001 [consulta: 06 de enero de 2006]. Disponible en Web: <http://www.hackxcrack.com/phpBB2/viewtopic.php?p=187386&start=#187386>

ALTADILL IZURA, Pello Xabier. IPTables Manual Práctico [en línea]. [consulta: 06 de enero de 2006]. Disponible en Web:<http://es.tldp.org/Manuales-LuCAS/doc-iptables-firewall/doc-iptables-firewall-html/#2>

KOCH, Werner. GNUPG [en línea]. 3 de enero del 2004 [consulta: 10 de enero del 2006]. Disponible en Web:<http://www.gnupg.org/(es)/index.html>

AMBROSI, Viviana. Cómo iniciar nuestro grupo de trabajo con GnuPG [en línea]. [consulta: 10 de enero del 2006]. Disponible en Web:<http://www.linti.unlp.edu.ar/tiki-read_article.php?articleId=142>

GARCIA, Pedro Luis. Configurar un HOTSPOT rápidamente con portal captivo chillispot [en línea]. 16 de julio del 2004 [consulta: 10 de enero del 2006]. Disponible en Web: <http://www.linuca.org/impresion.phtml?nIdNoticia=288>

CHILLISPOT. ChilliSpot - Open Source Wireless LAN Access Point Controller. Spice up your HotSpot with Chilli [en línea]. [consulta: 10 de enero del 2006]. Disponible en Web: <http://chillispot.org/index.html>

CHILLISPOT. ChilliSpot - Open Source Wireless LAN Access Point Controller. Spice up your HotSpot with Chilli [en línea]. [consulta: 10 de enero del 2006]. Disponible en Web: <http://chillispot.org/features.html>

ACMHUNTER. Configurando e Instalando el Kernel de Linux [en línea]. 07 de mayo del 2004 [consulta: 11 de enero del 2006]. Disponible en Web: <http://www.aqpglug.org.pe/documentos/09050442404.html>

EDUARDO, Basado en el artículo original GAT is MySQL? MySQL AB. ¿Qué es MySQL? [en línea]. 26 de agosto del 2002 [consulta: 12 de enero del 2006]. Disponible en World Wide Web: <http://www.mysql-hispano.org/page.php?id=2>

CASARES, Claudio. Tutorial de SQL [en línea]. 07 de septiembre del 2004 [consulta: 12 de enero del 2006]. Disponible en Web: <http://www.maestrosdelweb.com/editorial/tutsql1/>

CORTES, Carlos. Ponle un Firewall a tu Linux. Iptables [en línea].19 de Septiembre del 2001 [consulta: 13 de enero del 2006]. Disponible en Web: <http://bulma.net/body.phtml?nIdNoticia=861>

GUIARTE MULTIMEDIA S.L. ¿Qué es el DNS? [en línea]. [consulta: 13 de enero del 2006]. Disponible en Web: <http://www.desarrolloweb.com/faq/50.php>

CORTES, Carlos. BULMA [en línea] . 27 de Septiembre 2001 [consulta: 22 de febrero del 2006]. Disponible en Web: <http://bulma.net/body.phtml?nIdNoticia=873>

CIBERAULA, Asociación Española de Internet. Una Introducción a Apache [en línea].Madrid, España 2004 [consulta: 14 de enero del 2006]. Disponible en Web: <http://linux.ciberaula.com/articulo/linux_apache_intro/>

SÁNCHEZ GONZÁLEZ, Juan Bautista. El servidor HTTP Apache [en línea]. [consulta:14 de enero del 2004]. Disponible en Web: <http://www.geocities.com/SiliconValley/Campus/2208/WEapache.html#dire>

Wikimedia Foundation. Servidor HTTP Apache [en línea]. 27 de enero del 2006, [consulta: 15 de febrero del 2006]. Disponible en Web: <http://es.wikipedia.org/wiki/Servidor_HTTP_Apache>

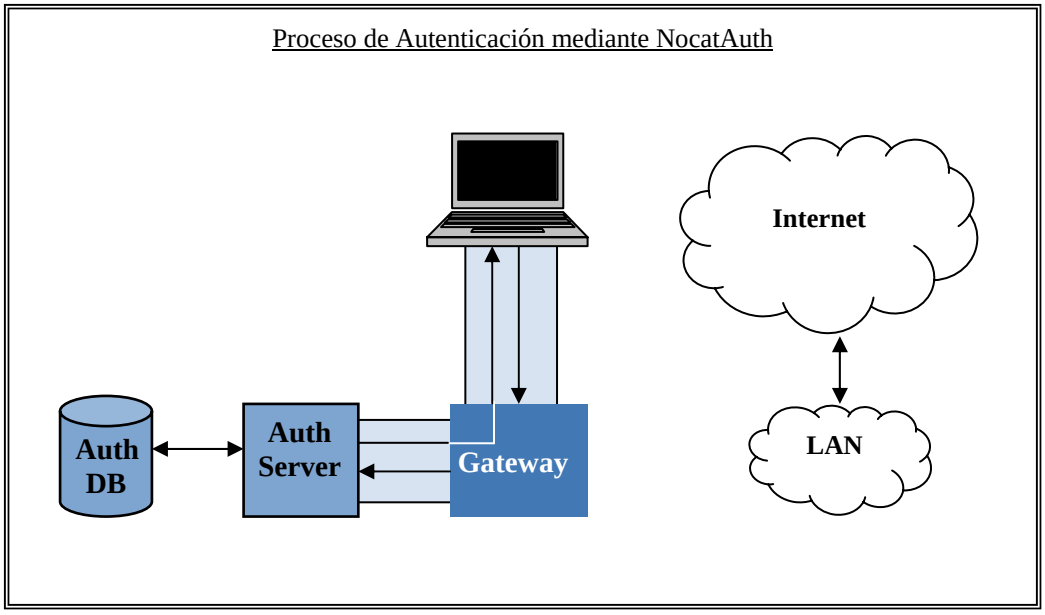
BankHacker. Tecnología de BankHacker [en línea]. 2000-2005, [consulta: 15 de febrero del 2006]. Disponible en Web: <http://www.bankhacker.com/tecnologia.phtml>

GARCIA CASTELLANO, Javier., CASTILLO VALDIVIESO, Pedro Angel., MELERO GUERVÓS, Juan Julián. Tutorial de Introducción a Perl [en línea]. Julio 2002, [consulta: 16 de febrero del 2006]. Disponible en World Wide Web: <http://flanagan.ugr.es/perl/index2.htm>

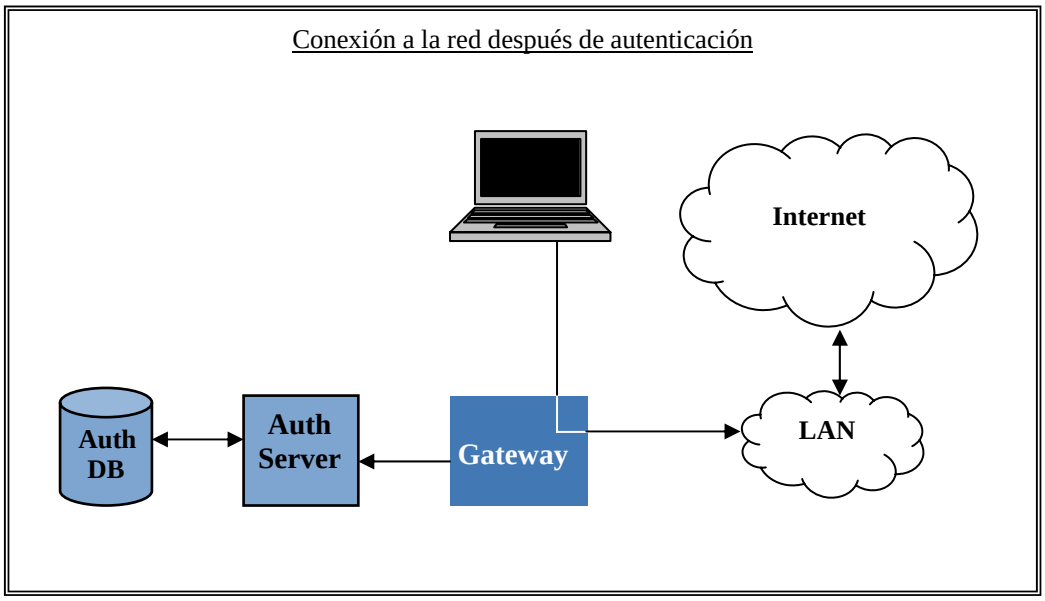
SOCHER,Guido. Perl Parte I [en línea]. 23 de mayo de 1999, [consulta: 16 de febrero del 2006]. Disponible en Web: <http://es.tldp.org/LinuxFocus/pub/mirror/LinuxFocus/Castellano/September1999/article114.html>

ANEXOS

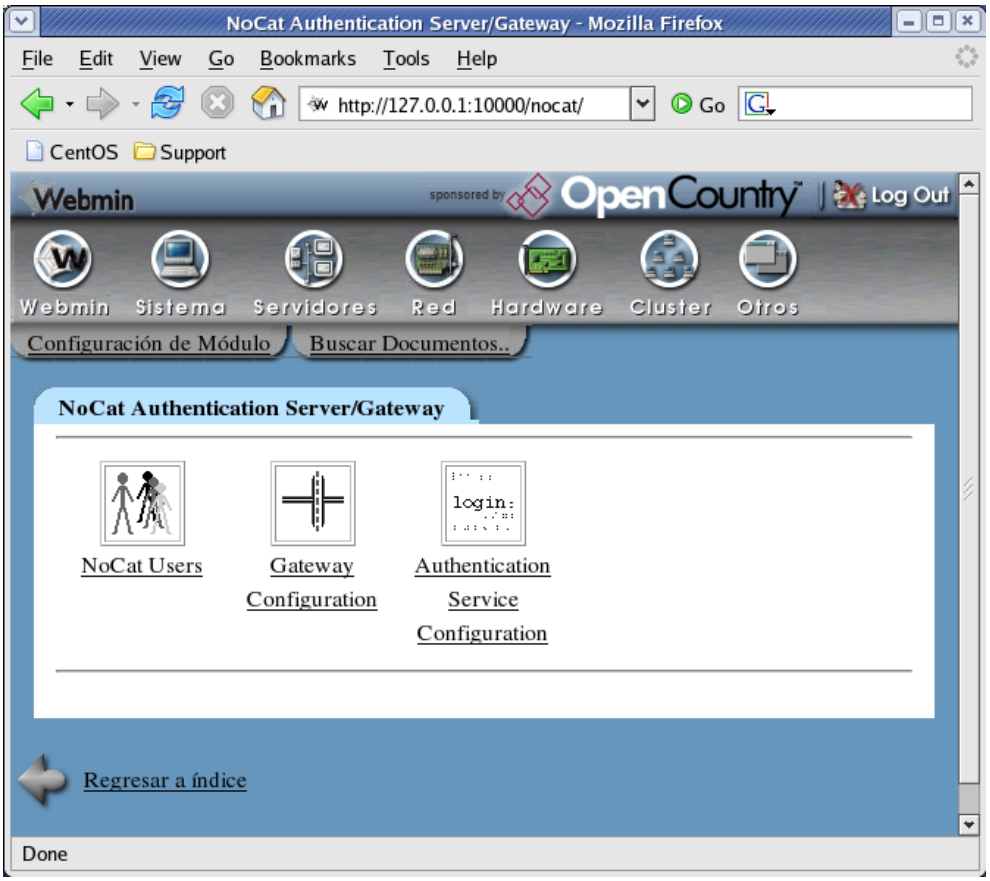
Anexo 1. Proceso de Autenticación mediante Nocat Auth



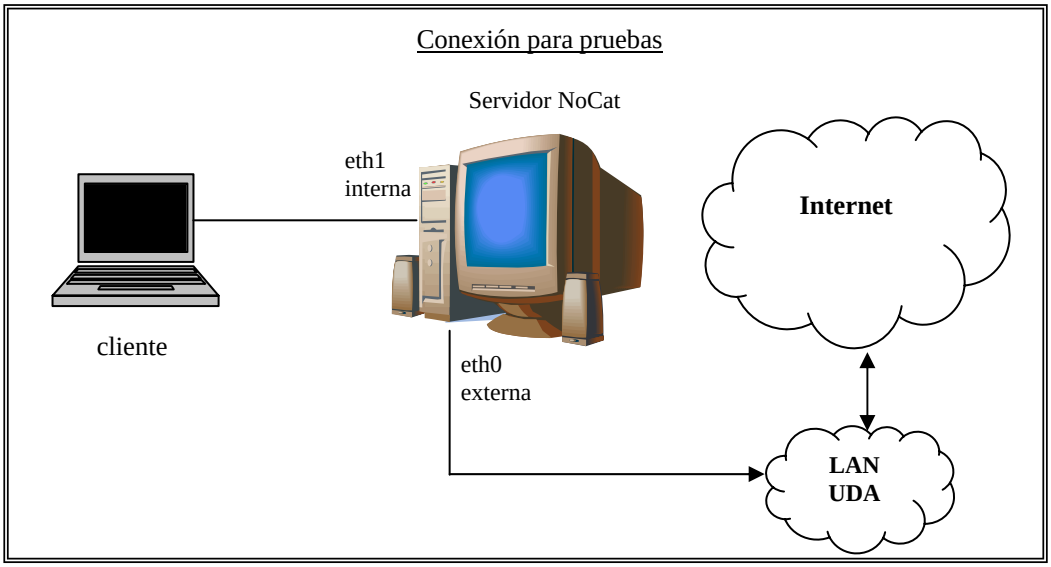
Anexo 2. Conexión a la red después de la autenticación



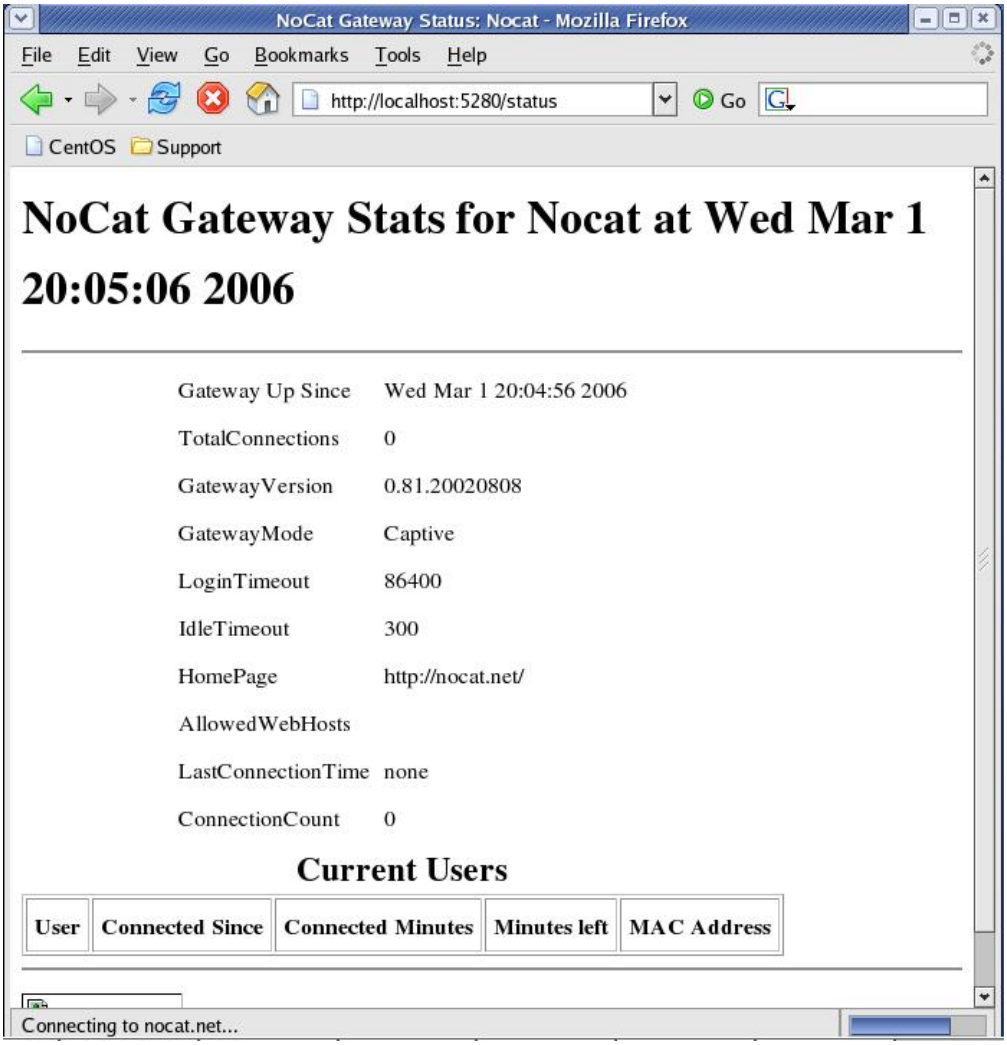
Anexo 3. Módulo NoCat para Webmin



Anexo 4. Estructura de conexión para la realización de pruebas



Anexo 5. Página de estadísticas de funcionamiento del Gateway NoCat



Anexo 6. Mensaje de alerta del browser cuando no esta levantado el Gateway NoCat.

Anexo 7. Estructura de las tablas member y network de la base de datos nocat.

```
mysql> desc member;
+-------------+---------------------+------+-----+-------------------+-------+
| Field       | Type                | Null | Key | Default           | Extra |
+-------------+---------------------+------+-----+-------------------+-------+
| url         | varchar(255)        | YES  |     | NULL              |       |
| description | text                | YES  |     | NULL              |       |
| created     | datetime            | YES  |     | NULL              |       |
| modified    | timestamp           | YES  |     | CURRENT_TIMESTAMP |       |
| status      | tinyint(3) unsigned | YES  |     | NULL              |       |
| login       | varchar(250)        |      | PRI |                   |       |
| pass        | varchar(255)        |      |     |                   |       |
| name        | varchar(255)        | YES  |     | NULL              |       |
+-------------+---------------------+------+-----+-------------------+-------+
8 rows in set (0.00 sec)

mysql> desc network;
+----------+-------------+------+-----+-------------------+-------+
| Field    | Type        | Null | Key | Default           | Extra |
+----------+-------------+------+-----+-------------------+-------+
| login    | varchar(250) |      | PRI |                   |       |
| network  | varchar(250) |      | PRI |                   |       |
| admin    | char(1)     | YES  |     |                   |       |
| created  | datetime    | YES  |     | NULL              |       |
| modified | timestamp   | YES  |     | CURRENT_TIMESTAMP |       |
+----------+-------------+------+-----+-------------------+-------+
5 rows in set (0.00 sec)
```

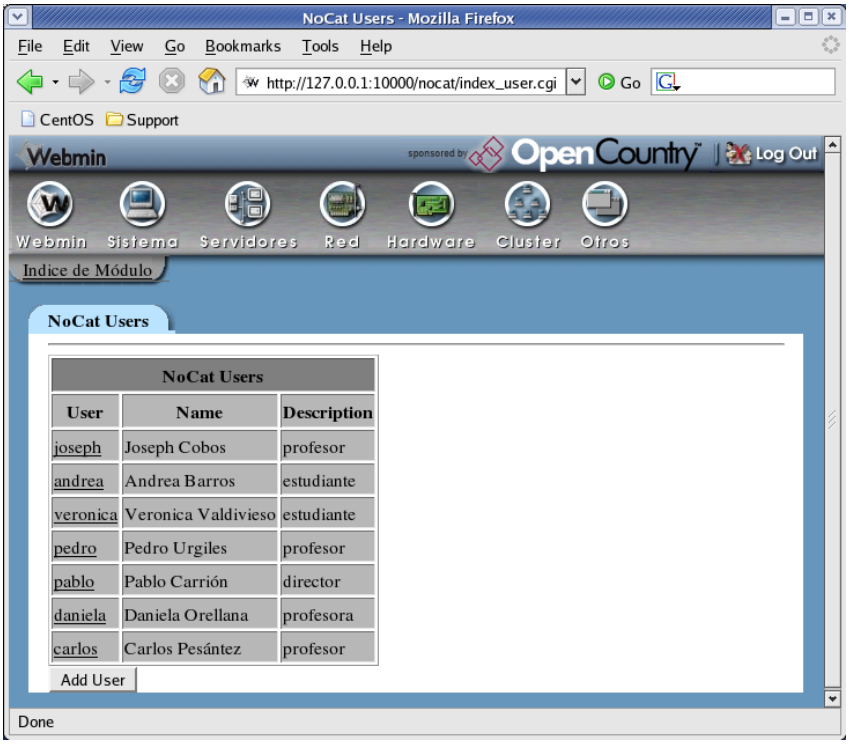
Anexo 8. Registro de la tabla member del usuario ‘paula’.

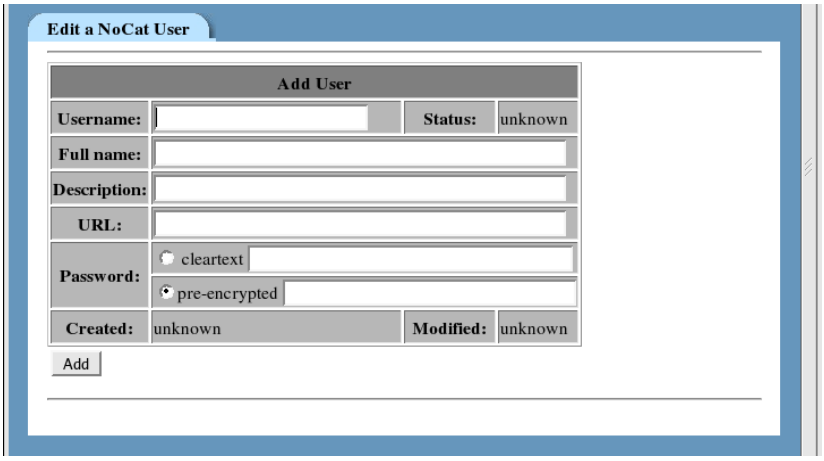
```
mysql> select * from member where login='paula';
+------+-------------+---------+---------------------+--------+--------+----------------------+------+
| url  | description | created | modified            | status | login  | pass                 | name |
+------+-------------+---------+---------------------+--------+--------+----------------------+------+
| NULL | NULL        | NULL    | 2006-02-28 22:51:39 |   NULL | paula  | yJO61okntFfb7TlGDmr9Yg | NULL |
+------+-------------+---------+---------------------+--------+--------+----------------------+------+
1 row in set (0.00 sec)
```

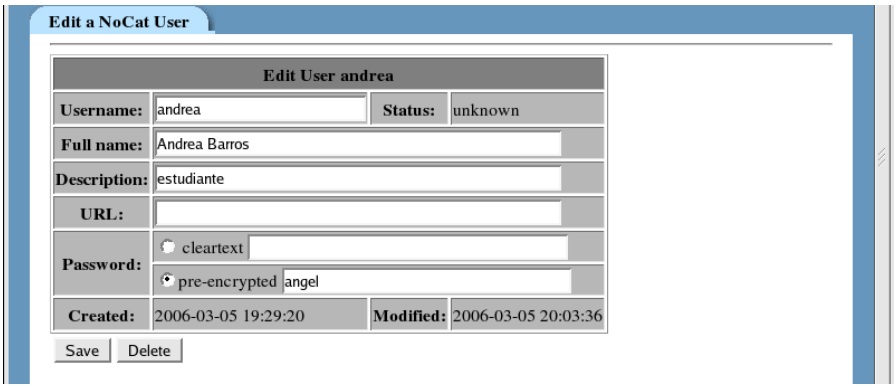
Anexo 9. Registro de la tabla network del usuario ‘paula’.

```
mysql> select * from network  where login='paula';
+--------+---------+-------+---------+---------------------+
| login  | network | admin | created | modified            |
+--------+---------+-------+---------+---------------------+
| paula  | members | 0     | NULL    | 2006-02-28 22:52:31 |
+--------+---------+-------+---------+---------------------+
1 row in set (0.00 sec)
```

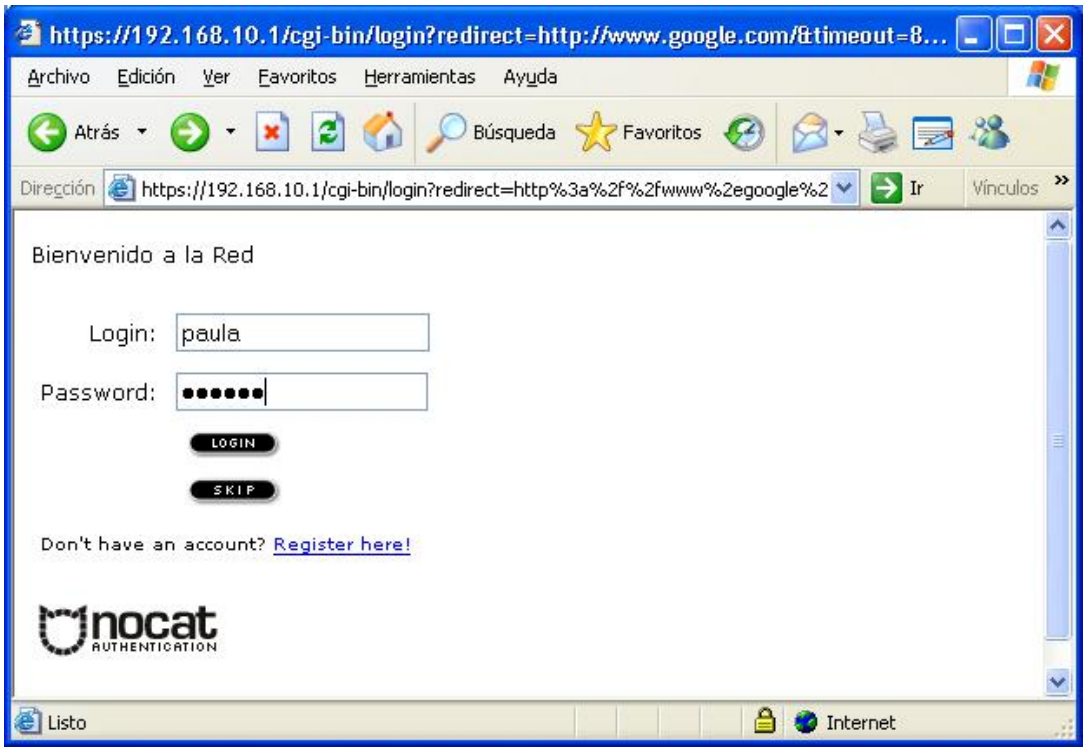
Anexo 10. Administración de usuarios con el módulo NoCat de Webmin



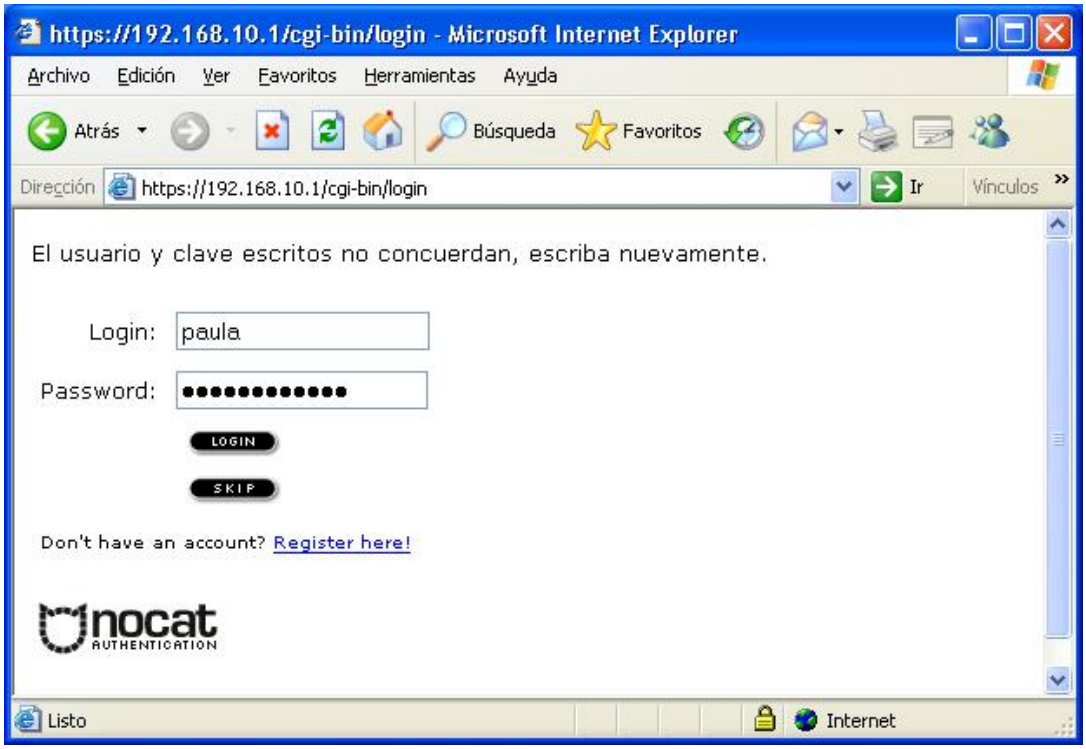




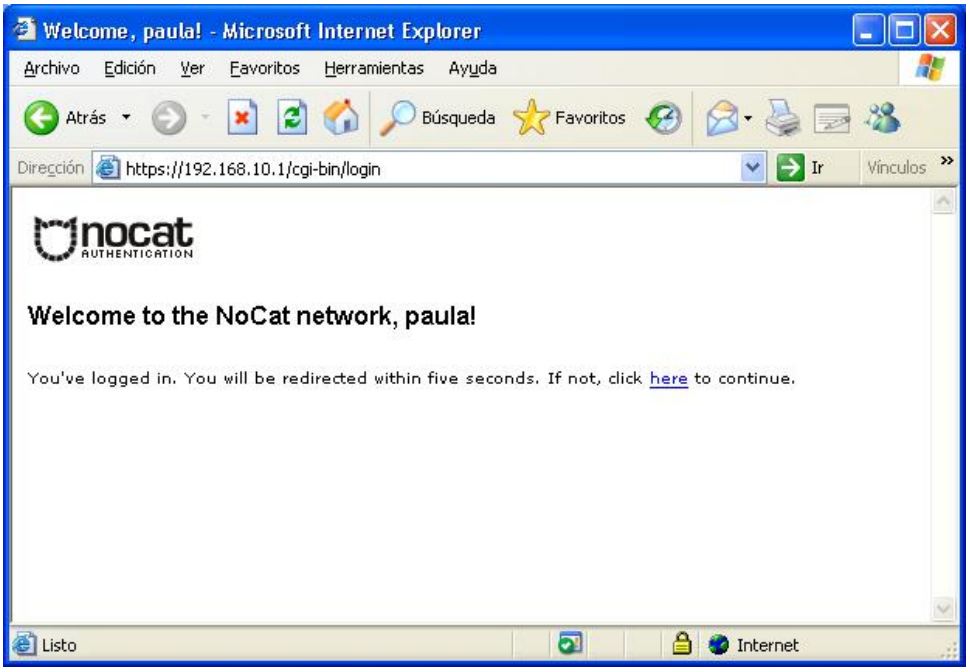
Anexo 11. Página de login para autenticación.



Anexo 12. Mensaje de no coincidencia de usuario y contraseña.



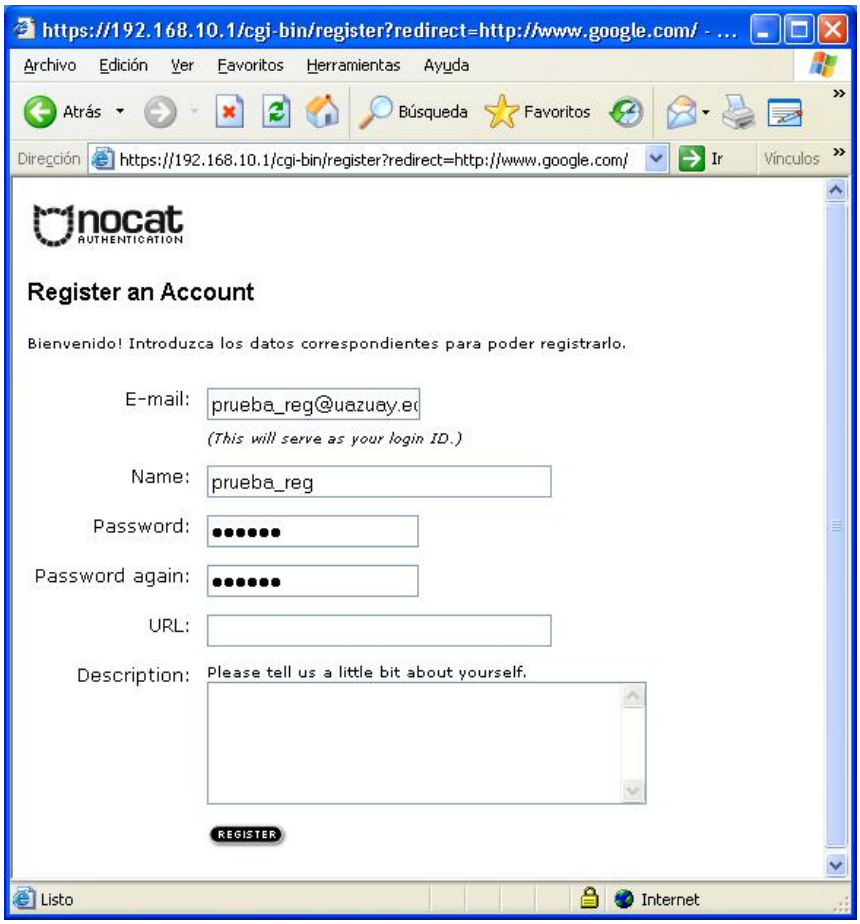
Anexo 13. Página de bienvenida de NoCat al usuario paula



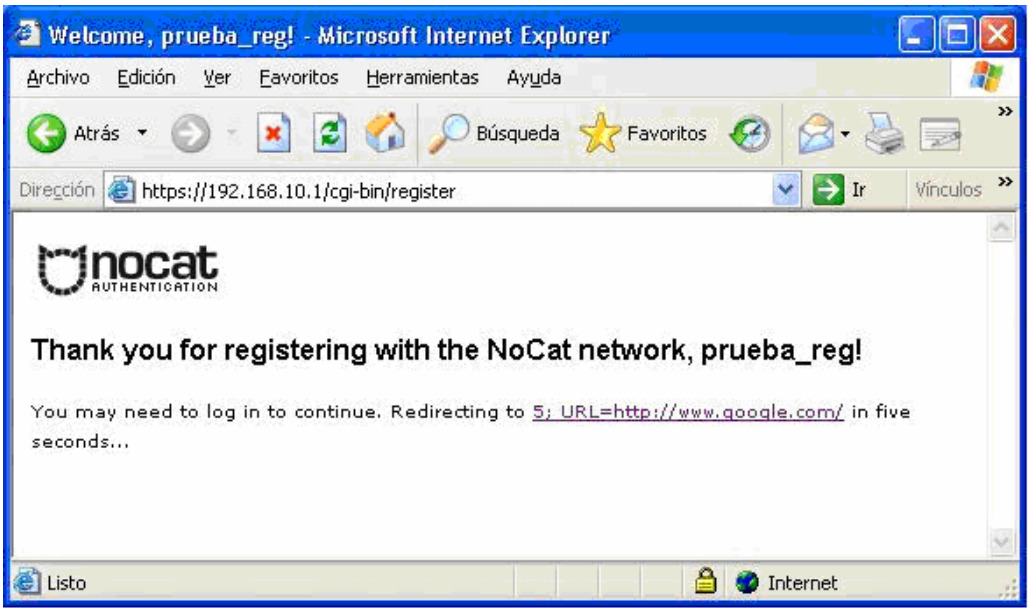
Anexo 14. Página de estado de conexión del cliente.

Anexo 15. Página www.google.com.ec redirigida luego de autenticación.

Anexo 16. Página de registro de un usuario nuevo.



Anexo 17. Página de registro exitoso para el usuario prueba_reg@uazuay.edu.ec



Anexo 18. Página de bienvenida de NoCat al usuario prueba_reg@uazuay.edu.ec