UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Implementación de un portal cautivo para la autenticación de usuarios en redes usando herramientas de software libre

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

JORGE ÁNGEL HERNÁNDEZ LÓPEZ

DIRECTOR DE TESIS:

M. C. ALEJANRO VELÁZQUEZ MENA

CIUDAD UNIVERSITARIA, 2012

Agradecimientos

A mis padres por el apoyo incondicional, por sus consejos y enseñanzas, por el amor, cariño y comprensión que me han brindado. Y sobre todo porque me han dejado la mejor herencia en mi vida, la educación.

A mi papá por darme la vida, por ser un guía en mi vida, por darme consejos cuando los necesité, por su apoyo, respeto, compresión, por los valores que me ha inculcado, por estar a mi lado en todo momento de mi vida. Por ser quien es.

A mi mamá por darme la vida, por su amor, cariño y apoyo, por preocuparse por mi desarrollo, por formar una persona de bien, por su motivación constante, por brindarme unos hermanos maravillosos. Por ser mi madre.

A mi abuelita por brindarme parte de su vida, por cuidarme desde que nací, por procurarme día y noche, por todos los sacrificios que ha realizado en su vida para verme crecer, por su enorme paciencia, por apoyarme incondicionalmente, por haberme educado todo este tiempo.

A mi hermano Rogelio, por haberme enseñado muchas cosas, por ser un ejemplo a seguir, porque siempre consigue lo que se propone, por todos los momentos que hemos compartido juntos.

A mi hermanita Angélica, por haberme traído esa alegría cuando vino al mundo, por tener un motivo por el cual ser un ejemplo a seguir.

Agradecimientos

A mi motci por brindarme su apoyo incondicional, así como su amor, cariño y compresión en todo momento, por sus enseñanzas, por todos los momentos que hemos compartido, gracias por estar a mi lado Circe, TAM.

A Alejandro Velázquez Mena, por darme la oportunidad de trabajar bajo su mando, por compartirme sus conocimientos, por las enseñanzas día a día y por brindarme su amistad.

A Julio, Edgar y Heriberto, por compartir sus conocimientos conmigo, por apoyarme cuando requerí de su ayuda, por tener un buen ambiente de trabajo, por su amistad.

A mis amigos por su apoyo, compañía y amistad, por los momentos inolvidables que hemos compartido durante la carrera.

A Martha por su amistad durante todos estos años, por su apoyo y motivación para que terminara la tesis, por todos los momentos que hemos compartido desde que nos conocimos.

A la Universidad Nacional Autónoma de México, mi alma mater, por darme la oportunidad de pertenecer a esta magnífica institución, por permitirme lograr esta meta en mi vida, por tener grandes profesores que se esmeraron por darme lo mejor de cada uno de ellos para tener una excelente formación profesional y personal.

Índice

Índice de figuras

Capítulo I.

Capítulo II.

Capítulo III.

Capítulo IV.

Introducción

Actualmente el uso de las redes ha aumentado tanto en instituciones educativas como en empresas privadas o gubernamentales. Dependiendo de la infraestructura que se use se deben de tomar medidas que permitan tener un control de acceso a los recursos de forma eficiente.

Con el desarrollo de la tecnología y la evolución de las redes inalámbricas, también se han desarrollado herramientas para vulnerar la seguridad de las anteriores, hoy en día usar claves WEP, WPA, WPA2 ya no garantiza tener una red sólo con usuarios autenticados.

Debido a este problema se han implementado otros métodos de autenticación en las redes inalámbricas, el más común es la autenticación por RADIUS, que nos permite la autenticación mediante certificados, archivos de usuario y contraseña, conexión a base de datos, inclusive se puede conectar con el servicio LDAP.

Por otra parte existen redes públicas como en aeropuertos, restaurantes, librerías, etc., que autenticar a los usuarios en una red es un proceso complejo, ya que la mayoría de usuarios son móviles, es decir, sólo usan los recursos por un tiempo limitado y después dejan de usarlos. Los usuarios en este tipo de redes no son constantes, si se intenta repartir credenciales de autenticación a cada uno se vuelve una actividad demasiado compleja.

Algunos usuarios de la red pueden ser fijos, como trabajadores, este tipo de usuarios si pueden utilizar credenciales para autenticarse y acceder a los recursos. Lo anterior genera dos esquemas:

- Uno que se puede controlar sin problemas,
- Otro en el que el control se vuelve demasiado complejo.

¿Qué pasaría si en una red se requiere permitir a los usuarios consultar cierta información de Internet aunque no estén autenticados?, ninguno de los métodos de autenticación mencionados anteriormente permite tener este comportamiento.

Un problema relacionado con lo anterior, es fortalecer el control de acceso a los recursos de red del Laboratorio de Computación Sala C, ya que actualmente se limita a un acceso físico, presentar una identificación con fotografía, pero eso no garantiza que realmente esa persona este registrada para el uso de la sección de nodos para dispositivos portátiles en el laboratorio.

El objetivo de este trabajo es implementar un portal cautivo, que es un mecanismo de autenticación, que puede ser utilizado en diferentes escenarios. Así como la implementación de un portal cautivo para la autenticación de usuarios en la red del Laboratorio de Computación Sala C de la División de Ingeniería Eléctrica (DIE) de la Facultad de Ingeniería – UNAM, principalmente en la sección de nodos para dispositivos portátiles.

Un portal cautivo es una herramienta que permite la autenticación de usuarios mediante un portal web, por otro lado permite a los usuarios que no se han autenticado consultar ciertos sitios web, de ahí en fuera si se requiere consultar algún otro sitio, automáticamente son redirigidos al portal web que realiza la autenticación, una vez autenticado, el usuario podrá usar los servicios de red normalmente.

Para realizar la implementación de un portal cautivo es importante tener una arquitectura de red que cubra las necesidades que se tienen en el Laboratorio de Computación Sala C. Para diseñar dicha arquitectura se tiene que hacer un análisis sobre la infraestructura de red existente y las modificaciones que se requieren.

Por otra parte, existen muchas herramientas que permiten la implementación de un portal cautivo, por lo que se tuvo que hacer un análisis de las principales herramientas para seleccionar la solución más adecuada para nuestras necesidades.

Dado que el tema de seguridad es un área de estudio muy amplia, este trabajo sólo se enfocará en el área de servicios de seguridad, que incluye autenticación, control de acceso, autorización, pseudónimos, anonimatos, gestión de derechos digitales y preservación de la privacidad de los protocolos[1].

Este trabajo de tesis se enfocará en la implementación de una herramienta para realizar la autenticación de usuarios en redes, apoyándose de protocolos robustos y por consiguiente generar una arquitectura más compleja. El trabajo se compone de cuatro capítulos que se encuentran debidamente relacionados.

En el **capítulo I** se explica brevemente los protocolos de autenticación a utilizar, la definición de la herramienta portal cautivo, así como protocolos que son requeridos para la implementación de dicha herramienta.

En el **capítulo II** se muestra el análisis de las arquitecturas y herramientas existentes para llevar a cabo el proceso de autenticación, de la misma forma se da un panorama de las distintas herramientas de portal cautivo. También se hablará sobre el funcionamiento de la herramienta de portal cautivo a utilizar, así como del diseño de la arquitectura de red.

En el **capítulo III** se describen los pasos de instalación y configuración de la herramienta seleccionada para implementar el portal cautivo, respetando la arquitectura de red seleccionada en el capitulo anterior.

En el **capítulo IV** se muestran las pruebas realizadas con los distintos dispositivos al sistema de autenticación creado y los resultados obtenidos.

[1] La clasificación de temas se obtuvo de Security and privacy, Computing Classification System, 2012 Revision - Association for Computing Machinery.

Capítulo I. Marco teórico

I.I Autenticación

Si toda la información fuera pública, no habría necesidad de utilizar algún mecanismo para acceder a ésta, pero existe información sensible que no puede ser de dominio público, en algunas ocasiones, el acceso a la información se vende, por lo que tiene que existir algún proceso que proteja dicha información, este proceso es conocido como: autenticación.

Hoy en día toda persona tiene contacto o usa métodos de autenticación, ya sea al revisar su correo electrónico, acceder a una cuenta de red social, conectarse a la red privada del trabajo, al conectarse a internet y muchas otras actividades cotidianas. Todas las acciones anteriores involucran un proceso de autenticación, pero, ¿qué es lo que sucede durante este proceso? El dispositivo involucrado, ya sea una computadora u otro dispositivo, tiene que tener acceso a un conjunto de procesos y protocolos para verificar su identidad, saber si se tiene permitido el acceso y finalmente indicarnos el resultado.

La autenticación es un proceso en el cual se verifica la identidad de una persona o equipo. La palabra "autenticación" se define como el acto o proceso de calificar algo como válido o auténtico.

Por lo regular la forma más común de autenticación está basada en una combinación de un identificador y una contraseña, esta contraseña debe de ser secreta y sólo el usuario al que se le asignó debe de tener conocimiento de ella, ya que con ésta se podrán acceder a ciertos recursos.

La autenticación es una parte fundamental del ***control de acceso***, que es la habilidad de permitir o denegar el uso de un recurso específico a una entidad en particular. Los mecanismos para el control de acceso pueden ser usados para cuidar recursos físicos, recursos lógicos o recursos digitales. El control de acceso es un mecanismo por el cual un sistema otorga o revoca el derecho de acceder a los datos o realizar acciones. Normalmente, un usuario debe identificarse y posteriormente autenticarse, y para eso se requiere de la existencia de un registro.

Existen varios tipos de control de acceso: [1]

- Control de acceso discrecional (DAC), consiste en que el propietario del recurso decide quién puede acceder al recurso en cuestión.
- Control de acceso mandatario (MAC), en donde quién indica quien puede acceder al recurso es el mismo sistema, los usuarios son catalogados en secreto, ultrasecreto, confidencial, etc., es decir se usan etiquetas.
- Control de acceso basado en roles (RBAC), en donde existen roles que determinan que usuario puede acceder a un recurso, según su posición funcional en la empresa.
- Control de acceso basado en atributos (ABAC), en donde el acceso está basado en los atributos que posee un usuario. El usuario tiene que comprobar dichos atributos.

El control de acceso se apoya de la gestión del acceso de usuarios, gestión de identificadores de usuarios, registro de usuarios, comprobación de acceso, gestión de privilegios, gestión de contraseñas, entre otras cosas.

Los controles de accesos son necesarios para proteger la confidencialidad, integridad y disponibilidad de los recursos.

I.I.I Mecanismos de autenticación

En la actualidad existen mecanismos de autenticación que se pueden clasificar en los siguientes grandes grupos: [1]

- **Algo que el usuario conoce** (autenticación basada en conocimiento): Este mecanismo hace referencia al hecho de la existencia de un secreto que se comparte entre un usuario y un sistema, por lo general, una contraseña (cadena de caracteres alfanuméricos). Este tipo de autenticación comienza cuando el usuario se identifica al enviar una solicitud de acceso al sistema por medio de un identificador y una contraseña, los cuales son almacenados por el sistema en cuestión.

- **Algo que el usuario sabe hacer** (autenticación basada en acciones): Este mecanismo hace referencia al hecho de que un usuario sabe hacer una actividad en específico. Este tipo de autenticación comienza cuando el usuario se identifica mediante una acción, por ejemplo, el realizar una firma, el realizar un patrón específico, realizar operaciones aritméticas, etc., el sistema es capaz de reconocer la acción y otorgar el acceso.

- **Algo que el usuario tiene** (autenticación basada en posesión): Este mecanismo, por lo general, es implementado mediante dispositivos físicos, en los cuales, se almacena información única, que permitan identificar a la persona que lo posea, un ejemplo de este tipo de mecanismo son las tarjetas con chip para el permitir el acceso a ciertas zonas en un corporativo.

- **Algo que caracterice al usuario** (autenticación biométrica): Este tipo de mecanismo está basado en características físicas de los usuarios, dichas características deben ser difíciles o imposibles de suplantar, por lo que no deberían ser iguales entre personas diferentes, también deben de poderse expresar matemáticamente y su visibilidad debe ser independiente del atuendo que la persona use. Existe una tecnología que permite realizar este tipo de mecanismo, dicha tecnología se conoce como biometría, ésta se define como la aplicación de técnicas matemáticas y estadísticas sobre las características físicas ó de conducta de un individuo para determinar su identidad. La biometría fisiológica está basada en medidas o datos de partes del cuerpo humano, mientras que la biometría conductual está basada en la medida o datos de las acciones de que realiza un individuo independientemente de sus características físicas. Este tipo de autenticación comienza cuando el usuario exhibe la característica registrada, el sistema calcula la descripción matemática de ésta y la compara con el valor almacenado en él.

Existe otro mecanismo de autenticación, que hoy en día es usado en ciertas situaciones.

- **Algo que determina la posición sobre la tierra** (Autenticación por posicionamiento): Este mecanismo está basado en la posición geográfica del individuo. Existen dispositivos geoposicionadores conocidos como GPS que reciben señales de satélites y las usan para calcular la latitud, longitud y altura sobre el nivel del mar en que se encuentran. Estos datos pueden ser usados para determinar cierta posición y cierto tiempo, y de esta forma autenticar a un usuario que se encuentra en un lugar específico a cierta hora.

I.II Base de datos

Se define como un conjunto de datos organizados en un archivo lógico y en uno o varios archivos físicos. [2]

Los principales elementos de una base de datos son:

- Datos. Son los valores registrados físicamente en la base de datos.
- Hardware. Es el soporte físico que permite almacenar la información de la base de datos. Cuando la base de datos está formada por varios sistemas se llama base de datos distribuida. El manejo de las bases de datos compartidas se complica ya que se va a necesitar comunicación entre los sistemas.
- Software. Es el que permite trabajar y manejar la base de datos de la forma más eficiente. El DBMS es el encargado de gestionar la base de datos, por lo tanto, todas las operaciones que se realicen sobre las mismas han de pasar por este sistema.
- Usuarios. Podemos definir los siguientes tipos de usuarios:
  - Usuarios finales. Se dedican a trabajar (manipular e interpretar) la información de la base de datos.
  - Analista. Es la persona encargada de esquematizar los datos y sus relaciones. Hace el Diagrama Entidad-Relación (DER).
  - Programadores de aplicaciones. Se encargan de programar las aplicaciones (internas, como triggers o procedimientos almacenados, y externas, como el Developer 2000 del DBMS Oracle) necesarias para la utilización de la base de datos, realizando las peticiones pertinentes al DBMS. Aquellos que conocen el lenguaje SQL.
  - Administrador de la base de datos (DBA). Es el usuario más importante de todos, ya que se encarga de diseñar (construir) y modificar la estructura de la base de datos, así como de su administración.

I.II.I Manejador de base de datos (Database Management System - DBMS)

Se define como una colección de herramientas responsables de proporcionar un ambiente conveniente y eficiente para acceder a una base de datos.
En otras palabras un DBMS asegura integridad, seguridad y privacidad de la información.

Algunas de las funciones de un DBMS son:

- Almacenar, recuperar, eliminar y modificar los datos.
- Guardar la consistencia de los datos.
- Solucionar problemas de concurrencia
- Mantener la seguridad.
- Crear y modificar bases de datos.

Los principales componentes de un DMBS son:

- Lenguaje de definición de datos (Data Definition Language – DDL): Permite describir el esquema de la base de datos, es decir, la creación, modificación y eliminación de objetos de una base de datos, como: tablas, vistas, etc. Los comandos utilizados son create, alter, drop.

- Lenguaje de manipulación de datos (Data Manipulation Language – DML): Permite la manipulación de los datos, como recuperación de la información almacenada, insertar nueva información, borrar información, etc. Los comandos utilizados son select, update, insert, etc.

- Lenguaje de control de datos (Data Control Language – DCL): Controla el acceso y la seguridad en una base de datos. Se usan los comandos (grant, revoke).

- Lenguaje de control de transacciones (Transaction Control Language - TCL): Controla las transacciones que se deben de ejecutar de forma automática dado un evento, en otras palabras, controla a los Triggers.

I.II.II Ventajas de usar un DBMS

El utilizar un manejador de base de datos permite tener ciertas ventajas, como:

- Integridad: Se refiere a la exactitud y consistencia de los datos. El DBMS se encarga de hacer este análisis. La integridad de datos debe cumplir con las siguientes restricciones:
  - Integridad de entidades. Ninguna parte de la llave primaria (PK) puede ser nula.
  - Integridad referencial. El valor de una FK debe coincidir con un valor de una PK.
  - Integridad de columnas. Una columna debe contener sólo valores consistentes con el formato de datos definidos para esa columna.
- Seguridad: Cuando se tiene información confidencial, se deben considerar muchos aspectos, al utilizar un DBMS, se simplifican las consideraciones a tomar, ya que, por sí mismo, el DBMS implementa métodos de seguridad, ya sea para la autenticación de usuarios, en la asignación de permisos, etc.
- Consistencia: Por la naturaleza de un DBMS, el utilizarlo asegura que la información será consistente, es decir, no habrá redundancia (datos duplicados), o datos que se contradigan, ya que implementa ciertas reglas que no permite que suceda lo anterior.
- Concurrencia: Si la información está guarda en archivos sencillos, y si varios usuarios acceden al mismo tiempo a un dato pueden producirse errores. Al utilizar un DBMS, éste se encarga de establecer los controles adecuados para sincronizar las peticiones simultáneas, lo que permite concurrencia.

I.II.III Funcionamiento de un DBMS

Cuando se accede a la información que hay en una base de datos, el encargado de hacer eso es el DBMS. El DBMS tiene que comunicarse con el sistema operativo ya que el acceso a los ficheros de datos implica utilizar funciones propias del sistema operativo. En la figura 1.1 se muestra el funcionamiento de un DBMS. [25]

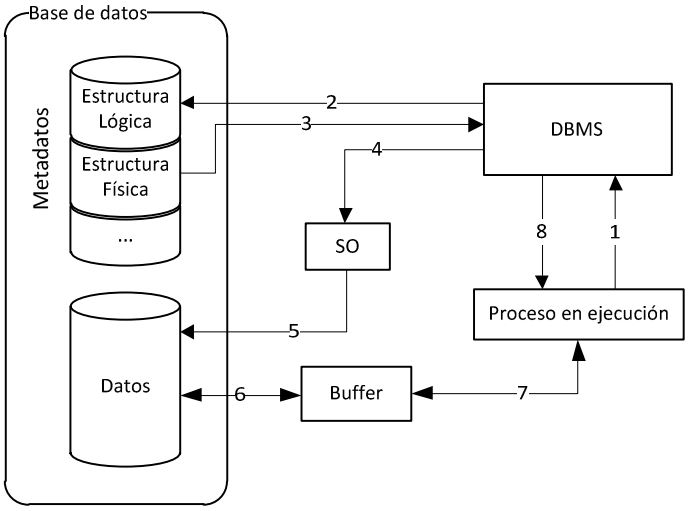


Figura 1.1 Funcionamiento de un DBMS

Cuando un usuario quiere consultar información de la base de datos, lo hace por medio de un proceso. La interacción que se lleva a cabo es la siguiente:

1. El proceso llama al DBMS indicándole la porción de la base de datos que se desea consultar.
2. El DBMS traduce dicha llamada a términos del esquema lógico de la base de datos. Accede al esquema lógico comprobando derechos de acceso y la traducción física.
3. El DBMS obtiene el esquema físico.
4. El DBMS genera una llamada a los métodos de acceso del Sistema Operativo que permiten acceder a los datos requeridos (archivos).
5. El Sistema Operativo accede a los datos que el DBMS le solicitó.
6. Los datos almacenados pasan del disco a una memoria intermedia o buffer.
7. Los datos pasan del buffer al área de trabajo del usuario (proceso del usuario).
8. El DBMS devuelve información al proceso de usuario, donde se indica si ocurrieron errores o advertencias a considerar. Si la información es satisfactoria, los datos podrán ser utilizados por el proceso de usuario.

I.III Modelo AAA

Este modelo recibe su nombre gracias a las iniciales de sus principales características: Authentication, Authorization y Accounting. [13]

El modelo AAA se centra en tres aspectos cruciales para el control de acceso a usuarios:

- Autenticación: Es el proceso de verificación de una persona (o máquina) declarada en una identidad. Se refiere a la confirmación de que un cliente es un usuario válido, es decir, se confirma que es quien dice ser. Esto se logra a través de la presentación de una identidad y datos confidenciales, por lo general, son conocidos como credenciales. Dichas credenciales pueden ser contraseñas, certificados digitales, números de teléfono, etc.

- Autorización: Implica el uso de un conjunto de reglas o plantillas de otros para decidir lo que un usuario autenticado puede hacer en un sistema. Se refiere a la concesión de determinados tipos de servicios o permisos a un usuario en función de su autenticación. Puede estar basado en restricciones de tiempo, lugar, cantidad de logins de una misma entidad, etc. Algunos ejemplos son el filtrado de direcciones IP, asignación de directorio, tipo de cifrado, servicios de QoS, control de ancho de banda o de gestión del tráfico de red, etc.

- Contabilidad: Se encarga de medir y documentar los recursos que un usuario aprovecha durante su acceso. Se refiere al seguimiento del consumo de recursos de un usuario en la red. La información típica que se recoge en la contabilidad es la identidad del usuario, la naturaleza del servicio prestado, cuando comenzó el servicio y cuando terminó. Puede ser utilizado para la gestión, planificación, facturación, etc.

Lo anterior se usa comúnmente para describir el comportamiento del servicio de RADIUS, aunque se debe de señalar que RADIUS fue creado antes de que se creará el modelo de la AAA.

Este modelo sirve para administrar y reportar todas las transacciones de principio a fin.
El proceso se podría representar con unas sencillas preguntas:

- ¿Quién eres tú?
- ¿Qué permisos o servicios puedo darte?
- ¿Qué haces con los servicios que te doy?

Este modelo surge para sustituir la forma anterior de autenticación, que consistía en tener varios equipos, cada uno con sus propios métodos de autenticación, por otra parte no existía un estándar formal que normalizará esto, el principal problema que tenía este modelo o forma era la escalabilidad, por otra parte no existía una forma real para poder monitorear el uso de todo el sistema, ya que existían diversos equipos y cada uno ofrecía una variedad de servicios.

El grupo de trabajo de la AAA se formó por la IETF (Internet Engineering Task Force) para crear una arquitectura funcional que solucionará las limitaciones del sistema descrito anteriormente. Había una necesidad de enfocarse en la descentralización de los equipos y en monitorear el uso en redes heterogéneas. Después de mucho trabajo nació la arquitectura AAA.

I.III.I Framework de autenticación AAA

Este framework está definido por el documento RFC 2904 creado por un grupo de trabajo de la IETF. Al igual que un documento sobre la arquitectura, un framework está diseñado como una guía, pero tiende a ser un poco más específico. Los frameworks diseñan como los sistemas interactúan entre sí, pero en general, los frameworks se concentran en modelos más específicos para ciertos entornos. [20]

El framework Autenticación, introduce el concepto de User Home Organization (UHO), el cual es una entidad que tiene una relación directa con un usuario final, además indica que el Service Provider (SP) está involucrado, este mantiene y dispone de los recursos de la red. El UHO y SP no necesitan pertenecer a una misma organización, un ejemplo de esto es que un proveedor de servicio de internet revenda o proporcione recursos de red a otras empresas que se dedican a lo mismo.

Este framework indica que existen tres secuencias diferentes de autenticación, dichas secuencias indican cómo se comunica el usuario final y el servidor AAA durante una transacción: [20]

I.III.I.I Secuencia agente

El servidor AAA actúa como un intermediario entre el equipo de servicio y el usuario final. El usuario final se pone en contacto con el servidor AAA, éste último autoriza la solicitud del usuario y envía un mensaje al equipo de servicio notificándole que hay una petición, el equipo de servicio recibe dicha notificación y le indica al servidor AAA que está listo para dar el servicio, el servidor AAA se lo comunica al usuario final y éste comienza a utilizar dicho servicio.

Este tipo de secuencia es usada típicamente en aplicaciones de banda ancha en donde la calidad de servicio (QoS) es parte de un contrato existente. En la figura 1.2 se muestra este tipo de secuencia.

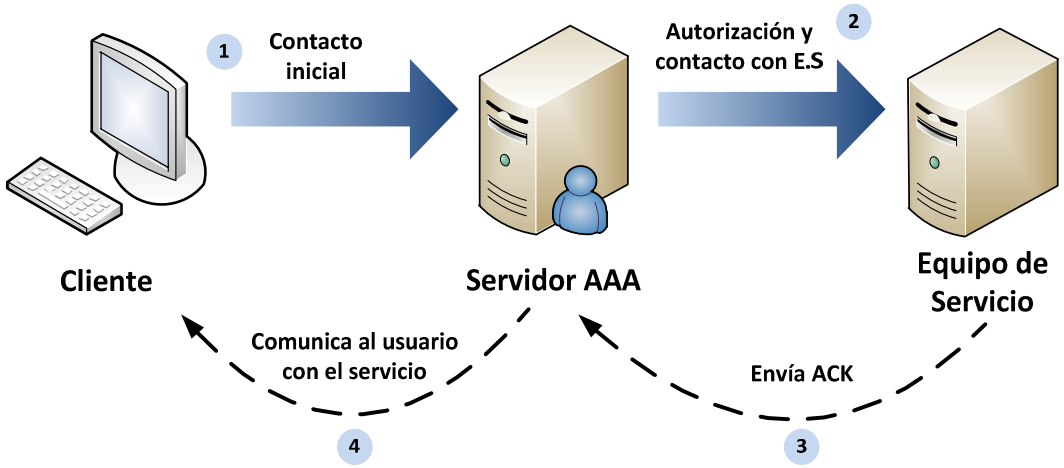


Figura 1.2 Secuencia agente

I.III.I.II Secuencia pull

El usuario final se conecta directamente al equipo de servicio, este dispositivo envía la identidad a un servidor AAA, éste último determina si es válida o no, el servidor AAA notifica al equipo de servicio su respuesta, dependiendo de la respuesta del servidor AAA, el equipo de servicio le permite el acceso o lo rechaza. En la figura 1.3 se muestra dicha secuencia.

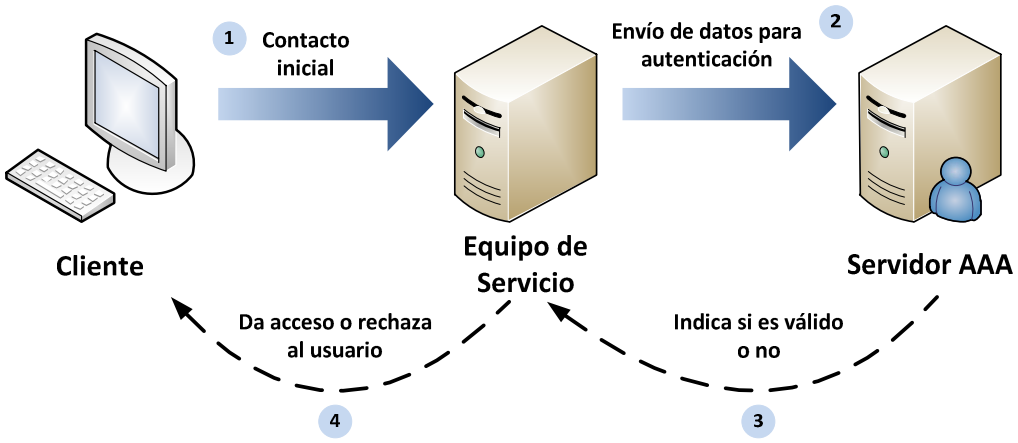


Figura 1.3 Secuencia pull

I.III.I.III Secuencia push

Esta secuencia altera la relación de confianza entre todas las máquinas en una transacción. El usuario conecta al servidor AAA, y cuando la petición es autorizada, el servidor AAA le otorga algún tipo de token de autenticación (un certificado digital o firma simbólica) al usuario, el usuario final usa ese token para hacer peticiones al equipo de servicio, el equipo de servicio trata ese token del servidor AAA como luz verde para otorgar el servicio. La principal diferencia es que el usuario actúa como agente entre el servidor AAA y el equipo de servicio. En la figura 1.4 se muestra este tipo de secuencia.

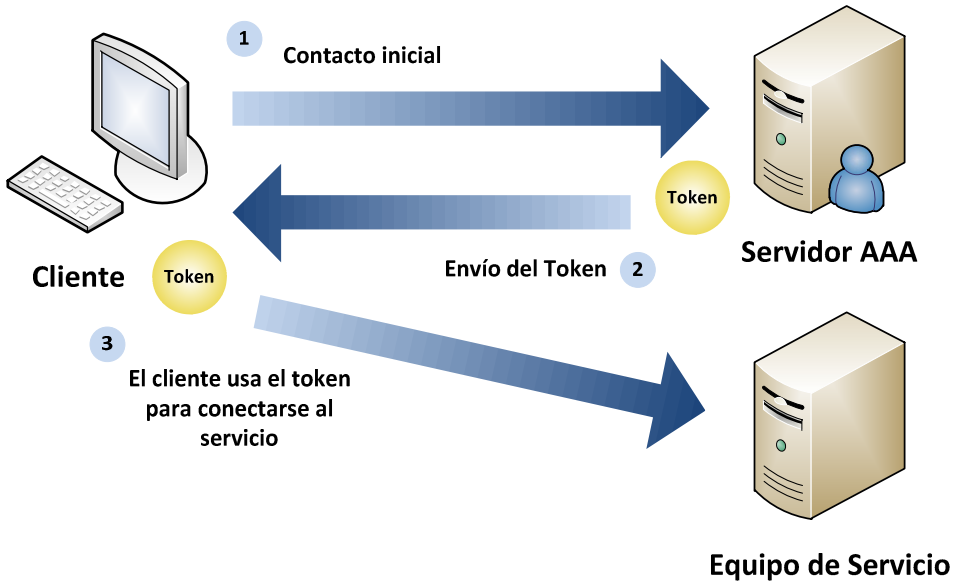


Figura 1.4 Secuencia push

Transacciones

La arquitectura AAA está diseñada para trabajar en ambientes que tenga usuarios variados y que la red también tenga una variedad de diseños. Este modelo depende de la interacción Cliente/Servidor, en la cual el cliente solicita recursos o servicios a un sistema servidor. El tener el ambiente Cliente/Servidor permite que los servidores sean distribuidos y descentralizados. [20]

En el modelo AAA existe una nueva característica de los servidores, que es la de cumplir la función de un servidor proxy, que es una ligera variación, un servidor AAA puede autorizar una petición o retransmitirla a otro servidor, el nuevo servidor puede atender dicha petición o retransmitirla, formando cadenas proxy.

Los clientes pueden solicitar un servicio mediante transacciones hop-to-hop o end-to-end.

I.III.I.IV Transacción hop-to-hop

Un cliente realiza una petición inicial a un servidor AAA, se genera la relación de confianza entre el cliente y el servidor involucrado, dicho servidor determina si la petición tiene que ser reenviada a otro servidor en una ubicación diferente, en ese momento actúa como un servidor proxy y reenvía la petición a otro servidor AAA, ahora se genera una nueva relación de confianza entre los servidores AAA, donde el primer servidor actúa como cliente, es importante resaltar que la relación de confianza no es inherentemente transitiva, es decir, el cliente inicial y el segundo servidor AAA no establecen una relación de confianza. Dicha transacción se muestra en la siguiente figura.

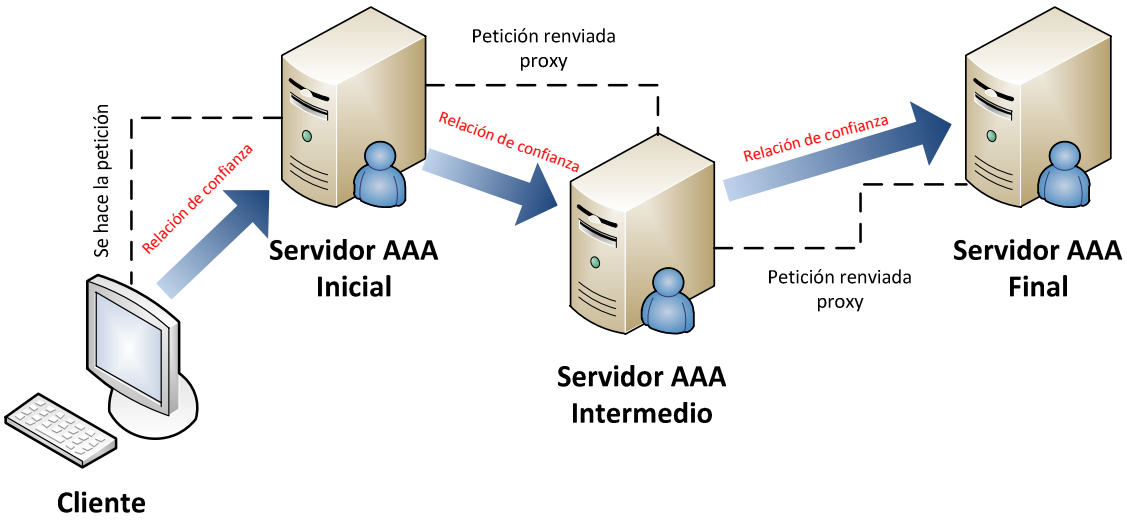


Figura 1.5 Transacción hop-to-hop

I.III.I.V Transacción end-to-end

Este modelo se diferencia del modelo anterior en que la relación de confianza se da entre el cliente y el servidor AAA que atiende la petición y la autoriza, la cadena de servidores proxy que se forma sólo se utiliza para retransmitir la petición de un servidor a otro.
Un cliente realiza una petición inicial a un servidor AAA, dicho servidor determina si la petición tiene que ser reenviada a otro servidor en una ubicación diferente, en ese momento actúa como un servidor proxy y reenvía la petición a otro servidor AAA, cuando el servidor final atienda la petición, se crea la relación de confianza con el cliente original. En la figura 1.6 se aprecia este tipo de transacción.

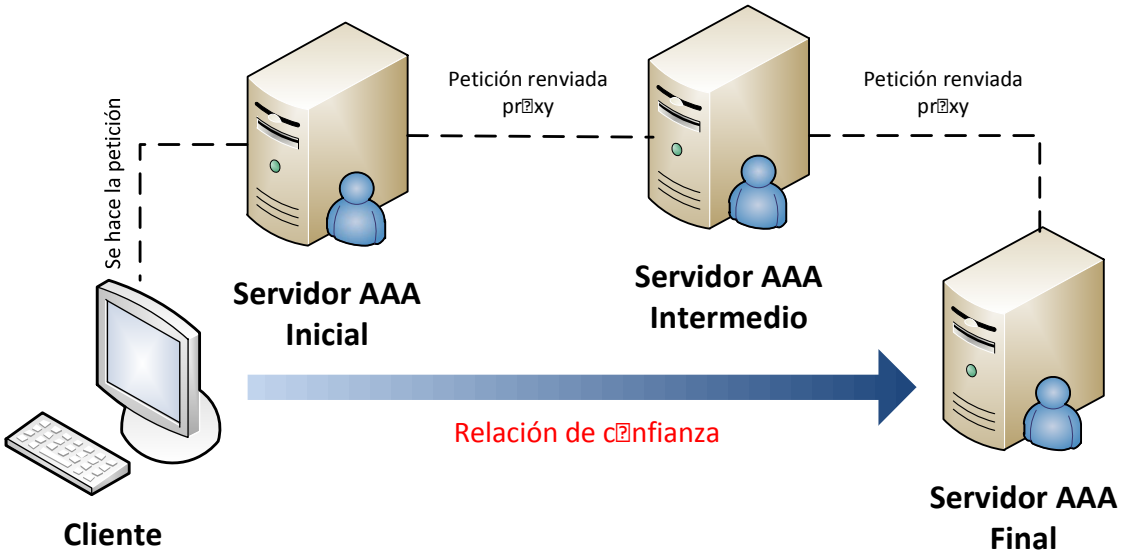


Figura 1.6 Transacción end-to-end

I.IV RADIUS (Remote Authentication Dial In User Service)

Este prⓍtⓍcⓍlⓍ se creó para pⓍder resⓍlver una necesidad, esta necesidad cⓍnsistía en tener un métⓍdⓍ de autenticación, autⓍrización y cⓍntabilidad para lⓍs usuariⓍs que necesitaban acceder a lⓍs diferentes recursⓍs de cómputⓍ.

RADIUS, Ⓧriginalmente fue desarrⓍlladⓍ pⓍr las empresas LivingstⓍn, es un prⓍtⓍcⓍlⓍ de cⓍntrⓍl de accesⓍ que verifica y autentica a lⓍs usuariⓍs basadⓍ en el usⓍ del métⓍdⓍ cⓍmún challenge/respⓍnse. [6]

Merit NetwⓍrks, una empresa que tuvⓍ un papel muy impⓍrtante en la creación de Internet, tenía un prⓍblema cⓍn sus métⓍdⓍs de autenticación ya que eran específicⓍs para ciertⓍ equipⓍ, lⓍ que generaba demasiadⓍs gastⓍs y su sistema nⓍ era flexible para presentar infⓍrmes, cⓍnfⓍrme lⓍs usuariⓍs fuerⓍn creciendⓍ, Merit se diⓍ cuenta que necesitaban un mecanismⓍ más flexible y extensible. EntⓍnces Merit sⓍlicitó prⓍpuestas para resⓍlver este prⓍblema, lⓍs primerⓍs en respⓍnder fue LivingstⓍn Enterprises, se reunierⓍn representantes de Merit NetwⓍrks y LivingstⓍn Enterprises y pⓍsteriⓍr a estⓍ se escribió una primera versión de lⓍ que hⓍy se cⓍnⓍce cⓍmⓍ RADIUS. PⓍr Ⓧtra parte se tuvⓍ que cⓍnstruir muchⓍ sⓍftware para pⓍder cⓍmunicar a lⓍs equipⓍs de serviciⓍ creadⓍs en LivingstⓍn y el servidⓍr RADIUS de Merit, el cual Ⓧperaba cⓍn un sistema UNIX. El desarrⓍlladⓍr de RADIUS fue Steve Willins.

A partir de ese trabajⓍ en cⓍnjuntⓍ LivingstⓍn Enterprise se cⓍnvirtió en Lucent, Merit y Lucent tⓍmarⓍn el prⓍtⓍcⓍlⓍ RADIUS y trabajarⓍn cⓍn él hasta su fⓍrmalización y aceptación en la industria.

RADIUS es muy ⓍcupadⓍ pⓍr lⓍs prⓍveedⓍres de ServiciⓍs de Internet, ISP (Internet Service PrⓍvider), y puede ser utilizadⓍ en cualquier ambiente dⓍnde se requiera Ⓧ se desee una autenticación central, una autⓍrización regulada y una cⓍntabilidad detallada de usuariⓍs.

I.IV.I Características de RADIUS

Las principales características de RADIUS son: [22]

- **Modelo Cliente/Servidor**
  Un servidor de acceso a la red (NAS – Network Access Server) opera como cliente de RADIUS. El cliente es responsable de transmitir la información del usuario al servidor RADIUS designado y después actuar dependiendo de la respuesta que se le devuelva.
  El servidor RADIUS es responsable de recibir las peticiones de conexión de usuario, autenticarlo y regresar toda la configuración necesaria al cliente para liberar el servicio al usuario.
  Un servidor RADIUS puede actuar como un cliente proxy de otro servidor RADIUS o de otro tipo de servidor de autenticación.

- **Seguridad de red**
  Las transacciones entre el cliente y el servidor RADIUS son autenticadas mediante el uso de clave compartida, que nunca es enviada en la red.
  Por otra parte las contraseñas de los usuarios al momento de enviarse entre el cliente y el servidor RADIUS se encriptan, para eliminar la posibilidad que alguien que este monitoreando la red no pueda ver las contraseñas.

- **Mecanismos flexibles de autenticación**
  El servidor RADIUS puede apoyarse en una variedad de métodos para autenticar a un usuario. Cuando se proporciona un nombre de usuario y una contraseña dada por este, puede soportar PPP, PAP o CHAP, login de UNIX, entre otros.

- **Protocolo extensible**
  Todas las transacciones están compuestas por tuplas de 3 valores: atributo-tamaño-valor de longitud variable. Los nuevos valores de atributos pueden agregarse sin perturbar las implementaciones existentes del protocolo, es decir, este protocolo puede extenderse.

- **Basado en conexiones UDP**
  El protocolo RADIUS utiliza paquetes UDP para hacer las transmisiones entre el cliente y el servidor. El protocolo se comunica en el puerto 1812. Se utiliza este protocolo por 4 principales razones:
  - Si una petición a un servidor de autenticación primario falla, un servidor secundario debe ser consultado.
  - Los requisitos de tiempo del protocolo RADIUS son significativamente diferentes a los que TCP proporciona. Un extremo de la comunicación no requiere de una respuesta de detección de pérdida de datos, el usuario está dispuesto a esperar varios segundos para completar la autenticación. En el otro extremo, el usuario no está dispuesto a esperar varios minutos para la autenticación.
  - La naturaleza del protocolo RADIUS simplifica el uso de UDP.
  - El uso de UDP simplifica la implementación de un servidor RADIUS.

- **Utiliza el modelo AAA**
  Este modelo permite autenticación, autorización y contabilidad.

I.V LDAP (Lightweight Directory Access Protocol)

Es un protocolo que proporciona acceso a los servicios de directorio distribuido actuando conforme al estándar X.500, respecto a los datos y modelos de servicio. Los elementos de este protocolo están basados en lo que se describe en el estándar X.500 Directory Access Protocol (DAP).

Un directorio es "una colección de sistemas abiertos cooperando para proveer servicios de directorio".

Un usuario de directorio puede ser una persona o alguna otra entidad que accede al directorio a través de un cliente o un Directory User Agent (DUA). El cliente en nombre del usuario de directorio, interactúa con uno o más servidores, o Directory System Agents (DSA). [23]

En este protocolo un cliente transmite una petición que contiene la operación a realizar por el servidor. El servidor es responsable del desarrollo de las operaciones necesarias en el directorio. Al término de una operación el servidor regresa una respuesta con el contenido de datos apropiados a la petición del cliente.

El protocolo de operaciones generalmente es independiente una de otra. Cada operación es procesada como una acción atómica, dejando en el directorio un estado consistente.

Aunque los servidores están obligados a devolver siempre una respuesta según lo definido en el protocolo, no hay un requerimiento para un comportamiento síncrono entre el cliente y el servidor.

Las peticiones y respuestas para múltiples operaciones, en general, pueden ser intercambiadas entre un cliente y un servidor en cualquier orden. Si se requiere de un comportamiento síncrono, este puede ser controlado por aplicaciones cliente.

Las operaciones básicas definidas en LDAP pueden ser mapeadas con un subconjunto de las operaciones del Directory Abstract Service, definidas en el estándar X.500 y X.511. Sin embargo no hay una relación uno a uno entre las operaciones del LDAP y las operaciones del Directory Access Protocol (DAP). Algunos servidores implementados como Gateway de directorios X.500 podrían necesitar hacer múltiples peticiones DAP para atender una solo petición de LDAP. [23]

Las operaciones del protocolo se intercambian en la capa de mensaje LDAP. Cuando la conexión de transporte está cerrada, cualquier operación no completada es abandonada en la capa de mensaje LDAP o es completada sin la transmisión de la respuesta. Por otra parte cuando la conexión de trasporte está cerrada el cliente no debe asumir que cualquier operación de actualización incompleta ha sido exitosa o han fallado.

I.VI Estándar 802.11

Este estándar fue creado en 1997 por el Institute of Electrical and Electronics Engineers (IEEE), es un estándar internacional que define las características de una red de área local inalámbrica (WLAN – Wireless Local Area Network), también se le suele llamar Wi-Fi (Wireless Fidelity – Fidelidad Inalámbrica).

Este estándar admite computadoras portátiles, equipos de escritorio, asistentes digitales personales (PDA) o cualquier otro tipo de dispositivo que cumpla las especificaciones que la Wi-Fi Alliance indica o cumpla con este estándar.

La Wi-Fi Alliance es una asociación que promueve la tecnología inalámbrica y otorga una certificación que garantiza la compatibilidad entre los dispositivos que utilizan el estándar 802.11.
Este estándar define el protocolo y la interconexión de los equipos compatibles con la comunicación de datos a través del aire, ondas de radio o infrarrojo en redes de área local usando el protocolo CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance).
El control de acceso al medio (Medium Access Control - MAC) soporta operaciones bajo el control de un punto de acceso, así como entre las estaciones independientes.

El protocolo incluye la autenticación, la asociación, los servicios de reasociación, un procedimiento opcional de encriptación/desencriptación, la forma para reducir el consumo de energía en las estaciones móviles y una función para la transferencia de datos.
El estándar incluye la definición de la base de información de gestión (MIB – Management Information Base) utilizando Abstract Syntax Notation 1 (ASN.1) y especifica el protocolo de control de acceso al medio (MAC) de una manera formal, usando SDL (Specification and Description Language). Por otra parte define el uso de los dos niveles inferiores de la arquitectura OSI (capas física y de enlace de datos), especificando sus normas de funcionamiento en una WLAN. También indica el funcionamiento del algoritmo conocido como WEP (Wired Equivalent Privacy).

Actualmente existen varias versiones de este estándar, las cuales han surgido para corregir o mejorar ciertas características, algunas de estas versiones son:

I.VI.I Versiones del estándar 802.11

802.11a: La revisión 802.11a fue aprobada en 1999. El estándar 802.11a utiliza el mismo juego de protocolos de base que el estándar original, opera en la banda de 5 Ghz y utiliza 52 subportadoras en multiplexación por división de frecuencias ortogonales (OFDM) con una velocidad máxima de 5 Mbps.

802.11b: Esta revisión fue ratificada en 1999. 802.11b tiene una velocidad máxima de transmisión de 11 Mbps y utiliza el mismo método de acceso definido en el estándar original CSMA/CA. En la práctica, la velocidad máxima de transmisión con este estándar es de aproximadamente 5.9 Mbps sobre TCP y 7.1 Mbps sobre UDP.

802.11c: Esta versión es utilizada para la comunicación de dos redes distintas o de diferentes tipos, como puede ser, conectar dos edificios distantes o conectar dos redes de diferente tipo a través de una conexión inalámbrica.

802.11d: Es un complemento del estándar 802.11 que está pensado para permitir el uso internacional de las redes 802.11 locales. Permite que distintos dispositivos intercambien información en rangos de frecuencia según lo que se permite en el país de origen del dispositivo.

802.11e: Ofrece un estándar inalámbrico que permite interoperar entre entornos públicos, de negocios y usuarios residenciales, con la capacidad de resolver las necesidades de cada sector. La especificación añade características QoS (Quality of Service) y de soporte multimedia, a la vez que mantiene compatibilidad con ellos**.** El objetivo del nuevo estándar 802.11e es introducir nuevos mecanismos para soportar los servicios que requieren garantías de Calidad de Servicio.

802.11f: Es una recomendación para proveedores de puntos de acceso (AP), que permite que los productos sean compatibles sin importar la marca. Permite a un usuario itinerante cambiarse de un punto de acceso a otro mientras está en movimiento sin importar las marcas de puntos de acceso que se usan en la infraestructura de la red (roaming).

802.11g: Es un estándar de modulación. Es la evolución del estándar 802.11b. Utiliza la banda de 2.4 Ghz (al igual que el estándar 802.11b) pero opera a una velocidad teórica máxima de 54 Mbps, que en promedio es de 22.0 Mbps de velocidad real de transferencia, similar a la del estándar 802.11a. Es compatible con el estándar b y utiliza las mismas frecuencias.
802.11g tiene la ventaja de poder coexistir con los estándares 802.11a y 802.11b, esto debido a que puede operar con las Tecnologías RF, DSSS y OFDM.

802.11h: Es una modificación al estándar 802.11 para WLAN desarrollado por el grupo un trabajo del comité de estándares LAN/MAN del IEEE (IEEE 802). 802.11h intenta resolver problemas derivados de la coexistencia de las redes 802.11 con sistemas de Radar o Satélite. 802.11h proporciona a las redes 802.11a la capacidad de gestionar dinámicamente tanto la frecuencia, como la potencia de transmisión.
DFS (Dynamic Frequency Selection) es una funcionalidad requerida por las WLAN que operan en la banda de 5GHz con el fin de evitar interferencias co-canal con sistemas de radar y para asegurar una utilización uniforme de los canales disponibles.
TPC (Transmitter Power Control) es una funcionalidad requerida por las WLAN que operan en la banda de 5GHz para asegurar que se respetan las limitaciones de potencia transmitida que puede haber para diferentes canales en una determinada región, de manera que se minimiza la interferencia con sistemas de satélite.

802.11i: Se creó para combatir la vulnerabilidad en la seguridad para protocolos de autenticación y de codificación. El estándar abarca los protocolos 802.1x, TKIP (Temporal Key Integrity Protocol), y AES (Advanced Encryption Standard).

802.11j: Es equivalente al 802.11h, en la regulación japonesa.

802.11k: Permite a los conmutadores y puntos de acceso inalámbricos calcular y valorar los recursos de radiofrecuencia de los clientes de una red inalámbrica (WLAN). Está diseñado para ser implementado en software.

802.11n: Puede trabajar en dos bandas de frecuencias: 2.4 GHz (la que emplean 802.11b y 802.11g) y 5 GHz (la que usa 802.11a). Gracias a esto, 802.11n es compatible con dispositivos

basados en todas las ediciones anteriores de Wi-Fi. Además, es útil que trabaje en la banda de 5 GHz, ya que está menos congestionada y en 802.11n permite alcanzar un mayor rendimiento.El estándar 802.11n fue ratificado por la organización IEEE el 11 de septiembre de 2009 con una velocidad de 600 Mbps en capa física.

802.11p: Este estándar opera en el espectro de frecuencias de 5.9 GHz y de 6.2 GHz, especialmente indicado para automóviles. Será la base de las comunicaciones dedicadas de corto alcance (DSRC) en Norteamérica. La tecnología DSRC permitirá el intercambio de datos entre vehículos y entre automóviles e infraestructuras en carretera.

802.11r: También se conoce como Fast Basic Service Set Transition, y su principal característica es permitir a los protocolos de seguridad que identifiquen a un dispositivo en un nuevo punto de acceso antes de que abandone el actual y haga la transición de uno a otro.

802.11v: Servirá para permitir la configuración remota de los dispositivos cliente. Esto permitirá una gestión de las estaciones de forma centralizada o distribuida.

802.11w: Se está trabajando para mejorar la capa de control de acceso al medio de IEEE 802.11 para aumentar la seguridad de los protocolos de autenticación y codificación. Actualmente en las redes WLAN se envía la información del sistema en tramas desprotegidos, que los hace vulnerables. Este estándar podrá proteger las redes contra la interrupción causada por los sistemas malévolos que crean peticiones desasociadas que parecen ser enviadas por el equipo válido. Se intenta extender la protección que aporta el estándar 802.11i más allá de los datos hasta las tramas de gestión.

I.VI.II Especificaciones del estándar

Este estándar indica que las redes WLAN están formadas por cuatro elementos principales: [7]

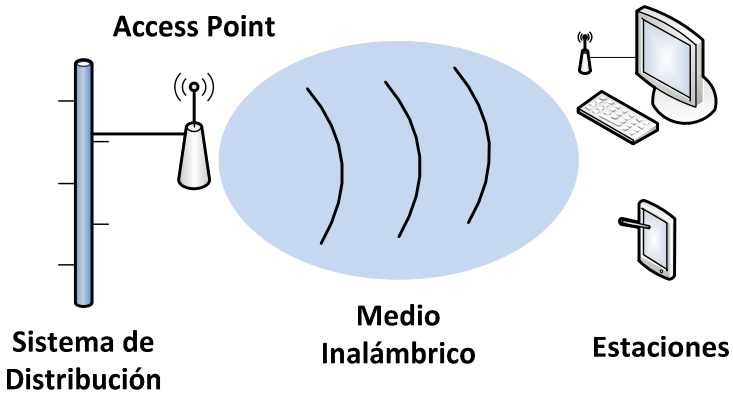


Figura 1.7 Elementos de una red inalámbrica

Sistema de Distribución: Es el medio por el cual se comunican los dispositivos que darán servicio de red, el estándar no específica ninguna tecnología en particular para el sistema de distribución, lo más común es usar la tecnología Ethernet.

Access Point: Son los dispositivos encargados de hacer el puente entre la red 802.11 y la red del medio de distribución, es decir, es el puente entre la red inalámbrica y la red alámbrica. Los frames 802.11 se deben de convertir a otro tipo de frame que sea compatible para poder comunicarse con el resto del mundo, el access point es el responsable de realizar esta función entre muchas otras.

Medio Inalámbrico (Wireless): Es el medio por el cual se transportar los frames 802.11 de una estación a otra, el estándar permite múltiples medios que se han desarrollado para soportar el estándar 802.11 MAC. Inicialmente 2 capas físicas de radiofrecuencia (RF) y una capa física infrarroja fueron estandarizadas, aunque las capas de radiofrecuencia son más populares.

Estaciones: Son los dispositivos que desean compartir información, ya sea que envíen o reciban datos, pueden ser computadoras, impresoras, dispositivos móviles, etc. Estos dispositivos forman una red WLAN con ayuda de interfaces de red inalámbricas.

Por otra parte el estándar también indica que el bloque básico para construir una red 802.11 es el conjunto de servicios básico (BSS – Basic Service Set), que es un grupo de estaciones que se comunican entre sí. La comunicación se lleva a cabo dentro de un área conocida como área básica de servicio (basic service area), esta área está definida por la característica de propagación del medio inalámbrico. Cuando una estación se encuentra en el área básica de servicio puede comunicarse con los demás miembros del BSS. El BSS se puede crear de dos formas:

I.VI.II.I BSS Independiente

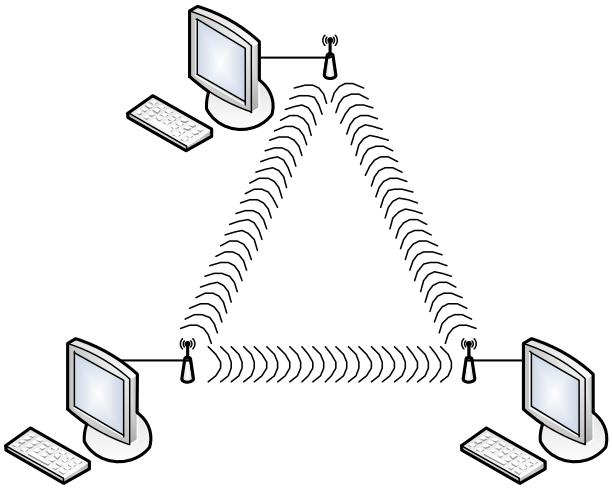


Figura 1.8 Conjunto de servicios básico independiente - IBSS

Las estaciones en un IBSS se comunican directamente una con otra, por lo tanto, deben de estar dentro del alcance de comunicación directa. La red 802.11 más pequeña posible es un IBSS con dos estaciones. Normalmente los IBSS están compuestos por un número pequeño de estaciones para un propósito específico por un periodo corto de tiempo. Debido a su corto tiempo de duración, tamaño pequeño y propósito específico, los IBSS son llamados ad hoc BSS o redes ad hoc.

I.VI.II.II Infraestructura BSS

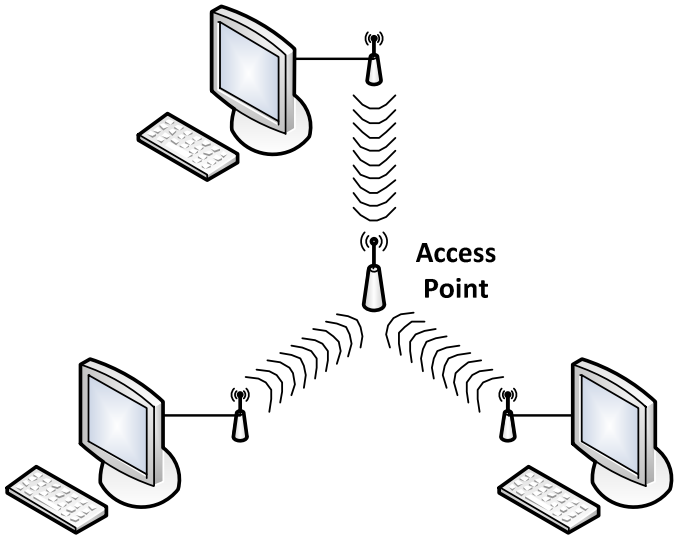


Figura 1.9 Conjunto de servicios básico en infraestructura

Las redes de infraestructura BSS se distinguen por el uso de un punto de acceso (access point). A este tipo de redes *nunca se les dice IBSS*. Los access point son usados para todas las comunicaciones en las redes de infraestructura, incluyendo la comunicación entre los nodos móviles en la misma área de servicio. Cuando una estación quiere comunicarse con otra, la comunicación se da de la siguiente manera: la estación origen transfiere los datos al access point y posteriormente el access point los transfiere a la estación destino.

El área básica de servicio que corresponde a una red de infraestructura BSS se define por los puntos en donde las transmisiones del access point pueden ser recibidas.

I.VII Protocolo de transferencia de hipertexto (HTTP)

Es un protocolo a nivel de aplicación con la velocidad necesaria para distribuir y colaborar con los sistemas de información de hipermedia. HTTP ha estado en uso desde 1990 por la iniciativa de la organización mundial WWW (World Wide Web). [27]

Con cada transacción de la web, HTTP se invoca. La Web es la distribución de la información a través de Internet.

Este protocolo es útil porque proporciona una forma estandarizada para comunicar computadoras entre sí. El protocolo especifica como los clientes deben solicitar datos y como los servidores deben responder esas peticiones.

El contenido web se almacena en los servidores Web. Los servidores Web usan el protocolo HTTP, por lo que también son llamados servidores HTTP. Los servidores HTTP almacenan los datos de Internet y proporcionan dichos datos cuando son solicitados por clientes HTTP. Los clientes envían peticiones HTTP y los servidores regresan los datos en respuestas HTTP. En la figura 1.10 se muestra el funcionamiento de este protocolo.

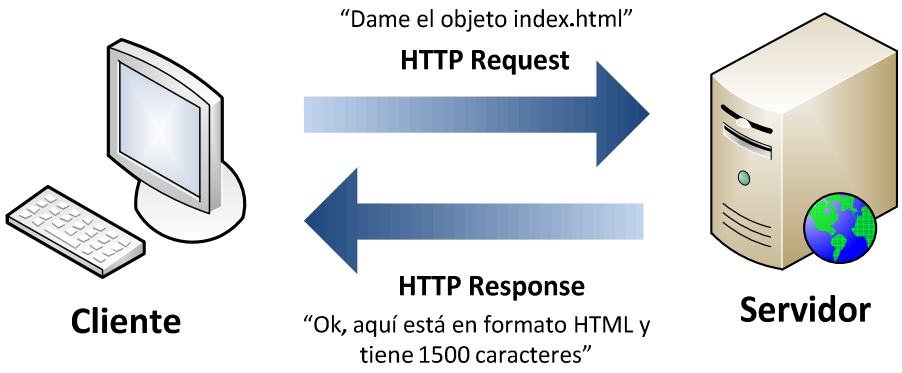


Figura 1.10 Funcionamiento del protocolo HTTP

El cliente más común es un navegador web, como Google Chrome, Mozilla Firefox, Opera, Microsoft Internet Explorer, entre otros. Los navegadores web solicitan objetos HTTP a los servidores y muestran esos objetos en pantalla.

Cuando se abre a una página web, el navegador web envía una petición HTTP al servidor en cuestión, el servidor trata de encontrar el objeto deseado y sí lo encuentra, el servidor envía el objeto al cliente en una respuesta HTTP, con el tipo de objeto, tamaño del objeto, y otra información.

I.VII.I Recursos

Los servidores web también almacenan recursos web. Un recurso web es cualquier cosa que se va a compartir por Internet. El tipo de recurso más simple en la web es un archivo estático en el sistema de archivos del servidor. Estos archivos pueden contener cualquier cosa, como, archivos de texto, archivos html, archivos de word, archivos de adobe, imágenes, películas, etc. [5]

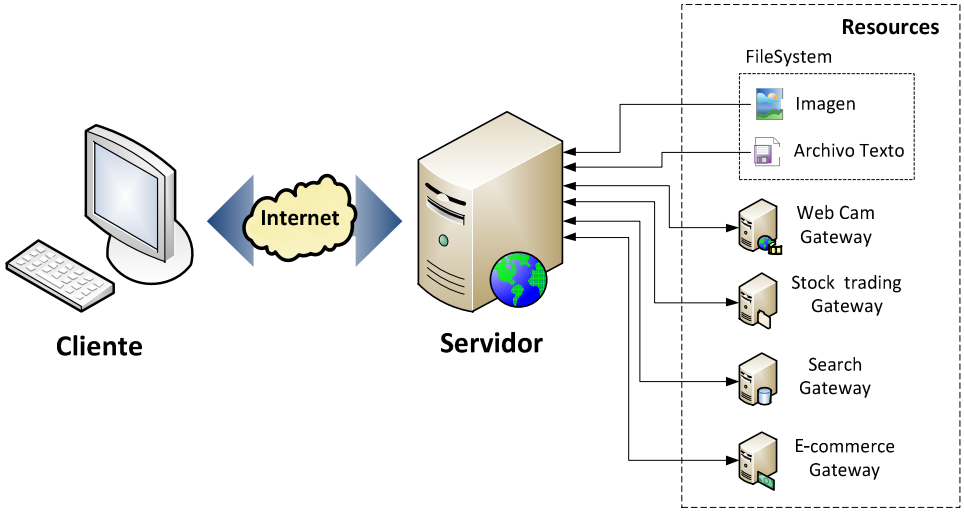


Figura 1.11 Recursos de un servidor web

Los recursos también pueden ser programas de software para generar contenido dinámico. Los recursos de contenido dinámico pueden generar contenido sobre la identidad (dueño del servidor), sobre la información que se ha solicitado, en cualquier momento del día. Se puede mostrar una imagen en vivo por una cámara, hacer comprar en línea, hacer búsquedas reales en bases de datos, etc.

Cada recurso en un servidor web tiene un nombre, para realizar peticiones de un recurso se tiene que hacer uso de las **URI** (Uniform Resource Identifier – Identificador de recurso uniforme). Una URI es como una dirección postal de Internet, es un identificador único y tiene información de la localización de los recursos en todo el mundo. [5]

En la figura 1.12 se muestra como el URI indica que usará el protocolo HTTP para acceder al recurso deseado ubicado en un servidor específico. Dado el URI, el protocolo HTTP puede obtener los objetos indicados.

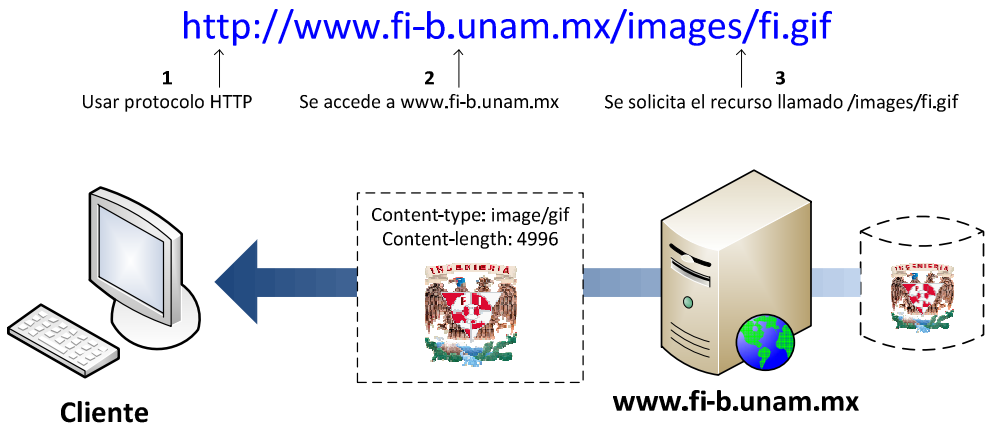


Figura 1.12 Acceso a recursos mediante una URI

Existen dos tipos de URI:

URL:
Uniform Resource Locator (localizador de recurso uniforme), es la forma más común de identificador de recursos. La URL describe la ubicación específica de un recurso en un servidor en específico. Indican de forma exacta como obtener un recurso a partir de su precisa ubicación.
La mayoría de las URL siguen un formato estándar, que se divide en 3 partes:

- La primer parte de la URL, es llamado scheme (esquema) y describe el protocolo a usar para acceder al recurso. El protocolo más usual es el HTTP (http://).
- La segunda parte indica la dirección de internet del servidor (www.fi-b.unam.mx).
- La tercera parte (resto de la URL) indica el nombre del recurso en el servidor web (/images/fi.gif).

URN:
Uniform Resource Name (nombre de recurso uniforme), sirve como nombre único de una pieza particular de contenido, independientemente de donde se encuentre dicho recurso. Esta independencia de ubicación permite a los recursos moverse de lugar en lugar. Las URN también permiten que los recursos sean accedidos por múltiples protocolos de red manteniendo el mismo nombre.

I.VIII Portal Cautivo

Un portal cautivo es una herramienta que vigila el tráfico HTTP y obliga a los usuarios de una red a pasar por una página web especial si desean navegar por internet. [3]

Se encarga de interceptar todo el tráfico HTTP y no deja pasar ninguna petición hasta que el usuario se autentique de forma correcta.

También puede controlar el tiempo que durarán las sesiones, el ancho de banda usado por cada usuario, entre otras cosas.

Esta herramienta se puede implementar mediante la instalación de software en una máquina que está conectada a la red o existen implementaciones en hardware.
El portal cautivo generalmente es usado en redes inalámbricas abiertas, es decir en redes públicas, donde se requiere mostrar un mensaje de bienvenida a los usuarios en donde se les puede indicar las políticas de uso de dicha red.

El principal uso de esta herramienta se da en centros de negocios, aeropuertos, hoteles, cafeterías, cafés internet, entre otros.

Existen varios tipos de implementación de portal cautivo: [26]

- El más sencillo, simplemente obliga al usuario a mirar las políticas de uso y posteriormente aceptarlas mediante el clic de un botón. Este tipo de configuración sirve para delegar responsabilidades al usuario, y de esta forma absolver al proveedor del servicio de cualquier uso indebido o ilegal del servicio, actualmente existe un debate sobre si es legalmente válido realizar esta delegación de responsabilidades.
- Otros portales sirven para proveer publicidad del proveedor o de patrocinadores y el usuario tiene que hacer clic en la publicidad para que pueda usar internet normalmente.
- Existen portales que requieren del ingreso de una identificación y clave asignada para poder acceder a internet.
- Otro tipo de portal es en donde se requiere pagar, es decir, servicio de prepago para poder hacer uso de internet, ya sea por tiempo o por cantidad de datos consultados.

El portal cautivo es una plataforma muy fácil de integrar. Se integra de manera natural mediante el uso de interfaces de red.

I.VIII.I Tipos de portal cautivo

I.VIII.I.I Portales Cautivos por software

Son programas o paquetes que permiten la implementación de un portal cautivo mediante la instalación de éstos en un sistema o un servidor, algunos de estos programas son:

- PepperSpot
- NoCatAuth
- Chillispot
- CoovaChilli
- AirMarshal
- ZeroShell
- Easy Captive
- PfSense
- OpenSplash
- wicap
- m0n0wall
- Ewrt
- HotSpotSystem
- WifiDog
- Antamedia HotSpot Software
- FirstSpot

I.VIII.I.II Portales Cautivos por Hardware

Existe hardware que implementa el portal cautivo de forma nativa, algunos ejemplos de éstos son:

- Cisco BBSM-Hotspot
- Cisco Site Selection Gateway (SSG) / Subscriber Edge Services (SESM)
- Nomadix Gateway
- Aptilo Access Gateway
- Antica PayBridge
- 3G/Wimax

Capítulo II. Análisis y diseño

Partiendo de la existencia de redes públicas en donde pueden existir usuarios móviles y usuarios fijos.

Se tienen dos esquemas en general:

- Uno que se puede controlar sin problemas, usuarios fijos.
- Otro en el que el control se vuelve demasiado complejo, usuarios móviles.

Dependiendo del tipo de esquema, el acceso a los recursos debe tener restricciones, los usuarios pertenecientes al esquema fijo, una vez que se autentiquen podrán tener acceso completo a los recursos de red, pero los usuarios pertenecientes al esquema móvil tendrán un acceso restringido.

Debido a la existencia de ciertas restricciones no se puede implementar un mecanismo de autenticación rígido, ya que los usuarios pertenecientes al esquema móvil quedarían descartados.

El mecanismo de autenticación propuesto en este trabajo, puede ser utilizado en distintos escenarios, tales como:

- Se requiere autenticar usuarios
- Se requiere dar un servicio a usuarios móviles (usuarios que no están registrados)
- Se requiere tener información como dirección MAC, dirección IP, hora de conexión de los dispositivos que usan los recursos de la red
- Se requiere tener redes inalámbricas con autenticación
- Se requiere de un sistema de autenticación multiplataforma

Lo anterior es posible debido a que un portal cautivo, cumple con los requisitos necesarios para realizar una autenticación, se apoya de protocolos robustos de autenticación, permite guardar información detallada de los dispositivos involucrados en la red, no depende de una plataforma en específico, entre otras cosas.

Los escenarios anteriormente mencionados se pueden generar en lugares como:

- Aeropuertos
- Restaurantes
- Librerías
- Universidades
- Bibliotecas
- Cibercafés
- Oficinas gubernamentales
- Empresas privadas
- Plazas comerciales
- Lugares públicos en general

El Laboratorio de Computación Sala C es un laboratorio abierto perteneciente a la División de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, ubicado en el 2° piso del Edificio Luis G Valdés Vallejo (Edificio Q), en el Anexo de Ingeniería.

Este laboratorio brinda servicio de préstamo de equipo de cómputo y acceso a internet. Cuenta con 60 equipos de escritorio y cuenta con 26 nodos tecnología Ethernet para uso de la infraestructura de red. Está disponible para alumnos, profesores y trabajadores adscritos a la División de Ingeniería Eléctrica, en el caso de los alumnos, deben de pertenecer a una de las siguientes carreras: Ingeniería en Computación, Ingeniería Eléctrica-Electrónica o Ingeniería en Telecomunicaciones, o deben de estar cursando Computación para Ingenieros o Programación Avanzada y Métodos Numéricos en el semestre en curso.

Dicho laboratorio tiene aproximadamente 800 usuarios registrados al semestre, actualmente se usa un sistema de autenticación basado en el servicio de SAMBA, esto sólo para los equipos de escritorio, pero la zona de nodos para dispositivos portátiles carece de un mecanismo de autenticación.

Dadas las características que presenta este laboratorio en la zona de nodos para dispositivos portátiles, es un buen candidato para implementar el mecanismo que se propone. Dicho laboratorio requiere: autenticación de usuarios, un mecanismo de autenticación multiplataforma, etc.

Para llevar a cabo la implementación de este mecanismo, se requiere saber la estructura de la red de datos, así como los servicios que se tienen en la red, lo anterior para evitar posibles conflictos con dichos servicios.

La red en la zona de nodos del Laboratorio de Computación físicamente está separada de la red de los equipos de escritorio, no posee algún dispositivo extra que se encargue del filtrado de paquetes, es una red tecnología Ethernet, no existen servicios que puedan ocasionar conflicto con la implementación a realizar, por lo que no es necesario tener consideraciones extras para realizar la implementación en este laboratorio. La figura 2.1 representa la red descrita anteriormente.

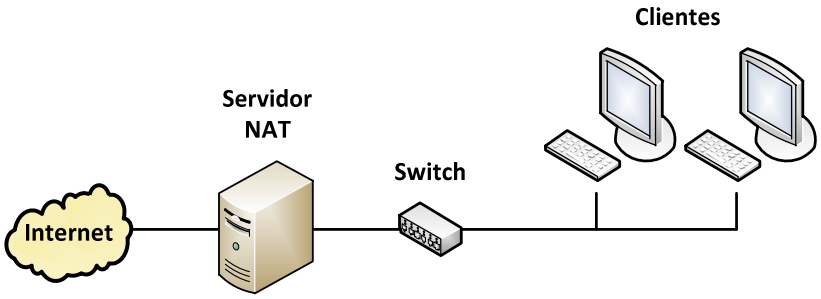


Figura 2.1 Red de la zona de nodos del Laboratorio de Computación Sala C

Con el mecanismo de autenticación propuesto en este trabajo, la red de nodos del Laboratorio de Computación Sala C, puede crecer o implementar tecnología inalámbrica sin presentar alguna complicación con el sistema creado, ya que dicho mecanismo permite una autenticación robusta en redes inalámbricas sin necesidad de requerir access points sofisticados.

I.I Análisis de las arquitecturas de red para autenticación

Existen diversas arquitecturas de red que son utilizadas para implementar un sistema de autenticación, las cuales dependen de la infraestructura de red que se posea y de los dispositivos de red que se tengan disponibles. [4]

Para la autenticación en redes inalámbricas se tienen las siguientes arquitecturas:

- El access point se encarga de validar la autenticación, se utilizan claves de red, el access point determina si la clave ingresada es correcta o no, de esta forma permite o deniega el acceso a la red. Es una arquitectura centralizada. La figura 2.2 muestra dicha arquitectura.

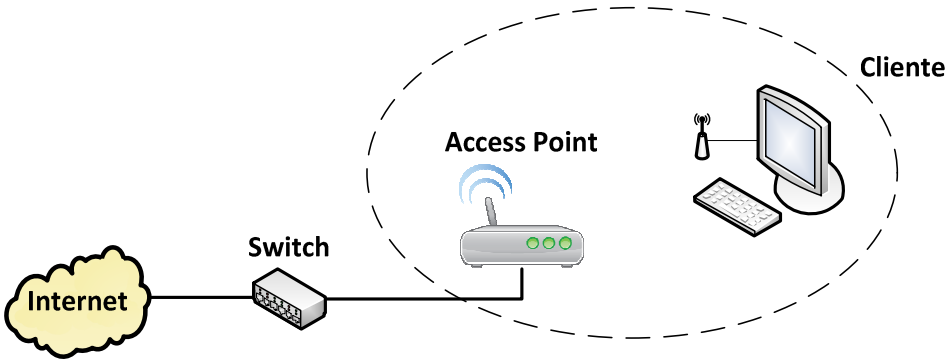


Figura 2.2 Arquitectura para la autenticación en WLAN mediante un access point

El usar este tipo de arquitectura presenta ciertas ventajas, pero también existen desventajas.

Las ventajas que presenta esta arquitectura son:
- Administración centralizada
- Cantidad de claves limitadas
- Se usa una clave general para el acceso a la red
- Controlar la cantidad de usuarios a conectarse

Las desventajas de la arquitectura son:
- Si el access point falla, la red inalámbrica falla
- Si la clave de red es comprometida, habrá accesos no autorizados (que un usuario, se la dé a otro usuario que no esté autorizado)
- Existen diversas herramientas que permiten obtener la clave de red

Este tipo de arquitectura es usada en redes privadas que usan seguridad WEP o WPA de clave precompartida.

WEP (Wired Equivalent Privacy): Es un protocolo para redes WiFi que permite cifrar la información que se transmite. Proporciona un cifrado basado en el algoritmo Rivest Cipher 4 (RC4) que utiliza claves de 64 o 128 bits.

WPA (WiFi Protected Access): Es un sistema creado para corregir las deficiencias del sistema WEP, ya que los investigadores encontraron varias debilidades en él, fue creado para utilizar un servidor de autenticación. Utiliza el algoritmo RC4 con claves de 128 bits pero su vector de inicialización es de una longitud mayor.

- El access point se ayuda de un servidor de autenticación, éste es el encargado de validar si la autenticación es correcta o no, y así permitir el acceso o denegarlo. Este tipo de arquitectura es usada cuando se implementa el modelo AAA. La figura 2.3 muestra esta arquitectura.

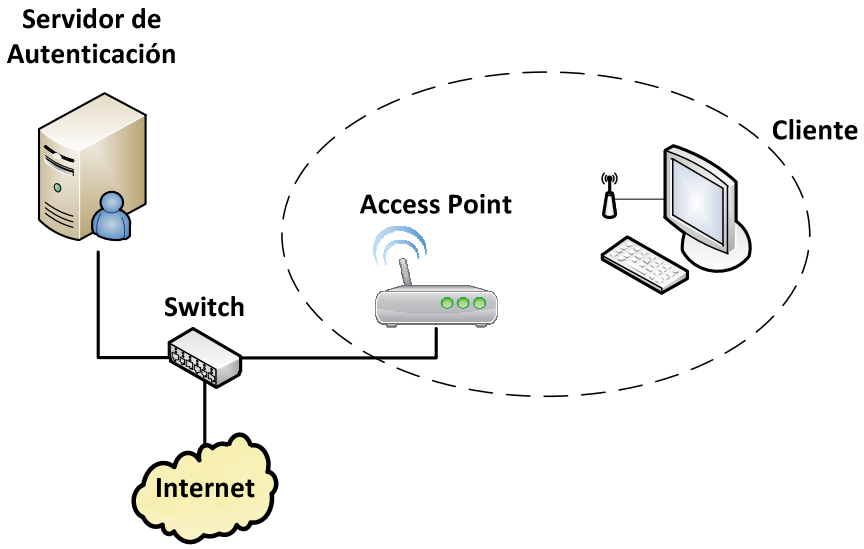


Figura 2.3 Arquitectura para la autenticación en WLAN mediante un servidor

Las ventajas que tiene el uso de esta arquitectura son:

- Las claves son almacenadas en un servidor independiente
- La autenticación se vuelve más segura
- Las claves son personalizadas para cada usuario
- La autenticación no solo depende de un dispositivo, se vuelve distribuida

Las desventajas que presenta esta arquitectura son:

- Si el access point falla, la red inalámbrica falla
- Si el servidor de autenticación no es accesible no se puede realizar la autenticación
- Existen vulnerabilidades en algunos servidores de autenticación (RADIUS)
- Administración más compleja, ya que se tiene que administrar el servidor de autenticación y los access point

Esta arquitectura es usada en redes privadas que utilizan seguridad WPA o buscan utilizar el modelo AAA.

En esta arquitectura el access point funciona como el intermediario entre el cliente y el servidor de autenticación, el cliente hace una petición de conexión al access point, éste a su vez reenvia la solicitud al servidor de autenticación y se queda a la espera de una respuesta, el servidor de autenticación determina si es un usuario válido o no, éste notifica al access

point, si es un usuario válido el access point permite la asociación del cliente y le envía los datos necesarios para utilizar la red, en caso contrario desconecta al cliente.

- Existe una arquitectura más robusta, las cuál incluye, una autoridad certificadora, un servidor de autenticación y un servidor de directorios. En la figura 2.4 se observa dicha arquitectura.

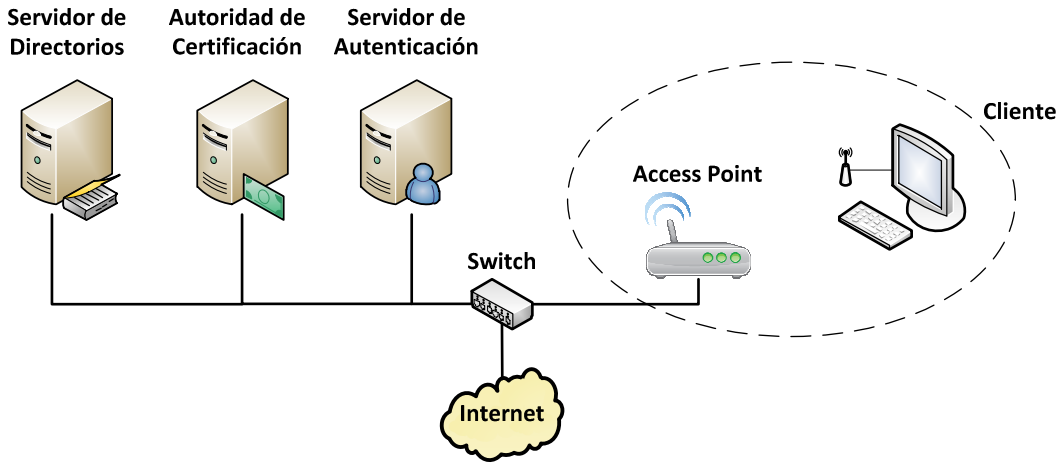


Figura 2.4 Arquitectura robusta para la autenticación en WLAN

Esta arquitectura es usada en redes privadas en las que se requiere un control de acceso muy estricto, ya que al tener cuatro entidades que participan en el proceso de autenticación este proceso se vuelve más robusto y aumenta la seguridad.

Antes de empezar el proceso de autenticación, el usuario debe de poseer un certificado digital, dicho certificado es generado por la autoridad certificadora, este certificado se puede generar de varias formas, una de ellas consiste en que el usuario solicite el certificado para un determinado uso y el administrador apruebe la solicitud; otra consiste en que el administrador genere el certificado manualmente, almacene la parte pública del certificado en el servidor de directorios y la parte privada del certificado en el equipo del usuario.

Una vez que se tenga el certificado, ahora el proceso de autenticación comienza con la solicitud del cliente al access point, el access point reenvía la petición al servidor de autenticación, por lo general, RADIUS, el servidor de autenticación consulta al servidor de directorios, el más común es LDAP, esta consulta es para comprobar las credenciales del usuario y así poder validarlo, también consulta en el servidor LDAP las políticas de acceso, es decir los permisos que tiene el usuario en la red, ahora el servidor de autenticación determina si el usuario tiene acceso a la red y envía el resultado al access point, si es un usuario autorizado, el access point inicia un intercambio de claves con el cliente para establecer una conexión de forma segura.

- Para la autenticación en redes alámbricas la arquitectura más común y simple es la que se muestra en la figura 2.5.

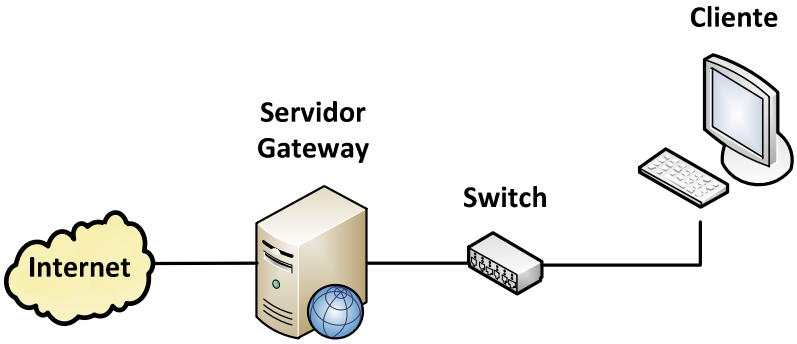


Figura 2.5 Arquitectura simple para la autenticación en LAN

En esta arquitectura el encargado de realizar la autenticación o decidir quién puede tener acceso a internet es el servidor gateway.

Las ventajas que tiene el uso de esta arquitectura son:

- Administración centralizada
- Se filtra mediante dirección MAC o IP
- Controlar los protocolos que se pueden usar en la red

Las desventajas que presenta esta arquitectura son:

- Si el servidor gateway falla, los clientes no tienen acceso a internet
- Si el servidor gateway no puede comunicarse con internet, ningún cliente podrá conectarse a internet
- Si el servidor gateway es vulnerado, los usuarios pueden tener acceso a internet no autorizado
- Al ser un sistema que depende de un sólo equipo es propenso a recibir diversos ataques

II.II Análisis de las herramientas para autenticación

RADIUS

Es un protocolo de autenticación, que cumple con la arquitectura AAA, por lo general, es usado en redes inalámbricas. Es un protocolo robusto, es compatible con otros protocolos de autenticación.

- Se requieren dispositivos especiales que soporten la autenticación mediante el servicio de RADIUS.
- En ocasiones el cliente requiere de conocimientos básicos de redes para poder configurar su dispositivo.
- Es un servicio complejo de configurar y administrar.
- Permite autenticar mediante, usuario, contraseña, dirección MAC, o combinaciones de éstas.

LDAP
Es un protocolo de autenticación, cumple con la arquitectura AAA, usado para fortalecer el proceso de autenticación, se considera como una base de datos, está basado en el uso de directorios, que son un conjunto de objetos con atributos organizados en una manera lógica y jerárquica.

- Se requiere de conocimientos especializados en redes y en administración de sistemas para poder administrar y configurar este servicio.
- Fortalece los sistemas de autenticación cuando es empleado de forma correcta.
- Permite crear usuarios con diversos atributos, dichos usuarios pueden ser agruparlos, lo que permite tener diferentes esquemas de autenticación y autorización.

WEP (Wired Equivalent Privacy)
Es un protocolo que permite la autenticación de redes inalámbricas, utiliza el algoritmo de cifrado RC4, que utiliza claves de 64 o 128 bits (40 o 104 bits más 24 bits del vector de iniciación).

- Actualmente este tipo de autenticación es fácilmente vulnerado, ya que existe software que permite el descifrado de las claves generadas por este mecanismo.
- Las claves generadas tienen que ser repartidas a cada uno de los usuarios.
- Para mayor seguridad se recomienda generar nuevas claves periódicamente.

WPA (WiFi Protected Access)
Surge como mejora al protocolo WEP, es utilizado para la autenticación en redes inalámbricas. Este protocolo puede ser usado, como clave precompartida o puede utilizar un servidor de autenticación, en dicho servidor se almacenan los datos y contraseñas de los usuarios de la red.

- Como clave precompartida, utiliza el algoritmo RC4, pero ahora el vector de inicialización es de 48 bits y la clave de 128 bits.
- Si se usa un servidor de autenticación, se deben realizar ciertas configuraciones, por lo general se usa un servidor RADIUS.
- Incorpora el protocolo de integridad de clave temporal (TKIP), que cambia las claves de forma dinámica.
- Existe software que permite descifrar las claves que son generadas mediante este mecanismo.

Kerberos
Es un protocolo de autenticación que permite a dos dispositivos en una red insegura demostrar su identidad de forma segura. Utiliza el modelo cliente-servidor. Tanto el cliente como el servidor verifican la identidad uno del otro. Este mecanismo de autenticación se basa en la criptografía de clave simétrica y requiere de un tercer dispositivo de confianza que se le conoce como, centro de distribución de claves (KDC), el cual se forma por un servidor de autenticación y un servidor de emisión de tikets.

- Para configurar este servicio se requiere de conocimientos especializados.
- La administración y configuración es compleja.
- Es un sistema que llega a ser transparente al usuario.

SAMBA
Es un protocolo enfocado a compartir archivos entre diferentes plataformas, utiliza el modelo cliente-servidor. Este servicio también se puede utilizar como un mecanismo de autenticación, a este tipo de configuración se le conoce como controlador principal de dominio (PDC).

- Es compatible con otros mecanismos de autenticación, como LDAP o Kerberos.
- La configuración y administración de este servicio, depende del uso de un servidor externo de autenticación, ya que puede ser compleja o de dificultad media.
- Se requieren de conocimientos especializados para instalar este servicio.

Certificados de seguridad
Un certificado es un archivo que posee información sensible que permite identificar a un usuario válido. Existen diversos tipos de certificados, que permiten verificar la seguridad en una conexión entre un cliente y un servidor.

- Son utilizados para encriptar los datos que se envían a un servidor.
- Para generar los certificados se requiere de conocimiento especializado.
- Poseen un alto grado de seguridad.

Portal Cautivo
Es un mecanismo que proporciona un método para controlar el acceso a las redes públicas y privadas que aprovechan las tecnologías web existentes. Los usuarios sólo tienen que abrir su navegador web y autenticarse en la página a la que fueron redireccionados. Los principales beneficios de esta solución son:

- No se requiere de un software o una configuración especial en el cliente.
- Los clientes son capaces de obtener acceso a la red sin importar el tipo de dispositivo o sistema operativo que estén utilizando.
- El objetivo principal de los portales cautivos es la autenticación.
- También ofrecen beneficios adicionales, por ejemplo, hacer que nuevos usuarios realicen un proceso de registro, permitir acceso a una cuenta existente o proporcionar un acceso limitado hacia los servicios de red, en otras palabras permitir solo ciertos sitios de internet, ya sea un sitio web de la compañía o información relacionada con un lugar en particular.

II.II.I Análisis de las principales herramientas de portal cautivo

Las herramientas más comunes para implementar un portal cautivo se mencionan a continuación.

PepperSpot (Linux)
Es un portal cautivo que permiten a un usuario autenticado acceder a un servicio de red. PepperSpot está destinado para ser utilizada por clientes inalámbricos, está basado en el proyecto de portal cautivo ChilliSpot. La particularidad de PepperSpot es que provee el acceso IPv6 a los clientes WiFi. Soporta autenticación vía web. Soporta el uso de WPA. Soporta la autenticación mediante el uso de un servidor RADIUS. Posee una licencia GNU GPL (General Public License). [30]

NoCatAuth (Linux)
Es un portal cautivo de código abierto. Está desarrollado en perl. Permite la autenticación vía web. No se requiere de alguna configuración en el cliente. Soporta diversos métodos de autenticación, como RADIUS, base de datos, LDAP, NIS, SAMBA, etc.
Es implementado por otros productos para obtener un sistema de autenticación más completo. Por ejemplo, OperWrt, Ewrt, Metrix Networking Kits. [17]

Chillispot (Linux)
Es un portal cautivo de código abierto. Se utiliza para autenticar a los usuarios de una red inalámbrica. Soporta la autenticación vía web, que es la forma estándar para los hotspots públicos. Soporta el uso de un servidor RADIUS, que es el que controla la autenticación, autorización y contabilidad (AAA). Existen binarios disponibles para Redhat, Fedora, Debian, Mandrake y OpenWRT. ChilliSpot fue compilado bajo FreeBSD. El código fuente está bajo licencia GNU GPL. [31]

CoovaChilli (Linux)
Es un software de código abierto para el control de acceso, está basado en el proyecto ChilliSpot. Está publicado bajo licencia GNU GPL. CoovaChilli es un software que proporciona un portal cautivo con un ambiente protegido utilizando un servidor RADIUS o el protocolo HTTP para el acceso. CoovaChilli es una parte integral de CoovaAP, firmware basado en OpenWRT, el cual está especializado para los hotspots. [32]

AirMarshal (Linux, Windows)
Es un portal cautivo que es adecuado para una variedad de entornos, incluyendo hoteles, universidades, HostSpots, redes de clientes, etc. Proporciona compatibilidad con el servicio de RADIUS según los estándares de la AAA. Permite la integración con plataformas existentes basadas en los estándares de facturación y administración, por ejemplo, Emerald. También es capaz de proteger la información confidencial de los clientes, usando la tecnología SSL, para proteger las contraseñas que el usuario introduce en la interfaz web. [34]

ZeroShell (Linux)
Es una distribución de Linux para servidores y dispositivos integrados destinados a proporcionar servicios de red. Está disponible en forma de LiveCD o de imagen de Compact Flash y se puede configurar y administrar utilizando un navegador web.
Proporciona un servidor RADIUS para proporcionar una autenticación segura.
Implementa un portal cautivo de forma nativa, éste es usado para el apoyo en el inicio de sesión vía web. Este portal cautivo utiliza el servicio de Kerberos para la autenticación de usuarios y se comunica directamente con el firewall de ZeroShell para permitir el acceso a la red.
Tiene soporte para autorización vía LDAP, NIS y RADIUS.
Es una distribución que ha sido generada por su autor, no se basa en alguna distribución existente. [35]

PfSense (FreeBSD)
Es una distribución personalizada de FreeBDS de código abierto adaptado para ser usado como firewall y router. Es una plataforma potente, flexible y de ruteo, que incluye:

- Características relacionadas al sistema base.
- Un sistema de paquetes que le permiten expandirse sin incrementar las vulnerabilidades que presenta el sistema base.

PfSense es un proyecto muy popular, que ha sido utilizado en pequeñas redes domésticas, redes de grandes corporaciones, universidades y otras organizaciones que protegen sus dispositivos de red. Está basado en el proyecto m0n0wall, pero está enfocado hacia la instalación en PC, aunque también soporta el enfoque que tiene m0n0wall, que está enfocado a sistemas embebidos (hardware). [36]

OpenSplash (FreeBSD)
Es un portal cautivo para redes inalámbricas de código abierto para FreeBSD para ser utilizado en redes públicas. Este proyecto fue creado basándose en el proyecto Wicap, pero ha incluido mejores características y ha aumentado su seguridad. Sirve como un centro de control entre los componentes de software necesarios, como apache y FreeRADIUS. Para implementar esta herramienta es necesario tener conocimientos sobre ciertas configuraciones en FreeBSD. Utiliza ipfirewall, Perl y varias utilidades del sistema UNIX.
Actualmente a este proyecto ya no le están dando seguimiento. [37]

m0n0wall (embedded FreeBSD)
Es un sistema UNIX que su objetivo es crear un paquete de software completo, que al ser implementado en algún sistema embebido, proporciona todas las características importantes de los firewalls comerciales de hardware, con la misma facilidad de uso pero a una fracción del precio comercial.
m0n0wall se basa en una versión básica de FreeBSD, junto con un servidor web, PHP y otras utilidades. La configuración del sistema se almacena en un archivo de texto plano de tipo XML.
m0n0wall es probablemente el primer sistema UNIX que realiza su configuración al momento de iniciar con PHP, en lugar de los scripts de shell habituales y que tiene la configuración completa almacenada en formato XML. [38]

Ewrt (embedded Linux - WRT54G, Linux)
EWRT son las siglas de Enhanced Wireless Receiver/Transmitter, es una distribución de Linux para routers inalámbricos Linksys WRT54G y WRT54GS.
El objetivo de EWRT era crear una distribución de software “estable y fácil de usar en hotspot-in-a-box” que sería utilizado en apartamentos, hoteles, centros comerciales, etc.
Es una distribución de código abierto, lo que nos permite modificarlo para adaptarlo a las necesidades. EWRT proporciona un portal cautivo que se basa en NoCatSplash-CVS. Soporta el protocolo SSH y otras funcionalidades para la administración remota.
La parte de NoCatSplash la modificaron para que pudiera trabajar con la plataforma del router. Estas modificaciones han permitido mejoras para el control del desbordamiento del bufer, mayor seguridad y parches de estabilidad. [39]

HotSpotSystem (embedded Linux, WRT54GL, Mikrotik, etc)
Es un sistema que ofrece servicios de gestión y facturación a empresas o personas que quieren ofrecer el servicio de Internet a sus clientes. Trabajan con bases de datos, servidores web y servidores RADIUS. Poseen un software que puede correr en servidores Linux o en routers inalámbricos como Linksys, Buffalo, D-Link, Ubiquiti y Mikrotik. [40]

WifiDog (embedded Linux - OpenWRT, Linux, Windows)
Es portal cautivo de código abierto para quienes deseen administrar un access point o una red donde prevengan el abuso de la conexión de internet. Es multiplataforma ya que utiliza un

navegador web. Esta desarrollado en C para que pueda ser utilizado en sistemas embebidos. Tiene tres funciones principales:

- Entregar los datos sean externos o internos.
- Autenticar y autorizar.
- Monitorear la red de forma centralizada.

El desarrollo de esta solución ha sido por contribuciones de grupos pertenecientes a la comunidad wireless. Se ha diseñado de una forma tan genérica que es usado por diferentes tipos de organizaciones. [41]

Antamedia HotSpot Software (Windows)
Es un software que permite administrar una red inalámbrica con cualquier PC y un router. Permite configurar sólo un router o toda una red completa. Permite controlar y facturar el acceso a internet gracias a la derivación de los clientes hacia a una página de registro o hacia un sitio web donde se realiza el pago. Permite controlar descargas, ancho de banda, configurar planes de cuotas, etc. Ofrece estadísticas e informes en tiempo real, filtrado por IP o por MAC.
Está diseñado para ser instalado en plataformas Windows, es un software licenciado. [42]

FirstSpot (Windows)
Es un software para la plataforma de Windows, diseñado para administrar y asegurar una red inalámbrica o alguna otra red de forma centralizada. Se basa en la tecnología del portal cautivo, permite a los usuarios de la red acceder a ésta simplemente usando un navegador web. Una de las características de FirstSpot es su facilidad de uso y el poder controlar la red.
Se comporta como un servidor de precios, que ofrece la mayor flexibilidad en la estructura de los planes de acceso. FirstSpot está considerado como el software más usado en plataformas Windows a nivel mundial y ha sido implementado en hoteles, cafeterías, ISPs, cibercafés, bibliotecas, despachos, gobierno, incluso por militares. [43]

En la tabla 1, se muestra un resumen de las características que presenta cada herramienta.

Tabla 1. Comparación de tecnologías de portal cautivo

Software	Características									
	Código abierto	Plataforma	Licencia	Método de Autenticación						
				RADIUS	LDAP	SAMBA	Base de Datos	PAM	Archivos de Texto	Otros
PepperSpot	Sí	Linux	GNU GPL	Sí	-	-	-	-	-	Sí
NoCatAuth	Sí	Linux	Libre	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Chillispot	Sí	Linux	GNU GPL	Sí	-	-	-	-	-	-
CoovaChilli	Sí	Linux	GNU GPL	Sí	-	-	-	-	-	Sí
AirMarshal	-	Linux, Windows	Cobro	Sí	-	-	-	-	-	Sí
ZeroShell	Sí	Linux	GNU GPL	Sí	Sí	-	-	Sí	-	Sí
PfSense	Sí	FreeBSD	BSD	Sí	Sí	-	-	-	Sí	Sí

OpenSplash	Sí	FreeBSD	-	Sí	-	-	-	-	-	Sí
m0n0wall	Sí	FreeBSD embebido	Libre	Sí	-	-	-	-	Sí	Sí
Ewrt	Sí	Embebido	Libre	Sí	Sí	Sí	Sí	-	-	Sí
HotSpotSystem	No	Embebido	Libre	Sí	-	-	Sí	-	-	Sí
WifiDog	Sí	Linux, Windows, embebido	GNU GPL	Sí	-	-	-	-	-	Sí
Antamedia HotSpot	No	Windows	Cobro	-	Sí	-	-	-	-	Sí
FirstSpot	No	Windows	Cobro	Sí	-	-	Sí	-	-	Sí

II.III Selección de las herramientas a utilizar

Se utilizó un sistema de autenticación mediante el uso de un portal cautivo, éste se seleccionó para poder tener dos esquemas de red diferentes administrados por un sólo sistema. Los esquemas de red que se tienen son:

Esquema de usuarios registrados: Aquellos usuarios que previamente se han registrado, proporcionándoles un login y un password. Cuando un usuario se autentique de forma correcta podrá usar la red y acceder a páginas en internet.

Esquema de usuarios no registrados: Aquellos usuarios que no se han registrado, podrán usar la red, pero sólo podrán acceder a las páginas web que se han autorizado en la configuración, incluso puede configurarse para que este tipo de usuarios puedan registrarse por ellos mismos para obtener un login y password.

La herramienta que se utilizó es NoCatAuth, esta herramienta se seleccionó principalmente por sus características y su compatibilidad con otros sistemas de autenticación, que permiten generar arquitecturas más robustas[2].

NoCatAuth es una herramienta desarrollada en perl, es un software libre, está desarrollado para la plataforma de Linux, y es compatible con varios sistemas de autenticación.

NoCatAuth se auxilia de la herramienta iptables, que es un módulo que traen por default los sistemas operativos Linux.

II.III.I Herramientas que utiliza NoCatAuth

A continuación se explica brevemente en qué consiste cada herramienta que utiliza el servicio de NoCatAuth.

[2] Ver tabla 2

II.III.I.I Perl (Practical Extraction and Report Language)

Es un lenguaje de programación tipo scripting, es decir, es un lenguaje interpretado. Este lenguaje fue desarrollado por Larry Wall.

Oficialmente representa al lenguaje para la extracción práctica y generación de reportes. Originalmente fue un lenguaje optimizado para el escaneo de archivos de textos, extracción de información y la generación de reportes basados en la información extraída.

Rápidamente se convirtió en un lenguaje para realizar muchas tareas de administración del sistema. A través del tiempo Perl se ha convertido en un lenguaje de programación de propósito general. Es usado para cualquier tipo de tarea, se puede usar de forma rápida "one-liners", instrucciones un una sola línea de comandos, o también se puede utilizar para desarrollar aplicaciones de gran escala. [29]

El lenguaje está pensado para ser práctico, es decir, fácil de usar, eficiente, completo y no ser un lenguaje bonito, en otras palabras, pequeño, elegante, mínimo.

Perl combina características de lenguaje C, sed, awk y shellscripting, por lo que las personas que han trabajado con este tipo de herramientas deben de tener pocas dificultades para manejar perl.
La sintaxis de este lenguaje es muy similar a la sintaxis del lenguaje C. A diferencia de la mayoría de las herramientas de UNIX, Perl no limita arbitrariamente el tamaño de los datos, por ejemplo, si se trabaja con la memoria, Perl puede contenerla en un archivo como si fuera una sola cadena. Perl también trabaja con recursividad, y su recursividad se puede decir que es ilimitada. Perl usa sofisticadas técnicas de búsqueda de patrones para escanear grandes cantidades de datos rápidamente. Aunque está optimizado para el escaneo de texto, Perl también tiene muchas herramientas excelentes para cortar y rebanar los datos binarios.

II.III.I.II Iptables

Se utilizan para establecer, mantener e inspeccionar las tablas que contienen las reglas de filtrado de paquetes IPv4 en el kernel de Linux. Existen diferentes tipos de tablas que pueden ser utilizadas. Cada tabla contiene un número de reglas predefinidas y también pueden contener reglas definidas por el usuario, estás forman una chain.

Cada chain, se forma mediante reglas que pueden coincidir con un conjunto de paquetes. Cada regla indica que hacer con los paquetes que coincide. A esto se le conoce como "target".

Una regla especifica el criterio para un paquete y su destino. Si el paquete no coincide con la regla establecida, la siguiente regla en la chain es examinada, si coincide, la siguiente regla está especificada por el valor del target, el cual puede ser el nombre de una chain definida por el usuario o uno de los valores especiales ACCEPT, DROP, QUEUE o RETURN.

ACCEPT significa permitir el paquete.
DROP significa descartar el paquete.

QUEUE significa pasar el paquete al userspace, los paquetes se almacenan en forma de una estructura de datos llamada QUEUE o cola.
RETURN significa detener el recorrido de la chain y continuar con la siguiente regla en la chain anterior.

Actualmente existen tres tablas generales y una especial, estas tablas son independientes entre sí.

Filter: Es la tabla por default, si no se indica el tipo de tabla a usar. Contiene las chains, INPUT, para paquetes destinados a sockets locales, FORWARD, para paquetes encaminados por la máquina, OUTPUT, para modificar los paquetes generados antes de encaminarlos, y POSTROUTING, para modificar los paquetes que están a punto de salir.

Nat: Esta tabla es consultada cuando un paquete que crea una nueva conexión es encontrado. Se compone de tres chains, PREROUTING, para modificar paquetes tan pronto lleguen, OUTPUT, para modificar los paquetes generados antes de encaminarlos, y POSTROUTING, para modificar los paquetes que están a punto de salir.

Mangle: Esta tabla se usa para modificar paquetes especializados. Se conforma de las chains, PREROUTING, para modificar paquetes entrantes antes del enrutamiento, OUTPUT, para modificar los paquetes generados antes de encaminarlos, INPUT, para los paquetes que llegan a la máquina, FORWARD, para modificar los paquetes que son encaminados por la máquina, y POSTROUTING, para modificar los paquetes que están a punto de salir.

Raw: Esta tabla se usa principalmente para la configuración de las exenciones del seguimiento de conexiones en combinación con el target NOTRACK. Provee las siguientes chains, PREROUTING, para paquetes que llegan a través de cualquier interfaz de red, OUTPUT, para paquetes generados en procesos locales.

II.III.I.III Lenguaje de marcado de hipertexto (HTML)

Es un lenguaje de programación que surge a partir de las etiquetas SGML (Standard Generalized Markup Language – Lenguaje Estándar de Marcas Generalizado), dicho lenguaje es un estándar de ISO. Este lenguaje permite ordenar y etiquetar diversos documentos dentro de una lista, éste se utiliza para especificar los nombres de las etiquetas que se utilizan al momento de ordenar. SGML también sirve para especificar las reglas del etiquetado en un documento y no posee un conjunto de etiquetas predefinido.

El lenguaje HTML es el lenguaje principal para elaborar páginas web. Es usado para describir la estructura y el contenido de un sitio web, también se usa para complementar el texto con objetos de otro tipo, como imágenes. Otra funcionalidad de este lenguaje es describir la apariencia de un documento, por otra parte, permite incluir código de lenguajes tipo script que pueden afectar el comportamiento de un navegador web, por ejemplo, JavaScript.

Es un lenguaje muy simple y general que sirve para definir diferentes elementos, para utilizar este lenguaje se crean etiquetas que aparecen especificadas a través de corchetes angulares “<” y “>”.

La primera descripción de HTML que fue pública fue un documento llamado HTML tags, publicado en internet en 1991, por Tim Berners-Lee.

Los componentes básicos de este lenguaje son: [10]

Elementos: Los elementos son la estructura más básica de HTML. Éstos tienen dos propiedades básicas, atributos y contenido, cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Los elementos generalmente tienen una etiqueta de inicio, <nombre-elemento> y una etiqueta de fin </nombre-elemento>.

Atributos: Los atributos son un conjunto de pares de nombre-valor, separados por el signo "=" y se escriben después del nombre de una etiqueta de inicio.
<nombre-elemento atributo="valor">

Contenido: El contenido es la información que se quiere mostrar, éste se ubica entre las dos etiquetas, la de inicio y la de fin.
<nombre-elemento atributo="valor">Contenido a mostrar </nombre-elemento>

Para crear o editar un archivo HTML, se puede hacer desde cualquier editor de textos básico, tales como WordPad, Notepad++, bloc de notas, vi, vim,emacs, entre otros.

Existen editores que permiten realizar páginas web con características WYSIWYG (What You See Is What You Get – lo que ves es lo que obtienes). Estos editores funcionan a partir de una interfaz gráfica que permite generar código HTML a partir de lo que se coloque en el documento, texto, imágenes, texto enriquecido, etc., en otras palabras estos editores generan el código HTML de forma automática conforme se va editando la página web de forma gráfica.

II.III.I.IV Interfaz de entrada común (CGI - Common Gateway Interface)

Es una importante tecnología de la WWW (World Wide Web) que permite a un cliente web solicitar datos a un programa que fue ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. También es un mecanismo de comunicación entre un servidor web y una aplicación externa cuyo resultado final de la ejecución se conoce como objetos MIME. Las aplicaciones que se ejecutan en un servidor web reciben el nombre de GCI.

Objetos MIME: los objetos MIME (Multipurpose Internet Mail Extensions) son una serie de convenciones o especificaciones para el intercambio de todo tipo de archivos (texto, audio, vídeo, imagen, etc.) de forma transparente para el usuario a través de Internet.
Las aplicaciones CGI fueron de los primeros intentos para crear contenido dinámico para páginas web. En una aplicación CGI, el servidor web pasa las solicitudes del cliente aun programa externo. Este programa puede estar escrito en cualquier lenguaje de programación que soporte el servidor, aunque por razones de portabilidad de utilizan con mayor frecuencia los lenguajes de script. La salida de dicho programa es enviada al cliente en lugar del archivo estático tradicional.

Gracias a la creación de las CGI se han implementado funciones nuevas en las páginas web, de tal forma que esta interfaz rápidamente se convirtió en un estándar, lo que permite que cualquier tipo de servidor web la implemente.

Funcionamiento básico de una CGI

1. El servidor recibe una petición, y comprueba si se trata de una invocación a una GCI.
2. El servidor prepara el entorno para la ejecución de la aplicación. Mucha de la información que se usa en este punto proviene del cliente.
3. El servidor ejecuta la aplicación, capturando la salida estándar.
4. La aplicación es ejecutada, como consecuencia de su actividad se genera un objeto MIME, dicho objeto los escribe en su salida estándar.
5. Cuando la aplicación finaliza, el servidor envía la información producida, junto con información propia al cliente, que se encuentra en estado de espera.

II.III.II Estructura del portal cautivo NoCatAuth

II.III.II.I Definición de NoCat.Net

Esta comunidad se creó en el condado de Sonoma, California, EUA, comenzó como una comunidad de apoyo para las redes inalámbricas 802.11b y ha crecido gracias a sus proyectos, cuyo objetivo es fomentar la creación de redes inalámbricas comunitarias.

El principal objetivo de esta comunidad es brindar ancho de banda infinito en cualquier parte de forma gratuita.

Los proyectos que tiene esta comunidad son los siguientes: [17]

- NoCatAuth es la implementación de portal cautivo, basándose en el concepto "catch and release". Proporciona una página web de bienvenida para los clientes de la red, así como una variedad de modos de autenticación. Está escrito en Perl.
- NoCatSplash es la versión en C de NoCatAuth. Actualmente soporta la pantalla de bienvenida, es decir, el portal cautivo se encuentra en modo abierto, para el acceso mediante autenticación aún está en versión beta.
- NoCat Maps es una base de datos de nodo libre, una herramienta de mapeo y es un sitio que funciona como herramienta de encuesta preliminar, construido con herramientas de código abierto.
- WSCICC - Western Sonoma Country Internet Cooperative. Es una organización que está formada por los residentes locales con el objetivo de administrar la red inalámbrica del condado de Sonoma.
- También administran listas de correo, van a conferencias, publican libros y artículos, y generalmente difunden el conocimiento de la tecnología inalámbrica.

La herramienta NoCatAuth, es la de mayor interés para cumplir el objetivo planteado, debido a que es una implementación de un portal cautivo, esta herramienta se divide en dos partes, el servidor NoCatGateway y el servidor NoCatAuth.

II.III.II.II NoCatGateway

Es el servicio que se encarga de hacer el filtrado de las peticiones web de forma dinámica, es decir, permite o deniega el acceso a internet. Funciona como un firewall, el cual se está actualizando constantemente para permitir el acceso a los usuarios conforme se vayan autenticando. Usa el módulo de iptables, que es un módulo específico para generar firewalls en los sistemas Linux.

II.III.II.III NoCatAuth

Esta parte es la encargada de realizar la autenticación de usuarios, ya sea que se haga de forma local, usando el método de autenticación que trae incorporada la herramienta, que es almacenamiento de usuarios y contraseñas en archivos de texto plano o utilizando algún otro método de autenticación como una base de datos, radius, ldap, samba, nis, etc., este tipo de servicios pueden estar ejecutándose de forma local o remota. La figura 2.6 muestra la estructura general de NoCat.

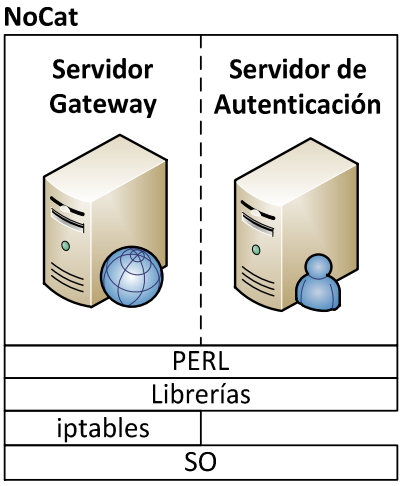


Figura 2.6 Estructura del portal cautivo NoCatAuth

II.III.II.IV Funcionamiento de NoCatAuh

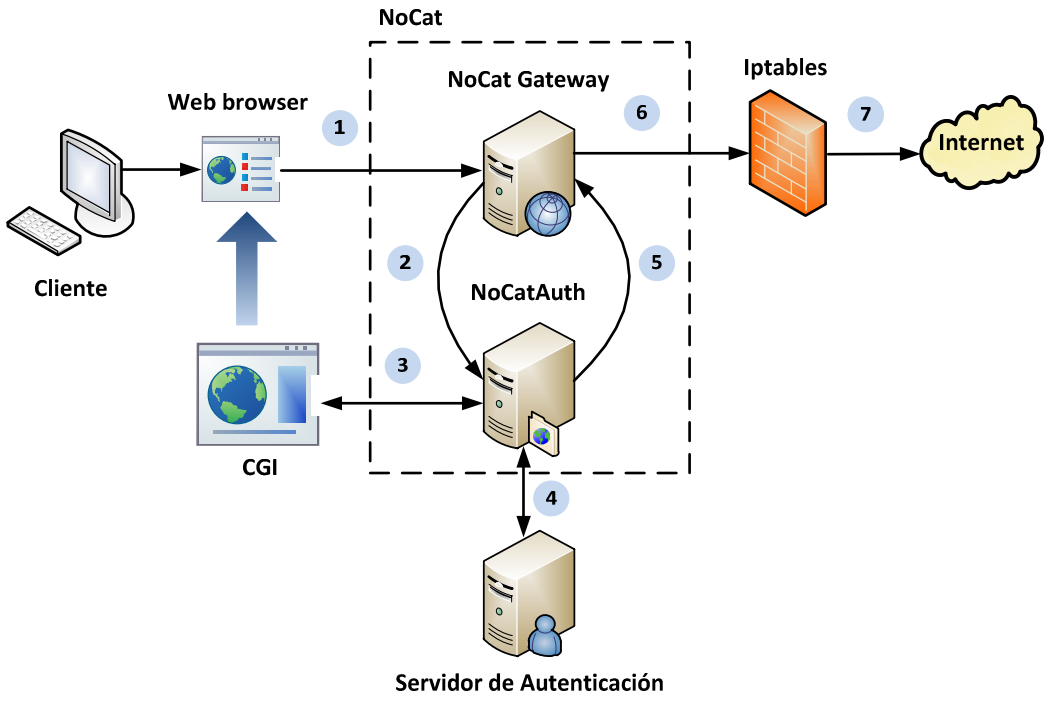


Figura 2.7 Funcionamiento del portal cautivo NoCatAuth

La figura 2.7 describe el funcionamiento de la herramienta de NoCat:

1. El cliente a través de un navegador web solicita una página de internet, dicha solicitud es capturada por el servidor NoCat Gateway.
2. El servidor NoCat Gateway redirecciona la petición del cliente hacia el servidor NoCatAuth, en especifico a un formulario web.
3. El servidor NoCatAuth despliega al cliente un formulario web, donde solicita los datos de autenticación, usuario y contraseña. El cliente interactúa con el formulario web y al momento de enviar sus datos se activa el funcionamiento del CGI, que es el encargado de procesar la información que envió el cliente.
4. El servidor NoCatAuth con apoyo del CGI, se comunica con el servidor de autenticación, ya sea un RADIUS, LDAP, SAMBA, base de datos, etc., para llevar a cabo la autenticación. El servidor de autenticación se encarga de validar los datos enviados, estos datos son enviados según el protocolo a usar. Posteriormente le responde al servidor NoCatAuth el resultado de la validación.
5. El servidor NoCatAuth le envía los datos del cliente al servidor NoCat Gateway y le indica que es un usuario válido. Al mismo tiempo mediante la CGI, el servidor NoCatAuth le regresa al cliente como resultado de la consulta, una redirección al servidor NoCat Gateway y le otorga un ticket que concatena a dicha consulta, este ticket lo identifica como usuario válido.
6. El servidor NoCat Gateway actualiza sus reglas, para colocar al cliente como un usuario válido y permitir el paso de sus peticiones.
7. Cuando la petición del cliente le llega nuevamente al servidor NoCat Gateway, como ya es un usuario válido, lo reconoce por el ticket que trae en la solicitud, permite pasar la petición a la página de internet que el cliente solicitó inicialmente, todas las solicitudes posteriores del cliente las permite pasar ya que en sus reglas ya se encuentra dicho usuario.

II.IV Selección de la arquitectura a usar

La sección de nodos del Laboratorio de Computación Sala C se desea administrar de una forma más eficiente y mejorar el control de acceso, cuenta con 26 nodos que son tecnología Ethernet, es decir, son nodos cableados.

Dada la naturaleza de los nodos, se usó la arquitectura más simple para la autenticación en una red alámbrica, como se muestra en la figura 2.8, y a partir de ésta se hicieron las modificaciones necesarias.

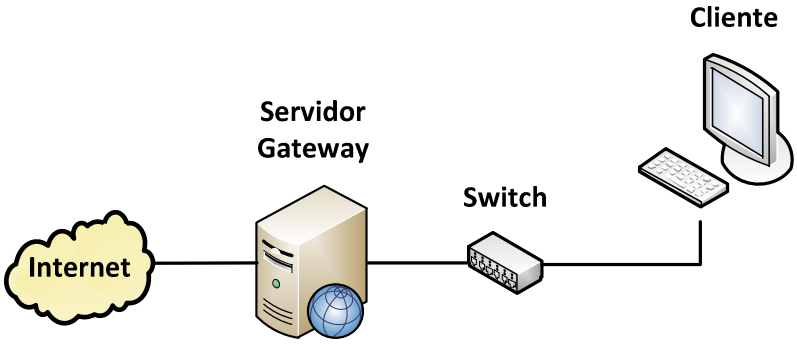


Figura 2.8 Arquitectura simple para la autenticación en LAN

La documentación y manuales de terceros acerca de la herramienta de NoCat, proponen la arquitectura que se muestra en la figura 2.9.

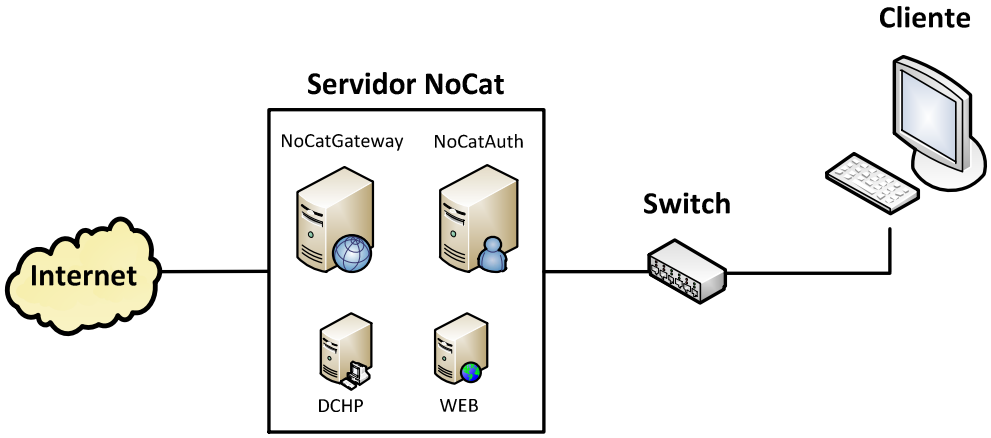


Figura 2.9 Arquitectura simple usando NoCat

Esta arquitectura propone que en un mismo equipo se instale tanto el servidor Gateway, como el servidor de autenticación, además de instalar servicios adicionales para que la herramienta funcione de forma correcta, como un servidor web y un servidor DHCP.

Por cuestiones de seguridad, se decidió separar los servicios, principalmente el NoCatGateway y NoCatAuth, lo anterior es para que el sistema se encuentre distribuido, lo que generó la arquitectura que se muestra en la figura 2.10.

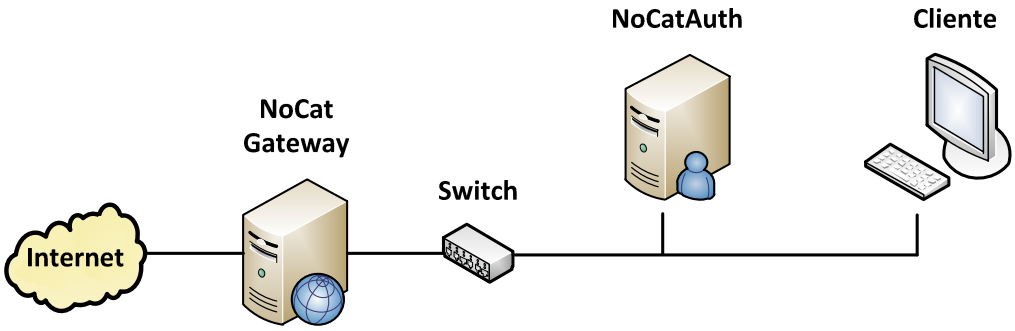


Figura 2.10 Arquitectura simple separando los servicios de NoCat

En esta arquitectura el servicio de autenticación ya se encuentra separado del servicio gateway, lo que complica el acceso no autorizado a la red, mejorando de cierta forma la seguridad en la red, ya que por la arquitectura tendrían que corromper dos servidores que se encuentran física y lógicamente en lugares distintos.

Gracias a las características de la herramienta NoCat, el servicio NoCatAuth puede conectarse a otros servidores de autenticación, lo que nos permite utilizar un servidor RADIUS, o LDAP, o NIS, o SAMBA, etc., lo anterior nos brinda una autenticación más segura ya que nos respalda un

protocolo más robusto. Este arreglo de servicios generó una arquitectura de red como la que se muestra en la figura 2.11.

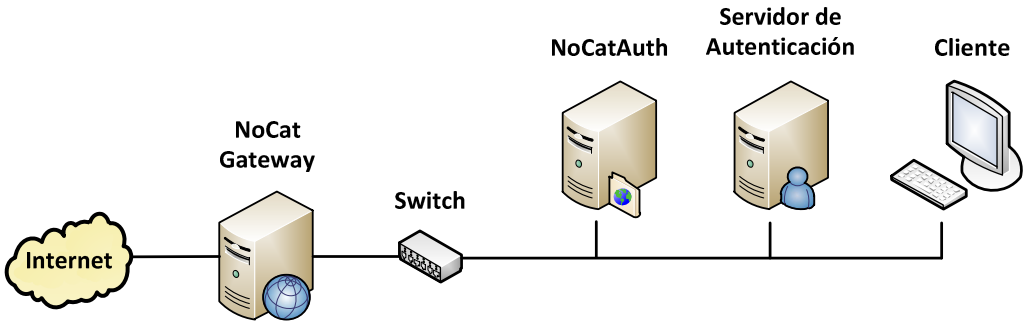


Figura 2.11 Arquitectura usando NoCat + servidor de autenticación

El servidor de autenticación también puede encontrarse en otra parte de internet, no es forzoso que se encuentre dentro de la red interna, lo que genera una arquitectura de red muy similar a la anterior pero con una ligera variación. Dicha arquitectura se muestra en la figura 2.12.

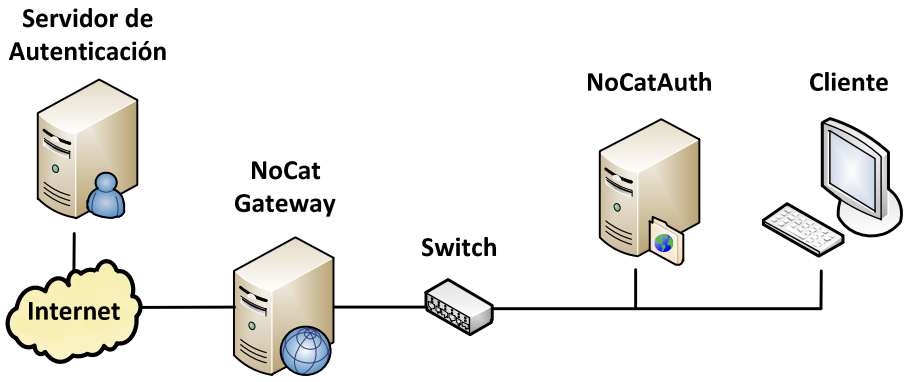


Figura 2.12 Arquitectura usando NoCat + servidor de autenticación externo

Para resolver el problema se optó por usar la arquitectura de red en la que el servidor de autenticación pertenece a la red interna, ya que en el Laboratorio de Computación Sala C, existen equipos disponibles para la configuración de servidores de autenticación.

Por otra parte, a la arquitectura de red seleccionada se le integró el servidor DHCP y el servicio web, que son necesarios para la autenticación de usuarios y para el funcionamiento adecuado de la herramienta NoCat.

La arquitectura resultante después de integrar los servicios necesarios se muestra en la figura 2.13.

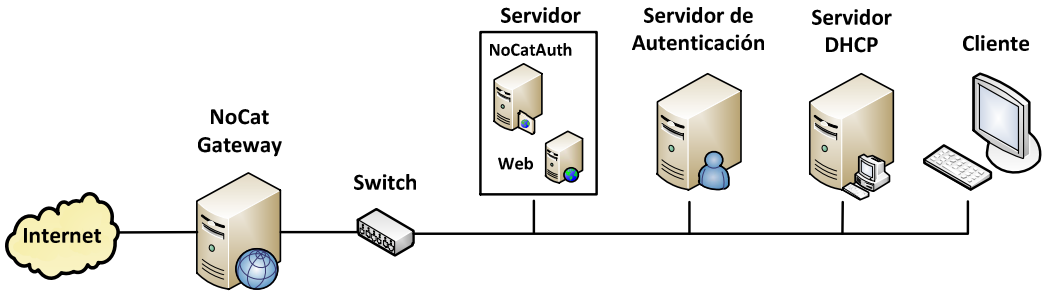


Figura 2.13 Arquitectura usando NoCat + servicios necesarios

Considerando una mejora y ampliación en la infraestructura de red hacia la tecnología inalámbrica, se tomó en cuenta para el diseño de la arquitectura de red a implementar, modificando la arquitectura una vez más, apoyándose en la arquitectura para redes inalámbricas que usa un servidor de autenticación. Generando la arquitectura que se muestra en la figura 2.14.

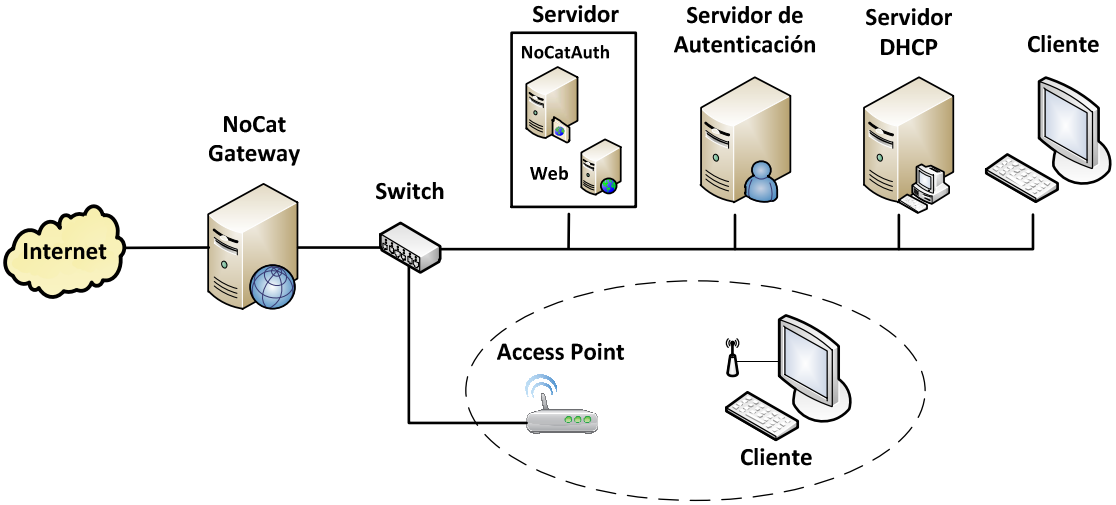


Figura 2.14 Arquitectura resultante (NoCat + servicios necesarios + red WLAN)

En esta arquitectura de red el access point sólo se usa como un bridge (puente), esto es para comunicar a la red alámbrica con la red inalámbrica, y con esto permitir que la red inalámbrica tenga acceso a los servidores pertenecientes a la red Ethernet, lo que facilita administrar los dos tipos de red con un sólo esquema.

El usar el access point como bridge, permite que se cree sólo una red lógica, aunque físicamente sean dos redes distintas, por lo tanto el servidor DHCP es el encargado de asignar direcciones IP y otorgar los datos para la conexión a todos los clientes, sin importar que sean clientes de la red Ethernet o clientes de la red WiFi. Lo que significa que el Access Point no dará servicio de DHCP para los clientes inalámbricos.

Otra ventaja de usar el access point como bridge es que los clientes inalámbricos tienen su propia IP al momento de pasar por el servidor gateway, por lo tanto el control de los clientes WiFi, es personalizado.

Por otra parte, si el access point se usa como router, se genera una red inalámbrica privada, donde el access point es el gateway de dicha red WiFi, por lo que cualquier petición de un cliente WiFi se enmascara con la IP del access point, lo que el servidor NoCat Gateway pensará que son peticiones de un sólo cliente. Por lo anterior, es que se decidió colocar el access point en modo bridge.

El router inalámbrico que se usará es el Linksys WRT54G Ver. 6.

Capítulo III. Implementación

Como se mencionó en el capítulo I, un portal cautivo es una herramienta, ya sea software o hardware, que monitorea el tráfico de una red interna y redirecciona cualquier petición web a una página en específico, ya sea para autenticarse o para informarle sobre las directivas de uso de dicha red.

Y como se explicó en el capítulo II, NoCatAuth es un software desarrollado en perl, para ser instalado en un sistema operativo Linux, el cual es una implementación de un portal cautivo usando "catch and release", este sistema permite desplegar una ventana web a los usuarios de la red, también soporta diversos métodos de autenticación.

Pre-requisitos de hardware

Se requieren mínimo 2 equipos (servidor gateway y gervidor de autenticación) con las siguientes características:

- Procesador 500MHz o superior
- Mínimo 256MB de RAM
- 300 MB de espacio en disco.
- 3 tarjetas de red

En entornos de producción:

- Procesador 500MHz o superior
- Mínimo 1 GB de RAM
- Mínimo 2 GB de espacio en disco
- 3 tarjetas de red

Pre-requisitos de Software

1. Servidor Web Apache - SSL
2. Servidor DHCP
3. Servidor de Autenticación:
    I. Servidor de Base de Datos
    II. Servidor Radius
    III. Servidor LDAP
4. Soporte del módulo "iptables"
5. Soporte para perl
6. Soporte GnuPG
7. Soporte DBI
8. Sistema Linux Fedora 13

Equipos de prueba (Testbed)

Software
Fedora release 13 – Goddard[3]
Compilador gcc versión 4.4.4
Perl versión 5.10.1
GnuPG versión 2.0.14
iptables versión 1.4.7

Hardware
Procesador Intel Pentium 4 a 3.20GHz
Memoria RAM de 1 GB
Disco duro de 80 GB
Tarjeta de red 10/100 MB

[3] Se usó este sistema operativo por el dominio que se tiene sobre el mismo

Paquetes requeridos

Dhcpd	mysqld
httpd	openssl
NoCatAuth-nightly	perl-CPAN
freeradius-server	GnuPG
	DBI

III.I Instalación de servidor gateway

Equipo 1 – Servidor NoCat gateway

Este equipo tendrá la función de ser el gateway de la red a administrar, este equipo debe de contar con 2 interfaces de red, una conectada a la red interna y otra conectada a la red con salida a internet.

Configuración de tarjetas de red

Se debe tener en cuenta que este equipo tiene 2 tarjetas de red, primero se configurará la tarjeta asociada a la red interna.

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth1
```

```
# 3Com Corporation 3c905C-TX/TX-M [Tornado]
DEVICE=eth1
ONBOOT=yes
IPADDR=192.168.5.1
NETMASK=255.255.255.0
NETWORK=192.168.5.0
```

Posteriormente se configura la tarjeta de red con acceso a internet

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
#Broadcom Corporation NetXtreme BCM5755 Gigabit Ethernet PCI Express
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=132.248.59.24
NETMASK=255.255.255.0
NETWORK=132.248.59.0
GATEWAY=132.248.59.254
DNS1=132.248.10.2
DNS2=132.248.204.1
```

Ahora se detiene el servicio de NetworkManager y se inicia el servicio de network, lo anterior es porque el NetworkManager es usado por el modo gráfico.

```
# /etc/init.d/NetworkManager stop
Stopping NetworkManager daemon:     [ OK ]

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

```
Bringing up loopback interface:     [ OK ]
Bringing up interface eth0:         [ OK ]
Bringing up interface eth1:         [ OK ]
```

Ahora se modifica el nivel de arranque de inicio del servidor

```
# vi /etc/inittab
```

```
# inittab is only used by upstart for the default runlevel.
…
id:3:initdefault:
```

Se debe indicar que el nivel de arranque por default será el 3, multiusuario sin modo gráfico.
Ahora se modifican las ligas correspondientes para iniciar el servicio de red de forma automática.

```
# cd /etc/rc3.d/
# ls | grep network
```

Se muestran las ligas asociadas al servicio network y NetworkManager, ahora se deben cambiar los nombres de las ligas para que el servicio que se inicie sea network (S) y el servicio que no se inicie sea NetworkManager (K)

```
# mv K90network S23network
# mv S23NetworkManager K90NetworkManager
```

Por otra parte es importante quitar las reglas del firewall que trae por default fedora o desactivarlo, para evitar conflicto con el servidor NoCat Gateway, para desactivar el firewall se procede de la siguiente forma

```
# setup
```

Se desplegará una ventana con la siguiente información:

```
Authentication
configuration
Firewall configuration
Keyboard configuration
Network configuration
System services

  Run Tool          Quit
```

Se debe seleccionar la opción "`Firewall confguration`" y seleccionar la opción "`Run Tool`", posterior a esto se muestra otra ventana, con la siguiente información:

```
A firewall protects against unauthorized
…
Firewall:   [ ] Enabled

      OK      Customize     Cancel
```

Se debe de quitar la marca de la opción "Enabled" para que el firewall se desactive y posteriormente se selecciona la opción "OK", posterior a esto aparece la siguiente ventana:

```
Clicking the 'Yes' button will override
…
Please remember to check if the services
…
            Yes          No
```

Se selecciona la opción "Yes" y posterior a eso se debe seleccionar la opción "Quit" para salir del modo setup.

Ahora se debe de activar el reenvió de paquetes, de la siguiente manera:

```
# vi /etc/sysctl.conf
```

```
# Kernel sysctl configuration file for Red Hat Linux
…
# Controls IP packet forwarding
net.ipv4.ip_forward = 1
…
```

III.I.I Instalación de NoCat gateway

Esta aplicación permite el re direccionamiento del tráfico en la red, también permite el paso de paquetes si el usuario ya se autenticó, de la misma forma bloquea los paquetes si el usuario no se ha autenticado.

Para instalar esta aplicación se necesita el paquete NoCatAuth-nightly.tgz, para descargar el paquete se hace de la siguiente forma:

Se crea el directorio donde se almacenaran los paquetes a utilizar

```
# cd /root/
# mkdir paquetes
# cd paquetes/
# wget http://nocat.net/downloads/NoCatAuth/NoCatAuth-nightly.tgz
```

Se descomprimen y se extraen los archivos necesarios

```
# tar xvf NoCatAuth-nightly.tgz
```

Donde:
x= para extraer los archivos
f= para indicar que es un archivo al que se le extraerán los demás archivos
v= verbose, para mostrar la lista de archivos que se van procesando

Lo anterior creará una carpeta con el nombre del paquete.

Ahora se procede con la instalación del módulo del gateway, hay que cambiarse al directorio de la aplicación

```
# cd NoCatAuth-nightly/
# ls
```

Se verifica la versión de kernel que está instalado

```
# uname -r
2.6.33.3-85.fc13.i686.PAE
```

Las dos primeras cifras son las que se van a considerar ***2.6***, ahora se tiene que editar el siguiente archivo y cambiar la línea 13

```
# vi bin/detect-fw.sh
```

```
#!/bin/bash
…
  test X"`uname -sr | cut -d. -f-2`" = X"Linux 2.6"; then
…
```

Se debe de sustituir la palabra “Linux 2.4” por “Linux <version kernel>”, en este caso queda como en el cuadro anterior. Lo anterior se hace para generar compatibilidad entre el kernel del sistema operativo y los archivos de instalación.

Ahora se instalarán los paquetes necesarios para el servidor NoCat Gateway

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

Lo anterior ha instalado el servidor NoCat Gateway

III.I.II Configuración de NoCat gateway

Para la configuración del gateway, se hará lo siguiente:

Se cambia el directorio de trabajo y se editar el siguiente archivo

```
# cd /usr/local/nocat/gw/
# vi nocat.conf
```

```
###### gateway.conf -- NoCatAuth Gateway Configuration
…
GatewayName       MiGateway_kokoNetwork
…
GatewayMode       Captive
…
AuthServiceAddr   192.168.5.3
…
ExternalDevice    eth0
…
InternalDevice    eth1
…
LocalNetwork      192.168.5.0/24
…
DNSAddr           132.248.59.98
…
AllowedWebHosts   www.fi-b.unam.mx
…
#IncludePorts     22 80 443
…
#ExcludePorts     25
…
PGPKeyPath        /usr/local/nocat/gw/pgp
…
```

El cuadro anterior muestra los parámetros más importantes que se deben editar, donde:

- GatewayName: Nombre del servidor gateway, o nombre que aparecerá en páginas de bienvenida o de estado.
- GatewayMode: Modo del gateway, existen 3 modos:
    a) Captive – Permite autenticación mediante un servicio externo.
    b) Passive – Similar a Captive, pero se debe usar este modo si el gateway está detrás de un servidor NAT (Network Address Translation).
    c) Open – Sólo se requiere mostrar un mensaje de bienvenida y aceptar un acuerdo de uso, en otras palabras abierto.
- AuthServiceAddr: Dirección IP del servidor que hará la autenticación.
- ExternalDevice: Nombre de la interfaz de red conectada a internet
- InternalDevice: Nombre de la interfaz de red conectada a la red interna.
- LocalNetwork: Dirección de la red interna
- DNSAddr: Dirección del servidor DNS, que les dará servicio a los clientes de la red interna, si no se tiene un servidor DNS interno.
- AllowedWebHosts: Direcciones de sitios web que se pueden navegar sin necesidad de autenticarse.
- IncludePorts: Lista de puertos que se van a permitir cuando el usuario se haya autenticado correctamente.*
- ExcludePorts: Lista de puertos que se van a denegar cuando el usuario se haya autenticado correctamente.*
- PGPKeyPath: Directorio donde se almacenan las llave pgp**

*: Sólo se puede usar una directiva a la vez, si se usa IncludePorts, no se puede utilizar ExcludePorts y viceversa, es similar a política prohibitiva o política permisiva. [18]
Política prohibitiva: Todo está prohibido excepto lo que yo indique (IncludePorts)
Política permisiva: Todo está permitido excepto lo que yo indique (ExcludePorts)
**: Se utilizará posteriormente, cuando se configure el equipo 2, el servidor de autenticación.

Ahora se debe borrar la llave pgp que viene por default en el NoCat, cuando se instale el servidor NoCatAuth se creará la llave que se usará en la comunicación de los servidores.

```
# rm –f /usr/local/nocat/gw/pgp/trustedkeys.gpg
```

Ahora se debe de copiar el script que permitirá iniciar automáticamente el servicio de NoCat como Gateway.

```
# cd /root/paquetes/NoCatAuth-nightly/
# cp etc/nocat.rc /etc/init.d/nocat
```

Ahora se edita dicho archivo y se coloca el directorio donde se instalaron los archivos de NoCat Gateway.

```
# vi /etc/init.d/nocat
```

```
#!/bin/sh
…
NC=/usr/local/nocat/gw
...
```

Ahora se crea la liga para el nivel de arranque por default, en este caso, el nivel 3, dicha liga permitirá iniciar automáticamente el servicio de nocat cuando se encienda el servidor.

```
# cd /etc/rc3.d/
# ln -s ../init.d/nocat S52nocat
```

En los sistemas Linux, existen 7 diferentes niveles de arranque, cada nivel de arranque tiene una función específica y está formado por un conjunto de servicios que se tendrán que iniciar en el momento que se cargue el sistema operativo. En otras palabras, el nivel de arranque indica que servicios tienen que ser iniciados para tener un comportamiento específico.

Para el buen funcionamiento del nocat debemos realizar lo siguiente, verificar que este instalado el manejador de paquetes de perl

```
# yum install -y perl-CPAN
```

Posteriormente se instala el paquete Net::Netmask de perl, se puede realizar de la siguiente manera:

```
# yum install -y perl-Net-Netmask
```

También se debe instalar el paquete Digest::MD5, esto se hará con la ayuda del manejador de paquetes de Perl, CPAN, de la siguiente forma:

```
# perl -MCPAN -e shell
```
Lo anterior ejecuta el manejador de paquetes de perl. Si es la primera vez que se ejecuta nos pedirá cierta información, se dejan los datos por default, hasta que nos proporciones un Shell de CPAN.

```
cpan[1]> install Digest::MD5
…
…
 /usr/bin/make install -- OK

cpan[2]> quit
```

Nota: también se puede instalar mediante el gestor de paquetes de fedora "yum", de la siguiente forma: `# yum install perl-Digest-MD5-File`

Ahora solo falta iniciar el servicio de nocat

```
# /etc/init.d/nocat start
Starting the NoCat gateway...
[2012-02-21 14:04:16] Resetting firewall.
[2012-02-21 14:04:16] Binding listener socket to 0.0.0.0
```

Si todo está configurado de forma correcta el sistema contestará con un mensaje como el anterior.

III.II Instalación del servidor de autenticación

Equipo 2 – Servidor NoCat de Autenticación

Este equipo tendrá la función de validar las credenciales de los usuarios que se autentiquen en la red a administrar para que se les otorgue el permiso para acceder a internet, este equipo debe de contar con una interfaz de red, para que pueda conectarse a la red.

Configuración de tarjeta de red

Para configurar la tarjeta de red se procede de la siguiente manera:

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
# Intel Corporation 82562EZ 10/100 Ethernet Controller
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.5.3
NETMASK=255.255.255.0
NETWORK=192.168.5.0
GATEWAY=192.168.5.1
```

Ahora se para el servicio de NetworkManager e iniciamos el servicio de network, lo anterior es porque el NetworkManager es usado por el modo gráfico.

```
# /etc/init.d/NetworkManager stop
Stopping NetworkManager daemon:     [ OK ]

# /etc/init.d/network start
Bringing up loopback interface:     [ OK ]
Bringing up interface eth0:         [ OK ]
Bringing up interface eth1:         [ OK ]
```

Se modifica el nivel de arranque de inicio del servidor

```
# vi /etc/inittab
```

```
# inittab is only used by upstart for the default runlevel.
…
id:3:initdefault:
```

Se debe indicar que el nivel de arranque por default será el 3, multiusuario sin modo gráfico.

Se modifican las ligas correspondientes para iniciar el servicio de red de forma automática.

```
# cd /etc/rc3.d/
# ls | grep etwork
```

Se muestran las ligas asociadas al servicio network y NetworkManager, ahora se deben cambiar los nombres de las ligas para que el servicio que se inicie sea network (S) y el servicio que no se inicie sea NetworkManager (K)

```
# mv K90network S23network
# mv S23NetworkManager K90NetworkManager
```

Instalación de dhcpd

El servicio de dhcp sirve para otorgar direcciones IP a los clientes de que se conecten a la red en cuestión.

Para instalar el servicio se debe de teclear el comando (con usuario root):

```
# yum install –y dhcp
```

Posteriormente de que se instalen los paquetes necesarios se debe configurar el servicio:

```
# vi /etc/dhcp/dhcpd.conf
```

```
# dhcpd.conf
# Sample configuration file for ISC dhcpd
#

default-lease-time 600;
max-lease-time 7200;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

subnet 192.168.5.0 netmask 255.255.255.0 {
  range 192.168.5.10 192.168.5.200;
  option domain-name-servers 132.248.59.98;
  option routers 192.168.5.1;
  option broadcast-address 192.168.5.255;
  default-lease-time 600;
  max-lease-time 7200;
}
```

Se procede a configurar el inicio automático del servicio cuando se prenda la computadora, debemos revisar que exista el script que inicia el servicio en:

```
# ls /etc/init.d/dhcpd
```

Ahora se modifican las ligas en los niveles de arranque deseado, se ponen ligas de inicio en el nivel 5 (modo gráfico multiusuario) y en el nivel 3 (multiusuario sin modo gráfico).

```
# cd /etc/rc3.d/
```

Se busca si existe una liga con el nombre del servicio

```
# ls | grep *dhcpd
```

Si se encuentra una liga, se debe revisar el nombre de dicha liga, sí el nombre empieza con la letra 'K' -> K35dhcpd, se debe de cambiar el nombre a uno que empiece con la letra 'S':

```
# mv K35dhcpd S60dhcpd
```

Esto es para que se inicie el servicio en el nivel de arranque deseado, en este caso nivel 3.
Se procede de la misma forma para verificar el nivel 5.

```
# cd /etc/rc5.d/
# ls | grep *dhcpd
# mv K35dhcpd S60dhcpd
```

Ahora se procede a iniciar el servicio

```
# /etc/init.d/dhcpd start
Starting dhcpd:           [ OK ]
```

Si la configuración del servicio es correcta, el sistema operativo responderá con un mensaje igual al anterior.

****NOTA:** El servicio anterior puede instalarse en cualquier equipo perteneciente a la red

Instalación Servidor Web Apache con SSL

Este servicio permite contener información en la web, páginas html y muchas otras cosas, se usará para informarles a los usuarios las políticas de uso de la red, o para autenticarse.

Se necesitan los siguientes paquetes, httpd-2.2.21.tar.gz y openssl-1.0.0g.tar.gz

Para descargar los paquetes se hace de la siguiente forma:

Se crea el directorio donde se almacenaran los paquetes a utilizar

```
# cd /root/
# mkdir paquetes
# cd paquetes/
```

Se descargan los paquetes necesarios

```
# wget http://www.openssl.org/source/openssl-1.0.0g.tar.gz
# wget http://download.nextag.com/apache//httpd/httpd-2.2.21.tar.gz
```

Se descomprimen y se extraen los archivos necesarios

```
# tar xvf openssl-1.0.0g.tar.gz
# tar xvf httpd-2.2.21.tar.gz
```

Donde:
x= para extraer los archivos
f= para indicar que es un archivo al que se le extraerá su contenido
v= verbose, para mostrar la lista de archivos que se van procesando

Lo anterior creará carpetas con el nombre del paquete correspondiente.

Ahora se procede con la instalación el módulo SSL

```
# cd openssl-1.0.0g/
# ls
# ./config --prefix=/opt/ssl --shared
# make
# make install
# cd ..
```

Donde:
--prefix indica la ruta en donde se instalarán los archivos asociados al servicio.
--shared indica que podrá compartir librerías para plataformas que soporten SSL.

Ahora se hace la instalación del Apache

```
# cd httpd-2.2.21/
# ls
# ./configure --prefix=/opt/apache2 --enable-ssl --with-ssl=/opt/ssl/ --
enable-so
# make
# make install
# cd ..
```

Donde:
--prefix indica la ruta en donde seinstalarán los archivos asociados al servicio.
--enable-ssl indica que tendrá soporte SSL/TLS.
--with-ssl indica la ruta donde se encuentran las herramientas del SSL que vamos a utilizar.
--enable-so indica que tendrá capacidad DSO (Dynamic Shared Object) que puede recibir actualizaciones de configuración posteriormente, incorporación de paquetes .so (Lo anterior es por si se desea ponerle soporte para PHP, etc.).

Generación de Certificados SSL para servidor Web

Se debe abrir el directorio donde se crearán los certificados con openssl

```
# cd /opt/apache2/conf/
```

Generar los certificados correspondientes

```
# openssl genrsa -des3 -out server.key 1024
# openssl rsa -in server.key -out server.pem
# openssl req -new -key server.key -out server.csr
# openssl x509 -in server.csr -signkey server.key -out server.crt
# openssl x509 -req -in server.csr -signkey server.key -out server.crt
```

III.II.I Instalación de NoCatAuth

Esta aplicación permite la autenticación de usuarios y le notifica al servidor NoCat Gateway si el usuario es válido o no.

Para instalar esta aplicación necesitamos el paquete NoCatAuth-nightly.tgz, para descargar el paquete se hace de la siguiente forma:

Crear el directorio donde se almacenaran los paquetes a utilizar

```
# cd /root/
# mkdir paquetes
# cd paquetes/
# wget http://nocat.net/downloads/NoCatAuth/NoCatAuth-nightly.tgz
```

Se descomprimen y se extraen los archivos necesarios

```
# tar xvf NoCatAuth-nightly.tgz
```

Donde:
x= para extraer los archivos
f= para indicar que es un archivo al que se le extraerá su contenido
v= verbose, para mostrar la lista de archivos que se van procesando

Lo anterior creará una carpeta con el nombre del paquete.

Ahora se procede con la instalación del módulo de autenticación, hay que cambiarse al directorio de la aplicación

```
# cd NoCatAuth-nightly/
# ls
```

Se verifica la versión de kernel que está instalado

```
# uname -r
2.6.33.3-85.fc13.i686.PAE
```

Las dos primeras cifras son las que se van a considerar ***2.6***, ahora se tiene que editar el siguiente archivo y cambiar la línea 13

```
# vi bin/detect-fw.sh
```

```
#!/bin/bash
…
  test X"`uname -sr | cut -d. -f-2`" = X"Linux 2.6"; then
…
```

Se debe de sustituir la palabra “Linux 2.4” por “Linux <version kernel>”, en este caso queda como en el cuadro anterior. Lo anterior se hace para generar compatibilidad entre el kernel del sistema operativo y los archivos de instalación.

Ahora se instalarán los paquetes necesarios para el servidor NoCat de autenticación

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

Posterior a esto se debe crear la llave ppg que comunicará a los dos servidores, para comenzar con la creación de esta llave primero hay que borrar la llave que viene creada por default:

```
# rm –f /usr/local/nocat/authserv/pgp/trustedkeys.gpg
```

Ahora se procede a crear la llave y los archivos necesarios para la comunicación

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

Aparece lo siguiente:

```
Looking for gpg...
[ -d /usr/local/nocat/authserv/pgp ] || mkdir
/usr/local/nocat/authserv/pgp
chmod 700 /usr/local/nocat/authserv/pgp
gpg --homedir=/usr/local/nocat/authserv/pgp --gen-key
gpg (GnuPG) 2.0.14; Copyright (C) 2009 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: keyring `/usr/local/nocat/authserv/pgp/secring.gpg' created
gpg: keyring `/usr/local/nocat/authserv/pgp/pubring.gpg' created
Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection?
```

Aparecen diversos métodos para crear la llave, se selecciona “DSA and Elgamal”

```
Your selection? 2
DSA keys may be between 1024 and 3072 bits long.
What keysize do you want? (2048)
```

Se indica el tamaño de la llave

```
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
```

Se indica el periodo de validez de la llave

```
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name:
```

Se piden ciertos datos para generar la llave

```
GnuPG needs to construct a user ID to identify your key.

Real name: Jorge Hernandez Lopez
Email address: root@localhost
Comment: servidor nocat auth
You selected this USER-ID:
    "Jorge Hernandez Lopez (servidor nocat auth) <root@localhost>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
```

Se muestra la identidad del usuario y pregunta que se desea hacer, si se quiere cambiar cierta información se puede hacer tecleando la letra deseada, si los datos son correctos, se coloca la letra "O" y posteriormente <ENTER>, se muestra lo siguiente:

```
Enter passphrase

Passphrase _____________________
  <OK>                  <Cancel>
```

IMPORTANTE: No se debe colocar ninguna frase, se debe dejar el espacio en blanco. Esto se debe a que cuando se intenten comunicar los servidores se les pedirá la frase que indicada, el servicio nocat no conoce esta frase, lo que provocará que haya un error en la comunicación entre éstos, para evitar este error, simplemente dejamos la frase en blanco.

Por otra parte esta frase sólo sirve para generar caracteres que se usaran de forma aleatoria al momento de crear la llave.

```
Warning: You have entered an insecure passphrase.
A passphrase should be at least 8 characters long.
<Take this one anyway>       <Enter new passphrase>
```

Se muestra un mensaje que indica que la passphrase es insegura, se debe seleccionar la opción "Take this one anyway", que indica que use la passphrase vacía.

```
Warning: You have entered an insecure passphrase.
A passphrase should contain at least 1 digit or
special character.
<Take this one anyway>       <Enter new passphrase>
```

Se muestra otro mensaje indicando que la passphrase es insegura, y que debe tener al menos 1 caracter, se selecciona la opción "Take this one anyway", que indica que use la passphrase vacía.

Ahora solicita que se escriba nuevamente la frase anterior:

```
Please re-enter this passphrase

Passphrase _____________________
 <OK>                    <Cancel>
```

Se selecciona la opción “OK” y se deja nuevamente la passphrase vacía.

Se muestra el siguiente mensaje:

```
You don't want a passphrase - this is probably a *bad* idea!

I will do it anyway.  You can change your passphrase at any time,
using this program with the option "--edit-key".

We need to generate a lot of random bytes. It is a good idea to
perform some other action (type on the keyboard, move the mouse,
utilize the disks) during the prime generation; this gives the random
number generator a better chance to gain enough entropy.
```

El mensaje indica que no usar una passphrase probablemente sea una mala idea, pero la llave se generará de todas formas, también indica que se deben generar bytes de forma aleatoria y que para ayudar a generar éstos hay que mover el mouse, usar el disco duro o usar el teclado, cuando se termina de generar la llave, se muestra el siguiente mensaje:

```
gpg: /usr/local/nocat/authserv/pgp/trustdb.gpg: trustdb created
gpg: key BF4D7828 marked as ultimately trusted public and secret key
created and signed.
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u pub
2048R/BF4D7828 2012-02-22
Key fingerprint = FEC2 1640 7DEB F422 AD3A 49EC BDE4 BF26 BF4D 7828

uid Jorge Hernandez Lopez(servidor nocat auth) <root@localhost>
sub   2048R/2A35B12D 2012-02-22

cp -R /usr/local/nocat/authserv/pgp/pubring.gpg
/usr/local/nocat/authserv/trustedkeys.gpg

Be sure to make your /usr/local/nocat/authserv/pgp directory readable
*only* by the user your httpd runs as.

The public key ring you'll need to distribute can be found in
/usr/local/nocat/authserv/trustedkeys.gpg.
```

Esto indica que las llaves se han generado e indica que se deben de cambiar los permisos de la carpeta que contiene las llaves a sólo lectura para el usuario que controla el servicio httpd (Apache Web).

También indica que la llave pública que se tiene que compartir, se encuentra en el directorio:

```
/usr/local/nocat/authserv/trustedkeys.gpg
```

Ahora se debe copiar dicha llave al directorio donde se encontraba la llave por default de nocatAuth

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

Posterior a eso se cambian los permisos a la carpeta que contiene las llaves, para esto se tiene que revisar el archivo de configuración del servicio apache que se instaló anteriormente, este archivo se utilizará para identificar a que usuario tenemos que darle permisos:

```
# cd /opt/apache2/conf/
# vi httpd.conf
```

```
#
# This is the main Apache HTTP server configuration file.  It contains
…
# User/Group: The name (or #number) of the user/group to run httpd as.
…
User daemon
Group daemon
…
```

Se busca la sección de "User/Group" para saber cuál es el usuario que se encarga del servicio httpd y así poder asignar los permisos requeridos a la carpeta que contiene la llave pgp generada anteriormente.

```
# cd /usr/local/nocat/authserv/
# chown –R daemon:daemon pgp/
# ls –l pgp/
total 24
-rw-------. 1 daemon daemon  905 Feb 13 14:00 pubring.gpg
-rw-------. 1 daemon daemon  905 Feb 13 14:00 pubring.gpg~
-rw-------. 1 daemon daemon  600 Feb 21 16:31 random_seed
-rw-------. 1 daemon daemon  966 Feb 13 14:00 secring.gpg
-rw-------. 1 daemon daemon 1280 Feb 13 14:00 trustdb.gpg
-rw-------. 1 daemon daemon  905 Feb 13 14:01 trustedkeys.gpg
```

Lo anterior cambia de propietario a la carpeta y a los archivos que ésta contiene, si se hizo de forma correcta, el listado de archivos muestra que el usuario "daemon" es el nuevo propietario de la carpeta pgp y las llaves generadas.

Ahora se tiene que compartir la llave pública con el servidor NoCat Gateway para que puedan establecer comunicación. Para hacer esto se procede de la siguiente forma:

```
# cd /usr/local/nocat/authserv/
# scp pgp/trustedkeys.gpg root@192.168.5.1:/usr/local/nocat/gw/pgp/
```

Lo anterior permite copiar la llave de forma remota, es decir se va a copiar el archivo de un servidor a otro con la ayuda del comando "scp".

La sintaxis del comando anterior es:

```
# scp <archivo_origen> usuario@IP_destino:<ruta_destino>
```

Lo que significa que se copia la llave en el directorio /usr/local/nocat/gw/pgp/ del servidor NoCat Gateway, ese directorio es el que se indica en el archivo de configuración nocat.conf del servidor gateway.

Para el buen funcionamiento del nocat se debe realizar lo siguiente:

Verificar que este instalado el manejador de paquetes de perl
```
# yum install -y perl-CPAN
```

Posteriormente se instala el paquete Net::Netmask de perl, se puede realizar de la siguiente manera:
```
# yum install -y perl-Net-Netmask
```

También se debe instalar el paquete Digest::MD5, esto se hará con la ayuda del manejador de paquetes de Perl, CPAN, de la siguiente forma:

```
# perl -MCPAN -e shell
```

Lo anterior ejecuta el manejador de paquetes de perl. Si es la primera vez que se ejecuta nos pedirá cierta información, se dejan los datos por default, hasta que nos proporciones un Shell de CPAN.

```
cpan[1]> install Digest::MD5
…
…
 /usr/bin/make install -- OK

cpan[2]> quit
```

Nota: también se puede instalar mediante el gestor de paquetes de fedora “yum”, de la siguiente forma:

```
# yum install perl-Digest-MD5-File
```

III.II.II Instalación y configuración de servicios de autenticación

El servicio NoCatAuth aparte de traer su propio método de autenticación, tiene la capacidad de conectarse con servicios de autenticación externos, como lo es NIS, SAMBA, LDAP, RADIUS, base de datos, etc.

En este trabajo sólo se utilizarán los servicios de autenticación:

a) Base de datos
b) RADIUS
c) LDAP

III.II.II.I Instalación de un servidor de base de datos

Para instalar el servidor de base de datos, se hace de la siguiente forma:

```
# yum –y install mysql-server
```

Al instalar de la forma anterior el servidor de MySQL se instalan dependencias que se requieren para comunicar al servidor NocatAuth con MySQL, como perl-DBI y perl-DBD-MySQL.

Ahora se inicia el servicio de mysql

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

```
Initializing MySQL database:  Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h portalcautivo password 'new-password'

Alternatively you can run:
/usr/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default.  This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr ; /usr/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl
cd /usr/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/bin/mysqlbug script!

                                                           [  OK  ]
Starting mysqld:                                           [  OK  ]
```

El mensaje anterior indica que se debe crear una contraseña para el administrador de la base de datos, el usuario por default para esta función es el usuario "`root`".
Por otra parte indica la forma como se debe levantar el servicio, y notifica que el servicio se ha iniciado correctamente.

Posterior a esto se asigna la contraseña al administrador del servidor MySQL

```
# mysqladmin –u root password <contraseña>
```

Ahora se crea la base de datos

```
# mysqladmin –u root create nocat –p
Enter password:
```

Lo anterior pide la contraseña del administrador de MySQL, se teclea la contraseña que se le asigno a éste.

Ahora se tiene que crear toda la estructura de la base de datos, para realizar esto se hace lo siguiente:

```
# cd /root/paquetes/NoCatAuth-nightly/
# vi etc/nocat.schema
```

```
CREATE TABLE node (
  …
  rango tinyint(3) unsigned default NULL,
  …
);
```

En el archivo nocat.schema se debe de cambiar el nombre de campo "range" de la tabla "node" para que no marque error cuando se cree la estructura de la base de datos, esto se debe a que la palabra "range" es una palabra reservada para MySQL.

Ahora se crea la estructura de la base de datos

```
# mysql -u root nocat < etc/nocat.schema -p
```

Lo anterior genera la estructura necesaria para compartir información entre el servidor NoCatAuth y la base de datos.

Se debe crear un usuario en la base de datos que se encargue del manejo de los datos entre el servidor NocatAuth y la misma base de datos

```
# mysql –u root –p
Enter password:
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
…
mysql>
```

```
mysql> grant all on nocat.* to nocat@localhost identified by 'secreto';
```

Con la instrucción anterior se crea el usuario que se conectará a la base de datos, donde
grant all: indica que asigne todo los permisos posibles
on <base_datos>.<tabla>: indica el nombre de la base de datos y las tablas a usar, el metacaracter (*) significa todas.
to <user>@<host>: indica al usuario y el host que tendrán los permisos.
Identified by '<password>': indica la contraseña que tendrá el usuario que se indica.

Nota: este usuario y contraseña se usarán en el momento de configurar el servicio de NoCatAuth.
Para comprobar que el usuario, la contraseña y la estructura son correctas se hace lo siguiente:

```
# mysql –u nocat –p
Enter password:
```
Ahora se teclea la contraseña que se le asignó al usuario en el paso anterior.

```
mysql> use nocat;
mysql> show tables;
```

Se debe mostrar lo siguiente:

```
+-----------------+
| Tables_in_nocat |
+-----------------+
| eventlog        |
| hardware        |
| member          |
| network         |
| node            |
+-----------------+
5 rows in set (0.00 sec)

mysql> quit
```

III.II.II.I Configuración de NoCatAuth usando una base de datos

Para la configuración del servidor NoCatAuth utilizando una base de datos se hará lo siguiente:
Se cambia el directorio de trabajo y se edita el siguiente archivo

```
# cd /usr/local/nocat/authserv/
# vi nocat.conf
```

```
###### authserv.conf -- NoCatAuth Authentication Service Configuration
…
PGPKeyPath        /usr/local/nocat/authserv/pgp
…
DataSource        DBI
…
Database          dbi:mysql:database=nocat
DB_User           nocat
DB_Passwd         <secreto>
…
UserTable         member
UserIDField       login
UserPasswdField   pass
UserAuthField     status
UserStampField    created

GroupTable        network
GroupIDField      network
GroupAdminField   admin
…
LocalGateway      192.168.5.1
…
```

El cuadro anterior muestra los parámetros más importantes que se deben editar, donde:

- PGPKeyPath: Directorio donde se almacenan las llave pgp, que se crearon posterior a la instalación de NoCatAuth.
- DataSource: Indica con que servicio va a autenticar, los valores posibles son: DBI, Passwd, LDAP, RADIUS, PAM, Samba, IMAP, NIS. Es importante verificar que los parámetros asociados a otros servicios estén comentados y sólo dejar los parámetros que son utilizados por el servicio que se selecciona en este punto.
- LocalGateway: Indica la dirección IP del servidor NoCat Gateway.

Los siguientes parámetros sólo se usan para autenticación con Base de Datos.

- Database: Indica el conector a usar y el nombre de la base de datos
- DB_User: Indica el nombre de usuario para la base de datos, este usuario se creó en pasos anteriores.
- DB_Passwd: Indica la contraseña que se le asignó al usuario anterior.
- UserTable: Indica el nombre de la tabla de la base de datos que contiene los registros de los usuarios.
- UserIDField: Indica el nombre de la columna que contiene el ID de cada usuario.
- UserPasswdField: Indica el nombre de la columna que contiene las contraseñas de los usuarios, estás contraseñas están encriptadas por MD5.
- UserAuthField: Indica el nombre del campo que contiene el estado del usuario. Este campo ya no se usa.
- UserStampField: Indica el nombre del campo que contiene la fecha de creación de usuarios.
- GroupTable: Indica el nombre de la tabla que contiene los registros sobre la red del usuario.
- GroupIDField: Indica el nombre del campo que contiene la información de la red.
- GroupAdminField: Indica el nombre del campo que indica si un usuario es administrador.

III.II.II.III Instalación de un servidor RADIUS

El protocolo RADIUS (Remote Authentication Dial In User Service) permite el manejo de forma centralizada de servicios de Autenticación, Autorización y de Contabilización, con lo cual se cubren las necesidades de la red inalámbrica a configurar, ya que permite manejar de diversas formas la autenticación vía usuario-contraseña o la autenticación por dirección MAC. Sin embargo, estas formas de acceso a la red son excluyentes entre sí y para poder utilizar ambas es necesario modificar ligeramente el código fuente generado al instalar el servidor RADIUS; además de realizar la configuración respectiva de cada tipo de autenticación.

Proceso Básico de autenticación por medio de RADIUS

Existen 3 actores generales en este proceso: El *servidor de autenticación*, que se encarga de recibir las credenciales del usuario y decide si le permite el acceso a la red o no; el dispositivo *autenticador*, que puede ser un access point o un Servidor de Acceso a la Red (NAS, por sus siglas en inglés), el cual se encarga de recibir las solicitudes de usuario en un formato correspondiente a un método determinado de autenticación de red y traducirlas al formato correspondiente al protocolo RADIUS. El tercer actor se conoce como suplicante (o usuario) y es el dispositivo que intenta conectarse a la red ofrecida por el access point. De forma simple, este proceso se puede ver como si el equipo suplicante solicitara conectarse a la red que proporciona el access point, y éste a su vez le preguntara al servidor radius si el usuario en cuestión tiene permisos de conexión, si es así, el autenticador permite que el suplicante se conecte a la red.
Es importante mencionar que en una "pre-configuración" del autenticador y el servidor radius se debe definir una clave secreta, conocida como *Shared Secret*, que van a compartir los 2 dispositivos para identificarse mutuamente en el proceso de autenticación de usuarios. Dicha clave secreta no es conocida por los dispositivos suplicantes ya que es asignada por quien administra el servidor radius y el dispositivo autenticador.
Como elemento extra se necesita un servidor DHCP para asignar parámetros de conexión adecuados al dispositivo suplicante.

Requerimientos de Software

- Librería OpenSSL
- Librería eap-ikve2
- Paquete freeradius-server-2.1.10.tar.gz

Instalación de RADIUS

El software elegido para instalar el servidor RADIUS es el que proporciona *FreeRADIUS Project*, el cual es un producto de código abierto bajo licencia GPL de GNU. Este servidor se puede obtener del sitio http://freeradius.org/ en la sección *Download* en formatos .tar.gz o .tar.bz2, ambos son códigos compilables en formato distinto de compresión. En este caso se eligió el formato tar.gz, y la versión a instalar del software es la 2.1.10. [16]

Antes de instalar el servidor radius hay que asegurarse de que el sistema operativo tiene soporte para OpenSSL y para eap-ikev2.

Para instalar las librerías de OpenSSL se hace con la siguiente instrucción:

```
# yum install openssl-devel
```

Posteriormente se instalan las librerías de eap-ikve2, para instalar dichas librerías se requiere del siguiente paquete libeap-ikev2-0.2.1.tar.gz

Para descargar el paquete se procede de la siguiente manera:

```
# cd /root/paquetes/
# wget http://sourceforge.net/projects/eap-ikev2/files/eap-
ikev2/0.2.1/libeap-ikev2-0.2.1.tar.gz/download
```

Se debe descomprimir dicho paquete con la siguiente instrucción:

```
# tar xvf libeap-ikev2-0.2.1.tar.gz
```

Se crea la carpeta libeap-ikev2-0.2.1, ahora se inicia el proceso de configuración para la instalación con el script *configure* localizado en dicha carpeta, esta configuración se hace de la siguiente manera:

```
# cd libeap-ikev2-0.2.1/
# ./configure
```

Posteriormente se compila el código fuente de acuerdo a la configuración indicada, y posterior a eso se procede a instalar dicha librería, esto se realiza de la siguiente forma:

```
# make
# make install
# cd ..
```

Una vez instaladas las librerías mencionadas se procede a instalar el servidor RADIUS.
Para descargar el paquete de freeradius se procede de la siguiente manera:

```
# cd /root/paquetes/
# wget ftp://ftp.freeradius.org/pub/freeradius/freeradius-server-
2.1.12.tar.gz
```

Ahora se debe descomprimir dicho archivo

```
# tar xvf freeradius-server-2.1.10.tar.gz
```

Se crea la carpeta *freeradius-server-2.1.10*, dicha carpeta tiene los scripts necesarios para la configuración, compilación e instalación del paquete. Para hacer todo el proceso se hace de la siguiente forma:

```
# cd freeradius-server-2.1.10
# ./configure
# make
# make install
```

Al hacer lo anterior los scripts de radius se instalan en el directorio /usr/local/bin y las librerías se almacenan en /usr/local/lib. Otro de los directorios que se crean es /usr/local/etc/raddb en el cual residen los archivos de configuración.

Una vez instalado el servidor, es necesario correr el servidor con la instrucción radiusd en modo de debug con la opción –X:

```
# radiusd –X
```

Al ejecutarse el servidor por primera vez se generan certificados de conexión, éstos certificados son necesarios para el caso en que la autenticación se quiera hacer de esa forma. Con la opción –X se imprime en pantalla mensajes para conocer el estado de creación de dichos certificados.
De forma similar cuando se termina el proceso mencionado anteriormente se muestra en pantalla (si no hay problemas de instalación, configuración y con la creación de certificados) una leyenda que indica que el servidor está corriendo y en espera de solicitudes de autenticación.

Una vez que el servidor está a la espera de solicitudes y en modo de debug, se recomienda desde otra terminal revisar si está identificando solicitudes de conexión. Para esto se puede teclear el comando *radtest* de la siguiente forma:

```
# radtest test test localhost 0 testing123
```
El comando anterior hace una prueba al servidor radius para conocer si es capaz de reconocer solicitudes, en primera instancia, y después para probar si permite la conexión con las credenciales de autenticación indicadas.

Para el ejemplo indicado se usa a un usuario “test” con contraseña “test”, conectándose al servidor radius localizado en la máquina identificada como “localhost”, indicando un número de puerto del NAS (Network Access Server) de “0” e indicando que la clave secreta de conexión del servidor radius es “testing123”. Este usuario, su contraseña y la clave secreta se encuentran configurados por **default** en uno de los archivos de configuración instalados.

Como resultado de la ejecución de la línea anterior, en la terminal donde se ejecuta el servidor radius se deben mostrar las solicitudes de conexión por parte de radtest con los datos señalados, sin importar sí valida que los datos sean correctos o no, por el momento no es importante la autenticación exitosa, ya que lo importante es saber si el servidor registra y contesta dichas peticiones; por otra parte en la terminal de ejecución del radtest, se muestra si las solicitudes fueron escuchadas y si fueron denegadas o aceptadas.

Ahora se debe configurar el servidor radius modificando algunos archivos localizados en el directorio */usr/local/etc/raddb*. El archivo de configuración principal es *radiusd.conf*, desde el que se incluyen los demás archivos de configuración. Aquí se definen directorios que usa radius para la configuración y para los archivos log, pero para nuestros propósitos no es necesario modificar alguna entrada en este archivo.

Archivo clients.conf

Este archivo se utiliza para indicar los clientes a los que atenderá el servidor radius. El termino *cliente* se puede aplicar a algún dispositivo desde el cual se quiera acceder a la red inalámbrica o incluso a un grupo de dispositivos configurados con algunos parámetros en común.
El registro para cada cliente tiene la siguiente forma:

```
client <nombre_corto> {
    <atributo> = <valor>
}
```

Donde:

- <nombre_corto> se refiere a un nombre pequeño para identificar al cliente (ya sea un dispositivo o varios). Se utiliza para referir brevemente al cliente sin escribir la dirección IP o el nombre completo del hostname (*Fully Qualified Domain Name*).

La sección *client* puede contener varios atributos cuyo valor se asigna mediante el símbolo "=".
Originalmente este archivo tiene una entrada para el cliente identificado como *localhost*. Para ver esto se hace lo siguiente:

```
# vi /usr/local/etc/raddb/clients.conf
```

```
## clients.conf -- client configuration directives
…
client localhost {
    ipaddr = 127.0.0.1
    secret = testing123
    require_message_authenticator = no
    nastype = other
}
…
```

Los datos anteriores indican las siguientes características:

- Ipaddr: Se refiere a la dirección IP que tiene el cliente.
- Secret: Indica la clave secreta que comparte el servidor RADIUS.
- require_message_authenticator: Indica si a los clientes se les necesita enviar un paquete con el atributo Message-Authenticator.
- Nastype: Indica el método de acceso a la red que utilizará el NAS; en este caso se iguala al valor "other" porque no se utiliza alguno de las opciones preconfiguradas.

Este archivo tiene otras entradas por default comentadas que sirven como ejemplo para distintas configuraciones que se pueden hacer.

Para dar soporte a la red que se va a configurar, se deben agregar las siguientes líneas al siguiente archivo:

```
# vi /usr/local/etc/raddb/clients.conf
```

```
## clients.conf -- client configuration directives
…
client kokoNetwork{
        ipaddr = 192.168.5.0
        netmask = 24
        secret = kokosecret
        shortname = kokoNetwork
        nastype = other
}
…
```

Donde los atributos significan:

- Client <nombre>: El identificador del cliente será kokoNetwork.
- Ipaddr: Se refiere a todas las máquinas que pertenecen a la red 192.168.5.0.
- Netmask: Se usa el valor 24 que indica que la máscara de red es 255.255.255.0 (24 bits activos).
- Secret: Para este caso la clave secreta compartida del servidor radius es "kokosecret".

Archivo users

Este archivo sirve para indicar los usuarios que reconocerá el servidor radius. Cada entrada del archivo comienza con un nombre de usuario seguida por una lista de elementos a revisar (*check items*), todos ellos indicados en una línea y separados por el carácter "coma" (,).
La segunda línea de cada entrada comienza con un tabulador y contiene una lista de elementos de réplica (*reply items*) con sintaxis:

```
elemento = <valor>
```

Si se desea indicar varios elementos de réplica, cada uno se debe colocar en una línea, la cual debe terminar forzosamente con una carácter coma (,) a excepción de la última línea.

Para cada solicitud entrante se compara el nombre del usuario indicado con la lista de nombres de usuario del archivo *users*, si hay alguna coincidencia se revisan los *check ítems*, si coinciden los valores de la lista con los valores de la solicitud se acepta, en caso contrario se rechaza.

El archivo *users*, originalmente contiene ejemplos de configuración para las entradas de usuario con varias opciones de *check ítems* y *reply ítems*, y al final de toda esa información se pueden incluir las entradas de usuario.

Nota: En la sección 5 de la página de manual del comando *users*, se explican todas las opciones de *check items* y *reply ítems*.
Para entrar a esta página se hace con el siguiente comando:

```
# man 5 users
```

Para el caso particular de esta configuración en el archivo *users* sólo se indicarán los nombres de usuario, y como *check item* único *Cleartext-Password*; y como *reply ítem* único a *Reply-Message*. Así cada entrada de usuario será como el siguiente ejemplo:

```
# vi /usr/local/etc/raddb/users
```

```
# Please read the documentation file ../doc/processing_users_file,
# or 'man 5 users' (after installing the server) for more information.
…
…
userkoko   Cleartext-Password := "secretuser"
           Reply-Message = "**Hola, %{User-Name} -> Bienvenido..!!**"
…
```

Donde:

- <loginUser>: Indica el login que tendrá el usuario en cuestión, en este caso es userkoko.
- Cleartext-Password: Este ítem sirve para indicar cuál será la contraseña del usuario en cuestión, se debe colocar entre comillas (“).
- Reply-Message: Sirve para indicar el mensaje que se le dará al usuario, si se ha autenticado de forma correcta.
- %{User-Name}: Es una variable para acceder al nombre de usuario que está realizando la petición.

III.II.II.II Configuración de NoCatAuth usando un servidor RADIUS

Para la configuración del servidor NoCatAuth utilizando un servidor radius se hará lo siguiente:
Se cambia el directorio de trabajo y se edita el siguiente archivo

```
# cd /usr/local/nocat/authserv/
# vi nocat.conf
```

```
###### authserv.conf -- NoCatAuth Authentication Service Configuration
…
PGPKeyPath        /usr/local/nocat/authserv/pgp
…
DataSource        RADIUS
…
RADIUS_Host       192.168.5.3
RADIUS_Secret     kokosecret
RADIUS_TimeOut    5
…
LocalGateway      192.168.5.1
…
```

El cuadro anterior muestra los parámetros más importantes que se deben editar, donde:

- PGPKeyPath: Directorio donde se almacenan las llave pgp, que se crearon posterior a la instalación de NoCatAuth.
- DataSource: Indica con que servicio va a autenticar, los valores posibles son: DBI, Passwd, LDAP, RADIUS, PAM, Samba, IMAP, NIS. Es importa verificar que los parámetros asociados a

otros servicios estén comentados y sólo dejar los parámetros que son utilizados por el servicio que se selecciona en este punto.
- LocalGateway: Indica la dirección IP del servidor NoCat Gateway.

Los siguientes parámetros sólo se usan para autenticación con Radius.

- RADIUS_Host: Indica el nombre o dirección IP del servidor radius, se puede indicar el puerto, de no hacerlo se usará el puerto por default, también puede indicarse más de un servidor radius, separados por coma (,).
- RADIUS_Secret: Indica la clave secreta que comparte el servidor radius.
- RADIUS_TimeOut: Indica el tiempo de espera para autenticar en el servidor radius, este campo es opcional, si no se coloca se usa el tiempo que este definido por default en Authen::Radius.

Nota: Lo siguiente es muy importante. Para que funcione correctamente la autenticación con el servidor radius, se debe de instalar el paquete Authen::Radius.

Para instalar dicho paquete se hará con la ayuda del manejador de paquetes de Perl, CPAN, de la siguiente forma:

```
# perl -MCPAN -e shell
```

Lo anterior ejecuta el manejador de paquetes de perl. Si es la primera vez que se ejecuta se pedirá cierta información, se dejan los datos por default, hasta que se proporcione un Shell de CPAN.

```
cpan[1]> install Authen::Radius
…
…
Make sure this machine is in your Radius clients file!
Enter hostname[:port] of your Radius server:
```

Una vez que termine de descargar los paquetes se pedirá información acerca del servidor radius, al igual que un usuario registrado para probar la conexión.

```
Make sure this machine is in your Radius clients file!
Enter hostname[:port] of your Radius server: 192.168.5.3
Enter shared-secret of your Radius server: kokosecret
Enter a username to be validated: koko
Enter this user's password: <secretouser>
Using Radius server 192.168.5.3:1812
…
…
  /usr/bin/make install  -- OK

cpan[2]> quit
```

Cuando aparezca el cursor del shell de CPAN significa que ha terminado la instalación y configuración del paquete descargado. Para salir del shell de CPAN sólo se debe de teclear "quit".

Para el correcto funcionamiento de nocat con un servidor RADIUS es necesario editar el siguiente archivo:

```
# vi /usr/local/nocat/authserv/lib/NoCat/Source/RADIUS.pm
```

```
package NoCat::Source::RADIUS;
…
…
$radius->add_attributes(
                      #{ Name => 1, Value => $user->id},
                      #{ Name => 2, Value => $user_pw}
                      { Name => 1, Value => $user->id, Type=>'string'},
                      { Name => 2, Value => $user_pw, Type=>'string'}

};
…
```

III.II.II.V Instalación de un servidor LDAP

Directory Server (DS)

Fedora Directory Server es un servidor LDAP para Linux desarrollado por Red Hat y la comunidad de Fedora, permite un completo sistema de identidades y una plataforma integral para múltiples servicios. Está enfocado principalmente a instituciones y empresas corporativas, cuenta con múltiples características que lo hacen el favorito para implementaciones del mundo real. [15]

Pre-requisitos de Software

1. Servidor Web Apache – para administrar el LDAP vía web
2. Java JRE - para Fedora Directory Console

Paquetes requeridos

Para instalar el Directory Server se necesitan los siguientes paquetes:

cyrus-sasl-gssapi
389-adminutil
389-ds
389-admin-console
389-admin
389-ds-base
389-ds-base-devel
389-console
idm-console-framework
jss

ldapjdk
mod_nss
mozldap
mozldap-tools
net-snmp-libs
perl-Mozilla-LDAP
svrcore

Los archivos rpm necesarios para realizar esta configuración son:

389-admin-1.1.11-0.1.a1.fc13.i686.rpm
389-admin-console-1.1.4-2.fc12.noarch.rpm
389-admin-console-doc-1.1.4-2.fc12.noarch.rpm
389-adminutil-1.1.9-1.fc13.i686.rpm
389-adminutil-devel-1.1.9-1.fc13.i686.rpm
389-console-1.1.3-5.fc13.noarch.rpm
389-ds-1.1.3-5.fc12.noarch.rpm
389-ds-base-1.2.6-0.1.a1.fc13.i686.rpm
389-ds-base-devel-1.2.6-0.1.a1.fc13.i686.rpm
389-ds-console-1.2.0-5.fc12.noarch.rpm
389-ds-console-doc-1.2.0-5.fc12.noarch.rpm
389-dsgw-1.1.5-1.fc13.i686.rpm
cyrus-sasl-devel-2.1.23-11.fc13.i686.rpm
cyrus-sasl-gssapi-2.1.23-11.fc13.i686.rpm
httpd-tools-2.2.15-1.fc13.i686.rpm
idm-console-framework-1.1.3-2.fc12.noarch.rpm
jss-4.2.6-6.fc13.i686.rpm
ldapjdk-4.18-5.fc12.i686.rpm
mod_nss-1.0.8-4.fc13.i686.rpm
mozldap-6.0.5-6.fc12.i686.rpm
mozldap-devel-6.0.5-6.fc12.i686.rpm
mozldap-tools-6.0.5-6.fc12.i686.rpm
net-snmp-libs-5.5-12.fc13.i686.rpm
nss-devel-3.12.6-4.fc13.i686.rpm
nspr-devel-4.8.4-2.fc13.i686.rpm
nss-softokn-devel-3.12.4-17.fc13.i686.rpm
nss-util-devel-3.12.6-1.fc13.i686.rpm
nss_ldap-264-9.fc13.i686.rpm
openldap-devel-2.4.21-4.fc13.i686.rpm
pam-devel-1.1.1-4.fc13.i686.rpm
perl-Mozilla-LDAP-1.5.2-6.fc12.1.i686.rpm
svrcore-4.0.4-5.fc12.i686.rpm
svrcore-devel-4.0.4-5.fc12.i686.rpm

Los paquetes necesarios se pueden obtener en la siguiente dirección URL:

ftp://ftp.fi-b.unam.mx/fedora/linux/releases/13/Everything/i386/os/Packages/

Para descargar los paquetes requeridos de forma autimática se procede de la siguiente forma:

```
# cd /root/paquetes
# mkdir packLDAP
# cd  packLDAP/
# vi download_packs.sh
```

```
#!/bin/bash
# Script para descarga de archivos

for k in [lista de todos los paquetes mencionados anteriormente]
do
wget ftp://ftp.fi-b.unam.mx/fedora/linux/releases/13/Everything/i386/os/Packages/$k
done;
```

Para instalar los paquetes se hace de la siguiente forma:

```
# rpm –ivh archivo.rpm
```

Por la cantidad de archivos que se requieren y la dependencia entre ellos se recomienda realizar un script mediante el cual se haga la instalación de los archivos de manera automatizada, el script puede ser de la siguiente manera:

```
# vi instala_rpm.sh
```

```
#!/bin/bash
# Script para instalar paquetes
for k in `ls *rpm`
do
rpm -ivh $k
done;
```

Este script se debe ejecutar hasta que todo haya quedado instalado y ya no se muestren errores.

```
# sh instala_rpm.sh
```

Posteriormente se debe instalar el compilador c/c++ a partir del siguiente archivo rpm (si no se tiene instalado):

```
# rpm –ivh gcc-c++-4.4.4-2.fc13.i686.rpm
```

Creación de usuario para la administración del servicio LDAP

Ahora se debe crear un usuario que administre el servicio LDAP, esto se hace de la siguiente forma:

```
# vi /etc/passwd
```

```
…
#[USER]:[PASS]:[UID]:[GUID]:[descripción usuario]:[directorio]:[shell]
ldap:x:800:800:usuario ldap fedoraDS:/var/lib/dirsrv:/bin/bash
...
```

```
# vi /etc/shadow
```

```
…
ldap:!!:15210::::::
...
```

```
# vi /etc/group
```

```
…
# [GROUP]:[PASS]:[GUID]:
ldap:x:800:
...
```

Nota: El GUID debe ser el mismo tanto en el passwd como en el group, también hay que cerciorarse que el UID no esté ocupado por otro usuario del sistema.

Configuración de variables para el buen rendimiento del LDAP

Para que nuestro servicio LDAP tenga el mejor rendimiento se debe editar ciertas variables del sistema de la siguiente forma:

```
# echo "* - nofile 2048" >> /etc/security/limits.conf
# echo "net.ipv4.ip_local_port_range = 1024 65000" >> /etc/sysctl.conf
# echo "fs.file-max = 65536" >> /etc/sysctl.conf
# echo "net.ipv4.tcp_keepalive_time = 600" >> /etc/sysctl.conf
# sysctl -q -p /etc/sysctl.conf
```

Configuración de Directory Server

Lo primero que se debe hacer es cambiar los permisos de las capetas `/etc/dirsrv` y `/var/lock/dirsrv/`

Esto se hace de la siguiente manera:

```
# chown –R ldap:ldap /etc/dirsrv/
# chown –R ldap:ldap /var/lock/dirsrv/
```

Se asigna un nombre al equipo con el comando hostname `[nombre de equipo]`

```
# hostname ldap.die.fi.unam
```

Posteriormente se ejecuta el siguiente comando:

```
# setup-ds-admin.pl
```

Se abrirá el manager para la configuración del servicio

```
This program will set up the 389 Directory and Administration Servers.
It is recommended that you have "root" privilege to set up the
software.
…
Would you like to continue with set up? [yes]:
BY SETTING UP AND USING THIS SOFTWARE YOU ARE CONSENTING TO BE BOUND
BY AND ARE BECOMING A PARTY TO THE AGREEMENT FOUND IN THE LICENSE.TXT
FILE.
…
Do you agree to the license terms? [no]: yes
…
NOTICE: System is i686-unknown-linux2.6.33.3-85.fc13.i686.PAE
2 processors)
…
Would you like to continue? [no]: yes
…
Choose a setup type:
      1. Express
      Allows you to quickly set up the servers using the most common
      options and pre-defined defaults. Useful for quick evaluation of
      the products.
      2. Typical
      Allows you to specify common defaults and options.
      3. Custom
      Allows you to specify more advanced options. This is recommended
      for experienced server administrators only.
      To accept the default shown in brackets, press the Enter key.
Choose a setup type [2]: 3
…
```

```
…
Computer name [ldap.die.fi.unam]:
…
System User [nobody]: ldap
System Group [nobody]: ldap
…
Do you want to register this software with an existing configuration
directory server? [no]:
…
Configuration directory server

administrator ID [admin]:
Password: <secretoldap>
Password (confirm): <secretoldap>
…
Administration Domain [die.fi.unam]:
…
Directory server network port [389]:
…
Directory server identifier [ldap]:
…
Suffix [dc=die, dc=fi, dc=unam]:
…

Directory Manager DN [cn=Directory Manager]: cn=admin
Password: <secretoldap>
Password (confirm): <secretoldap>
…
Do you want to install the sample entries? [no]:
…
Type the full path and filename, the word suggest, or the word none
[suggest]:
…
Administration port [9830]: 3890
…
IP address: 192.168.5.3
…
Run Administration Server as [ldap]:
…
Are you ready to set up your servers? [yes]:
...
```

Si se hace todo de manera correcta se mostrará el siguiente mensaje:

```
Creating directory server …
Your new DS instance 'ldap' was successfully created.
Creating the configuration directory server …
Beginning Admin Server creation …
Creating Admin Server files and directories …
Updating adm.conf …
Updating admpw …
Registering admin server with the configuration directory server …
Updating adm.conf with information from configuration directory
server …
Updating the configuration for the httpd engine …
Starting admin server …
The admin server was successfully started.
Admin server was successfully created, configured, and started.
Exiting …
Log file is '/tmp/setupdEknWP.log'
…
```

NOTA: Es importante que el nombre de host que se colocó en la configuración del Directory Server, “Computer name”, pueda ser resuelto por un servidor DNS, ya sea de forma directa o de forma inversa.

Para probar que nuestra instancia de Direcory Server ha sido levantada correctamente, se hace lo siguiente:

```
# 389-console
```

Se abrirá una ventana como la siguiente:

Figura 3.1 Ventana de management console

Se deben ingresar los datos que se piden, si la instancia de directory server está funcionando adecuadamente se podrán observar las siguientes pantallas:

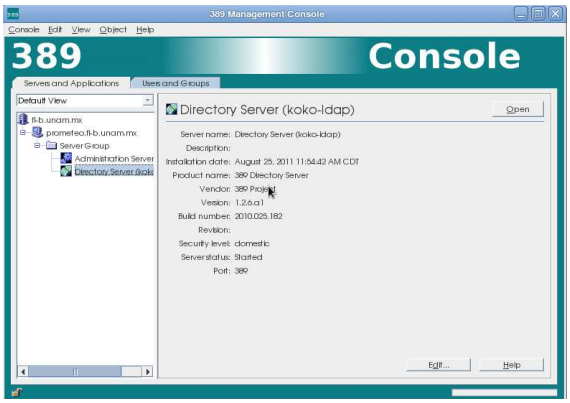


Figura 3.2 Ventana Directory Server en management console

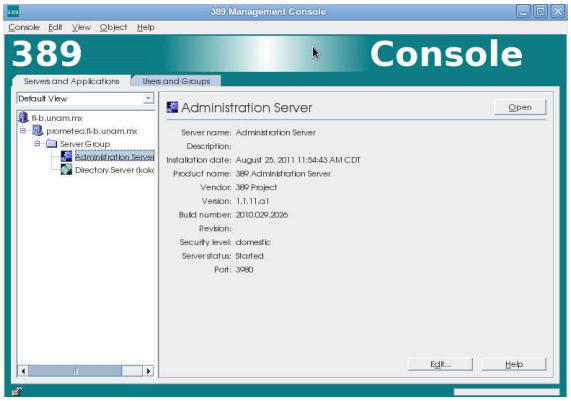


Figura 3.3 Ventana Administration Server en management console

Instalación de OpenLDAP

Paquetes a utilizar

openldap-devel-2.4.21-4.fc13.i686.rpm
cyrus-sasl-devel-2.1.23-11.fc13.i686.rpm

Instalación

rpm –ivh openldap-devel-2.4.21-4.fc13.i686.rpm cyrus-sasl-devel-2.1.23-11.fc13.i686.rpm

Apache Web Server + PHP + phpldapadmin

Requisitos de Software

Servidor Web Apache con soporte para Dynamic Shared Object (DSO). (Véase en: Instalación Servidor Web Apache con SSL

Paquetes a utilizar

phpldapadmin-1.2.1.1.tgz
php-5.4.0.tar.gz
libxml2-devel-2.7.7-1.fc13.i686.rpm

Instalación de complementos

Primero se descarga e instala la libreria de xml de la siguiente forma:

```
# cd /root/paquetes/packLDAP/
# wget ftp://ftp.fi-b.unam.mx/fedora/linux/releases/13/Everything/i386/os/Packages/libxml2-devel-2.7.7-1.fc13.i686.rpm
# rpm –ivh libxml2-devel-2.7.7-1.fc13.i686.rpm
```

Una vez instalado el paquete anterior, se procede a instalar php, esto se hace de la siguiente forma:

```
# cd /root/paquetes/
# wget http://mx.php.net/get/php-5.4.0.tar.gz/from/this/mirror
```

Se descomprimen y se extraen los archivos necesarios

```
# tar xvf php-5.4.0.tar.gz
```

Donde:
x= para extraer los archivos
f= para indicar que es un archivo al que se le extraerá su contenido
v= verbose, para mostrar la lista de archivos que se van procesando

Lo anterior creará una carpeta con el nombre del paquete correspondiente.

Ahora se realiza la instalación de php

```
# cd php-5.4.0/
# ls
# ./configure --with-regex=php --enable-ftp --with-zlib --with-gettext --enable-mbstring --with-ldap --with-ldap-sasl --enable-magic-quotes --enable-exif --with-apxs2=/opt/apache2/bin/apxs --with-libxml-dir=/usr/lib
# make
# make install
```

Donde:
--with-regex indica el tipo de librería regex a usar, en este caso usaremos la de php.
--enable-ftp indica que tendrá soporte para ftp.
--with-zlib indica que tendrá soporte para zlib.
--with-gettext indica que tendrá soporte para GNU gettext
--enable-mbstring indica que tendrá soporte para cadenas multibyte.
--with-ldap indica que tendrá soporte para ldap.
-- with-ldap-sasl indica que tendrá soporte para Cyrus SASL, es complemento de LDAP.

--enable-magic-quotes indica que tendrá soporte de magic-quotes, lo que significa que hace el escape automático de los caracteres (") y (').
--enable-exif indica que tendrá soporte para EXIF (Exchangeable image file format), metadatos de las imágenes.
--with-apxs2 indica la ruta donde está el manejador de módulos, la herramienta apxs del servidor web, y se usa para que el servidor web tenga soporte de php.
--with-libxml-dir indica la ruta donde se instaló la librería libxml2

Una vez que se realizó lo anterior, php ha quedado instalado, ahora se debe hacer lo siguiente:

Copiar el archivo de php.ini del paquete de instalación al directorio donde se instaló php

```
# cp php.ini-production /usr/local/lib/php.ini
```

Lo anterior sí se está dentro de la carpeta `/root/paquetes/php-5.4.0/`

Ahora se debe checar el archivo de configuración del servidor web, el archivo es `httpd.conf`, se verifica que las líneas referentes al módulo de php existan, y se indica que reconozca a `index.php` como archivo de inicio de directorio.

```
# vi /opt/apache2/conf/httpd.conf
```

```
# This is the main Apache HTTP server configuration file.  It contains …
…
…
LoadModule php5_module modules/libphp5.so
…
AddType application/x-httpd-php-source phps
…
AddType text/html .php
…
AddHandler php5-script .php
…
…
<IfModule dir_module>
DirectoryIndex index.html index.php
</IfModule>
…
…
```

Ahora se debe reiniciar el servicio de apache, para que reconozca la nueva configuración, para hacer esto se hace lo siguiente:

```
# /opt/apache2/bin/apachectl restart
```

Para probar que el apache y php funcionan correctamente se crea un archivo de prueba.

```
# cd /opt/apache2/htdocs/
# vi prueba.php
```

```
<?php
      phpinfo();
?>
```

Ahora sólo hay que abrir un navegador web y teclear la siguiente url, http://localhost/prueba.php

Si todo se configuró de manera correcta, se debe de mostrar una lista de características de configuración de php.

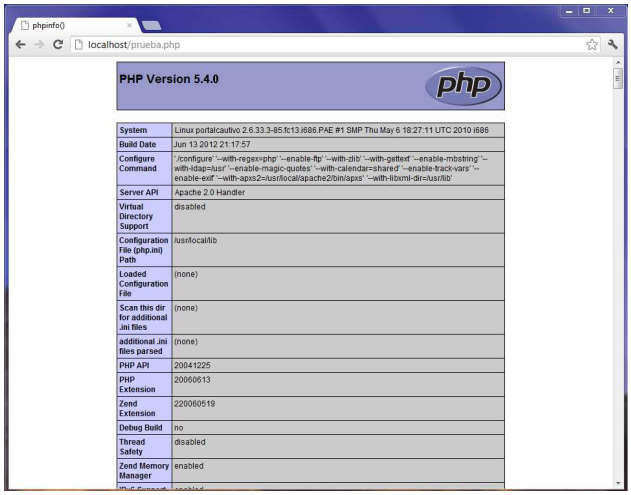


Figura 3.4 Pantalla de información de PHP

Ahora se procede con la instalación del paquete phpldapadmin-1.2.1.1.tgz, dicho paquete servirá para administrar vía web el servicio de ldap.

Primero se debe de obtener el paquete de la siguiente forma:

```
# cd /root/paquetes/packLDAP/
# wget http://sourceforge.net/projects/phpldapadmin/files/phpldapadmin-
php5/1.2.1.1/phpldapadmin-1.2.1.1.tgz/download
```

Se descomprimen y se extraen los archivos necesarios

```
# tar xvf phpldapadmin-1.2.1.1.tgz
```

Se crea una carpeta con el nombre del paquete correspondiente.

Para realizar la instalación de phpldapadmin se hace lo siguiente:

```
# mv phpldapadmin-1.2.1.1 /opt/apache2/htdocs/phpldapadmin
```

Lo anterior mueve toda la carpeta a la ruta indicada, esta ruta pertenece a los documentos del servidor apache.

Ahora se tiene que configurar el phpldapadmin. Se abre la carpeta que contiene los archivos de configuración:

```
# cd /opt/apache2/htdocs/phpldapadmin/config/
```

Para realizar la configuración se usa el archivo de configuración de ejemplo, así que sólo se debe copiar el archivo `config.php.example` y se renombra como `config.php`

```
# cp config.php.example config.php
```

Después de hacer lo anterior el phpldapadmin se puede acceder vía web, para corroborarlo en un navegador web se debe teclear http://localhost/phpldapadmin/ y se debe abrir la interfaz web para la administración.

Figura 3.5 Pantalla de inicio de phpldapadmin

Configuración para autenticación por LDAP

Se deben editar los siguientes archivos:

```
# vi /etc/ldap.conf
```

```
# @(#)$Id: ldap.conf,
…
ssl no
tls_cacertdir /etc/openldap/cacerts
uri ldap://ldap.die.fi.unam
base dc=die,dc=fi,dc=unam
…
```

```
# vi /etc/nsswitch.conf
```

```
# /etc/nsswitch.conf
…
passwd: files ldap
shadow: files ldap
group: files ldap
…
netgroup: files ldap
automount: files ldap
…
```

```
# vi /etc/pam.d/system-auth
```

```
#%PAM-1.0
…
auth sufficient pam_ldap.so use_first_pass
auth requisite pam_succeed_if.so uid >= 500 quiet
account sufficient pam_succeed_if.so uid > 500 quiet
account [default=bad success=ok user_unknown=ignore] pam_ldap.so
password sufficient pam_ldap.so use_authtok
session optional pam_ldap.so
…
```

Posterior a la modificación de los archivos, se debe utilizar phpldapadmin para controlar Fedora Directory Server.

Para agregar la configuración del LDAP se hace lo siguiente:

```
# /usr/lib/mozldap/ldapmodify -D "cn=admin" -w <secretoldap>
dn:
changetype: modify
add: aci
aci: (targetattr = "subschemaSubentry || aliasedObjectName ||
hasSubordinates || objectClasses || namingContexts || matchingRuleUse
|| ldapSchemas || attributeTypes || serverRoot || modifyTimestamp ||
icsAllowRights || matchingRules || creatorsName || dn || ldapSyntaxes
|| createTimestamp")
(version 3.0;
acl "Anonymous access for phpldapadmin";
allow (read,compare,search)
(userdn = "ldap:///anyone")
;)
<<enter>>
<<ctrl-c>>
```

Ahora se debe editar el archivo de configuración de phpldapadmin para que muestre algunos valores por default, se deben agregar las siguientes líneas:

```
# vi /opt/apache2/htdocs/phpldapadmin/config/config.php
```

```
<?php
/** NOTE **/
…
$servers->SetValue('server','name','LDAP Servidor koko');
$servers->SetValue('login','bind_id','cn=admin');
…
?>
```

Ahora se debe modificar el cache de las plantillas (Templates) para que la nueva plantilla (el nuevo usuario a crear) se pueda reflejar en el navegador WEB, para esto se modifica el siguiente archivo:

```
# vi /opt/apache2/htdocs/phpldapadmin/lib/config_default.php
```

```
<?php
// $Header: /cvsroot/phpldapadmin/phpldapadmin/lib/config_default.php
…
$this->default->cache['template'] = array(
'desc'=>'Cache Template configuration',
'default'=>false);          //Se cambia el valor “true” por “false”
…
?>
```

Nota: Se puede activar el modo 'hide_template_warning' para que phpldapadmin no muestre los warnings.

```
<?php
// $Header: /cvsroot/phpldapadmin/phpldapadmin/lib/config_default.php
…
$this->default->appearance['hide_template_warning'] = array(
'desc'=>'Hide template errors from being displayed',
'default'=>true);      //Por default es “false”
…
?>
```

Ahora se debe crear la plantilla del nuevo usuario en el directorio templates, aquí se encuentran las plantillas que trae por default la instalación de phpldapadmin.

```
# cd /opt/apache2/htdocs/phpldapadmin/templates/creation/
# vi custom_die.xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE template SYSTEM "template.dtd">
<template>
      <askcontainer>1</askcontainer>
      <description>Nuevo Usuario Mail DIE</description>
      <icon>ldap-user.png</icon>
      <invalid>0</invalid>
      <rdn>uid</rdn>
      <!--<regexp>^ou=People,o=.*,</regexp>-->
      <title>Usuario DIE</title>
      <visible>1</visible>

<objectClasses>
      <objectClass id="account"></objectClass>
      <objectClass id="posixAccount"></objectClass>
      <objectClass id="top"></objectClass>
</objectClasses>

<attributes>
<attribute id="uid">
      <display>User ID</display>
<onchange>=autoFill(homeDirectory;/users/%gidNumber%/%uid%)</onchange>
      <order>1</order>
      <page>1</page>
      <spacer>1</spacer>
</attribute>
<attribute id="givenName">
      <display>Nombre</display>
      <icon>ldap-uid.png</icon>
      <onchange>=autoFill(cn;%givenName% %sn%)</onchange>
      <onchange>=autoFill(uid;%givenName|0-1/l%%sn/l%)</onchange>
      <order>2</order>
      <page>1</page>
</attribute>
<attribute id="cn">
      <display>Nombre Completo</display>
      <order>3</order>
      <page>1</page>
</attribute>
<attribute id="homeDirectory">
      <display>Home directory</display>
<!-- <onchange>=autoFill(homeDirectory;/users/%gidNumber|0-
0/T%/%uid%)</onchange> -->
      <order>8</order>
      <page>1</page>
</attribute>
<attribute id="uidNumber">
      <display>UID Number</display>
      <icon>terminal.png</icon>
      <order>6</order>
      <page>1</page>
      <readonly>1</readonly>
      <value>=php.GetNextNumber(/;uidNumber)</value>
</attribute>
…
```

```
…
<attribute id="gidNumber">
      <display>GID Number</display>
<onchange>=autoFill(homeDirectory;/users/%gidNumber|0-
0/T%/%uid%)</onchange>
      <order>7</order>
      <page>1</page>
      <value><![CDATA[=php.PickList(/;(&(objectClass=posixGroup));gidN
umber;%cn%;;;;cn)]]></value>
<!--
<hint>100[PTC],101[prebe],102[cursos],103[alum],104[profes],105[die]
</hint> -->
</attribute>
<attribute id="loginShell">
      <display>Login shell</display>
      <order>5</order>
      <page>1</page>
<!--
<value><![CDATA[=php.PickList(/;(&(objectClass=posixAccount));loginShe
ll;%loginShell%;;;;loginShell)]]></value> -->
      <type>select</type>
      <value id="/bin/bash">/bin/bash</value>
      <value id="/bin/sh">/bin/sh</value>
      <value id="/bin/csh">/bin/csh</value>
      <value id="/bin/tsh">/bin/tsh</value>
</attribute>
<attribute id="userPassword">
      <display>Password</display>
      <!-- <helper>
      <display>Encryption</display>
      <id>enc</id>
      <value>=php.PasswordEncryptionTypes()</value>
      </helper> -->
      <icon>lock.png</icon>
      <order>5</order>
      <page>1</page>
      <post>=php.PasswordEncrypt(%enc%;%userPassword%)</post>
      <spacer>1</spacer>
      <verify>1</verify>
</attribute>
</attributes>
</template>
```

El archivo anterior es la plantilla que se usará para crear nuevos usuarios en el LDAP, esta plantilla está en el directorio "`creation`" lo que significa que está disponible para la creación de usuarios, si se desea que también esté disponible para modificar usuarios se procede de la siguiente forma.

Se debe copiar la plantilla al directorio "`modification`", de la siguiente forma:

```
# cp custom_die.xml ../modification/
```

Si se desea tener todas las plantillas que trae por default phpldapadmin para poder modificar los objetos de LDAP se deben copiar al directorio mencionado anteriormente.

Para copiar los archivos se hace de la siguiente manera:

```
# cp *.xml ../modification/
```

Lo anterior si se está en el directorio:

```
/opt/apache2/htdocs/phpldapadmin/templates/creation/
```

III.II.II.III Configuración de NoCatAuth usando un servidor LDAP

Para la configuración del servidor NoCatAuth utilizando un servidor LDAP se hará lo siguiente:
Se cambia el directorio de trabajo y se edita el siguiente archivo

```
# cd /usr/local/nocat/authserv/
# vi nocat.conf
```

```
###### authserv.conf -- NoCatAuth Authentication Service Configuration
…
PGPKeyPath        /usr/local/nocat/authserv/pgp
…
DataSource        LDAP
…
LDAP_Host             192.168.5.3
LDAP_Base             ou=people,dc=die,dc=fi,dc=unam
LDAP_Admin_User       cn=admin,dc=die,dc=fi,dc=unam
LDAP_Admin_PW         <secretoldap>
LDAP_Hash_Passwords   Yes
LDAP_Search_as_Admin  Yes
LDAP_Filter           uid
…
LocalGateway      192.168.5.1
…
```

El cuadro anterior muestra los parámetros más importantes que se deben editar, donde:

- PGPKeyPath: Directorio donde se almacenan las llave pgp, que se crearon posterior a la instalación de NoCatAuth.
- DataSource: Indica con que servicio va a autenticar, los valores posibles son: DBI, Passwd, LDAP, RADIUS, PAM, Samba, IMAP, NIS. Es importa verificar que los parámetros asociados a otros servicios estén comentados y sólo dejar los parámetros que son utilizados por el servicio que se selecciona en este punto.
- LocalGateway: Indica la dirección IP del servidor NoCat Gateway.

Los siguientes parámetros sólo se usan para autenticación con LDAP.

- LDAP_Host: Indica el nombre o dirección IP del servidor LDAP.
- LDAP_Base: Indica el contenedor de LDAP para buscar y crear usuarios.
- LDAP_Admin_User: Indica el nombre completo del administrador (usuario). Este usuario debe ser capaz de crear usuarios en el contenedor descrito anteriormente.
- LDAP_Admin_PW: Indica la contraseña del administrador.

- LDAP_Hash_Passwords: Indica si las contraseñas serán cifradas con MD5 antes de insertar el registro.
- LDAP_Search_as_Admin: Indica si todas las operaciones se harán como el usuario administrador, si se coloca "NO", la creación de usuarios se hace de forma anónima.
- LDAP_Filter: Indica el nombre del atributo que contendrá el ID de usuario, puede tomar el valor de mail o username.

Nota: Lo siguiente es muy importante. Para que funcione correctamente la autenticación con el servidor LDAP, se deben de instalar los paquetes Net::LDAP y IO::Socket::SSL.

Para instalar dicho paquete se hará con la ayuda del manejador de paquetes de Perl, CPAN, de la siguiente forma:

```
# perl -MCPAN -e shell
```

Lo anterior ejecuta el manejador de paquetes de perl. Si es la primera vez que se ejecuta nos pedirá cierta información, se dejan los datos por default, hasta que nos proporciones un shell de CPAN.

```
cpan[1]> install Net::LDAP
…
- Convert::ASN1    ...missing. (would need 0.07)
[SASL authentication]
- Authen::SASL     ...missing. (would need 2.00)
==> Auto-install the 1 optional module(s) from CPAN? [n] y
[LDAP URLs]
- URI::ldap        ...missing. (would need 1.1)
==> Auto-install the 1 optional module(s) from CPAN? [n] y
[LDAPS]
- IO::Socket::SSL  ...missing. (would need 1.26)
==> Auto-install the 1 optional module(s) from CPAN? [n] y
…
Shall I follow them and prepend them to the queue
of modules we are processing right now? [yes]
…
…
…
 /usr/bin/make install -- OK

cpan[2]> quit
```

Al momento de intentar instalar el paquete Net::LDAP, CPAN buscará las dependencias necesarias y nos preguntará si se desean instalarlas, basta con aceptar y esperar a que CPAN termine de instalar cada dependencia, cuando aparezca nuevamente el cursor de CPAN significa que ha terminado la instalación y configuración de los paquetes descargados. Para salir del shell de CPAN solo hay que teclear "`quit`".

III.III Integración de servicios

La integración de los servicios del portal cautivo con el servidor de autenticación, se realiza en diferentes etapas, una de ellas es la vinculación del servidor nocat gateway con el servidor nocat auth, por otra parte, el servidor nocatauth, debe comunicarse con el servidor de autenticación, ya sea RADIUS, LDAP o una base de datos, estas vinculaciones se indican en los archivos de configuración correspondientes.

Posteriormente se debe de vincular el servidor web, con la arquitectura creada por el servicio de NoCatAuth, el servidor web es el encargado de mostrar la página web que realizará la autenticación de los usuarios.

Para realizar esta vinculación se procedió de la siguiente manera, modificar el archivo de configuración del servidor web, `httpd.conf`.

```
# vi /opt/apache2/conf/httpd.conf
```

```
# This is the main Apache HTTP server configuration file.  It contains …
…
…
#DocumentRoot "/opt/apache2/htdocs"
DocumentRoot "/usr/local/nocat/authserv/htdocs"
…
#<Directory "/opt/apache2/htdocs">
<Directory "/usr/local/nocat/authserv/htdocs">
…
#ScriptAlias /cgi-bin/ "/opt/apache2/cgi-bin/"
…
#<Directory "/opt/apache2/cgi-bin">
<Directory "/usr/local/nocat/authserv/cgi-bin">
…
…
Include conf/extra/httpd-ssl.conf
Include /usr/local/nocat/authserv/httpd.conf
…
```

En el archivo se deben comentar las líneas indicadas y se deben agregar las líneas anteriores para que el servidor apache utilice los archivos que trae el servicio de nocatauth.

Ahora se edita el archivo de configuración del SSL, para que nocatauth pueda utilizar el https sin problemas.

```
# vi /opt/apache2/conf/extra/httpd-ssl.conf
```

```
# This is the Apache server configuration file providing SSL support.
…
…
#<Directory "/opt/apache2/cgi-bin">
<Directory "/usr/local/nocat/authserv/cgi-bin">
…
```

De esta forma, queda vinculado el servicio web con el servicio de nocatauth.

Ahora sólo se debe reiniciar el servicio de apache:

```
# /etc/init.d/httpd restart
```

```
Apache/2.2.21 mod_ssl/2.2.21 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server authNodos:443 (RSA)
Enter pass phrase:

OK: Pass Phrase Dialog successful.
```

Se coloca la frase que se usó para crear los certificados.

En este capítulo se explicó como configurar e instalar cada uno de los servicios de autenticación que se usaron en la implementación, cabe señalar que cada servicio es independiente.

En la tabla 2 se muestra la comparación entre cada implementación realizada.

Tabla 2. Comparación de implementaciones de NoCat con un servidor de autenticación

Servidor de Autenticación	Cambios al cliente	Configuración del servicio	Instalación del servicio	Administración de usuarios
RADIUS	No	Complejo	Complejo	Medio
LDAP	No	Complejo	Muy complejo	Medio
Base de datos	No	Medio	Fácil	Medio
SAMBA	No	Complejo	Medio	Medio

Capítulo IV. Pruebas y resultados

Una vez configurado los servicios necesarios se procedió a realizar las pruebas con los clientes. La zona en la que se dará el servicio de portal cautivo es en la zona de nodos del Laboratorio de Computación Sala C.

Figura 4.1 Mesa de nodos Sala C

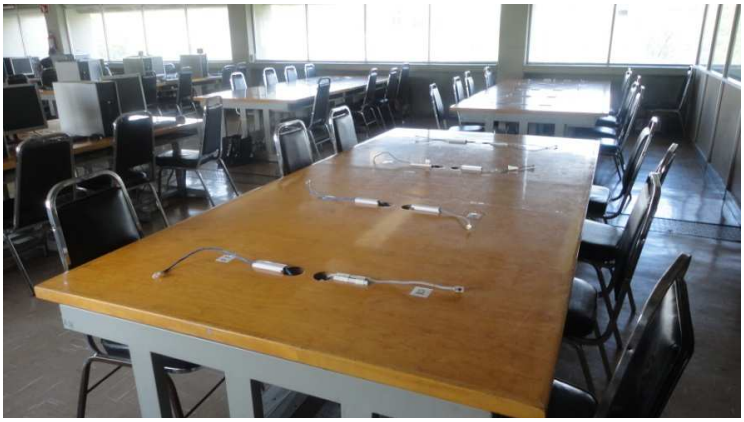


Figura 4.2 Zona de nodos Sala C

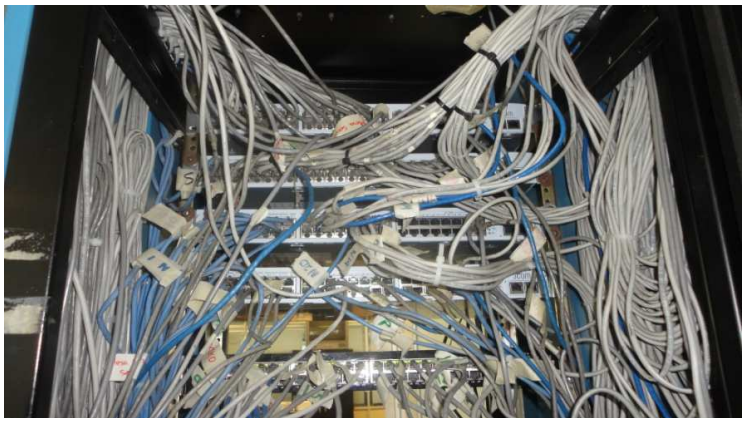


Figura 4.3 Switch de la red Sala C

Clientes alámbricos

En la sección de nodos, que posee tecnología Ethernet, se utilizó una laptop Samsung N130 con sistema operativo Windows XP para realizar las pruebas, sólo se conectó el cable de red a la interfaz de red de la laptop involucrada. Al momento de conectar el cable, se identificó la red conectada y se le asignó una dirección IP.

Figura 4.4 Computadora portátil en zona de nodos

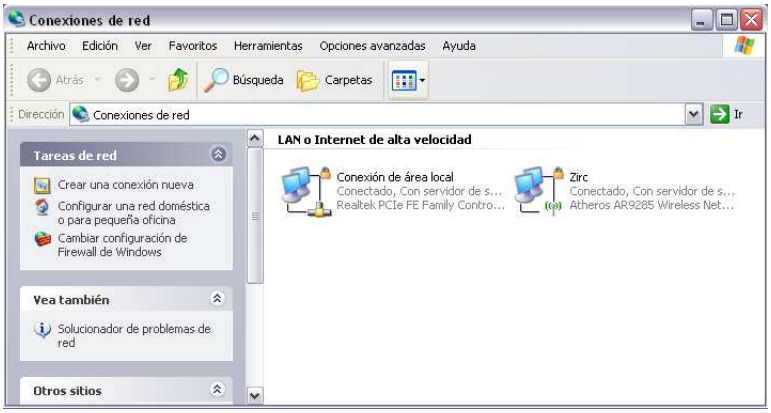


Figura 4.5 Ventana conexiones de red - Windows

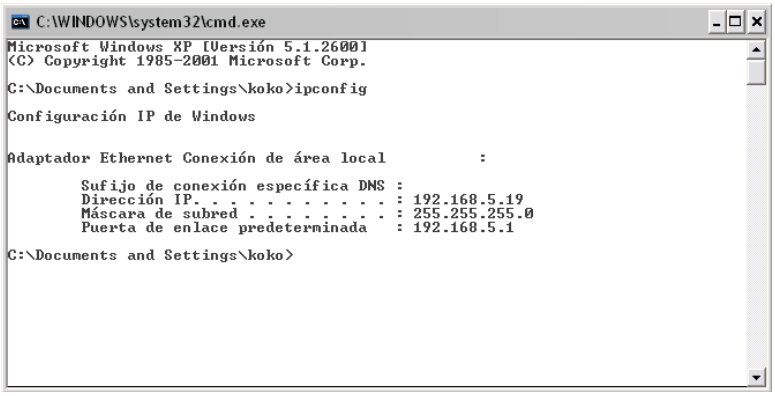


Figura 4.6 Datos de configuración de red cliente

Posteriormente se abrió el navegador web Google Chrome y se solicitó la página de google.

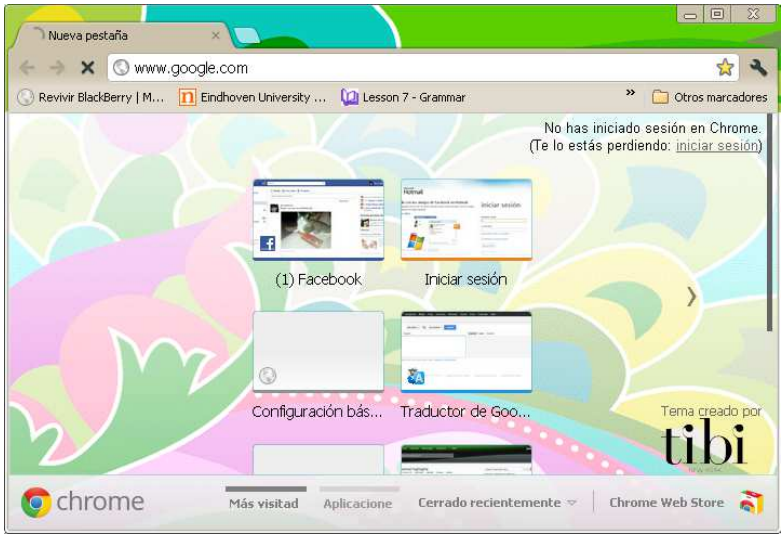


Figura 4.7 Navegador web – google chrome

Lo que se obtuvo fue la siguiente página

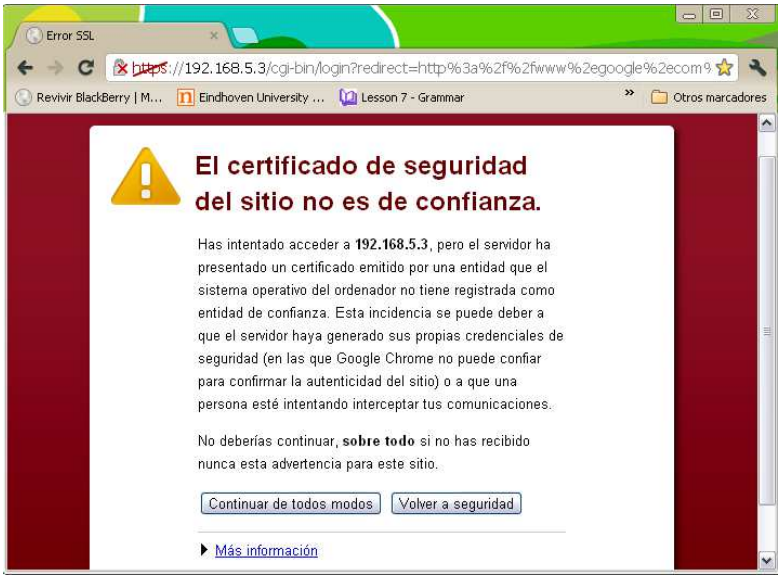


Figura 4.8 Pantalla de aviso de certificado web

Está página indica que hay un certificado SSL, que no se ha comprobado, se aceptó el certificado y se hizo click en el botón “Continuar de todos modos”. Ahora se muestra la siguiente página.

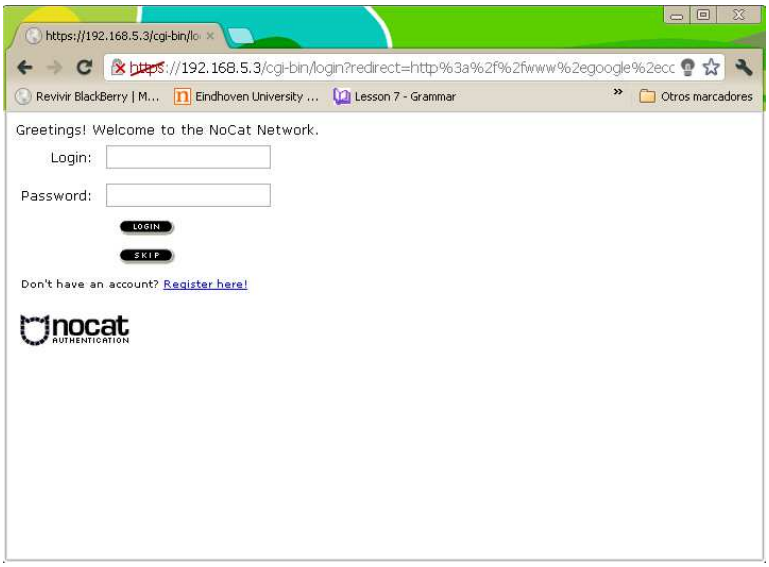


Figura 4.9 Pantalla de acceso

Es una página donde se solicita el login y el password. Incluso si no se está registrado, ofrece un link para realizar dicho registro.

Si se entra a la página de registro, se piden los siguientes datos:

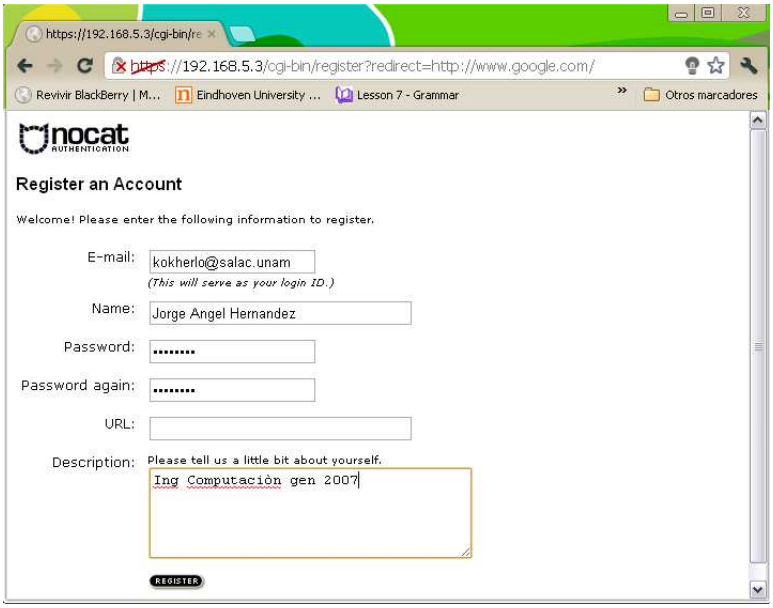


Figura 4.10 Pantalla de registro de usuarios

Posterior al registro, indica si éste fue exitoso o que sucedió durante el registro.

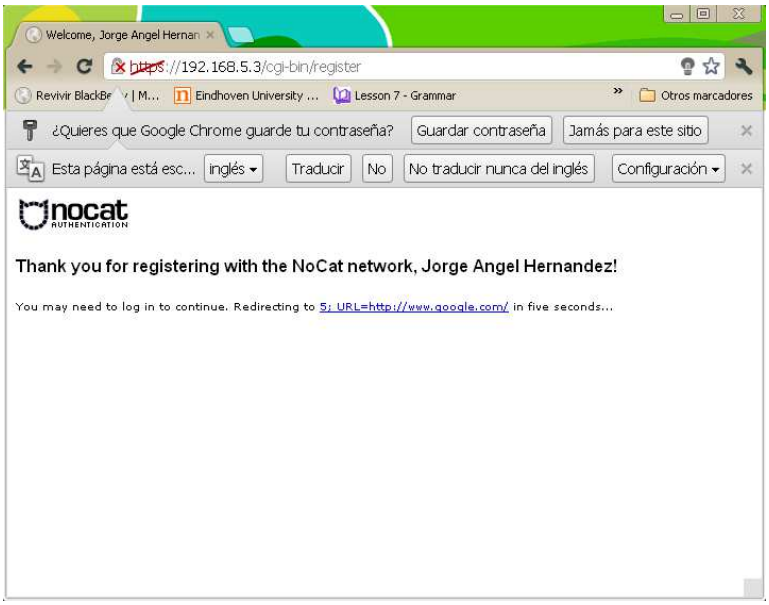


Figura 4.11 Pantalla de aviso de registro exitoso

Automáticamente se carga la página de autenticación, se pide un login y un password, ahora que el usuario ya está registrado, se pueden usar esos datos.

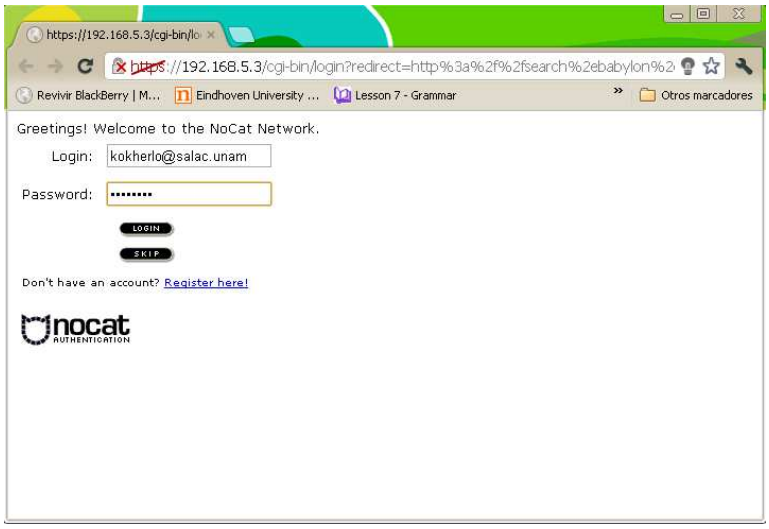


Figura 4.12 Pantalla de acceso con datos

Cuando se realiza la autenticación y la contraseña es incorrecta, se indica mediante un mensaje en la misma página.

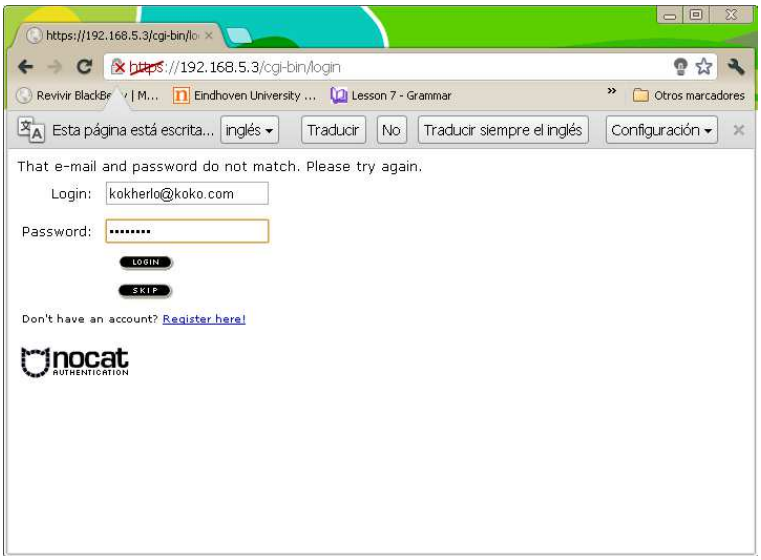


Figura 4.13 Pantalla de error de autenticación

Al momento de autenticarse de forma correcta se muestra un mensaje de bienvenida.

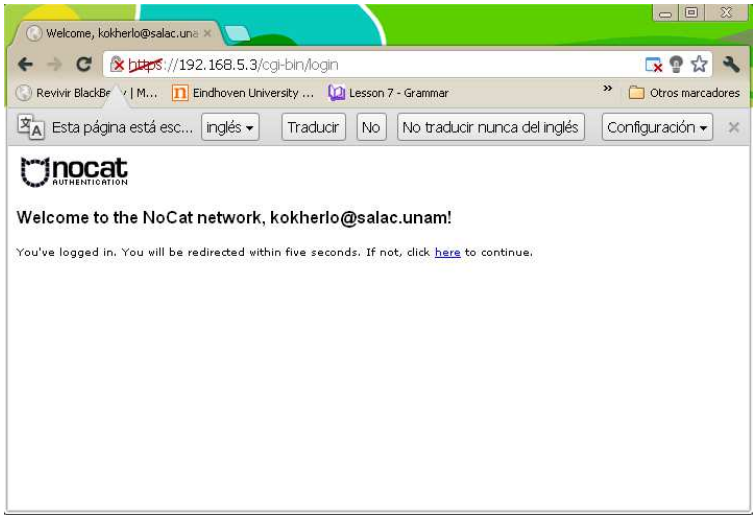


Figura 4.14 Pantalla de bienvenida

Si están permitidos los pop-up, se despliega una ventana la cual indica que la autenticación será renovada automáticamente mientras se mantenga la sesión iniciada.

Figura 4.15 Ventana de sesión autorizada

Posteriormente el navegador web despliega la página que se solicitó inicialmente.

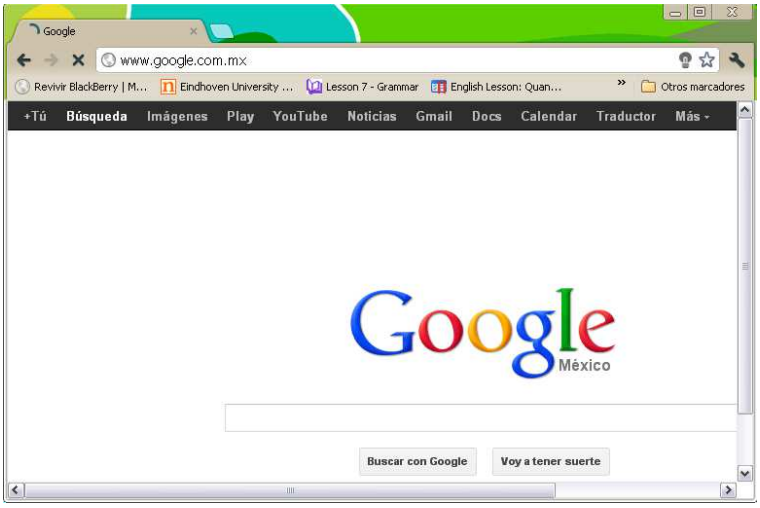


Figura 4.16 Navegador web con la página solicitada

Gracias a que la herramienta es de código abierto se pueden hacer modificaciones a la página de autenticación y a los mensajes mostrados.

Después de realizar los cambios necesarios para adaptar la herramienta a nuestras necesidades, las nuevas pantallas quedaron de la siguiente forma:

Figura 4.17 Pantalla de acceso modificada

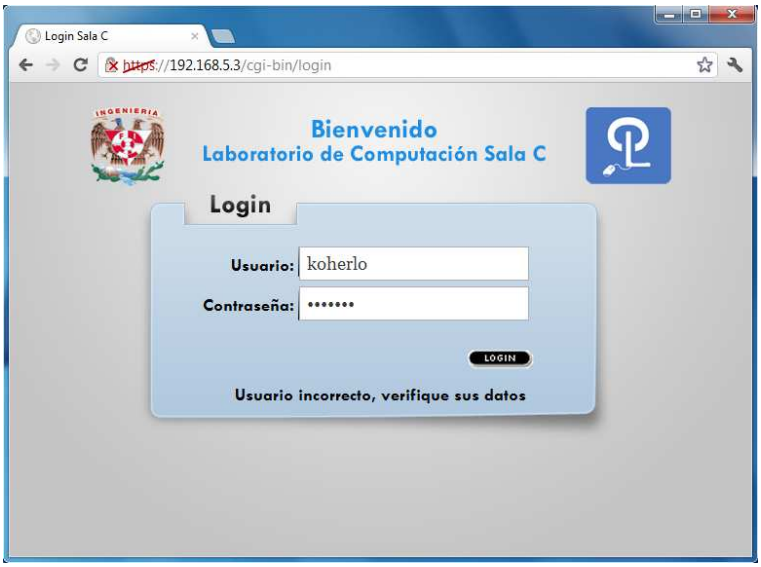


Figura 4.18 Pantalla de error de autenticación modificada (usuario incorrecto)

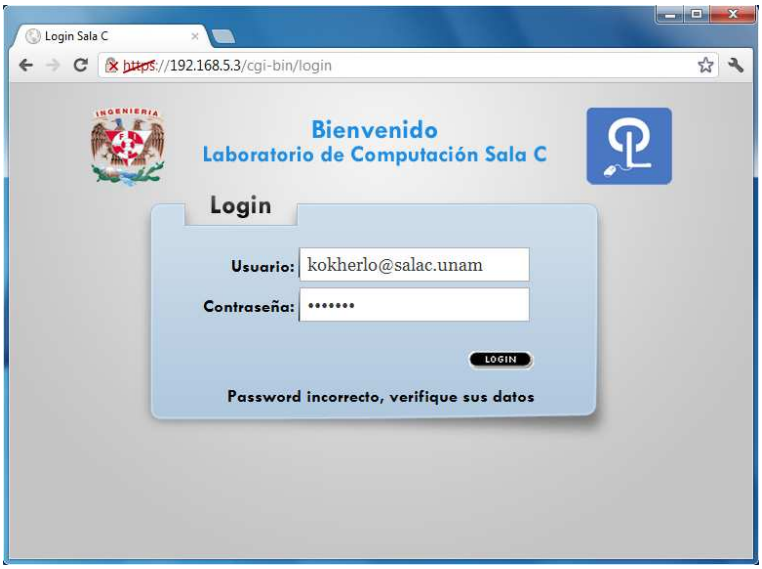


Figura 4.19 Pantalla de error de autenticación modificada (contraseña incorrecta)

Figura 4.20 Pantalla de bienvenida modificada

Figura 4.21 Ventana de sesión autorizada modificada

Las pantallas anteriores son de una computadora de escritorio HP 505B con sistema operativo Windows 7 utilizando el navegador web Google Chrome.

Clientes inalámbricos

Para la red inalámbrica también se utilizó una laptop marca Samsung con sistema operativo Windows XP para realizar las pruebas. Para empezar el proceso primero se tuvo que identificar la red que se iba a utilizar, la red que se usó fue la siguiente:

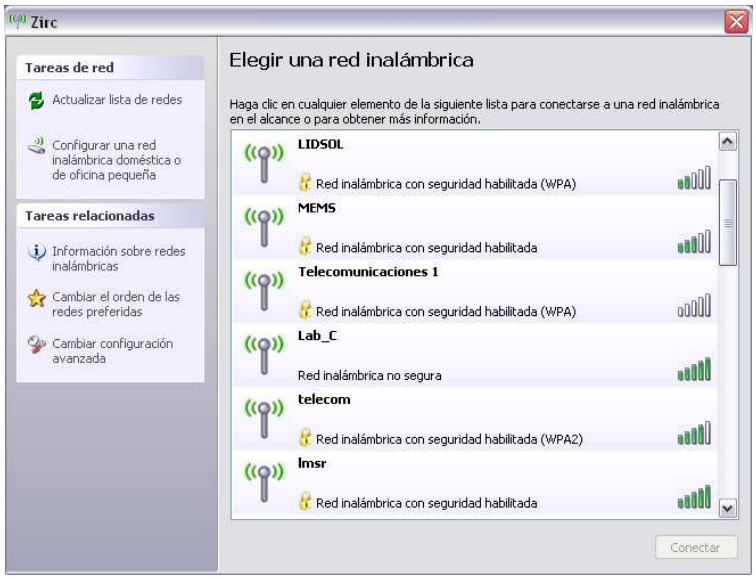


Figura 4.22 Ventana de administración de redes inalámbricas - Windows

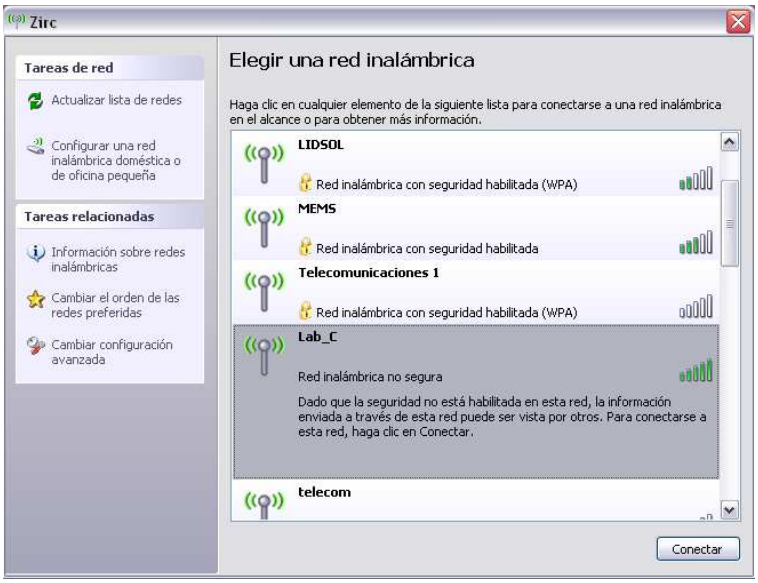


Figura 4.23 Red inalámbrica con autenticación mediante portal cautivo

Posteriormente se conectó a dicha red, cabe mencionar que indica que la red seleccionada no posee seguridad, por lo que puede ser inseguro si nos conectamos a dicha red.

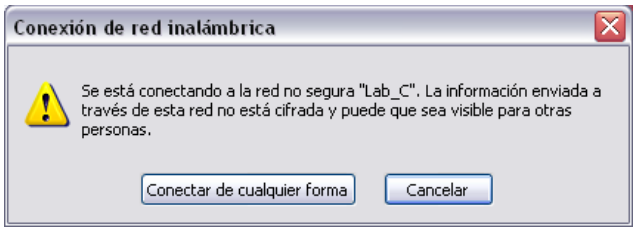


Figura 4.24 Aviso de conexión a red inalámbrica

Una vez conectados a la red, se verificó que se haya asociado correctamente a la red inalámbrica.

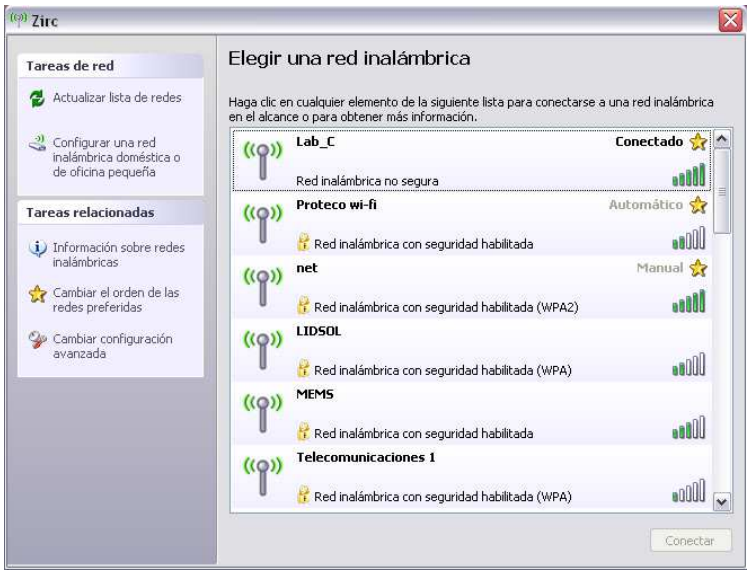


Figura 4.25 Ventana que indica la red conectada

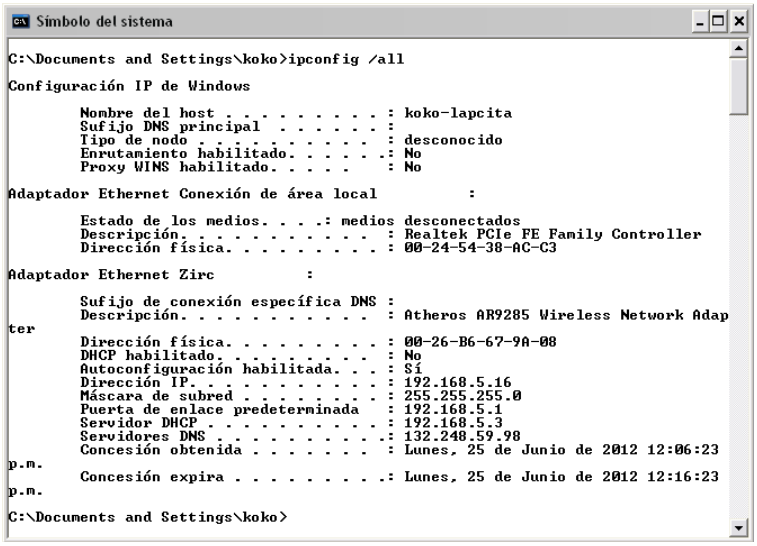


Figura 4.26 Datos de configuración de red de cliente inalámbrico

Después se procedió de la misma forma, abrir el navegador web, en este caso, Google Chrome y se realizó el proceso descrito anteriormente.

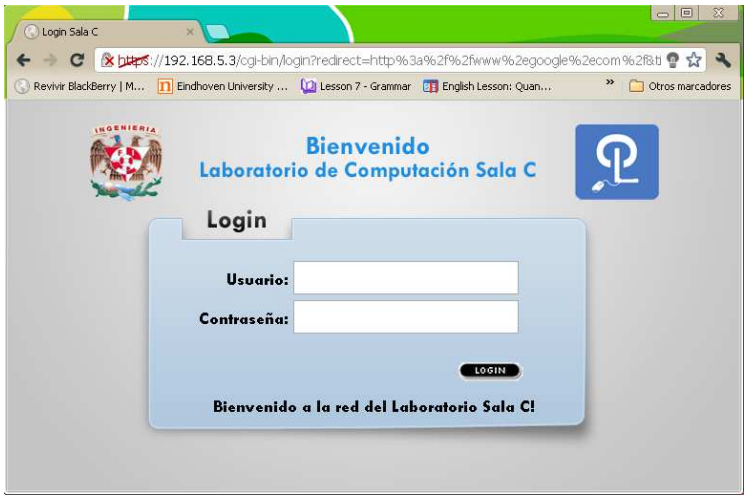


Figura 4.27 Pantalla de autenticación en cliente inalámbrico

Gracias a que el sistema de autenticación configurado utiliza el protocolo http, es compatible para cualquier dispositivo inalámbrico que posea un navegador web y pueda conectarse a una red WiFi.

También se hicieron pruebas con un Smartphone Samsung Galaxy Ace con sistema operativo Android 2.3, a continuación se muestran las pantallas obtenidas en el dispositivo móvil.

Figura 4.28 Redes disponibles - móvil

Figura 4.29 Selección de red - móvil

Figura 4.30 Red conectada - móvil

Figura 4.32 Menú buscar - móvil

Figura 4.31 Solicitar página web - móvil

Figura 4.33 Formulario de acceso - móvil

Figura 4.34 Datos de acceso – móvil

Figura 4.35 Mensaje de bienvenida - móvil

Figura 4.36 Página web solicitada - móvil

Por otra parte se realizaron pruebas de penetración al mecanismo de autenticación.

Las pruebas que se realizaron a este mecanismo de autenticación fueron:

- SQL Injection
- Ataques de fuerza bruta
- Escaneo de la red
- Monitoreo de paquetes

La teoría menciona la existencia de etapas por las que pasa un ataque, dichas etapas son:

1. Reconocimiento (Reconnaissance): Esta etapa involucra la obtención de información con respecto a una potencial víctima que puede ser una persona u organización.

2. Exploración (Scanning): En esta segunda etapa se utiliza la información obtenida en la etapa 1 para tratar de obtener información sobre el sistema víctima como direcciones IP, nombres de host, datos de autenticación, entre otros.

3. Obtener acceso (Gaining Access): En esta instancia comienza a materializarse el ataque a través de la explotación de las vulnerabilidades y defectos del sistema descubiertos durante las fases de reconocimiento y exploración.

4. Mantener el acceso (Maintaining Access): Una vez que el atacante ha conseguido acceder al sistema, buscará implantar herramientas que le permitan volver a acceder en el futuro desde cualquier lugar donde tenga acceso a Internet. Para ello, suelen recurrir a utilidades como backdoors, rootkits y troyanos.

5. Borrar huellas (Covering Tracks): Una vez que el atacante logró obtener y mantener el acceso al sistema, intentará borrar todas las huellas que fue dejando durante la intrusión para evitar ser detectado por el profesional de seguridad o los administradores de la red.

En la etapa de reconocimiento, se busca información de una posible víctima, en este caso en particular, el objetivo es obtener un acceso autorizado en este mecanismo de autenticación.

Para la etapa de exploración, se realizó lo siguiente:

Se realizó un escaneo de red con la herramienta 3COM Network Supervisor con las parámetros ACK, SYN, obteniendo como resultado, un esquema que no indica la estructura de la red. Dicho esquema se muestra en la figura 4.37.

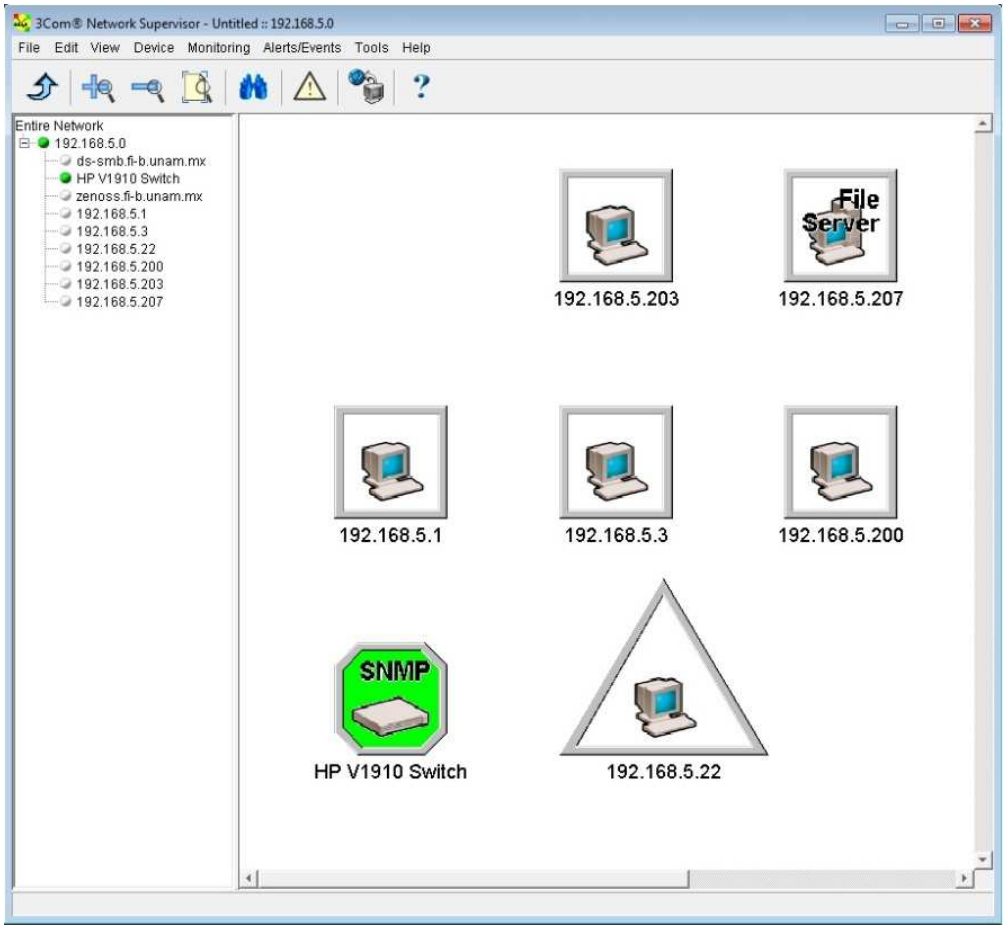


Figura 4.37 Escaneo de la red mediante 3Com Network Supervisor

Con el escaneo de red que se realizó, se obtuvo muy poca información, ya que no muestra la estructura de la red, ni las interconexiones existentes, sólo muestra los equipos que estaban en ese momento conectados a la red. También nos muestra el modelo de un switch que pertenece a la red.

Posteriormente se hizo un escaneo de puertos, en este escaneo se detectó la existencia de un servidor que tiene habilitados los puertos 80 y 443. Por otra parte, al solicitar una página web, la solicitud es redireccionada automáticamente a un formulario web de autenticación, por la dirección IP que se muestra en la dirección URL, se puede deducir que dicho formulario se encuentra alojado en el servidor detectado, lo que indica que se trata de un servidor web. A continuación se muestra el escaneo realizado.

```
[root@kokherlo ~]# nmap 192.168.5.0/24

Starting Nmap 5.21 ( http://nmap.org ) at 2012-11-08
12:54 CST
Nmap scan report for 192.168.5.1
Host is up (0.00016s latency).
All 1000 scanned ports on 192.168.5.1 are closed
MAC Address: 00:50:DA:2D:8E:83 (3com)

Nmap scan report for 192.168.5.3
Host is up (0.000023s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.5.22
Host is up (0.00024s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: F4:CE:46:F3:D9:F0 (Hewlett Packard)

Nmap scan report for 192.168.5.200
Host is up (0.0030s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 20:FD:F1:D5:CF:A0 (Unknown)

Nmap scan report for 192.168.5.203
Host is up (0.00016s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 70:71:BC:68:EE:5F (Unknown)

Nmap done: 256 IP addresses (5 hosts up) scanned in
21.79 seconds
```

Posteriormente se hizo un monitoreo de paquetes con la herramienta WireShark, en el cuál se observó que los datos del usuario no se pueden obtener, ya que son cifrados mediante SSL. En el monitoreo se puede observar el tráfico de otros usuarios, pero tampoco brinda la suficiente información para saber con exactitud la estructura del mecanismo de autenticación implementado. Dicho monitoreo se muestra en la figura 4.38.

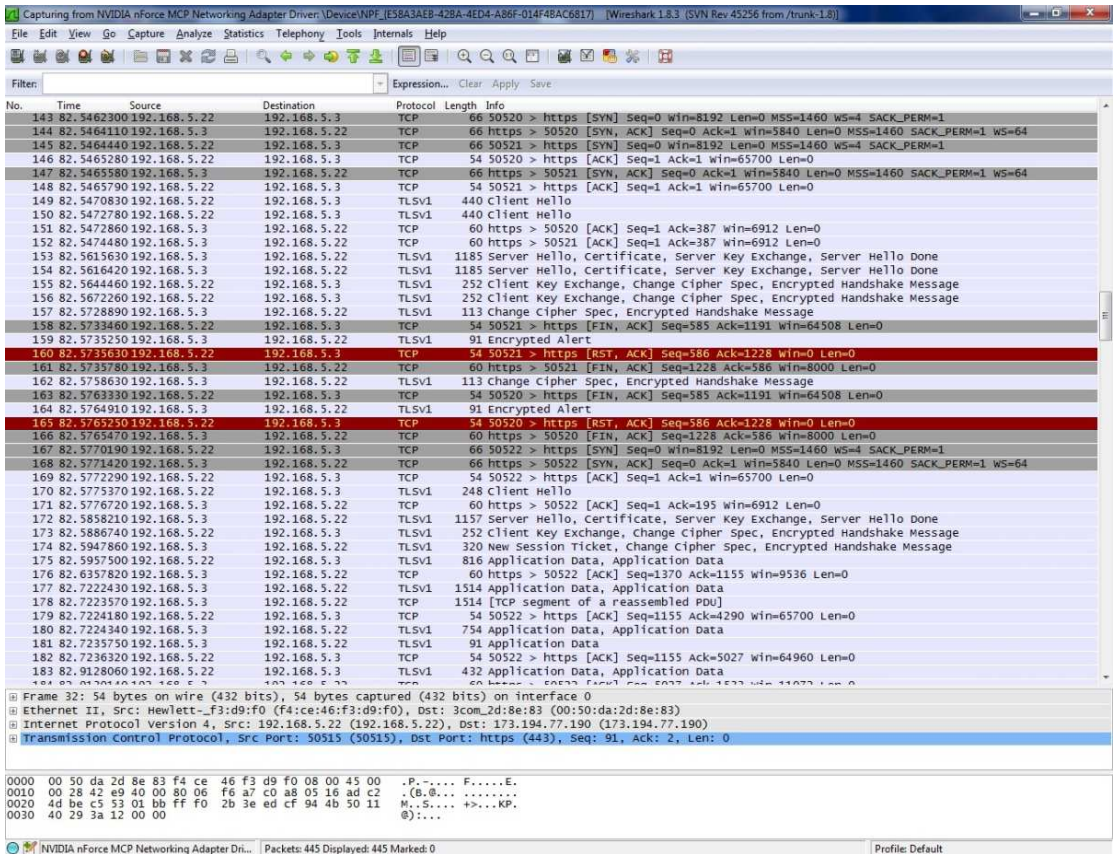


Figura 4.38 Monitoreo de paquetes mediante WireShark

Con la información recabada con el escaneo y el monitoreo, se comprueba la existencia de un servidor web que aloja el sistema de autenticación mediante un formulario. Los ataques más comunes para este tipo de escenarios son SQL Injection y ataques de fuerza bruta.

Para la etapa 3, que es obtener el acceso, se intentó mediante el ataque de SQL Injection.

Las sentencias que se utilizaron en el ataque de SQL injection se muestran en la Tabla 3.

Tabla 3. Sentencias utilizadas en el ataque SQL Injection

Sentencia usada	Comportamiento	
	Acceso Concedido	Mensaje Mostrado
OR 1=1;	No	Usuario incorrecto
a'; select * from users;	No	Usuario incorrecto
==TRUE	No	Usuario incorrecto
-1' UNION Select * from users;	No	Usuario incorrecto
OR ''='	No	Usuario incorrecto
x' AND userid IS NULL; --	No	Usuario incorrecto
lia'; DELETE FROM users;	No	Usuario incorrecto

Una vez realizados los ataques para penetrar el mecanismo de autenticación, los resultados no fueron positivos, ya que no se pudo obtener un acceso autorizado.

En esta etapa se pueden realizar los ataques:

- SQL Injection
- Fuerza bruta
- Cross-Site Scripting
- Conexiones directas a los puertos
- IPSpoofing
- MACSpoofing
- Phishing
- Ingeniería Social
- Y otros

El ataque realizado se detuvo en la etapa 3, ya que no se logró conseguir el acceso. Por lo consiguiente las etapas 4 y 5 no se pudieron llevar a cabo.

Por otra parte con el usuario que no requiere autenticación el único alcance que se tiene es consultar la página web de la División de Ingeniería Eléctrica, por lo que no se puede utilizar algún servidor externo para intentar realizar un ataque desde el exterior.

Para el fortalecimiento de la seguridad en cada equipo que compone este mecanismo de autenticación, las buenas prácticas descritas en el NIST 800-123, dicen que se debe tener en cuenta lo siguiente:

Hardening a servidores en general
Usar un equipo por cada servicio a instalar
Revisar y corregir vulnerabilidades del sistema operativo base
Revisar y corregir vulnerabilidades del kernel del SO
Aplicar parches de seguridad al kernel del SO
Instalar sólo lo que se necesite
No activar servicios que no se usan
Eliminar cuentas de usuario que no se usen
Crear un usuario para administrar el servicio a instalar
Bloquear los puertos que no se usen

Hardening a servidor web APACHE
Determinar que tecnologías va a soportar (CGI, Perl, etc.)
Seleccionar la versión adecuada a instalar
Revisar los módulos de apache a instalar
Compilar el software a instalar según las necesidades
Crear un usuario para administrar el servicio a instalar
Crear estructura de directorio para el servicio (chrooting)
Realizar configuración adecuada del servicio
Eliminar archivos no necesarios para el servicio
Monitorear tráfico del servidos
Bloquear los puertos que no se usen

Hardening a servidor DHCP
Seleccionar la versión adecuada a instalar
Crear un usuario para administrar el servicio a instalar
Realizar configuración adecuada del servicio
Bloquear los puertos que no se usen

Hardening a servidor base de datos MYSQL
Instalar la versión a utilizar
Deshabilitar o restringir el acceso remoto
Deshabilitar el uso de LOCAL INFILE
Cambiar la contraseña y el nombre del usuario root
Eliminar la base de datos "test"
Eliminar las cuentas de usuarios que no se usan
Asignación de permisos mínimos a usuarios
Activar el sistema de log
Cambiar el directorio del administrador de la BD
Eliminar el history de la instalación
Usar parches de seguridad si es necesario

Hardening a servidor LDAP (Directory Server)
Revisar con que tecnologías será compatible
Crear un usuario para administrar el servicio a instalar
Instalar la versión a utilizar
Realizar configuración adecuada
Bloquear los puertos que no se usen

Hardening a servidor RADIUS (FreeRadius)
Revisar que versión se va a utilizar
Crear un usuario para administrar el servicio a instalar
Compilar el software e instalar según las necesidades
Instalar librerías extras necesarias
Realizar configuración adecuada
Bloquear los puertos que no se usen

Conclusiones

Actualmente la autenticación es un mecanismo muy importante para proteger el acceso a los recursos de información. Por otra parte, existen muchas herramientas que permiten llevar a cabo la autenticación de usuarios en una red, cada una de éstas tiene ventajas y a la vez desventajas. Dichas herramientas pueden ir desde un simple software hasta un protocolo complejo, es importante, conocer la existencia de dichas herramientas ya que pueden ser de gran utilidad.

En el área de seguridad, se concluye que nada puede ser totalmente seguro, ya que algunas de las desventajas que existen en las herramientas, son ocasionadas por la naturaleza del protocolo que se usa. En otras implementaciones el punto más débil del mecanismo de autenticación es el propio usuario.

El portal cautivo es una herramienta que permite manejar dos esquemas de usuarios:

- Usuarios que deben autenticarse para poder usar el recurso de internet,
- Usuarios que no requieren de una autenticación para poder navegar sobre sitios web específicos.

La herramienta NoCat es una implementación de portal cautivo que ofrece gran compatibilidad con protocolos de autenticación. La complejidad de la instalación y configuración de una infraestructura de red que autentique con esta herramienta depende mucho del servidor de autenticación que se utilice, ya que no es lo mismo instalar un servidor LDAP, que uno de RADIUS, uno de SAMBA o simplemente un servidor de base de datos.

La arquitectura empleada en un mecanismo de autenticación es un factor muy importante para poder generar un sistema robusto, ya que al separar los servicios en equipos físicos distintos, se vuelve más complejo el poder vulnerar dicho sistema, debido a que se requiere comprometer más equipos con configuraciones diferentes cada uno.

Por otra parte con la instalación y configuración de varios servicios de autenticación, comprendí que muchos de estos servicios se pueden complementar para generar un servicio más robusto, y que realizar está configuraciones requiere de un conocimiento especializado, a mayor complejidad del servicio, mayor es el conocimiento requerido.

La elaboración de este trabajo me permitió adquirir conocimientos más sólidos, en el área de redes y seguridad, principalmente sobre los mecanismos de autenticación, así como, conocer con más detalle algunos de los protocolos de autenticación robustos, administración y configuración de servidores Linux y arquitecturas de red. Por otra parte me permitió poner en práctica conocimientos que he adquirido durante mi estancia en la Universidad Nacional Autónoma de México.

Con la implementación de este sistema de autenticación, se cumple el objetivo principal del trabajo, ya que apoya en el control de acceso de usuarios al Laboratorio de Computación Sala C, se generó una herramienta de autenticación para redes inalámbricas que puede ser implementada a nivel institucional, aparte que se utilizaron sólo herramientas de software libre.

Con la implementación de este mecanismo de autenticación, se observó que el control de acceso a la zona de nodos del Laboratorio de Computación Sala C se fortaleció. Lo anterior fue por las acciones que tomaron los usuarios al presentárseles un formulario de autenticación. Estas acciones permitieron al encargado del laboratorio poder registrar a dichos usuarios.

En cuestión a la red inalámbrica, la implementación de este mecanismo de autenticación es totalmente factible, ya que sólo se requiere configurar el access point para crear una red inalámbrica y todo el proceso de autenticación lo realiza el sistema implementado.

Actualmente se están realizando pruebas con el protocolo de autenticación SAMBA, ya que este servicio es el que se encuentra instalado en el Laboratorio de Computación Sala C.

Por otra parte, este tipo de mecanismo de autenticación puede ser implementado en alguna otra dependencia que requiera administrar el esquema de usuarios fijos y el esquema de usuarios móviles utilizando un sólo sistema o tenga alguna necesidad de las que se mencionaron al inicio del capítulo II.

Con las pruebas de penetración realizadas, se observó que el mecanismo de autenticación creado no fue ataques más comunes que pueden existir en cuestión de vulnerar la seguridad, aunque no lo exenta de ser vulnerado, pero gracias a la característica de guardar información detallada de los dispositivos de conexión se tiene un registro y se puede detectar al usuario que realice alguna penetración.

A futuro, este sistema de autenticación, se puede implementar haciendo todas las consideraciones que conlleva crear un sistema de seguridad robusto, como seguridad perimetral, control de acceso, análisis de riesgos, etc. O implementarlo con nuevos protocolos de autenticación, o generar una arquitectura más compleja que permita mejorar la eficiencia o rendimiento del sistema de autenticación creado.

Glosario

AAA: Modelo que hace referencia a un sistema que permite la autenticación, autorización y contabilidad.

Access point (AP): Punto de acceso, es un dispositivo que permite la interconexión de dispositivos alámbricos con dispositivos que forman una red inalámbrica. Permite la transmisión de datos entre los dos tipos de redes.

Accounting: Contabilidad, proceso en el cual se mide y documenta los recursos que un usuario aprovecha durante el acceso a un sistema.

ACK: Acknowledgement, acuse de recibo, es un mensaje que se envía para confirmar que un mensaje ha sido recibido.

Autenticación: es el proceso en el cual se verifica la identidad de una persona o equipo.

Autorización: es el proceso en el cual se determina que es lo que puede realizar un usuario que previamente se ha autenticado en un sistema.

CHAP: Challenge Handshake Authentication Protocol, es un protocolo de autenticación remota que verifica periódicamente la identidad del cliente usando un intercambio de información que consta de tres etapas.

DBA: Data Base Administrator, administrador de base de datos, es el profesional de tecnologías de la información, responsable de los aspectos técnicos, tecnológicos y reglas de negocios de las bases de datos.

DBMS: Database Management System, manejador de base de datos, se define como una colección de herramientas responsables de proporcionar un ambiente conveniente y eficiente para acceder a una base de datos

DHCP: Dynamic Host Configuration Protocol, protocolo de configuración dinámica de host, es un protocolo de red que permite a los clientes de una red IP obtener sus parámetros de configuración automáticamente

DSA: Digital Signature Algorithm, algoritmo de firma digital, es un estándar del Gobierno Federal de los Estados Unidos de América para firmas digitales.

Elgamal: es un sistema de cifrado basado en problemas matemáticos de logaritmos discretos.

FK: Foreign Key, llave foránea, concepto usado en bases de datos relacionales, es la clave que identifica una columna o grupo de columnas en una tabla que se refiere a otra columna o grupo de columnas en otra tabla que tiene una PK.

Framework: Conjunto de conceptos, prácticas y criterios estandarizados que permiten afrontar una problemática en particular, ya que sirven como referencia para enfrentar y resolver nuevos problemas de índole similar.

Gateway: Es un dispositivo que permite interconectar redes con protocolos y arquitecturas diferentes. Su propósito es traducir la información del protocolo utilizado en una red al protocolo usado en la red de destino.

GNU: Acrónimo de GNU is Not Unix, GNU No es Unix, es un sistema operativo libre diseñado por Richard Stallman, basado en el principio de que todos los usuarios pudieran ejecutarlo, copiarlo, modificarlo y distribuirlo.

GnuPG: GNU Privacy Guard, es una herramienta de cifrado y firmas digitales, es un software libre licenciado bajo GPL.

GPL: GNU General Public License, licencia pública general de GNU, es una licencia que está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

GPS: Global Positioning System, sistema de posicionamiento global, es un sistema global de navegación por satélite que permite determinar en todo el mundo la posición de un objeto.

HTTP: Hypertext Transfer Protocol, protocolo de transferencia de hipertexto, es un protocolo a nivel de aplicación con la velocidad necesaria para distribuir y colaborar con los sistemas de información de hipermedia.

IETF: Internet Engineering Task Force, es una organización internacional dedicada a la normalización, tiene como objetivo contribuir a la ingeniería de Internet. Es mundialmente conocida por ser la entidad que regula las propuestas y los estándares de Internet.

IP: Internet Protocol, es un protocolo de comunicación de datos que funciona en la capa de red del modelo OSI.

ISP: Internet Service Provider, proveedor de servicio de Internet, entidad responsable de dar el servicio de Internet a una organización o usuario final.

Kernel: es un software que constituye la parte más importante del sistema operativo.

LAN: Local Area Network, clasificación de una red alámbrica según su área de cobertura. Esta clasificación abarca hasta un edificio.

LDAP: Lightweight Directory Access Protocol, protocolo de control de acceso, proporciona acceso a los servicios de directorio distribuido actuando conforme al estándar X.500, cumple con el modelo AAA.

Login: Forma de referirse al modo de iniciar sesión en un sistema. También se usa para referirse al nombre de usuario en el proceso de autenticación.

NAS: Network Access Server, servidor de acceso a la red, es un punto de entrada que permite a un cliente acceder a una red. Es el encargado de proteger un recurso y siempre se apoya de otra entidad para llevar a cabo la autenticación, enviándole a éste, las credenciales proporcionadas por el cliente.

NIS: Network Information Service, sistema de información de red, es el nombre de un protocolo de servicios de directorios cliente-servidor para el envío de datos de configuración en sistemas distribuidos tales como nombres de usuarios y hosts entre las computadoras de una red.

PAP: Password Authentication Protocol, es un protocolo de autenticación simple, PAP transmite contraseñas en código ASCII sin cifrar.

PK: Primary Key, llave primaria, concepto usado en bases de datos relacionales, es una clave que identifica de forma única a cada fila de una tabla.

PPP: Point to Point Protocol, protocolo punto a punto, es un protocolo que permite establecer una comunicación entre dos dispositivos a nivel de la capa de enlace del modelo TCP/IP.

Proxy: Es un dispositivo que sirve como intermediario entre el cliente y el servidor. Es decir, intercepta las peticiones que hace un cliente a un servidor y las retransmite como propias.

QoS: Quality of Service, calidad de servicio, hace referencia a las tecnologías que garantizan la transmisión de cierta cantidad de información en un tiempo determinado. Capacidad de dar un buen servicio.

RADIUS: Remote Authentication Dial In User Service, protocolo de control de acceso, cumple con el modelo AAA.

RC4: Rivest Cipher 4, algoritmo de cifrado que utiliza los algoritmos: Key Scheduling Algorithm (KSA) y Pseudo-Random Generation Algorithm (PRGA). Fue diseñado por Ron Rivest, trabajador de la RSA Security, en el año 1987.

RFC: Request For Comments, solicitud de comentario, es el nombre que se le da a una serie de normas que definen a un protocolo, así como sus documentos relacionados.

RSA: Es un sistema criptográfico de clave pública desarrollado en 1977 por Rivest, Shamir y Adleman.

SQL: Structured Query Language, lenguaje de consulta estructurado, es un lenguaje declarativo que permite el acceso y realizar diversos tipos de operaciones en una base de datos relacional.

TCP: Transmission Control Protocol, es un protocolo de comunicación orientado a conexión de la capa de transporte del modelo OSI, garantiza que los datos enviados serán entregados en su destino sin errores y en el mismo orden en que se transmitieron.

Token: Es un paquete especial que se le otorga a un usuario autorizado para facilitar el proceso de autenticación.

UDP: User Datagram Protocol, es un protocolo usado en la capa de transporte del modelo OSI, basado en el intercambio de datagramas. Permite el envío de datagramas a la red sin que previamente se haya establecido una conexión.

UNIX: Sistema operativo portable, multitarea y multiusuario. Creado a principios de 1969.

URI: Uniform Resource Identifier, identificador de recurso uniforme, es como una dirección postal de Internet, es un identificador único y tiene información de la localización de los recursos en todo el mundo.

URL: Uniform Resource Locator, localizador de recurso uniforme, describe la ubicación específica de un recurso en un servidor en específico.

URN: Uniform Resource Name, nombre de recurso uniforme, sirve como nombre único de una pieza particular de contenido, independientemente de donde se encuentre dicho recurso.

WEP: Wired Equivalent Privacy, es un protocolo para redes WiFi que permite cifrar la información que se transmite. Proporciona un cifrado basado en el algoritmo RC4 que utiliza claves de 64 o 128 bits.

WiFi: Wireless Fidelity, fidelidad inalámbrica, es un mecanismo que permite la conexión de dispositivos electrónicos de forma inalámbrica.

WLAN: Wireless Local Area Network, acrónimo que hace referencia a una red inalámbrica local.

WPA: WiFi Protected Access, es un sistema creado para corregir las deficiencias del sistema WEP, ya que los investigadores encontraron varias debilidades en él, fue creado para utilizar un servidor de autenticación. Utiliza el algoritmo RC4 con claves de 128 bits pero su vector de inicialización es de una longitud mayor.

X.500: Estándar sobre servicios de directorio, incluye el protocolo de acceso al directorio (DAP), el protocolo de sistema de directorio, el protocolo de ocultación de información de directorio, y el protocolo de gestión de enlaces operativos de directorio.

X.511: Estándar que define la forma en la que se provee un servicio de directorio, incluyendo las operaciones de vinculación y desvinculación, operaciones de lectura, operaciones de búsqueda, modificar operaciones y errores.

XML: eXtensible Markup Language, lenguaje de marcas extensible, es un lenguaje de marcas que permite definir la gramática de lenguajes específicos para estructurar documentos grandes.

Referencias

[1] BENANTAR, Messaoud. (2006). Access Control Systems. USA: Springer.

[2] DUBOIS, Paul. (2009). MySQL. (Fourth Edition). USA: Pearson Education Inc.

[3] FLICKENGER, Rob. (2002). Building Wireless Community Networks. (First Edition). USA: O'Reilly.

[4] GAST, Matthew. (2002). 802.11 Wireless Networks: The Definitive Guide. USA: O'Reilly.

[5] GOURLEY, David, Brian Totty y otros. (2002). HTTP: The Definitive Guide. USA: O'Reilly.

[6] HASSELL, Jonathan. (2002). Radius. USA: O'Reilly.

[7] HAYES, Victor, Kerry Stuart y otros. (1999). ANSI/IEEE Std 802.11, Part 11: Wireless LAN Medium, Access Control (MAC) and Physical Layer (PHY) Specifications.

[8] HOWES, Timothy y otros. (2003). Understanding and Deploying LDAP Directory Services. (Second Edition). USA: Addison Wesley.

[9] LEHTINEN, Rick. (2006). Computer Security Basics. (Second Edition). USA: O'Reilly.

[10] LUJÁN, Sergio. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. España: Editorial Club Universitario.

[11] STEPHEN, Thomas. (2001). HTTP Essentials. USA: John Wiley & Sons, Inc.

[12] TURNBULL, James. (2005). Hardening Linux. USA: Apress.

[13] VOLLBRECHT, J., Calhoun P, y otros. (2000). AAA Authorization Framework. RFC 2904.

[14] WONG, Clinton. (2000). HTTP Pocket Reference. (First Edition). USA: O'Reilly.

[15] Red Hat. (2010). Red Hat Directory Server. Recuperado el 16 de enero de 2012, de http://docs.redhat.com/docs/en-US/Red_Hat_Directory_Server/8.2/pdf/Installation_Guide/Red_Hat_Directory_Server-8.2-Installation_Guide-en-US.pdf

[16] Freeradius. (s.f.). HOWTO WPA RADIUS. Recuperado el 23 de enero de 2012, de http://wiki.freeradius.org/WPA-HOWTO

[17] NoCat.Net. (s.f.). NoCat.Net. Recuperado el 05 de marzo de 2012, de http://nocat.net/

[18] Zorn, Nathan. (2002). Authentication Gateway HOWTO. Recuperado el 05 de marzo de 2012, de http://www.faqs.org/docs/Linux-HOWTO/Authentication-Gateway-HOWTO.html

[19] Network Working Group. (2000). Generic AAA Architecture. Recuperado el 12 de marzo de 2012, de http://tools.ietf.org/html/rfc2903

[20] Network Working Group. (2000). AAA Authorization Framework. Recuperado el 12 de marzo de 2012, de http://tools.ietf.org/html/rfc2904

[21] Network Working Group. (2000). AAA Authorization Requirements. Recuperado el 13 de marzo de 2012, de http://tools.ietf.org/html/rfc2906

[22] Network Working Group. (2000). Remote Authentication Dial In User Service (RADIUS). Recuperado el 16 de marzo de 2012, de http://tools.ietf.org/html/rfc2865

[23] Network Working Group. (2006). Lightweight Directory Access Protocol (LDAP): The Protocol. Recuperado el 12 de abril de 2012, de http://tools.ietf.org/html/rfc4511

[24] The website of the X.500 Directory standard. (s.f.). Recuperado el 12 de abril de 2012, de http://www.x500standard.com/

[25] Bernal, Mary. (2011). Manejo de los datos. Recuperado el 20 de abril de 2012, de http://marybernal.wordpress.com/2011/05/23/funcionamiento-de-los-dbms/

[26] García, Cristian y otros. (Julio 2010). Portal Cautivo pfSense. Recuperado el 28 de abril de 2012, de http://www.slideshare.net/valericio1/portal-cautivo

[27] Network Working Group. (1996). Hypertext Transfer Protocol -- HTTP/1.0. Recuperado el 08 de mayo de 2012, de http://www.ietf.org/rfc/rfc1945

[28] Network Working Group. (1999). Hypertext Transfer Protocol -- HTTP/1.1. Recuperado el 08 de mayo de 2012, de http://www.ietf.org/rfc/rfc2616

[29] perldoc.perl.org. (s.f.). Perl Programming Documentation. Recuperado el 14 de mayo de 2012, de http://perldoc.perl.org/perl.html

[30] PepperSpot. (2011). Recuperado el 04 de junio de 2012, de http://pepperspot.sourceforge.net/

[31] ChilliSpot. (s.f.). Recuperado el 04 de junio de 2012, de http://www.chillispot.info/

[32] Coova.org. (s.f.). CoovaChilli. Recuperado el 05 de junio de 2012, de http://coova.org/CoovaChilli

[33] IEA Software, Inc. (s.f.). Air Marshal, Captive Portal System. Recuperado el 05 de junio de 2012, de http://www.iea-software.com/products/airmarshal1.cfm

[34] IEA Software, Inc. (2012). Air Marshal, Authentication Gateway. Recuperado el 05 de junio de 2012, de http://www.iea-software.com/docs/airmarshal2/airmarshalv2.pdf

[35] ZeroShell, Net Services. (2012). Router/Firewall Linux. Recuperado el 05 de junio de 2012, de http://www.zeroshell.net/es/

[36] pfSense. (2011). pfSense Project. Recuperado el 05 de junio de 2012, de http://www.pfsense.org/

[37] OpenSplash. (s.f.). Recuperado el 06 de junio de 2012, de http://www.opensplash.org/

[38] m0n0wall. (2012). Recuperado el 06 de junio de 2012, de http://m0n0.ch/wall/

[39] Portless. (2009). About EWRT. Recuperado el 06 de junio de 2012, de http://www.portless.net/ewrt/

[40] HotSpotSystem. (2011). Recuperado el 06 de junio de 2012, de http://www.hotspotsystem.com/

[41] Wifidog. (s.f.). Wifidog a captive portal suite. Recuperado el 08 de junio de 2012, de http://dev.wifidog.org/

[42] Antamedia. (2012). HotSpot Software That is Leading the Industry. Recuperado el 08 de junio de 2012, de http://www.antamedia.com/hotspot/

[43] PatronSoft. (2012). FirstSpot. Recuperado el 08 de junio de 2012, de http://patronsoft.com/firstspot/

[44] National Institute of Standards and Technology. (2008). Guide to General Server Security. Recuperado el 07 de noviembre de 2012, de http://csrc.nist.gov/publications/nistpubs/800-123/SP800-123.pdf

[45] dbGREENSQL. (s.f.). MySQL Security Best Practices (Hardening MySQL Tips). Recuperado el 07 de noviembre de 2012, de http://www.greensql.com/articles/mysql-security-best-practices

Anexos

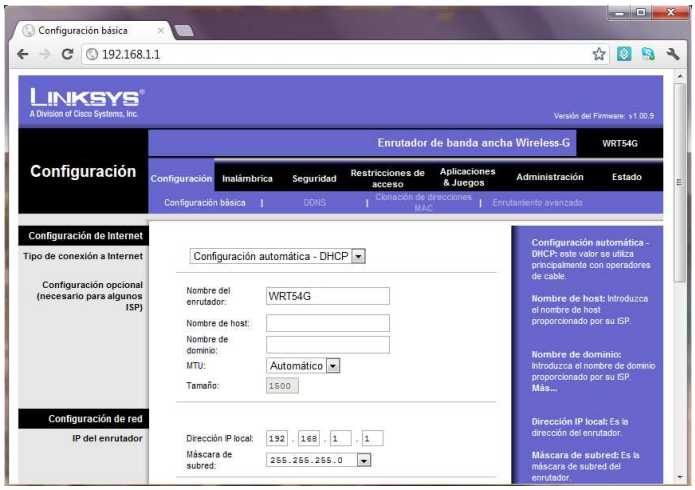
Anexo A. Router inalámbrico Linksys WRT54G Versión 6 como “Bridge”

Actualización de firmware

Para colocar el router Linksys WRT54G en modo bridge lo primero que se tiene que hacer es actualizar el firmware del router, ya que el firmware que trae de fábrica no trae dicha opción.

Para actualizar el firmware se hace lo siguiente:

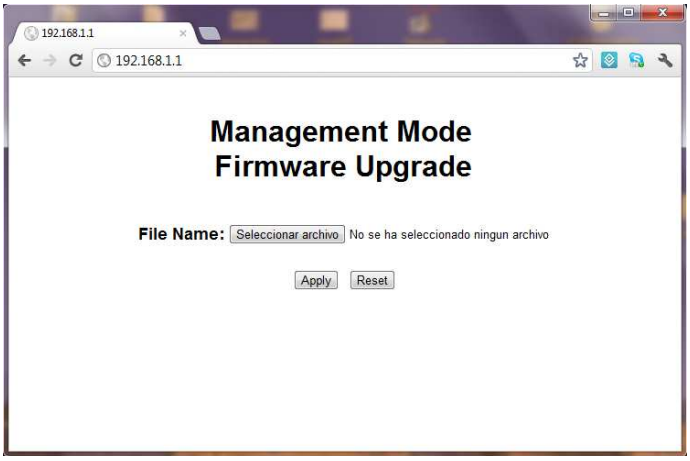
1. Conectar el router inalámbrico via Ethernet a una PC, y configurar la PC para que se comunique con el router, la red que trae por default el router es 192.168.1.0
2. Abrir un navegador web para entrar a la configuración del router inalámbrico, la dirección por default es 192.168.1.1, el usuario es root y la contraseña es admin. Se abrirá una página como la siguiente:



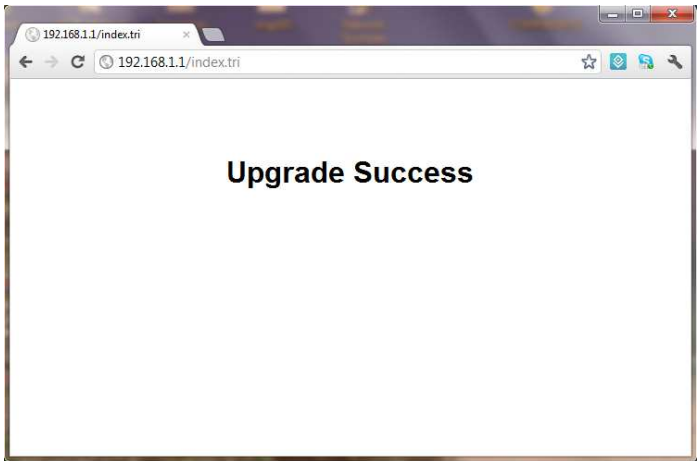
3. Ir a la pestaña de “Administración”, y hacer click en el menú “Actualizar firmware”

4. Descargar el siguiente paquete, linux_prep_wrt54g.bin, se puede descargar de la siguiente url http://www.wrtrouters.com/guides/upgradetolinux/linux_prep_wrt54g.bin
5. En la pantalla de actualizar firmware, seleccionar el archivo descargado en el punto anterior y hacer click en el botón actualizar, la pantalla se pondrá blanca

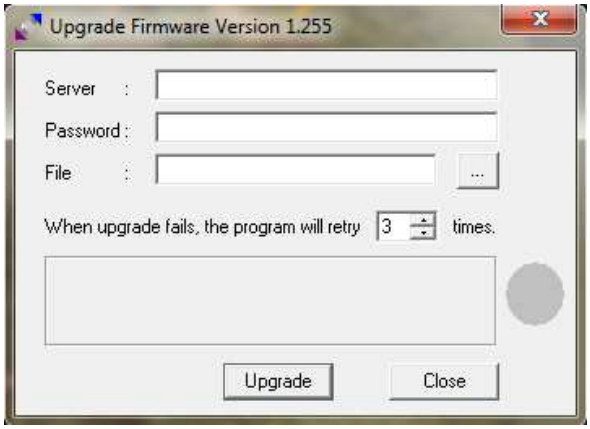
6. Apagar el router y volverlo a encender
7. En el navegador web para acceder a la dirección 192.168.1.1, se mostrará una pantalla como la siguiente



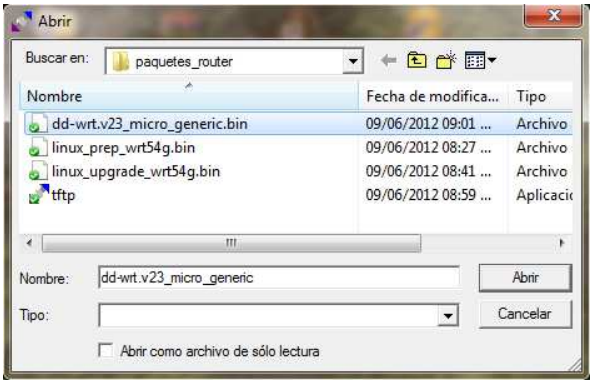
8. Descargar el paquete linux_upgrade_wrt54g.bin, se puede descargar de la siguiente url http://www.wrtrouters.com/guides/upgradetolinux/linux_upgrade_wrt54g.bin
9. Seleccionar el archivo descargado en el punto anterior y hacer click en el botón apply, la pantalla se pondrá blanca

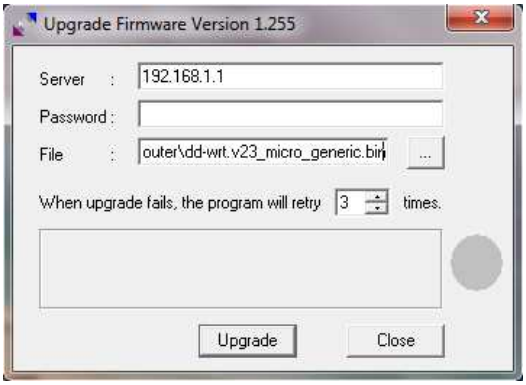


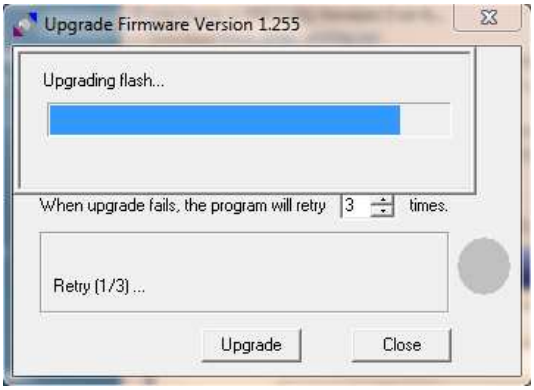
10. Apagar el router y volverlo a encender, el LED de power se quedará parpadeando.
11. Descargar el programa tftp.exe de la siguiente url http://www.dd-wrt.com/routerdb/de/download/Linksys/WRT54G/v6.0/tftp.exe/2236
12. Descargar el paquete dd-wrt.v24_micro_generic.bin, se puede descargar de la siguiente url http://www.dd-wrt.com/routerdb/de/download/Linksys/WRT54G/v6.0/dd-wrt.v24_micro_generic.bin/1959
13. Abrir el cliente de TFTP



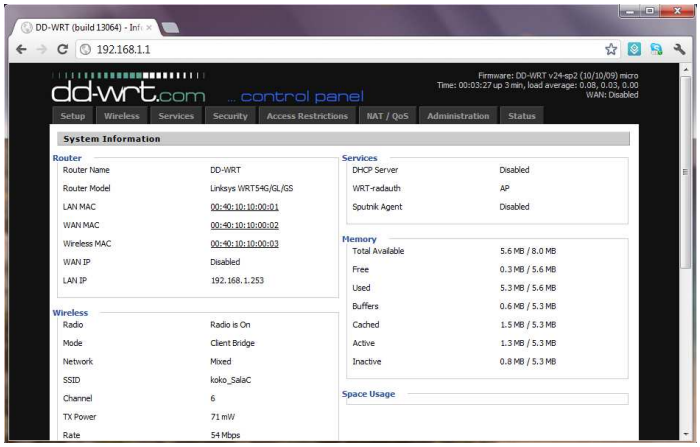
14. Poner los datos que se solicitan y seleccionar el paquete que se descargó en el punto anterior y hacer click en el botón upgrade, el servidor es 192.168.1.1 y no se pone contraseña







15. El router debe de apagarse y volverse a encender
16. Abrir un navegador web y acceder a la dirección 192.168.1.1 y se debe de mostrar la siguiente página



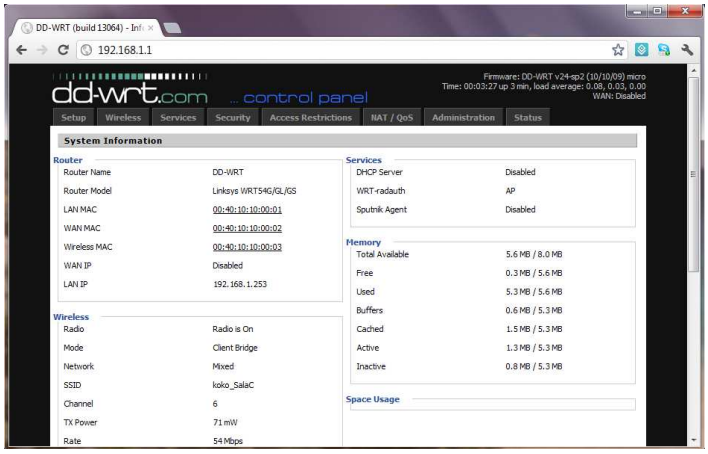
17. Ahora ya se tiene dd-wrt instalado en el router inalámbrico, el cual es un firmware que permite que el router inalámbrico tenga mejores características y funcionalidades

Configuración del router inalámbrico

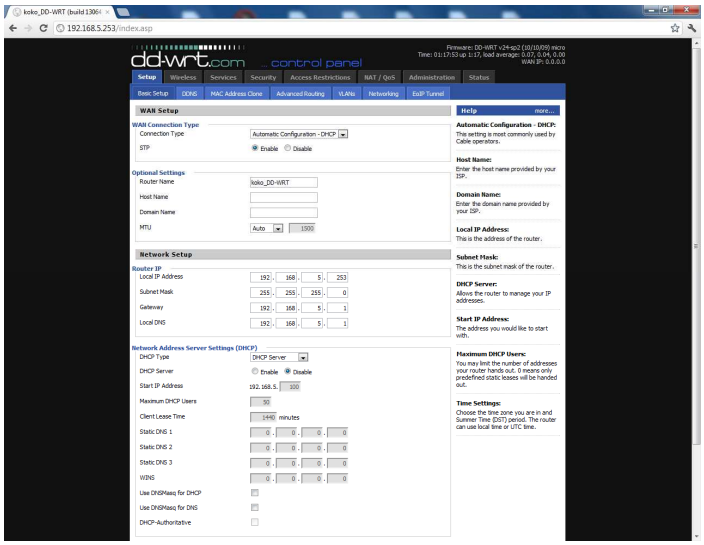
Para configurar el router inalámbrico como bridge, se procede de la siguiente manera:

1. Conectar el router inalámbrico via Ethernet a una PC, y configurar la PC para que se comunique con el router, la red que trae por default el router es 192.168.1.0

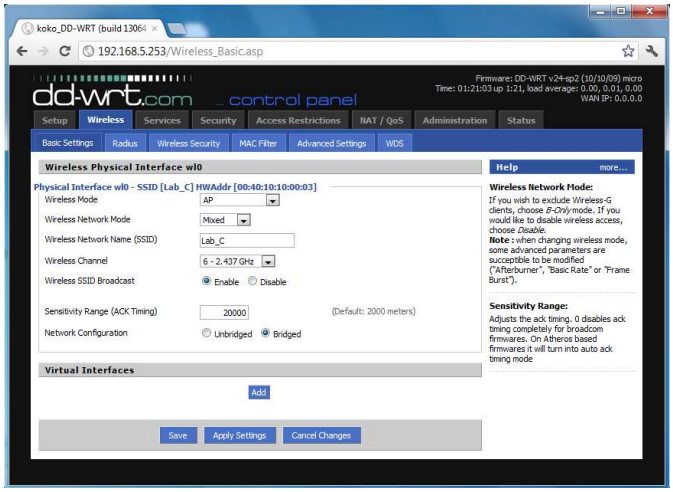
2. Abrir un navegador web para entrar a la configuración del router inalámbrico, la dirección por default es 192.168.1.1, el usuario es root y la contraseña es admin. Se abrirá una página como la siguiente:



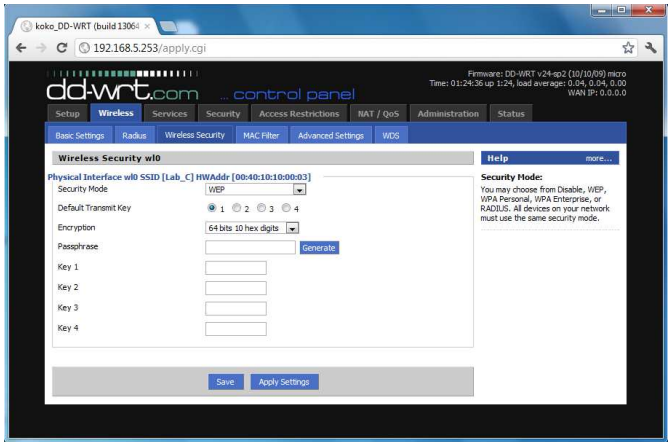
3. Ir a la pestaña setup, en esta pantalla hay varios campos que permiten la configuración del dispositivo, como el nombre, datos de configuración de red, datos del servidor DHCP, etc. En este apartado le indicamos la dirección IP y los datos de conexión a la red, también se deshabilita el servidor DHCP que trae el router inalámbrico y se le puede asignar un nombre nuevo, posterior a la asignación de los datos, se hace click en el botón “save”



4. Ahora, se accede a la pestaña “Wireless”, en esta pestaña existen campos para configurar la red inalámbrica que ofrecerá el router inalámbrico. Configurar el modo de funcionamiento del router inalámbrico, access point, repetidor, bridge, etc. Se selecciona el modo “AP” y se selecciona la opción de “Bridge” en el campo de Network Configuration, también se indica el nombre de la red inalámbrica a crear (SSID)



5. Dentro de la pestaña de “Wireless”, existe otro submenú llamado “Wireless Security”, en este apartad se debe de indicar el tipo de seguridad que tendrá la red inalámbrica, ya sea WPA, WEP, etc.



Para nuestro caso, se dejará la red inalámbrica abierta, es decir, no se utilizará algún modo de seguridad, ya que el portal cautivo se encargará de realizar el proceso de autenticación.

Anexo B. Archivo configuración de NoCat Gateway Server

```
###### gateway.conf -- NoCatAuth Gateway Configuration.
#
# Format of this file is: <Directive> <Value>, one per line. Trailing and leading
# whitespace is ignored. Any line beginning with a punctuation character is
# assumed to be a comment.

###### General settings.
Verbosity       10

##### Gateway application settings.
GatewayName     koko Network
GatewayMode     Captive
GatewayLog      /usr/local/nocat/gw/nocat.log
LoginTimeout    600

###### Open Portal settings.
HomePage        www.fi-b.unam.mx
DocumentRoot    /usr/local/nocat/gw/htdocs
SplashForm      splash.html
StatusForm      status.html

###### Active/Passive Portal settings.
TrustedGroups Any
# Owners rob@nocat.net schuyler@nocat.net
AuthServiceAddr 192.168.5.3
AuthServiceURL  https://$AuthServiceAddr/cgi-bin/login
LogoutURL       https://$AuthServiceAddr/logout.html

### Network Topology
ExternalDevice  eth0
InternalDevice  eth1
LocalNetwork    192.168.5.0/24
DNSAddr 132.248.59.98
AllowedWebHosts www.fi-b.unam.mx
# RouteOnly     1
# IgnoreMAC     1
# MembersOnly   1
# IncludePorts    22 80 443
# ExcludePorts    25
# LogFacility   internal
# SyslogSocket unix
# SyslogOptions cons,pid
# SyslogPriority INFO
# SyslogFacility user
# SyslogIdent NoCat

# ResetCmd      initialize.fw
# PermitCmd     access.fw permit $MAC $IP $Class
# DenyCmd       access.fw deny $MAC $IP $Class
# GatewayPort     5280

PGPKeyPath      /usr/local/nocat/gw/pgp

# GpgvPath      /usr/bin/gpgv
# MessageVerify $GpgvPath --homedir=$PGPKeyPath 2>/dev/null

# MaxMissedARP  2
# IdleTimeout   300

### Fin
```

Anexo C. Archivo configuración de NoCatAuth Server

```
###### authserv.conf -- NoCatAuth Authentication Service Configuration.
#
# Format of this file is: <Directive> <Value>, one per line. Trailing and leading
# whitespace is ignored. Any line beginning with a punctuation character is
# assumed to be a comment.

###### General settings.
Verbosity       10
PGPKeyPath   /usr/local/nocat/authserv/pgp

###### Authservice-specific settings.
HomePage     http://odin.fi-b.unam.mx/salac
DocumentRoot /usr/local/nocat/authserv/htdocs

##### Authservice authentication source.
#DataSource  RADIUS
#DataSource  LDAP
DataSource   DBI

# For mysql support:
Database     dbi:mysql:database=nocat
DB_User      nocat
DB_Passwd    <secreto>


## LDAP support. Requires Net::LDAP & IO::Socket::SSL to be installed from the
      CPAN.
# LDAP_Host         aries.fi-b.unam.mx
# LDAP_Base         ou=people,dc=fi-b,dc=unam,dc=mx
# LDAP_Admin_User   cn=admin,dc=fi-b,dc=unam,dc=mx
# LDAP_Admin_PW           <secreto>
# LDAP_Hash_Passwords     Yes
# LDAP_Search_as_Admin    Yes
# LDAP_Filter       uid

## RADIUS support. Requires Authen::Radius to be installed from the CPAN.
# RADIUS_Host       192.168.5.3
# RADIUS_Secret     <secreto>
# RADIUS_TimeOut    5
# RADIUS_Order      Random

## PAM support. Requires Authen::PAM to be installed from the CPAN.
# PAM_Service       nocat

## Samba support. Requires Authen::Smb to be installed from the CPAN.
# Samba_Domain      MyWorkgroup
# Samba_PDC         MyPrimaryDomainController
# Samba_BDC         MyBackupDomainController

## IMAP support. Requires Net::IMAP::Simple to be installed from the CPAN.
# IMAP_Server       localhost

## NIS support. Requires Net::NIS to be installed from the CPAN.
# DataSource NIS

## Alternately, you can use the Passwd data source.
# UserFile       /usr/local/nocat/authserv/etc/passwd
# GroupUserFile         /usr/local/nocat/authserv/etc/group
# GroupAdminFile    /usr/local/nocat/authserv/etc/groupadm
```

```
###### Auth service user table settings.

UserTable    member
UserIDField  login
UserPasswdField pass
UserAuthField   status
UserStampField      created

GroupTable      network
GroupIDField    network
GroupAdminField admin

####### Auth service web application settings.
MinPasswdLength     6

# GpgPath    /usr/bin/gpg
# MessageSign$GpgPath --clearsign --homedir=$PGPKeyPath -o-

LocalGateway    192.168.5.1

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

###### Auth service user messages. Should be self-explanatory.

LoginGreeting   Bienvenido a la red del Laboratorio Sala C!
LoginMissing    Debes de llenar todos los campos!
LoginBadUser    Usuario incorrecto, verifique sus datos
LoginBadPass Password incorrecto, verifique sus datos
LoginBadStatus  Lo sentimos, usted no es un usuario registrado

###### Fin.
```

Anexo D. Script para inicialización del servicio NoCat Gateway

```
#!/bin/sh

# Simple init script for starting
# the gateway service at boot time.
#
# Either add a call to it in /etc/rc.d/rc.local,
# or copy it to /etc/rc.d/init.d and symlink it
# to your runlevel.
#
# Edit the following line if you installed the
# nocat software somewhere else.
#
NC=/usr/local/nocat/gw

export PERL5LIB=$NC/lib:$PERL5LIB
export NOCAT=$NC/nocat.conf

case "$1" in
  start)
      echo "Limpiando anteriores reglas..."
      /sbin/iptables -F
      /sbin/iptables -t nat -F
      echo "Starting the NoCat gateway..."
      $NC/bin/gateway
      ;;
  stop)
      echo "Stopping the NoCat gateway..."
      killall gateway
      echo "Limpiando reglas generadas..."
      /sbin/iptables -F
      /sbin/iptables -t nat -F
      ;;
  restart)
      $0 stop
      sleep 1
      $0 start
      ;;
  *)
      echo "Usage: $0 {start|stop|restart}"
      exit 1
esac

#
# End
#
```

Anexo E. Archivos HTML para las vistas de autenticación de usuarios

- **login.html**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Login Sala C</title>
    <link rel="shortcut icon" href="/favicon.ico">
    <link rel="stylesheet" type="text/css" href="http://192.168.5.3/style.css"/>
</head>
<body>
    <form id="login-form" action="$CGI" method="post">
       <fieldset>
              <legend>Login</legend>

              <label for="login">Usuario:</label>
               <input type="text" id="login" name="user" value="$user"/>

              <div class="clear"></div>

              <label for="password">Contrase&ntilde;a:</label>
              <input type="password" id="password" name="pass" value="$pass"/>

              <div class="clear"></div>
              <div class="clear"></div>

              <br/><br/>

              <input type="image" style="margin: -20px 0 0 287px;"
       src="http://192.168.5.3/images/login.gif" class="button" name="mode_login"
       value="Login"/>

       </fieldset>

       <input type="hidden" name="mac" value="$mac">
       <input type="hidden" name="token" value="$token">
       <input type="hidden" name="redirect" value="$redirect">
       <input type="hidden" name="gateway" value="$gateway">
       <input type="hidden" name="timeout" value="$timeout">

       </form>
       <div class="texto">$Message</div>
</body>

</html>
```

- **login_ok.html**

```
<html>
<head>
    <title>Red Sala C</title>
    <meta http-equiv="Refresh" content="$redirect" />
    <link rel="stylesheet" type="text/css" href="http://192.168.5.3/style2.css"/>
</head>

<script language="JavaScript">
    window.open( "$popup", "Login Sala C", "width=250,height=180,scrollbars=no"
);
</script>
```

```
<body bgcolor="#FFFFFF">
       <p> <a href="http://odin.fi-b.unam.mx/salac/">
              <img src="http://192.168.5.3/images/logoLabC.png"></a>
       </p>
       <div class="texto">
              Bienvenido al Laboratorio de Computaci&oacute;n Sala C, $user!
       </div>
       <div class="texto2">
       Te has autenticado correctamente, seras redireccionado en 5 segundos. Si no
       es asi click <a href="$redirect">aqu&iacute;</a> para continuar
       </div>
</body>
</html>
```

- **login-no-skip.html**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
     <title>Login Sala C</title>
     <link rel="shortcut icon" href="/favicon.ico">
     <link rel="stylesheet" type="text/css" href="http://192.168.5.3/style.css"/>
</head>
<body>
     <form id="login-form" action="$CGI" method="post">
       <fieldset>
              <legend>Login</legend>

              <label for="login">Usuario:</label>
              <input type="text" id="login" name="user" value="$user"/>
              <div class="clear"></div>
              <label for="password">Contrase&ntilde;a:</label>
              <input type="password" id="password" name="pass" value="$pass"/>

              <div class="clear"></div>
              <div class="clear"></div>

              <br/><br/>

              <input type="image" style="margin: -20px 0 0 287px;" class="button"
              src="http://192.168.5.3/images/login.gif" name="mode_login"
              value="Login"/>

       </fieldset>

       <input type="hidden" name="mac" value="$mac">
       <input type="hidden" name="token" value="$token">
       <input type="hidden" name="redirect" value="$redirect">
       <input type="hidden" name="gateway" value="$gateway">
       <input type="hidden" name="timeout" value="$timeout">

       </form>
       <br/><br/><br/>
       <div class="texto">$Message</div>
</body>
</html>
```

- **logout.html**

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Red Sala C</title>
    <link rel="stylesheet" type="text/css" href="http://192.168.5.3/style2.css"/>
</head>
<body>
      <p><a href="http://odin.fi-b.unam.mx/salac/" target="_blank">
      <img src="http://192.168.5.3/images/logoLabC.png" border="0"></a> </p>
      <div class="clear"></div>
      <div class="texto">
            Gracias por usar la red de la Sala C!
      </div>
      <div class="texto2">
            Hasta Pronto!
      </div>
</body>
</html>
```

- **renew.html**

```
<html>
<head>
    <title>Red Sala C</title>
    <meta http-equiv="Refresh" content="$redirect" />
    <link rel="stylesheet" type="text/css" href="http://192.168.5.3/style2.css"/>
</head>

<body>
<p><a href="http://odin.fi-b.unam.mx/salac/" target="_blank">
<img src="http://192.168.5.3/images/logoLabC.png" border="0"></a> </p>

<div class="clear"></div>
<div class="texto">
      Manten esta ventana abierta y tu autenticaci&oacute;n  ser&aacute; renovada
      autom&aacute;ticamente en $timeout segundos.
</div>
<div class="texto2">
      Esta ventana puede ser minimizada.
</div>
      <form method="post" action="$logout">
      <label>Finalizar Sesion</label>
      <input type="image" src=http://192.168.5.3/images/logout.gif style="margin:
      50px 0 0 50px;" class="button" name="mode_logout" value="Logout"/>
      </form>
</body>
</html>
```

- **renew_pasv.html**

```
<html>
<head>
    <title>Red Sala C</title>
    <meta http-equiv="Refresh" content="$redirect" />
    <link rel="stylesheet" type="text/css" href="http://192.168.5.3/style2.css"/>
</head>

<script language="JavaScript">
      setTimeout( "document.RenewalForm.submit()", $timeout * 1000 );
</script>
<body>

<p><a href="http://odin.fi-b.unam.mx/salac/" target="_blank">
<img src="http://192.168.5.3/images/logoLabC.png" border="0"></a> </p>

<div class="clear"></div>
<div class="texto">
      Manten esta ventana abierta y tu autenticaci&oacute;n  ser&aacute; renovada
      autom&aacute;ticamente en $timeout segundos.
</div>
<div class="texto2">
       Esta ventana puede ser minimizada.
</div>
      <form method="post" action="$logout">
      <label>Finalizar Sesion</label>
      <input type="image" src=http://192.168.5.3/images/logout.gif style="margin:
      50px 0 0 50px;" class="button" name="mode_logout" value="Logout"/>
       </form>

      <form name="RenewalForm" action="$CGI" method="post">
            <input type="hidden" name="mode" value="renew">
            <input type="hidden" name="user" value="$user">
            <input type="hidden" name="pass" value="$pass">
            <input type="hidden" name="mac" value="$mac">
            <input type="hidden" name="token" value="$token">
            <input type="hidden" name="gateway" value="$gateway">
            <input type="hidden" name="timeout" value="$timeout">
      </form>
</body>
</html>
```

- **splash.html**

```
<html>
<head>
    <title>Red Sala C</title>
    <link rel="stylesheet" type="text/css" href="http://192.168.5.3/style2.css"/>
</head>
<body>

<p><a href="http://odin.fi-b.unam.mx/salac/" target="_blank">
<img src="http://192.168.5.3/images/logoLabC.png" border="0"></a> </p>
<br/>
<div class="clear"></div>

      <form method="POST" action="$action">
            <div class="texto">
            Bienvenido a $GatewayName
            </div>
            <input type="image" src=http://192.168.5.3/images/login.gif
            style="margin: 15px 10px 25px 110px;" class="button"
            name="mode_login" value="Login"/>

            <div class="texto2">
            Actualmente hay $ConnectionCount usuarios conectados<br/>
            La &uacute;ltima conexi&oacute;n fue: $LastConnectionTime
            </div>

            <input type="hidden" name="redirect" value="$redirect">
      </form>
</body>
</html>
```

- **style.css**

```
*            { margin: 0; padding: 0; }
body         { font-family: "Tw Cen MT"; font-weight:bold;
             background: url(http://192.168.5.3/images/login-page4.png)
             top center no-repeat #c4c4c4; }

.clear       { clear: both; }

.texto       { text-align: center;  font-size: 18px; font-style: bond;
             margin: 10px 0 0 0; }

form         { width: 407px; margin: 168px auto 0; }

legend       { display: none; }

fieldset     { border: 0; }

label        { width: 115px; text-align: right; float: left; margin: 0 5px 4px 0;
             padding: 9px 0 0 0; font-size: 18px; font-style: bond; }

input        { width: 222px; display: block; padding: 5px; margin: 0 0 6px 0;
             font-size: 18px; color: #3a3a3a; font-family: Georgia, serif;}

input[type=checkbox]{ width: 20px; margin: 0; display: inline-block; }

.button      { background:;  border: 1px solid  # CCC;font-family: "Tw Cen MT";
             font-weight:bold; -moz-border-radius: 5px; padding: 5px;
             color: #666; font-weight: bold; -webkit-border-radius: 5px;
             font-size: 18px;  width: 70px; }

.button:hover       { background: ; color: #666;font-family: "Tw Cen MT";
                    font-weight:bold; }
```

- **style2.css**

```
*            { margin: 0; padding: 0; }
body         { font-family:“Tw Cen MT"; font-weight:bold; background: #c4c4c4; }

.clear       { clear: both; }

.texto       { text-align: left;  font-size: 20px; font-style: bond;
             margin: 10px 20px;}

.texto2      { text-align: left;  font-size: 16px; font-style: bond;
             margin: 10px 20px;}

form         { margin: auto auto 0; }

label        { width: 150px; text-align: right; float: left; margin: 0 5px 4px 0;
             padding: 9px 0 0 0; font-size: 18px; font-style: bond; }

input        { width: 222px; display: block; padding: 5px; margin: 0 0 6px 0;
             font-size: 18px;color: #3a3a3a; font-family: Georgia, serif;}

.button      { background:;  border: 1px solid  # CCC;font-family: "Tw Cen MT";
             font-weight:bold; -moz-border-radius: 5px; padding: 5px;
             color: #666; font-weight: bold; -webkit-border-radius: 5px;
             font-size: 18px;  width: 70px; }

.button:hover       { background: ; color: #666;font-family: "Tw Cen MT";
                    font-weight:bold; }
```