ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA,

REDES Y COMUNICACIÓN DE DATOS

PROYECTO DE GRADO PARA OBTENCIÓN DEL TÍTULO EN

INGENIERÍA ELECTRÓNICA

AUTORES:

ASTUDILLO CABRERA JAIME JAVIER

TROYA ESTRELLA ANDRÉS SEBASTIÁN

TEMA: "DETECCIÓN DE VULNERABILIDADES EN REDES INALÁMBRICAS 802.11i, MEDIANTE EL ANÁLISIS DE TRÁFICO DE LA CAPA DE ENLACE"

DIRECTOR: ING. CARLOS ROMERO

CODIRECTOR: ING. FABIÁN SÁENZ

SANGOLQUÍ, JULIO DE 2014

“UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE”

INGENIERÍA EN ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS

CERTIFICADO

Ing. Carlos Romero

Ing. Fabián Sáenz

CERTIFICAN

Que el trabajo titulado “Detección de vulnerabilidades en redes inalámbricas 802.11i, mediante el análisis de tráfico de la capa de enlace.”, realizado por Jaime Javier Astudillo Cabrera y Andrés Sebastián Troya Estrella, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a Jaime Javier Astudillo Cabrera y Andrés Sebastián Troya Estrella que lo entreguen al Doctor Nikolai Espinosa, en su calidad de Coordinador de la Carrera.

Sangolquí, 21 de julio de 2014

_____	_____
Ing. Carlos Romero	Ing. Fabián Sáenz
DIRECTOR	CODIRECTOR

“UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE”

INGENIERÍA EN ELECTRÓNICA, REDES Y COMUNICACIÓN DE DATOS

DECLARACIÓN DE RESPONSABILIDAD

Jaime Javier Astudillo Cabrera
Andrés Sebastián Troya Estrella

DECLARAMOS QUE:

El proyecto de grado denominado **“DETECCIÓN DE VULNERABILIDADES EN REDES INALÁMBRICAS 802.11i, MEDIANTE EL ANÁLISIS DE TRÁFICO DE LA CAPA DE ENLACE”,** ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 21 de julio de 2014

Jaime Javier Astudillo Cabrera	Andrés Sebastián Troya Estrella

“UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE”

INGENIERÍA EN ELÉCTRICA, REDES Y COMUNICACIÓN DE DATOS

AUTORIZACIÓN

Nosotros, Astudillo Cabrera Jaime Javier y Troya Estrella Andrés Sebastián

Autorizamos a la Universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la institución del trabajo **“DETECCIÓN DE VULNERABILIDADES EN REDES INALÁMBRICAS 802.11i, MEDIANTE EL ANÁLISIS DE TRÁFICO DE LA CAPA DE ENLACE.”**, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, 21 de julio de 2014

Jaime Javier Astudillo Cabrera	Andrés Sebastián Troya Estrella

DEDICATORIA

Este Proyecto lo dedico a mi madre Eliana, y a mi hermana Daniela, quienes con su amor, alegría, paciencia y coraje, me han dado la fuerza para superar todos los obstáculos y la fe para alcanzar mis metas. A mis Abuelos Celín y Carlos, que no me pueden acompañar en este momento por su pronta partida, pero su espíritu y cariño siempre permanece en mi corazón.

Jaime Javier Astudillo Cabrera

Dedico este esfuerzo a Dios porque de el viene la sabiduría, inteligencia, inspiración y fuerza para superar cualquier obstáculo que se presente. A mis padres Wladimir y María Elena a quienes amo con todo mi corazón. Ofrezco también esta dedicatoria con mucho cariño a mi prometida Sophie por caminar siempre a mi lado. Por último, pero no menos importante dedico este logro a mis hermanos, abuelitos, tíos, primos y amigos, porque de una u otra manera han sabido transferirme sabios concejos, buenos valores y ejemplos que me transforman permanentemente en una mejor persona para poder ser instrumento de bendición a muchos.

Andrés Sebastián Troya Estrella

AGRADECIMIENTO

Primero agradecer a Dios por brindarme la posibilidad de vivir y cobijarme con su protección a diario. A mi madre Eliana por su abnegada labor y su infinito amor, A mi Padre Jaime por brindarme su protección, su apoyo, y por sus infinitas enseñanzas, a los dos por darme la oportunidad de alcanzar esta triunfo, y darme esa gran herencia que es la educación, .

Agradezco de manera especial a mis abuelitas Enma, y Fabiola, mujeres luchadoras, fuertes, cuyos concejos y regaños llenos experiencia y afecto me han dado la sabiduría para seguir adelante. A mis profesores, por guiar mi camino, fundamentado en un crecimiento tanto personal como profesional, al infundir en mi sus conocimientos y valores. Finalmente a mis amigos y compañeros, quienes han dejado huella en este largo trayecto.

Jaime Javier Astudillo Cabrera

En primer lugar agradezco a Dios por su favor y bendición sobre mi vida, cada logro se lo dedico a el con todo mi fervor para su honra y gloria. En segundo lugar, a mi familia por su profundo amor, apoyo incondicional, aliento y motivación para alcanzar mis objetivos. Finalmente a mi amigo y compañero de tesis por toda su dedicación y sacrificio para cumplir con esta meta que se ve reflejada en nuestro trabajo de investigación.

Andrés Sebastián Troya Estrella

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

En los últimos años se ha evidenciado el aumento del uso de redes inalámbricas debido a las características y ventajas que ofrecen sobre las redes cableadas; entre ellas: la movilidad de los usuarios manteniendo conectividad constante a la red local, la facilidad de incrementar el tamaño de la red y la configuración de diferentes topologías entre ellas el estándar *802.11i*.

Mediante la implementación de un escenario de prueba, se estudia el estándar *802.11i* por medio de la captura de las tramas de Administración y Control. Se utiliza un analizador de paquetes como *AirPcap Nx* o Kismet para interceptar información de la red inalámbrica que pueda ser analizada en Wireshark. Una vez obtenida esta información, se realizan ataques éticos utilizando software especializado como BT5r3 (BackTrack*)* para simular problemas en la seguridad como denegación de servicio *(DoS)*.

PALABRAS CLAVE

1. 802.11i
2. KISMET
3. AIRPCAP NX
4. DOS
5. WIRESHARK

ABSTRACT

In recent years there has been an increasing use of wireless networks because of the characteristics and advantages over wired networks; including: the mobility of users maintaining constant connectivity to the local network, allowing great flexibility, easy increase in the size of the network and the configuration of different topologies to meet the needs that satisfy connectivity.

Today, wireless networks suffer from certain vulnerabilities that leave exposed a network intrusions. The *IEEE* published several encrypted security mechanism to prevent unauthorized intrusion of external agents such as *WEP* and *WPA*, but they had several shortcomings and despite its implementation, the networks were still being violated; Thus, the *IEEE* began developing a new safety standard known as *802.11i*, which would provide enough security to *WLANs*.

Through the implementation of a test scenario, we aim to analyze the vulnerabilities in the *802.11i* standard. Using the *AirPcap Nx* adapter cards, we seek to capture the management and control packets from an *AirPcap* session for analysis in Wireshark. Once this information is obtained, ethical hacking attacks are conduct using specialized software such as BT5r3 (BackTrack) to demonstrate security breach problems under this scheme.

CAPÍTULO I

FUNDAMENTO TEÓRICO

1.1. Introducción

La implantación de redes inalámbricas se ha vuelto popular en los últimos años, tanto en el ámbito personal, como en el corporativo, es así que se han utilizado para ampliar la cobertura y suministrar servicios a los espacios públicos. La expansión de las aplicaciones para redes inalámbricas en dispositivos móviles, computadoras portátiles, impulsarán la utilización de este tipo de redes y en la mayoría de los casos han reemplazando a las redes cableadas. Para todo los ambientes en que las redes pueden ser instaladas, cabe la posibilidad de que los usuarios se conecten a una red segura e insegura; en la segunda, los usuarios no deseados accedan a la red mediante la generación de ataques, pueden violentarla y afectar el desempeño de la red inalámbrica.

Las redes de área local inalámbricas pueden ser consideradas mucho más allá de ser un sustituto de las redes cableadas, como una extensión de las mismas, ya que por medio de estas redes (inalámbricas) se puede tener acceso a los servicios que se ofrecen en un entorno de red. Existen varias ventajas que no ofrecen los sistemas cableados: la principal, es ofrecer al usuario gran movilidad sin perder conectividad; entre otras, la facilidad de instalación.

Considerando la proliferación de redes inalámbricas, uno de los temas más destacados a considerar es el de la seguridad, cuyo objetivo principal es aislar los actos deseables, y la prevención de actos potencialmente

perjudiciales para la red, de forma que si se producen, hagan el menor efecto dañino posible. Entre las actividades más destacadas que se pueden efectuar para proteger estas redes están: la identificación y autenticación de usuarios, detección de intrusos de la red, análisis de riegos y clasificación de datos presentes en la red. Hay que poner mayor énfasis en los factores que inciden dentro del comportamiento de la red, mediante el estudio de diferentes escenarios y del planteamiento de soluciones rápidas y efectivas para asegurar la integridad de los datos y confidencialidad de la red.

1.2. Justificación

Cada día se maneja un volumen mayor de información digital, a través de redes de área local inalámbricas *(WLAN)*, las compañías confían cada vez más en la tecnología para brindar seguridad a sus redes, es por eso que se ha desarrollado tecnologías como: *WPA*, *WPA2* y el estándar *802.11i*. Las mismas definen un protocolo de seguridad inalámbrico en respuesta a serias debilidades descubiertas en previas generaciones de la encriptación de *Wi-Fi*, tal como: es la privacidad equivalente a conexión por cable *(WEP)*. *WPA* y *WPA2* operan en dos modos: Personal, el cual utiliza *PSK*; Enterprise, el cual toma ventaja del servidor *RADIUS*. Los dos modos de acceso *Wi-Fi* protegido, son vulnerables al ataque *"Hole196"*, incluso la red más segura *802.1x* con autenticación y *AES*. Esta vulnerabilidad permite, entre otras cosas, a cualquiera con acceso a la red inalámbrica, desencriptar y robar información confidencial de cualquier otro cliente que se encuentre conectado a la misma red, por medio de la inyección de tráfico malicioso que compromete los dispositivos autorizados. En la actualidad *WPA2* es la forma de encriptación y autentificación más sofisticada y fuerte de todas las implementadas y estandarizadas, sin embargo, no es perfecta o exenta de errores en su protocolo como se ha mencionado anteriormente.

Este proyecto es importante, ya que analizará las vulnerabilidades en el estándar *802.11i*. Los principales beneficiarios de esta investigación serían:

corporaciones, gobiernos, instituciones académicas, etc. Los establecimientos que utilizan este estándar, considerarían estar protegidas por las medidas internas establecidas por cada una de ellas, asumiendo que ningún usuario mal intencionado podría vulnerar su red. Es por ello que es de vital importancia tener presente cuales son las falencias de la seguridad en esta tecnología, logrando así, la preservación de la confidencialidad de la información en formato digital.

1.3. Objetivos

1.3.1. General

Determinar las vulnerabilidades de una red inalámbrica que utiliza el estándar *802.11i*, mediante el análisis del tráfico en la capa de enlace, de los paquetes de Control, Administración y Datos.

1.3.2. Específicos

- Buscar información sobre las características y utilidades que ofrecen las tarjetas *AirPcap Nx Adapter* para la captura de tráfico de la red inalámbrica a implementarse en el escenario de prueba.

- Establecer un escenario de prueba para redes inalámbricas *WiFi 802.11i*, con soporte *Enterprise* y servidor *RADIUS*, con el fin de obtener información del funcionamiento y desempeño de esta red, en un comportamiento normal e intrusivo.

- Elaborar un análisis exhaustivo de los campos en las cabeceras de los paquetes de control, administración y datos, para poder determinar cuáles contienen información valiosa sobre posibles vulnerabilidades dentro del estándar *802.11i*.

- Evaluar los riesgos en la seguridad del escenario de prueba, que permitan proponer recomendaciones para disminuir las posibles intrusiones o ataques, que las redes actuales bajo este estándar puedan ser víctimas.

1.4. Redes de área local *wireless (WLAN)*

1.4.1. Características

Es necesario, previo a la determinación de las características de este protocolo señalar que el estándar *802.11* cuenta con diferentes especificaciones, y cada una con sus particularidades propias. El grupo de especificaciones definidas para el estándar *802.11* fue desarrollado por el *IEEE* (Instituto de Ingenieros Eléctricos y Electrónicos), con el fin de establecer comunicación, entre una estación inalámbrica con un cliente, o, a su vez entre dos clientes inalámbricos.

En la actualidad existen cuatro especificaciones *802.11a*, *802.11b*, *802.11g*, *802.11n*, todas ellas utilizan el protocolo Ethernet. La primera norma aprobada fue *802.11a*, trabaja a velocidades máximas de 54 Mbps dependiendo de la distancia, se propaga en el espectro de los 5 GHz, con modulación *OFDM*, tiene 12 canales no solapados, 8 para red inalámbrica, y 4 para conexiones punto a punto; la segunda norma 802.11b se propaga en el espectro de los 2,4 GHz banda sin licencia, con velocidad máxima de 11 Mbps, con modulación *DSSS*, cabe señalar que la velocidad en práctica , considerando las interferencia, disminuirían hasta 5,9 Mbps en *TCP* y 7,1 Mbps en *UDP*; la tercera especificación. 802.11g, que se propaga sobre frecuencias de 2,4 GHz sin necesidad de licencia, y a cortas distancias con velocidades máximas de 54 Mbps, integrando dos tipos de modulación *DSSS*, y *OFDM*, por lo que es compatible con la norma *802.11b*; y, finalmente 802.11n, es el nuevo y revolucionario sistema de redes *wireless*, que traen un considerable aumento del ancho de banda disponible: Estableciendo una comparación con los estándares antes mencionados en

los cuales se alcanzaba tasas de transferencia de datos de hasta 22 Mbps (Reales en el mayor de los casos comprobados), las redes que trabajan sobre el estándar 802.11n, alcanzan un *troughput* de 120 a 130 Mbps (Reales), es decir supera las capacidades de las redes de cable *Fast-Ethernet*, comúnmente utilizados en lugares de trabajo.

1.4.2. Métodos de Autenticación

Las redes inalámbricas siempre se encuentran sujetas a vulnerabilidades y ataques por el hecho de la emisión de señales radioeléctricas en el aire, es así que durante el paso de tiempo se ha implementado nuevos estándares, y protocolos para reforzar esta debilidad. Además de establecer estándares como *802.11i* que presenta un escenario robusto para la validación de usuarios por medio de un servidor, se crearon también métodos de autenticación que refuerzan este estándar mediante el cifrado de información necesaria al momento de realizar la autenticación. Los métodos *EAP* han sido desarrollados mejorando cada vez más la seguridad de la información, y reducir la incidencia de los ataques, pero de igual forma cada método presenta deficiencias con respecto a las características que ofrece, a continuación se realiza una corta descripción de cada uno de ellos.

- **EAP-MD5:** es uno de los métodos más conocidos, proporciona un nivel básico de seguridad, mediante un algoritmo de cifrado, es usado para la combinación de un elemento secreto y un desafío hacia el suplicante para comprobar si el suplicante conoce el elemento secreto. No propone una autenticación mutua entre suplicante y autenticador, pese a que no es fácil conseguir un paquete de este tipo, uno capturado puede obtener la información necesaria como obtener el desafío y la respuesta, donde por medio de un ataque de diccionario pueda coincidir con la identidad del suplicante con lo que se tendría todas las credenciales para ingresar a la red.

- **EAP-LEAP:** Es un método desarrollado por CISCO para el uso en dispositivos en el estándar 802.11, este método utiliza una contraseña para inicio de sesión como un secreto compartido entre suplicante y autenticador, es decir ofrece una autenticación mutua, esta característica no permite ataques de *MITM*, ya que se cifran los intercambios de datos con claves *WEP* aleatorias, las claves de sesión son exclusivos de los usuarios y no se comparten entre ellos. Si bien es cierto elimina los ataques de *MITM* pero sigue siendo vulnerable a ataques de diccionario por el hecho de utilizar en MS CHAPV2, para proteger las credenciales del usuario.

- **EAP-TLS:** El método más recomendable y atractivo para la protección de una red *WLAN*, ya que elimina la mayoría de los ataques conocidos como *MITM*, ataques de diccionario, ya que no se basa en intercambio de contraseñas entre el suplicante y autenticador, sino en una autenticación muta mediante el intercambio de certificados digitales, también genera y distribuye claves de cifrado basado en usuarios y sesiones para establecer las conexiones, es por ello que las credenciales del usuario jamás son reveladas. En el proceso de intercambio de certificados en donde el proceso inicia en el que suplicante envía su identidad hacia el autenticador y este encapsula la identidad y la encapsula en un mensaje *RADIUS* para que sea validada por el servidor, de ser correcta el servidor envía un certificado hacia el autenticador,, quien al recibir el mensaje lo envía hacia el suplicante, el suplicante valida la información del certificado, y envía su certificado hacia el servidor en donde adjunta la contraseña secreta, el autenticador toma el mensaje lo vuelve a encapsular en un mensaje *RADIUS* y lo envía hacia el servidor, quien valida la información y de ser correcta responde con un mensaje *RADIUS Access-Accept*, hacia el autenticador y a su vez al suplicante para tener accesos a la red. Siendo el intercambio de certificados su principal característica de seguridad frente a los ataques y vulnerabilidades, esta a su vez se convierte en su primera desventaja ya que la distribución e instalación de los certificados de los usuarios se convierte en una tarea dificultosa en redes de media o gran

escala, por el proceso que debe llevarse, de no ser así el usuario queda vulnerable a los ataques antes mencionados, por el hecho de que se estaría establecido una conexión de otro tipo y ya no *EAP–TLS*, en los siguiente capítulos se describirá a mayor detalle el funcionamiento de este método así, como el proceso de implementación.

- **EAP-TTLS:** Considerado una extensión de *EAP-TLS*, donde el proceso de autenticación consiste en dos fases. En la primera fase el autenticador utiliza el certificado para validar al suplicante, la diferencia con *TLS* es que solo se requiere el certificado del servidor, posterior a esto se establece un túnel seguro donde se intercambia información para realizar la autenticación del usuario por el túnel seguro establecido, para la segunda fase se debe considerar que *EAP-TTLS* mantiene el oculto la identidad anónima del suplicante para establecer el túnel en la primera fase, más no la identidad del suplicante para esta fase, en esta fase se permite el uso de protocolos basado en contraseñas heredadas en base de datos, protegiendo así la seguridad de estos protocolos contra ataques de *MITM*.

- **EAP-PEAP:** Mantiene el mismo principio que *EAP-TLS* y *EAP-TTLS,* como el intercambio de información para autenticación, envuelta en un túnel seguro, donde protege de varios ataques, la diferencia con *EAP-TTLS*, es que este método solo puede utilizar *EAP* para la fase dos.

A continuación se presenta una tabla que resume las características de los métodos antes descritos.

Tabla. 1

Características de métodos EAP

	EAP-MD5	EAP-LEAP	EAP-TLS	EAP-TTLS	EAP-PEAP
Autenticación de servidor	Ninguna	Contraseña	Certificado público	Certificado público	Certificado público
Autenticación de suplicante	Contraseña	Contraseña	Certificado público	MSCHAP(v2)	EAP
Generación de llave dinámica	No	Si	Si	Si	Si
Facilidad de despliegue	Fácil	Difícil	Difícil	Moderado	Moderado
Rendimiento general	Bajo	Moderado	Bueno	Bueno	Bueno

1.4.3. Configuraciones

- **Red AD HOC**

La Universalización, el crecimiento de los dispositivos portátiles y la difusión de tecnología por medio de medios inalámbricos, han permitido la creación y la innovación de nuevas configuraciones, como son las redes *wireless AD HOC*, las mismas que representan una solución viable, para aquellos casos en los que no es posible contar con una red que contenga infraestructura de comunicación fija.

Las redes *AD HOC* son un tipo especial de redes, pues no cuentan con control central ni sin conexión a una red de comunicación mundial. Este tipo de redes, las que se les podría considerar como de infraestructura o control, son aquellas formadas por los nodos que participan en la red, las ventajas más llamativas que presenta son: intrínseca facilidad y velocidad de instalación por lo que resulta más versátil, tomando en cuenta que cada nodo que pertenece a la red creada, tiene la capacidad de un *router*. La principal característica es la movilidad de los nodos que se encuentran conectados mediante un enlace, que a su vez se vuelve inestable, dependiendo de la seguridad y dispositivos que conforman la red.

Citando un par de aplicaciones en donde se utiliza redes *Ad-Hoc* son: Transferencia de datos, archivos, imágenes, convenciones y análisis de datos en terrenos catastróficos.

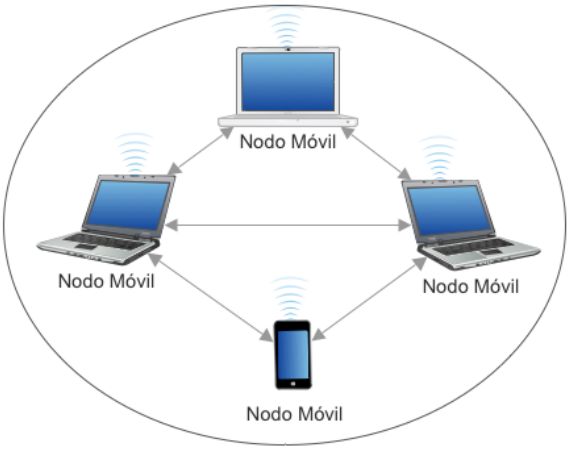


Figura. 1 Red Ad – Hoc

- **Red Infraestructura**

El modo infraestructura tiene una gran diferencia con respecto a la configuración anterior: las estaciones se conectan a un punto de acceso a través de un enlace inalámbrico, considerando que el sistema tiene una topología definida, la movilidad de los equipos no afecta la estabilidad del enlace.

Todas las configuraciones que se encuentran formadas por el punto de acceso y las estaciones ubicadas dentro del área de cobertura, forman una célula, nombrada conjunto de servicios básicos o *BSS*. Cada célula tiene un identificador propio *BSSID*, el cual corresponde a la dirección MAC del punto de acceso, o de los equipos dentro de la célula.

Este tipo de configuración de red es la más usada en la actualidad, y se puede considerar que funciona como una red tipo cliente-servidor, donde los clientes suelen ser los ordenadores personales que se conectan al servidor,

llamado punto de acceso en este caso. Por ende, cuando una estación quiere enviar algún tipo de mensaje hacia otra, la envía al punto de acceso, y este, se encarga de enviarlo hacia la estación de destino, definiéndose así como un sistema completamente centralizado; considerando esta especialidad una de las principales desventajas, ya que de surgir una caída del punto de acceso inalámbrico provocaría la desconexión total de la red.

Es por esto que se podría definir que la zona de cobertura local es equivalente a la zona de cobertura que tenga el punto de acceso.

Otra desventaja relevante, puede ser el crecimiento acelerado de las estaciones hasta un número crítico, cercano al límite de tolerancia del equipo de acceso, podría causar una disminución considerable del rendimiento.

Tomando en cuenta que las redes inalámbricas son *half-duplex*, y dos elementos de la red no pueden transmitir a la vez, de ésta forma se economiza el ancho de banda. También es posible establecer conexión entre puntos de acceso, mediante cable o *wireless* para aumentar el área de cobertura de la red, esta topología resulta ideal para permitir el acceso a Internet o a una red local a los ordenadores inalámbricos existentes.

Figura. 2 Red Infraestructura

1.4.4. Tipos de protocolos 802.11

Las tarjetas *AirPcap Nx Adapter* utilizadas en el escenario son compatibles para el monitoreo de redes con los protocolos *802.11a, b, g* y *n*, los mismos que son descritos a continuación.

- **802.11a:** Lanzada en el año de 1999, establece el funcionamiento de redes inalámbricas que trabajan sobre la frecuencia de 5 Ghz lo que favorece al protocolo porque existe menor interferencia, pero a su vez se presenta como desventaja ya que los equipos que se deseen conectar a la red están obligados a estar cerca del *AP*. Define 12 canales sin solapamiento, 8 para redes inalámbricas y 4 para conexiones punto a punto, capacidad de manejar velocidades de 6, 9, 12, 18, 24, 36, 48, 54 Mbps, utiliza modulación *OFDM*.

- **802.11b:** Lanzada en el año de 1999, el método de modulación que utiliza es el de espectro de difusión de secuencia directa complementaria y utiliza la llave de código complementario *(CCK)*, dispone de tres canales que no superponen en industrial, científico, médico, alcanza velocidades de hasta 11 Mbps, y el acceso al medio se da por medio del protocolo *CSMA/CA*.

- **802.11g:** Lanzada en el año 2003, utiliza modulaciones *DSSS*, *OFDM*, opera en el rango de frecuencia de 2.4 GHz, con velocidades de hasta 54 Mbps, compatible con *802.11b*, por trabajar en el mismo rango de frecuencias existen problemas de solapamiento de canales en las dos tecnologías por lo que se necesito mayor regulación para no provocar interferencias. Llego a proliferar el mercado debido a la compatibilidad que presentaba con otros estándares.

- **802.11n:** Presenta una notable mejora en comparación a los estándares anteriores, considerando que ofrece tasas teóricas de

transferencia de hasta 600 Mbps, pero en el mercado solo se conocen dispositivos que alcancen velocidades de hasta 300 Mbps. Además pese a ser basado en estándares anteriores incorpora el estándar *MIMO (Multiple Inputs, Multiple Outputs*), el que permite la utilización de más de un canal para transmisión y recepción a la vez. Ofrece la particularidad de utilizar *Channel Bounding* la cual permite ampliar el canal de 30 Mhz a 40 Mhz, por lo que amplía el ancho de banda e incrementa y permite mayor transmisión de datos.

1.4.5. Beneficios

Las redes *LAN* inalámbricas presentan varias ventajas sobre las redes *LAN* convencionales, siendo una de las más importantes la movilidad, que ofrece a las estaciones conectadas, la facilidad de poder desplazarse de un lado a otro dentro del área de cobertura, sin perder la conexión a la red, estos beneficios son evidentes para clientes que disponen de computadoras portátiles, dispositivos móviles, *PDA*, *smartphones*, ya que permite utilizar los recursos de la red sin estar atado a la restricción de los cables.

Se pueden detallar varios beneficios, de los cuales la planificación para la implantación para una red cableada es primordial, ya que se debe definir espacios para equipos, rutas de cable, a diferencia de las redes *wireless* cuya planificación estaría enfocada a la ubicación de las estaciones dentro del área de cobertura del punto de acceso, las dimensiones del mismo son reducidas, en comparación a los equipos de redes *LAN*. La tecnología *wireless* permite a una red alcanzar lugares donde los cables no llegan, o donde el coste de los mismos es muy alto. Por ejemplo, en espacios abiertos como jardines o piscinas, o para establecer comunicación entre oficinas ubicadas en edificios próximos, considerando la escalabilidad; las redes *WLAN* permiten ser configurados en distintas topologías que permiten adaptarse a las necesidades de cada situación, las configuraciones de los dispositivos *WLAN* pueden ir desde pequeñas redes con un número

reducido de usuarios, a grades infraestructuras con miles de usuarios con áreas de cobertura mayores, como campus universitarios o fábricas.

1.5. WIFI: 802.11i

1.5.1. Funcionamiento del protocolo

La rápida propagación de las redes inalámbricas basadas en los estándares *802.11* proporciona un nivel adicional de complejidad al problema de la seguridad de redes. Aunque los estándares incorporan ciertas funciones de seguridad, añadiendo diferentes mecanismos de protección, las redes inalámbricas representan un punto extremadamente vulnerable en la seguridad de una red. Considerando que la principal vulnerabilidad de una red inalámbrica es el hecho de que cualquier persona puede acceder a los datos que transitan por la red, debido a que utilizan un medio inseguro para sus comunicaciones, y que no existe ningún medio físico que imposibilite el acceso, se podría obtener de forma sencilla las tramas que circulan de forma pública en la red de área local inalámbrica *(WLAN)*, estas podrían ser utilizadas para obtener información que vulneren la seguridad de dicha red.

Para solventar las necesidades de seguridad de las redes inalámbricas se creó el estándar *802.11i* el cual presenta una mejora importante con respecto a los estándares predecesores de seguridad, es así que para solucionar las diferentes falencias que presentaban estos estándares. La norma *802.11i*, contiene una norma estándar de encriptación avanzada *(AES)*, que soporta claves de 128, 192 y 256 bits. La *AES* es una forma de codificación más desarrollada que se encuentra en la actual especificación de acceso protegido para redes inalámbricas. En principio garantiza que la información de las tramas que es enviada por estas redes esté encriptado y no pueda ser dañada por alguien que la intercepte, *802.11i*, introduce el llamado protocolo de red segura robusta *(RSN)* para establecer una

comunicación segura. A su vez implementa un protocolo extensible de autenticación *(EAP).* Este estándar tiene dos modos de funcionamiento en los que puede trabajar: el primero, es *WPA2* Enterprise basado en el protocolo *802.1x*, que utiliza tres elementos que son: suplicante, autenticador y servidor de autenticación, en donde tanto el servidor de autenticación como el suplicante generan dos claves aleatorias denominada llave maestra por parejas *(PMK)* durante la fase de autorización y autenticación de *802.1x*. Una vez finalizada la fase de autenticación, el servidor de autenticación y el cliente tienen *PMK* idénticas, pero el punto de acceso *(AP)* no, por lo tanto, a través del uso del discado de autenticación remota de usuarios en el servidor *(RADIUS)*, copia la clave del servidor de autenticación al *AP*. El protocolo no especifica el método de envío de la clave entre ambos dispositivos llegados hasta este punto, aún no se permite la comunicación si no que deben generar nuevas claves, en función de la *PMK*, para ser usadas en relación al cifrado y a la integridad, formando un grupo de cuatro claves, llamado clave transitorio por parejas *(PTK)* con una longitud de 512 bits. Para asegurar el tráfico *broadcast*, se crea claves de grupos de 256 bits llamado grupo de clave maestra *(GMK)* usado para crear la llave de encriptación del grupo *(GEK*) y la llave de integridad del grupo *(GIK)* de 128 bits de longitud cada una. Las cuatro claves forman la llave del grupo transitorio *(GTK)*. La última parte es demostrar que el AP tiene *PMK* idéntico, para ello lo valida el servidor de autenticación. Este proceso se realiza cada vez que es asociado un cliente con un *AP*. El segundo modo de operación, es WPA2 con una llave pre-compartida (PSK), pensado para entornos personales, el mismo que evita el uso de dispositivos externos de autenticación.

1.5.2. Ataques al protocolo

Conociendo todo el proceso o modo de funcionamiento que realiza el estándar y el intercambio de números aleatorios que se llevan a cabo entre un cliente y el *AP* para la autenticación y asociación, un atacante que quiera

vulnerar una red *WPA2-PSK*, va a tratar de capturar ese intercambio de números, para que una vez conocidos estos, junto con el identificador de establecimiento de servicio *(SSID)*, las direcciones de control de acceso al medio *(MAC)* del cliente y el AP de la red, pueda obtener la frase o secreto compartido que se utilizó. Una vez que el atacante tenga la clave compartida, se podrá conectar a la red.

Existen diferentes tipos de ataques para las redes inalámbricas, principalmente relacionados con la capa de enlace, de las cuales se citan los más importantes: *AP spoofing* llamado también asociación maliciosa, en este ataque el intruso se hace pasar por un punto de acceso, engaña al cliente y hace parecer que se está conectado a una red *WLAN* verdadera; envenenamiento del protocolo de resolución de direcciones *(ARP poisoning)* u hombre en el medio *(MITM)*, es un ataque en el que se supervisa una comunicación entre dos partes y falsifica la información para hacerse pasar por una de ellas; *WLAN scanners*, llamado también ataque de vigilancia, consiste en recorrer un área donde se desea realizar una intrusión para descubrir redes *WLAN* activas, para realizar ataques y una posterior adquisición de información; denegación de servicios *(DoS),* siendo este uno de los más importantes ataques, su objetivo principal es el de paralizar o desactivar una *WLAN,* es decir es un ataque realizado sobre la disponibilidad de la red.

Con el transcurso del tiempo se han creado diferentes herramientas para poder vulnerar la seguridad de las redes, para diagnosticar, evitar y contrarrestar estos ataques, es imprescindible monitorear las conexiones inalámbricas y detectar posibles ataques, es por esto que en definitiva, quizás la mejores herramienta con las que se puede contar para prevenir cualquier amenaza informática, es la de monitorización constante de equipos y sistemas.

Pese a las características especiales del estándar *802.11i*, que pretende proteger al máximo a la red, *nVidia* ha creado *CUDA*, el cual permite la posibilidad de romper mediante la fuerza bruta las claves que se usan en conexiones inalámbricas con encriptación *WPA2*. A pesar de que *WPA2* es la forma de encriptación y autentificación más segura que se utilizadas hoy en día, fue descubierto en el 2010 una vulnerabilidad que llamaron "*Hole196*", que se suma a la ya descrita anteriormente, para vulnerar la información confidencial a través de la red inalámbrica

La eventualidad de que se produzca el ataque, es un punto de máximo interés para las empresas, principalmente con los ataques de negación de servicio, ya que estos afecta directamente al desempeño de la red y deja al entorno de red sin servicios.

1.6. Tarjetas AirPcap Nx Adapter

1.6.1. Características

La evaluación de la seguridad en una red inalámbrica puede resultar desafiante si no se cuenta con las herramientas necesarias que, además, entiendan el lenguaje de las ondas de radio para facilitar su estudio, con esto se pretende obtener información de los protocolos inalámbricos y así realizar el análisis correspondiente dentro del escenario de prueba planteado.

Las tarjetas *AirPcap Nx Adapter* fueron escogidas para este proyecto debido a que cuentan con especificaciones que permiten el monitoreo de tráfico de la capa de enlace en redes inalámbricas 802.11i con pérdida de paquetes mínima. Estas características se explican a continuación:

Los adaptadores *Nx* son fácilmente instalados para funcionar en ambientes virtuales como *VMWare*, Sistemas operativos como: Windows,

Macintosh o Linux. Esto permite flexibilidad para el estudio en cualquier ambiente.

La tecnología de red inalámbrica que permite establecer la conexión y que puede ser monitoreada por las tarjetas *Nx* abarcan el estándar *IEEE 802.11 a/b/g/n*, el cual es un set de especificaciones de control de acceso al medio *(MAC)* y la capa física *(PHY)* que permite que uno o más dispositivos usando un método de distribución inalámbrica se conecten a una red *WLAN*.

Las frecuencias de operación de las tarjetas cubren el espectro de los 2.412 – 2.484 GHz (b/g/n) y 4.920 – 5.825 GHz (a/n) con esto se logra cubrir todos los canales dentro de cada banda que aportan información relevante para el escenario de prueba dentro del estándar *802.11i*.

Entre los esquemas de modulación que las tarjetas cobijan para redes inalámbricas se tiene; *CCK (Complementary Code Keying)* que es empleada en las especificaciones del *IEEE 802.11b*, *OFDM (Orthogonal Frequency-Division Multiplexing)* utilizada en el *IEEE* 802.11a, HT-OFDM para IEEE 802.11a/g/n y MCS 0-15 *(Modulation and Coding Scheme)* mandatorio para estándares 802.11n.

Cuenta con dos antenas incorporadas, adicional tiene conectores *MC-Card* integrados para antenas externas opcionales, esto permite extender el radio de cobertura del adaptador *Nx* mejorando el rendimiento en los entornos más exigentes.

Este quipo, permite descifrar los algoritmos de seguridad *WEP, WPA PSK, WPA2 PSK* en populares herramientas de análisis como: Wireshark al capturar paquetes de la red *WLAN,* Kismet para descubrir las redes disponibles dentro del radio de cobertura, *Aircrack-ng* o *Cain y Abel* para vulnerar las redes con el objetivo de encontrar y resolver problemas de seguridad; análisis de paquetes, uso del ancho de banda en el canal,

transmisión de errores, fallas en la asociación y autenticación de acceso en *AP's (Access Point)*, descubrir aplicaciones sin cifrar e investigaciones de baja taza de trasmisión.

La velocidad de transmisión de datos es de 54 Mbps para *802.11a/b/g*, 130 Mbps en canales de 20 MHz y 300 Mbps en canales de 40 MHz para *802.11n*. Cabe mencionar que permite un rendimiento de captura superior con un mínimo de pérdidas de paquetes, especialmente al capturar múltiples canales en redes *802.11n*.

Fácil configuración y flexibilidad sin requerir asociación con el *AP*.

Las tarjetas *AirPcap Nx* son versátiles y portables con forma y dimensiones similares a un *pendrive* compatible con puertos *USB 2.0* Tipo A y *UBS 1.1.*

Figura. 3 Kit de tarjeta AirPcap Nx

Las tarjetas *AirPcap Nx* tienen un *Chipset (Atheros AR9170)* y *Driver (carl9170)*, totalmente compatible con BT5r3 y Wireshark.

1.6.2. Funcionamiento de las tarjetas AirPcap Nx Adapter

AirPcap es un adaptador que capta la totalidad o un conjunto filtrado de tramas *WLAN* y entrega los datos a la plataforma de Wireshark para su

análisis. Una vez *AirPcap* está instalado, Wireshark muestra una barra de herramientas especial que proporciona el control directo del adaptador *AirPcap* durante la captura de datos inalámbrica.

AirPcap es también el nombre de una familia de productos que incluye *AirPcap Classic, AirPcap Tx, AirPcap Ex* y *AirPcap N*. Esta familia de productos representa la primera abierta, asequible y fácil de implementar soluciones de captura de paquetes *802.11 WLAN* para la plataforma de Windows, OSX y Linux. Los diferentes miembros de la familia *AirPcap + Wireshark* proporcionan información acerca de los protocolos inalámbricos y señales de radio, lo que le permite capturar y analizar el tráfico inalámbrico *802.11a/b/g/n* de bajo nivel, incluyendo los marcos de control, marcos de gestión, y la información de potencia.

1.6.3. AirPcap Nx y Wireshark

AirPcap es un adaptador especial para el análisis de tráfico de red inalámbrica. Con *AirPcap Nx* se pueden escanear múltiples canales en paralelo para el análisis de datos y solución de problemas de redes inalámbricas. Sin embargo, el alcance de estas herramientas se limita en el análisis de *WLAN* si no se cuenta con el adaptador *AirPcap Nx.* Es por esto que *AirPcap* proporciona una solución de hardware para la captura de paquetes inalámbricos plenamente integrado en el software Wireshark.

El adaptador *USB AirPcap Nx* es compatible con *802.11a/b/g/n*, y es adecuado tanto para la captura de paquetes como para la inyección de paquete. Para pruebas de carga, una antena interna y dos conectores *MC* para antenas externas y optimizar la detección del tráfico *802.11* esto con la finalidad de capturar los paquetes utilizando Wireshark mediante el controlador de captura *WinPcap*. Este se comunica con el adaptador de red *LAN / WLAN* y se ejecuta en "modo promiscuo" capturando los paquetes de

datos, de gestión y control para llevar a cabo el análisis de los datos procedentes de una sesión de captura *AirPcap*.

CAPÍTULO II

DISEÑO E IMPLEMENTACIÓN DEL ESCENARIO DE PRUEBA

2.1 Diseño del escenario

El siguiente capítulo describe la implementación de un escenario de prueba, junto con todos los materiales y métodos necesarios para desarrollar las diferentes pruebas y ataques generados por BT5*,* esto con el fin de determinar las vulnerabilidades por medio del análisis en Wireshark de las tramas o paquetes capturados por medio de las tarjetas *AirPcap Nx* entre el *AP*, los usuarios y el servidor.

2.1.1 El escenario 802.11i

El grupo *IEEE 802.11i* especifica *802.1x* para la autenticación de usuarios. *802.1x* es un marco de estándares abiertos para autenticar las estaciones inalámbricas conocidos como suplicantes *(supplicants)* con un servidor de autenticación en la red cableada mediante un punto de acceso inalámbrico llamado autenticador *(authenticator)*. El servidor de autenticación conocido como *RADIUS* (*Remote Authentication Dial-In User Service)* mantiene registros detallados de los usuarios para limitar el acceso a la red de los no autorizados. Estos tres elementos conforman el escenario físico de este entandar.

En la figura que se muestra a continuación, se puede identificar estos tres dispositivos. Los suplicantes: MacBook Air, iPhone 5s, LG, Asus e iPad; el autenticador: *AP* y finalmente el servidor de autenticación: *FreeRADIUS*.

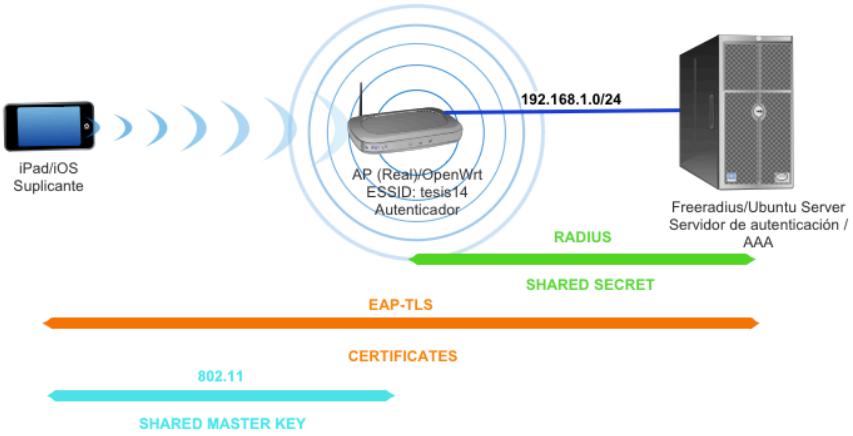


Figura. 4 Escenario 802.11i

2.1.2 Presentación de la topología del escenario de prueba

A continuación se presenta un arreglo de topología con configuraciones específicas para la evaluación y análisis de los escenarios de prueba en comportamiento normal e intrusivo.

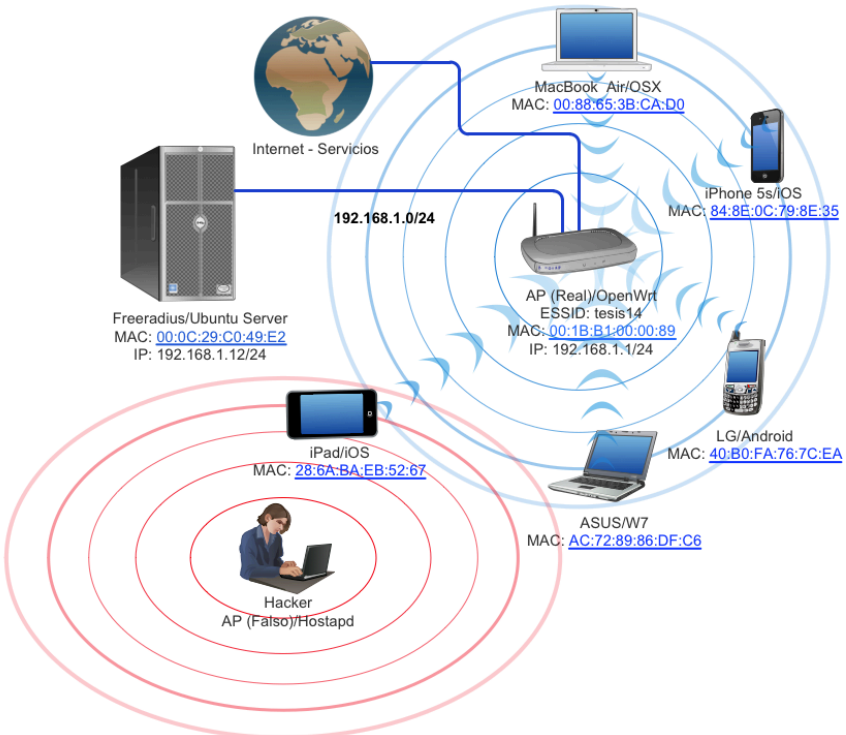


Figura. 5 Topología del escenario de prueba

Toda la información relevante de los equipos y la topología se muestra como sigue:

Tabla. 2

Direccionamiento, ESSID y BSSID

Dispositivo	Interfaz	Dir. IP	Máscara	Gateway	Tipo de red	BSSID	ESSID	Modo
Freeradius/Ubuntu Server	eth0	192.168.1.12	255.255.255.0	192.168.1.1	LAN	00:0C:29:C0:49:E2	N/A	N/A
AP (Real)/OpenWrt	br-lan	192.168.1.1	255.255.255.0	N/A	LAN	00:0D:B9:19:66:14	N/A	N/A
	eth1	NAT	255.255.255.0	N/A	WAN	00:0D:B9:19:66:15	N/A	N/A
	wifi0	N/A	N/A	N/A	N/A	00:1B:B1:00:00:89	tesis14	Master
	wifi1	N/A	N/A	N/A	N/A	00:1B:B1:00:00:AE	Wireless2	Monitor
MacBook Air/OSX	NIC	DHCP	255.255.255.0	192.168.1.1	WLAN	00:88:65:3B:CA:D0	N/A	N/A
iPhone 5s/iOS	NIC	DHCP	255.255.255.0	192.168.1.1	WLAN	84:8E:0C:79:8E:35	N/A	N/A
LG/Android	NIC	DHCP	255.255.255.0	192.168.1.1	WLAN	40:B0:FA:76:7C:EA	N/A	N/A
Asus/W7	NIC	DHCP	255.255.255.0	192.168.1.1	WLAN	AC:72:89:86:DF:C7	N/A	N/A
iPad/iOS	NIC	DHCP	255.255.255.0	192.168.1.1	WLAN	28:6A:BA:EB:52:68	N/A	N/A
AP (Falso)/Hostapd	wlan0	N/A	N/A	N/A	N/A	00:11:22:33:44:55	tesis14	Master
	mon0	N/A	N/A	N/A	N/A			Monitor
Macbook Pro/OSX	Ethernet	192.168.1.36	255.255.255.0	192.168.1.1	LAN	3C:07:54:35:59:F2	N/A	N/A

2.1.3 Esquema del escenario de prueba

El esquema para el escenario de prueba se resume en la siguiente figura para su implementación.

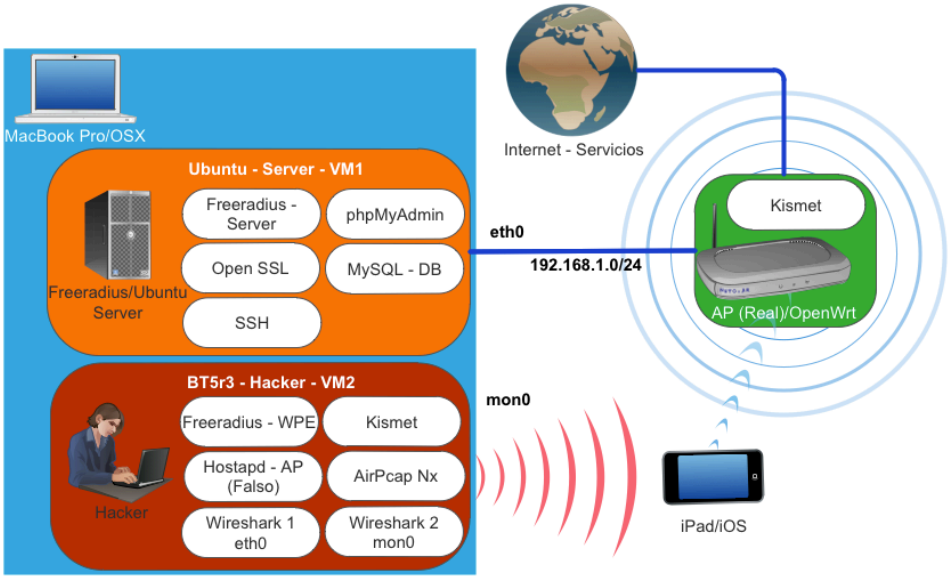


Figura. 6 Esquema del escenario de prueba

Como se puede ver en la figura del esquema del escenario de prueba, en un computador MacBook Pro se realizará la instalación de dos máquinas virtuales *(Virtual machines-VM):* La primera, llamada *Ubuntu–Server–VM1* que será preparada para todo lo referente al servidor de autenticación *RADIUS* y la conectividad con el *AP;* La segunda, llamada *BT5r3–Hacker–VM2* que será implementada con todo lo necesario para la captura de paquetes en un escenario de prueba normal o para realizar ataques en el caso del escenario de prueba intrusivo.

Adicional, se debe preparar el autenticador, en este caso llamado *AP* (Real)/OpenWrt con la instalación de un software para la captura de paquetes y la habilitación del *access point* al internet o servicios. Se debe verificar la correcta comunicación con la *VM1* para el acceso al servidor y la *VM2* para la captura de paquetes para su posterior análisis.

Finalmente, se tiene que contar con los dispositivos suplicantes para nuestra red como el equipo iPad/iOS que se muestra en el esquema anterior.

2.2 El equipo anfitrión para el escenario de prueba

Se ha seleccionado al equipo anfitrión u hospedador para la implementación de este escenario de prueba a una laptop MacBook Pro, la misma fue seleccionada por contar con las siguientes características:

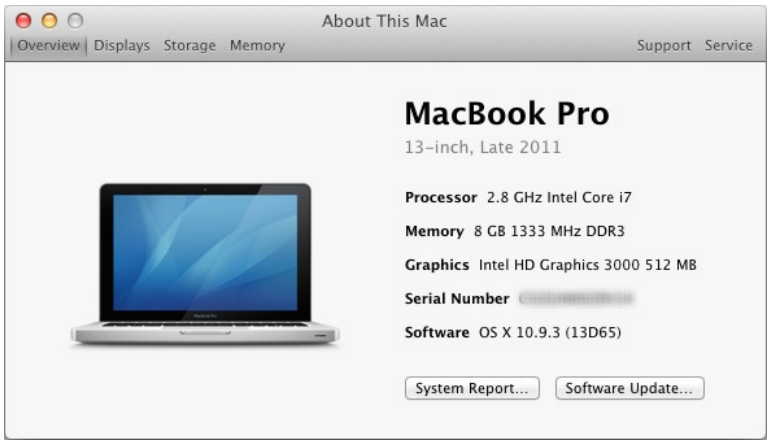


Figura. 7 Características equipo anfitrión

Adicional, este equipo cuenta con un disco duro interno de 750 GB, esto permite fácilmente contener la información de ambas máquinas virtuales necesarias para implementar el escenario de prueba. Características tales, como las del procesador de 2.8 GHz Intel Core i7 y la memoria RAM de 8GB 1333 Mhz DDR3 permiten el procesamiento fluido y estable de las máquinas virtuales cuando estas se usen simultáneamente.

2.3 Implementación Ubuntu – Server – VM1

Para esta máquina virtual, se instala la última versión de Ubuntu Server (14.04 LTS) de 64 bits de nombre código: *trsusty.* Una vez instalada, se debe obtener una pantalla como esta:

Figura. 8 Ubuntu Server 14.04 TLS

Los siguientes comandos son recomendados después de la instalación de Ubuntu para su correcto funcionamiento y actualización:

>> espe14@ubuntu:~$ sudo –i

[sudo] password for espe14: espetesis14

>> root@ubuntu:~# tasksel

Se seleccionan las opciones que se muestran y <Ok>

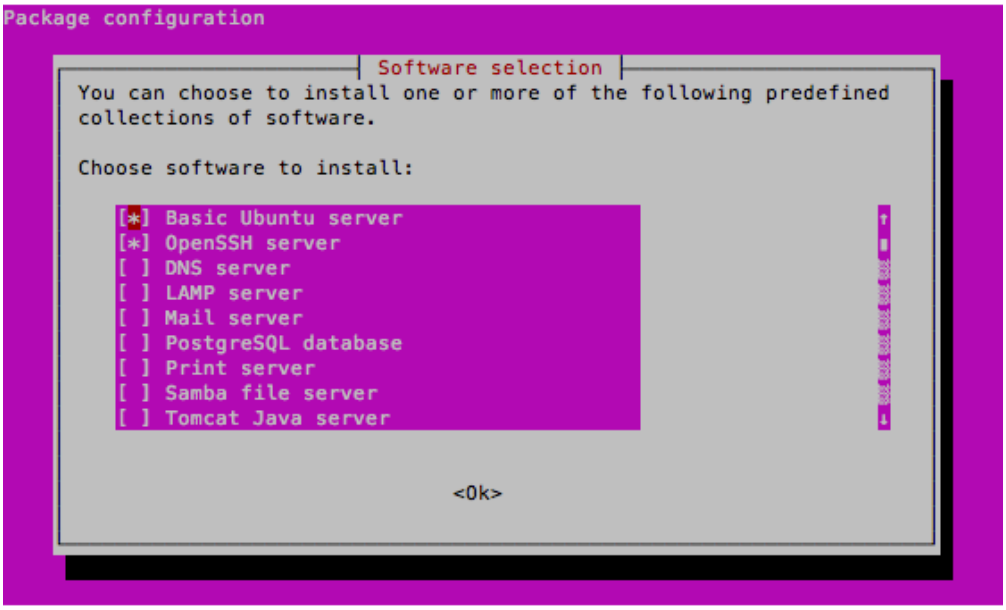


Figura. 9 menú tasksel

>> root@ubuntu:~# apt-get update && apt-get upgrade && apt-dist upgrade

El esquema a preparar en esta máquina virtual es la siguiente:

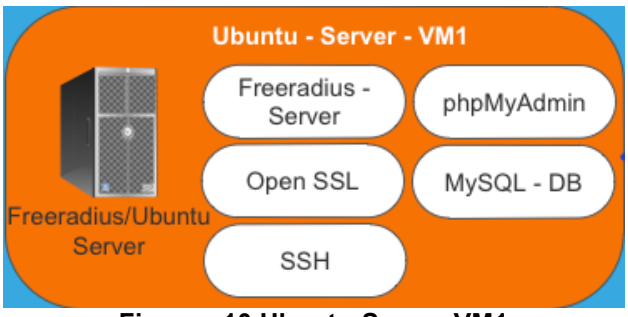


Figura. 10 Ubuntu-Server-VM1

En los pasos siguientes se realizará la instalación paso a paso de los módulos contenidos en el esquema de esta máquina virtual.

2.3.1 SSH

Secure Shell (SSH) es un protocolo de red de cifrado para la comunicación segura de datos a distancia. Se utiliza para la ejecución remota de comandos y otros servicios de red seguros entre dos equipos en

red. Esto permite, entre otras cosas, la facilidad de copiar y pegar líneas de comandos en ambientes sin interfaz gráfica o poco amigable con el usuario como lo es el servidor Ubuntu.

Se instala de la siguiente forma:

>> espe14@ubuntu:~$ sudo –i
[sudo] password for espe14: espetesis14
>> root@ubuntu:~# apt-get install ssh

Desde *Terminal* en el sistema operativo de Apple-OSX se puede establecer la comunicación segura con el servidor ingresando el comando:

>> ssh espe14@192.168.1.12
espe14: es el nombre del usuario registrado en el servidor.
192.168.1.12: es la dirección IP del servidor

Se solicitará la clave (*password*) del servidor, se ingresa la misma y se presiona *enter*.

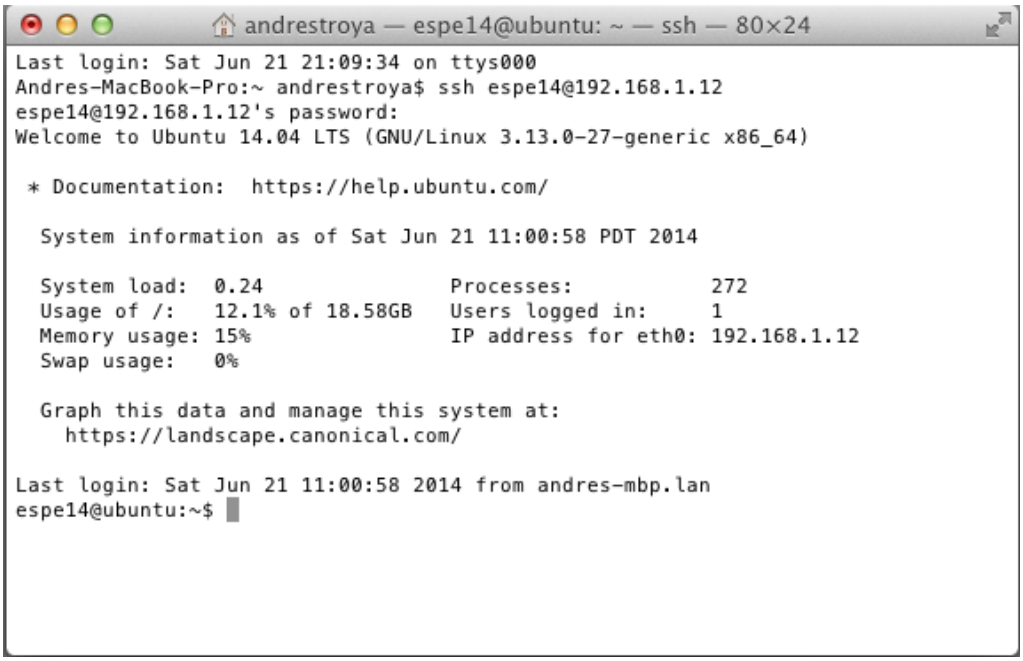


Figura. 11 SSH desde Terminal en OSX al Servidor Ubuntu

2.3.2 FreeRADIUS – Server

FreeRADIUS es el servidor *RADIUS* de mayor despliegue en el mundo. Es la base y referencia para múltiples ofertas comerciales, su distribución es gratuita.

En el servidor Ubuntu escribimos las siguientes líneas de comando para la instalación:

>> espe14@ubuntu:~$ sudo –i

[sudo] password for espe14: espetesis14

>> root@ubuntu:~# apt-get install freeradius freeradius-utils freeradius-mysql

>> root@ubuntu:~# freeradius –v

-v: Imprime información sobre la versión del servidor

```
espe14@ubuntu:~$ freeradius -v
freeradius: FreeRADIUS Version 2.1.12, for host x86_64-pc-linux-gnu, built on Fe
b 24 2014 at 14:57:57
Copyright (C) 1999-2011 The FreeRADIUS server project and contributors.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
You may redistribute copies of FreeRADIUS under the terms of the
GNU General Public License.
For more information about these matters, see the file named COPYRIGHT.
espe14@ubuntu:~$
```

Figura. 12 freeradius-v

>> root@ubuntu:~# /etc/init.d/freeradius restart

>> root@ubuntu:~# /etc/init.d/freeradius stop

>> root@ubuntu:~# freeradius –s –X –f

```
radiusd: #### Opening IP addresses and Ports ####
listen {
        type = "auth"
        ipaddr = *
        port = 0
}
listen {
        type = "acct"
        ipaddr = *
        port = 0
}
listen {
        type = "auth"
        ipaddr = 127.0.0.1
        port = 18120
}
 ... adding new socket proxy address * port 58340
Listening on authentication address * port 1812
Listening on accounting address * port 1813
Listening on authentication address 127.0.0.1 port 18120 as server inner-tunnel
Listening on proxy address * port 1814
Ready to process requests.
```

Figura. 13 freeradius –s –X -f

Al momento de ejecutar *freeradius –s –X –f* se obtiene la imagen que se muestra anteriormente, donde se comprueba que la instalación fue exitosas y el servidor está listo para procesar las peticiones.

2.3.3 OpenSSL

Al establecer el escenario de prueba bajo el estándar 802.11i, se requiere de un protocolo de autenticación. EAP *(Extensible Authentication Protocol)* es el mecanismo oficial adoptado para la autenticación en las redes inalámbricas para las conexiones punto a punto en este modelo.

En los escenarios, se va a implementar la autenticación *EAP-TLS (EAP-Transport Layer Security)*. Esto debido a que el protocolo *TLS,* es considerado como uno de los más seguros dentro de los estándares de *EAP* disponibles en la actualidad.

El protocolo *TLS* está basado en la arquitectura *802.1x/EAP*. Los componentes involucrados en este proceso de autenticación son, como ya se ha mencionado: el suplicante (equipos de los usuarios), el autenticador *(AP)* y el servidor de autenticación *(RADIUS)*. El suplicante y el servidor *RADIUS* deben soportar la autenticación *EAP-TLS*. El *AP* por otra parte,

debe dar soporte a los procesos de autenticación (este desconoce el tipo de protocolo *EAP*).

El servidor *RADIUS* proporciona su certificado al cliente y solicita el certificado del cliente. El cliente, valida el certificado de servidor y responde con un mensaje de respuesta de *EAP* que contiene su certificado e inicia la negociación para las especificaciones de cifrado. Después de validar el certificado del cliente, el servidor responde con las especificaciones de cifrado para la sesión.

OpenSSL es requerido para crear los certificados que permitan la autenticación mutua entre el usuario y el servidor *RADIUS*. El proceso para la instalación se muestra a continuación:

>> espe14@ubuntu:~$ sudo –i
[sudo] password for espe14: espetesis14
>> root@ubuntu:~# cd /etc/ssl
>> root@ubuntu:/etc/ssl# mkdir /etc/ssl/PKI
>> root@ubuntu:/etc/ssl# cd
>> root@ubuntu:~# apt-get install openssl
>> root@ubuntu:~# openssl version
version: muestra la versión de oppenssl instalada

```
root@ubuntu:~# openssl version
OpenSSL 1.0.1f 6 Jan 2014
root@ubuntu:~#
```

Figura. 14 openssl version

>> root@ubuntu:~# cd /etc/ssl
>> root@ubuntu:/etc/ssl# nano openssl.cnf

Se edita el archivo openssl.cnf y se cambia en la sección [CA_default] el parámetro dir a ./PKI

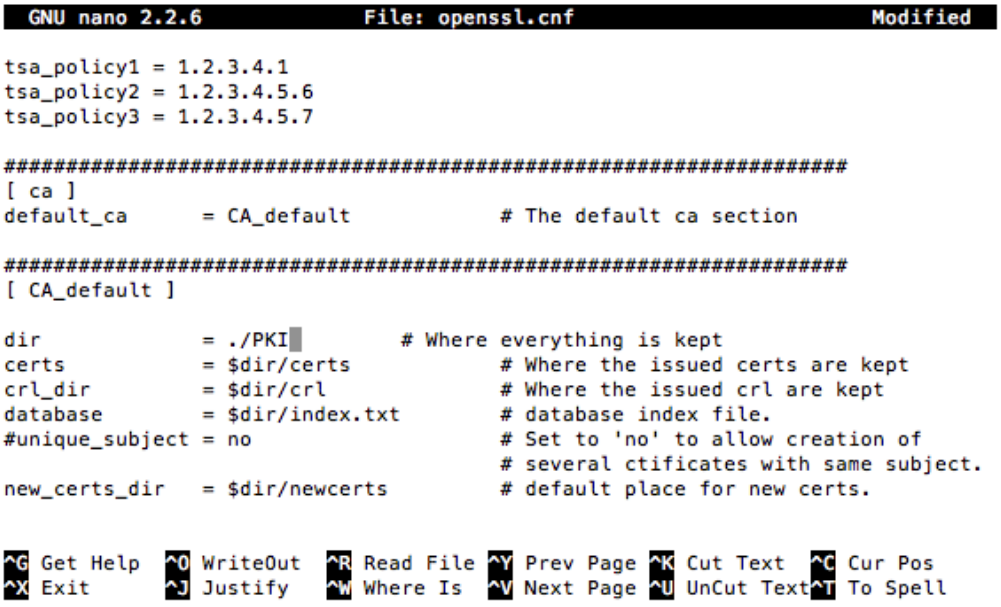


Figura. 15 modificación de openssl.cnf

>> root@ubuntu:/etc/ssl# cd

>> root@ubuntu:~# cd /usr/lib/ssl/misc/

>> root@ubuntu: /usr/lib/ssl/misc/# nano CA.sh

Se edita el archivo *CA.sh* y luego se cambia el parámetro *CATOP* a *./PKI*

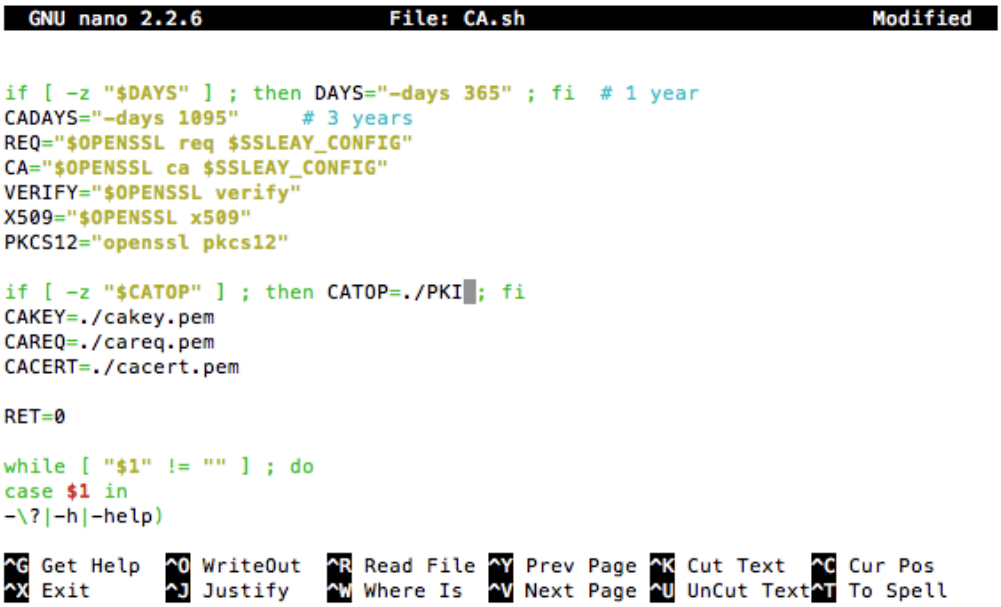


Figura. 16 Modificación de CA.sh

>> root@ubuntu: /usr/lib/ssl/misc/# cd

>> root@ubuntu:~# cd /etc/ssl

>> root@ubuntu:/etc/ssl# /usr/lib/ssl/misc/CA.sh –newca

Se crea el certificado *CA* y se establece una clave, en este caso, *Verifying – Enter PEM pass phrase:* espe14

```
Making CA certificate ...
Generating a 2048 bit RSA private key
....+++
.....................................................
.....................................................
.....................................................+++
writing new private key to './PKI/private/./cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Figura. 17 Creación del certificado CA

Se ingresa la información como se muestra en la siguiente figura, además, se debe especificar un *CN (Common Name)* para el servidor:

```
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:EC
State or Province Name (full name) [Some-State]:PICHINCHA
Locality Name (eg, city) []:QUITO
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ESPE
Organizational Unit Name (eg, section) []:REDES
Common Name (e.g. server FQDN or YOUR name) []:SERVIDOR
Email Address []:info@espe.edu.ec

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:espe14
An optional company name []:ESPE
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for ./PKI/private/./cakey.pem:
Check that the request matches the signature
Signature ok
```

Figura. 18 Poblar la información del certificado SERVIDOR

A continuación se muestra los detalles o resumen del certificado creado:

```
Certificate Details:
        Serial Number: 16842820547415286464 (0xe9bdb80d0056a2c0)
        Validity
            Not Before: Jun  3 20:07:31 2014 GMT
            Not After : Jun  2 20:07:31 2017 GMT
        Subject:
            countryName               = EC
            stateOrProvinceName       = PICHINCHA
            organizationName          = ESPE
            organizationalUnitName    = REDES
            commonName                = SERVIDOR
            emailAddress              = info@espe.edu.ec
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                DA:CA:A8:64:89:E5:6A:E9:EF:46:65:E9:EB:C3:53:3A:F9:41:2F:28
            X509v3 Authority Key Identifier:
                keyid:DA:CA:A8:64:89:E5:6A:E9:EF:46:65:E9:EB:C3:53:3A:F9:41:2F:2
8

            X509v3 Basic Constraints:
                CA:TRUE
Certificate is to be certified until Jun  2 20:07:31 2017 GMT (1095 days)
```

Figura. 19 Detalles del certificado SERVIDOR

Se realiza la solicitud de firma para el certificado del servidor:

>> root@ubuntu:/etc/ssl# openssl req –new –nodes –keyout PKI/server_key.pem –out PKI/server_req.pem –days 730 –config openssl.cnf

Se firma la petición de certificado del servidor:

>> root@ubuntu:/etc/ssl# openssl ca –config openssl.cnf –policy policy_anything –out PKI/server_cert.pem –infiles PKI/server_req.pem

```
        Validity
            Not Before: Jun  3 20:16:23 2014 GMT
            Not After : Jun  3 20:16:23 2015 GMT
        Subject:
            countryName               = EC
            stateOrProvinceName       = PICHINCHA
            localityName              = QUITO
            organizationName          = REDES
            organizationalUnitName    = ESPE
            commonName                = SERVIDOR
            emailAddress              = info@espe.edu.ec
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                44:F5:E8:12:63:AF:C4:24:15:02:A9:EF:97:6D:E6:D0:B4:4B:B4:9A
            X509v3 Authority Key Identifier:
                keyid:DA:CA:A8:64:89:E5:6A:E9:EF:46:65:E9:EB:C3:53:3A:F9:41:2F:2
8

Certificate is to be certified until Jun  3 20:16:23 2015 GMT (365 days)
Sign the certificate? [y/n]:
```

Figura. 20 Firma del certificado creado

Sign the certificate? [y/n]: y

1 out of 1 certificate requests certified, commit? [y/n]: y

>> root@ubuntu:/etc/ssl# cp PKI/server_cert.pem PKI/server_cert.pem-backup

>> root@ubuntu:/etc/ssl# cat PKI/server_key.pem PKI/server_cert.pem > PKI/server_keycert.pem

Se crea una petición de firma de certificado del cliente:

>> root@ubuntu:/etc/ssl# openssl req –new –keyout PKI/client_key.pem –out PKI/client_req.pem –days 730 –config openssl.cnf

```
Generating a 2048 bit RSA private key
.................................+++
.............................................................+++
writing new private key to 'PKI/client_key.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Figura. 21 Generación de llave privada RSA

Verifying – Enter PEM pass phrase: espe14

Se ingresa la información como se muestra en la siguiente figura, además, se debe especificar un *CN (Common Name)* para el cliente:

```
writing new private key to 'PKI/client_key.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:EC
State or Province Name (full name) [Some-State]:PICHINCHA
Locality Name (eg, city) []:QUITO
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ESPE
Organizational Unit Name (eg, section) []:REDES
Common Name (e.g. server FQDN or YOUR name) []:USUARIO
Email Address []:info@espe.edu.ec

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:espe14
An optional company name []:ESPE
```

Figura. 22 Detalles del certificado USUARIO

Se firma la solicitud del certificado del cliente para usuarios Mac OSX, Linux y Android:

>> root@ubuntu:/etc/ssl# openssl ca –config openssl.cnf –policy policy_anything –out PKI/client_cert.pem –infiles PKI/client_req.pem

```
        Validity
            Not Before: Jun  3 20:24:34 2014 GMT
            Not After : Jun  3 20:24:34 2015 GMT
        Subject:
            countryName               = EC
            stateOrProvinceName       = PICHINCHA
            localityName              = QUITO
            organizationName          = ESPE
            organizationalUnitName    = REDES
            commonName                = USUARIO
            emailAddress              = info@espe.edu.ec
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                E8:5C:65:CF:68:C9:6E:56:DB:7D:F4:40:D8:6F:C1:AB:D4:36:24:EB
            X509v3 Authority Key Identifier:
                keyid:DA:CA:A8:64:89:E5:6A:E9:EF:46:65:E9:EB:C3:53:3A:F9:41:2F:2
8

Certificate is to be certified until Jun  3 20:24:34 2015 GMT (365 days)
Sign the certificate? [y/n]:
```

Figura. 23 Firma del certificado creado

Sign the certificate? [y/n]: y

1 out of 1 certificate requests certified, commit? [y/n]: y

Se exportan los certificados P12 para clientes Windows y Mac:

>> root@ubuntu:/etc/ssl# openssl pkcs12 –export –in PKI/client_cert.pem –inkey PKI/client_key.pem –out PKI/client_cert.p12 –clcerts

```
Enter pass phrase for PKI/client_key.pem:
Enter Export Password:
Verifying - Enter Export Password:
```

Figura. 24 Ingreso de clave para exportar certificado

Enter pass phrase for PKI/client_key.pem: espe14

Enter Export Password: espe14

Verifying –Enter Export Password: espe14

Se exportan los certificados P12 para clientes Android:

>> root@ubuntu:/etc/ssl# openssl pkcs12 –export –in PKI/client_cert.pem –inkey PKI/client_key.pem –certfile PKI/cacert.pem –name “Wifi” –out PKI/client_cert.p12

```
Enter pass phrase for PKI/client_key.pem:
Enter Export Password:
Verifying - Enter Export Password:
```

Figura. 25 Ingreso de clave para exportar certificado

Enter pass phrase for PKI/client_key.pem: espe14

Enter Export Password: espe14

Verifying –Enter Export Password: espe14

Configuración para *FreeRADIUS*:

>> root@ubuntu:/etc/ssl# cp /etc/ssl/PKI/cacert.pem /etc/freeradius/certs/cacert.pem

>> root@ubuntu:/etc/ssl# cp /etc/ssl/PKI/server_keycert.pem /etc/freeradius/certs/server_keycert.pem

>> root@ubuntu:/etc/ssl# cd

>> root@ubuntu:~# cd /etc/freeradius/certs/

>> root@ubuntu:/etc/freeradius/certs/# openssl dhparam –check –text -5 512 –out dh

```
Generating DH parameters, 512 bit long safe prime, generator 5
This is going to take a long time
...+.............................+.+..+.............................+.....
.........+.....................+.....................+.............
.....+.............................+............+.....................
....+.......+....+.................................................+.....
..............+.....................................................
.....................................+.............+.....................
..............+.....................................................+.......
...+.....................................................................+.....
...................+...+................+..+........+.....+...+.............
+.....................................................................+.....
.....................................................................+....+
..+.......................+..............+............+............+......
.............................+..............+...........+.+......+..........
..........+......+.................+.+.................+.....................
.....................................................................
..+.....................++++++++++++++++
DH parameters appear to be ok.
```

Figura. 26 Generación de parámetros DH

>> root@ubuntu:/etc/freeradius/certs/# dd if=/dev/urandom of=random count=2

```
2+0 records in
2+0 records out
1024 bytes (1.0 kB) copied, 0.0278108 s, 36.8 kB/s
```

Figura. 27 Cuenta igual a 2

>> root@ubuntu:/etc/freeradius/certs/# chown freerad dh

>> root@ubuntu:/etc/freeradius/certs/# chmod o-w dh

>> root@ubuntu:/etc/freeradius/certs/# cd

>> root@ubuntu:~# cd /etc/freeradius/

>> root@ubuntu:/etc/freeradius# rm –R eap.conf

>> root@ubuntu:/etc/freeradius# nano eap.conf

Se debe ingresar toda la información que se muestra en la figura siguiente, para definir los parámetros y el uso del protocolo *TLS* en el archivo *eap.conf.*

```
 GNU nano 2.2.6                File: eap.conf

eap {
        default_eap_type = tls
        timer_expire     = 60
        ignore_unknown_eap_types = no
        cisco_accounting_username_bug = no

        tls {
                certdir = ${confdir}/certs
                cadir = ${confdir}/certs
                private_key_password = whatever
                private_key_file = ${certdir}/server_keycert.pem
                certificate_file = ${certdir}/server_keycert.pem
                CA_file = ${cadir}/cacert.pem
                dh_file = ${certdir}/dh
                random_file = ${certdir}/random
                fragment_size = 1024
                include_length = yes
                check_cert_cn = %{User-Name}
                cipher_list = "DEFAULT"
        }

peap {
        default_eap_type = mschapv2
}

mschapv2 {
}

}

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Figura. 28 Ingreso de parámetros en eap.conf

Más adelante se ve la instalación de los certificados obtenidos en esta sección para las diferentes plataformas: Mac OSX, iOS, Windows y Android.

2.3.4 MySQL

MySQL es un sistema de libre acceso de código abierto *RDBMS (Relational Database Management System)* que utiliza lenguaje *SQL* (Structured Query Language). *SQL* es el lenguaje más popular para añadir, acceder y gestionar contenidos en una base de datos. Es el más conocido por su rápido procesamiento, fiabilidad probada, facilidad y flexibilidad en su uso.

Para realizar la instalación de MySQL, se procede con las siguientes líneas de comando:

>> espe14@ubuntu:~$ sudo –i
[sudo] password for espe14: espetesis14
>> root@ubuntu:~# apt-get install mysql-server freeradius-mysql

Se ingresa una nueva clave de acceso para este servidor MySQL. Esto permite que el usuario con privilegios *“root”* pueda administras la base de datos *(DB).*

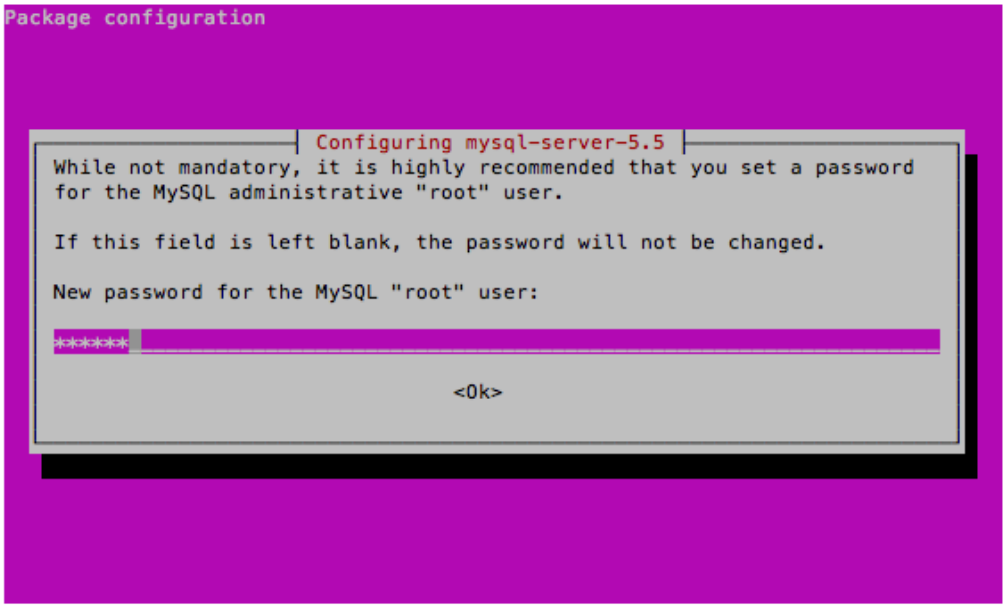


Figura. 29 Configurando el servidor MySQL

Se repite la clave ingresada para el servidor. Esto con la finalidad de verificar que sea la misma.

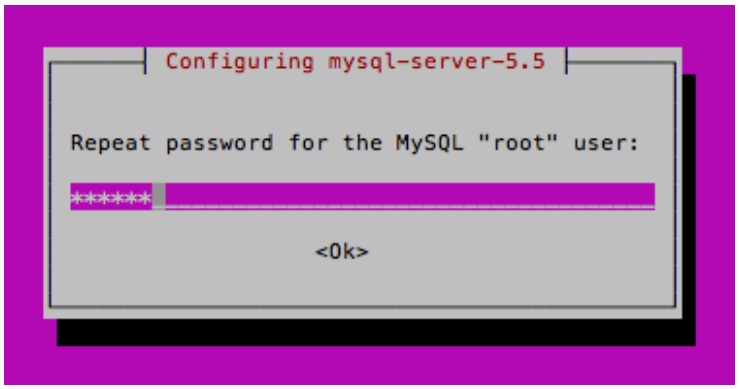


Figura. 30 Verificación de clave para MySQL

>> root@ubuntu:~#mysql --version

--version: muestra la versión de MySQL instalada

```
root@ubuntu:~# mysql --version
mysql  Ver 14.14 Distrib 5.5.37, for debian-linux-gnu (x86_64) using readline 6.
3
root@ubuntu:~#
```

Figura. 31 Versión de MySQL

>> root@ubuntu:~# echo "create database radius;" | mysql –u root –p

Después de este comando, se solicitará el ingreso de la clave de la base de datos *(Enter password:)*. En este caso: *espe14*. Se ingresa la clave cada vez que se solicite para realizar los cambios en la *DB*.

>> root@ubuntu:~# echo "grant all on radius. to radius@'localhost' identified by 'radius'; flush privileges;" | mysql –u root –p*

>> root@ubuntu:~# mysql –u root –p radius < /etc/freeradius/sql/mysql/schema.sql

>> root@ubuntu:~# mysql –u root –p radius < /etc/freeradius/sql/mysql/nas.sql

>> root@ubuntu:~# mysql –u root –p radius < /etc/freeradius/sql/mysql/ippool.sql

>> root@ubuntu:~# cd /etc/freeradius

>> root@ubuntu:/etc/freeradius# nano radiusd.conf

Para incluir el módulo *sql.conf* se borra el comentario de la línea que se muestra en la siguiente figura al remover el #.

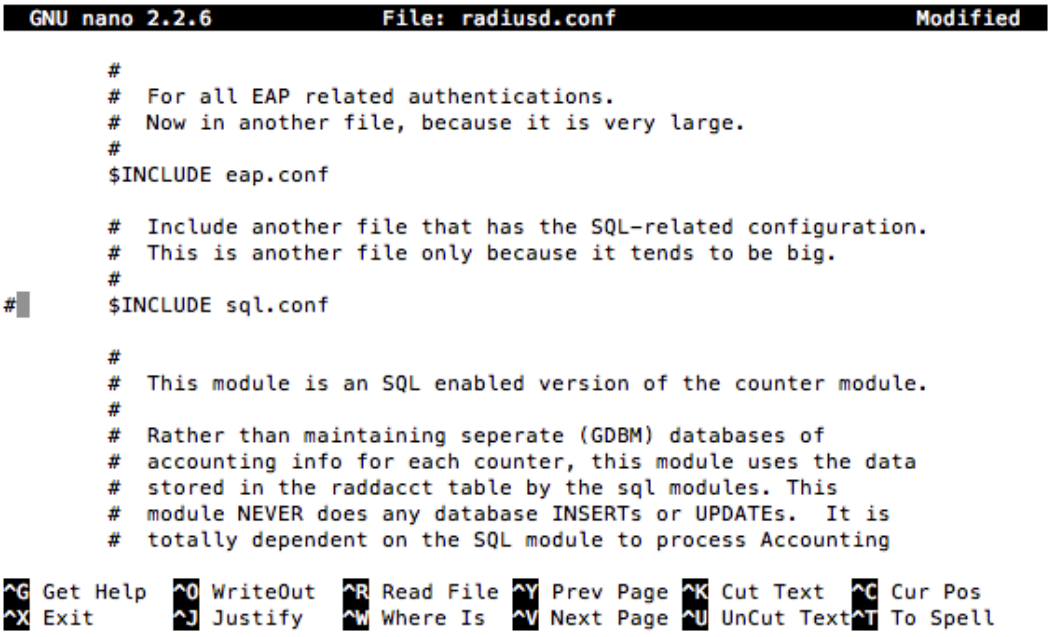


Figura. 32 Cambios en radiusd.conf

>> root@ubuntu:/etc/freeradius# nano sql.conf

Se edita el archivo *sql.conf* y se cambia el parámetro *password* a *“radius”.*

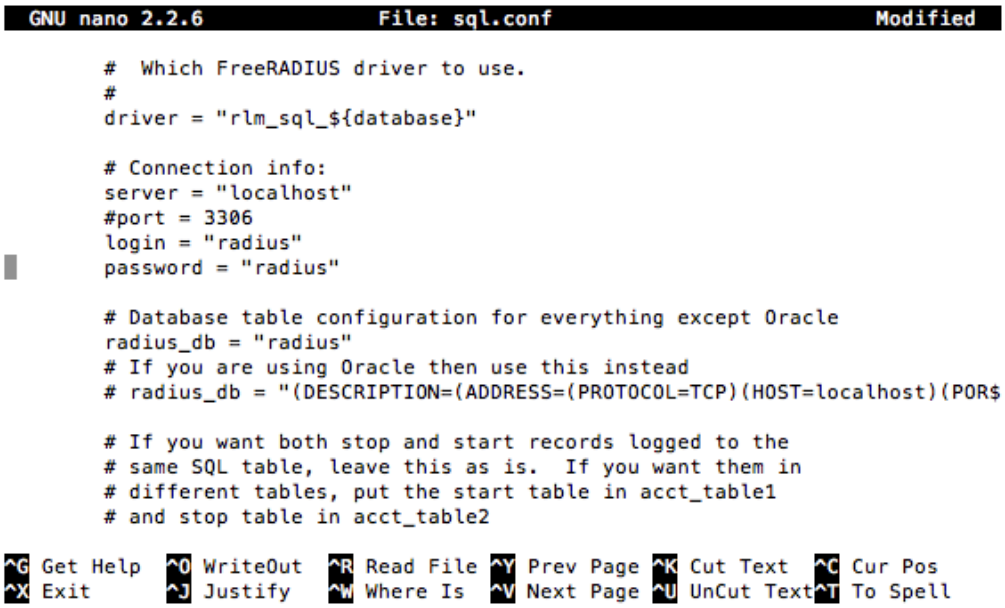


Figura. 33 Cambios en sql.conf en password

Se cambia el parámetro *readclients* a *yes*.

Figura. 34 Cambios en sql.conf en readclients

>> root@ubuntu:/etc/freeradius# cd

>> root@ubuntu:~# cd /etc/freeradius/sites-available/

>> root@ubuntu:/etc/freeradius/sites-available# nano inner-tunnel

Se Incluye el módulo *sql* en la sección de *authorized en el archivo de inner-tunnel,* des comentando la línea que se muestra en la siguiente figura al remover el #.

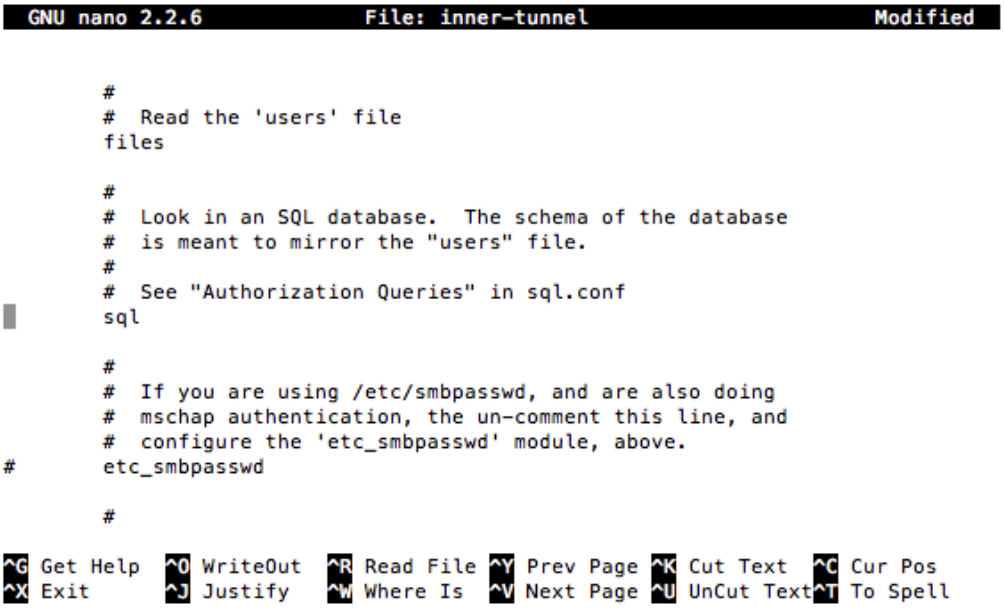


Figura. 35 Cambios en inner-tunnel en readclients

También, se incluye el módulo *sql* en la sección de *session* en el archivo de *inner-tunnel,* des comentando la línea que se muestra en la siguiente figura al remover el #.

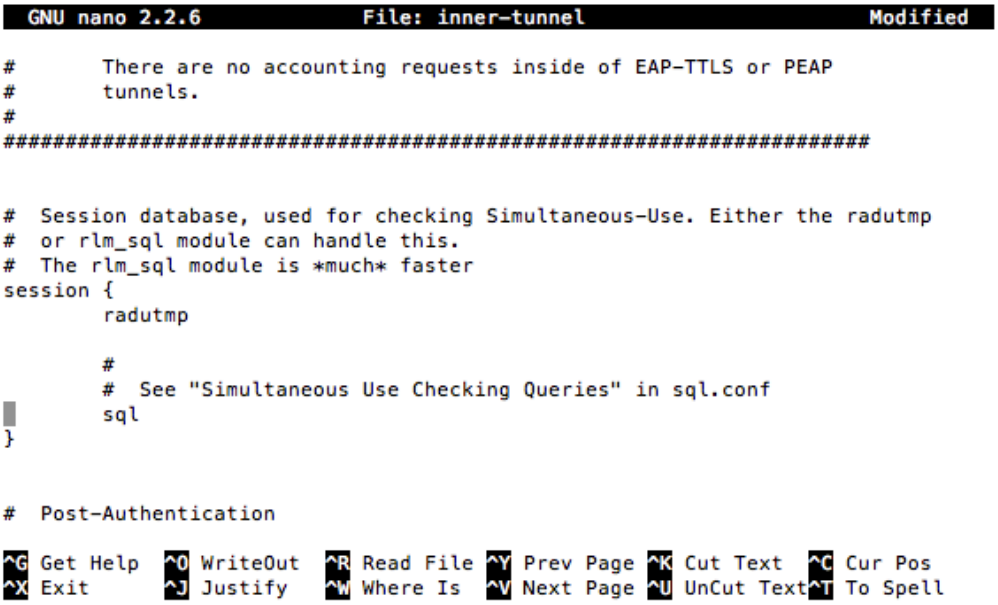


Figura. 36 Cambios en inner-tunnel en session

>> root@ubuntu:/etc/freeradius/sites-available# nano defaul

Se tiene que incluir el módulo *sql* en la sección de *authorized* en el archivo de *default*, des comentando la línea que se muestra en la siguiente figura al remover el #.

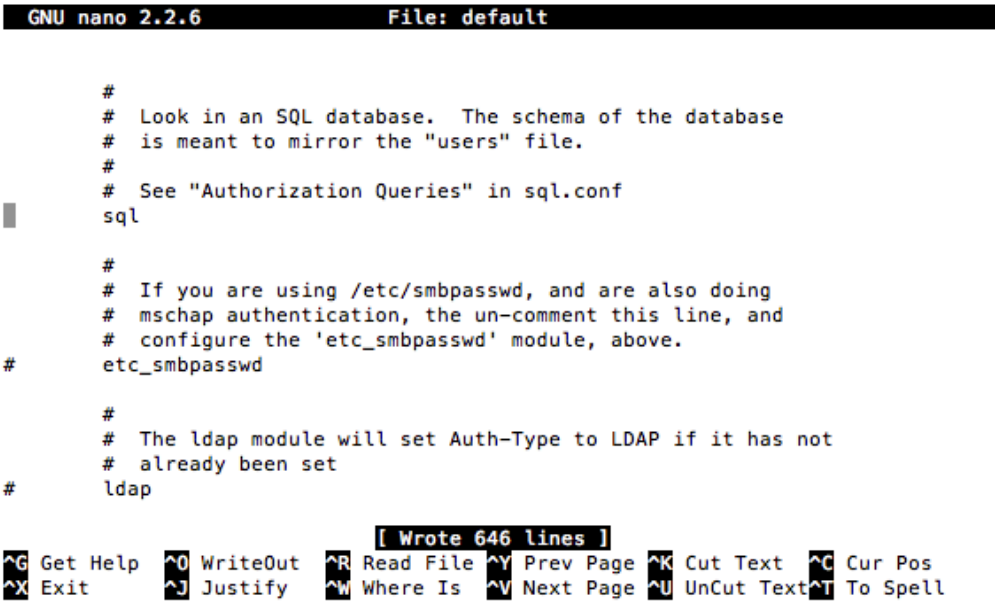


Figura. 37 Cambios en default en authorized

Incluimos el módulo *sql* en la sección de *accounting* en el archivo de *default*, des comentando la línea que se muestra en la siguiente figura al remover el #.

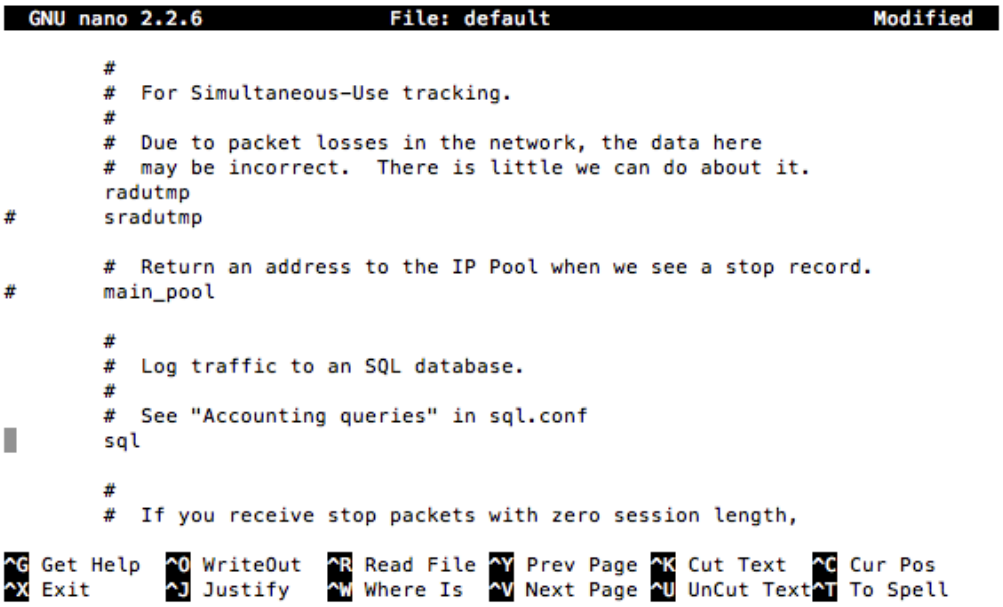


Figura. 38 Cambios en default en accounting

El módulo *sql* es habilitado en la sección de *session* en el archivo de *default,* des comentando la línea que se muestra en la siguiente figura al remover el #.

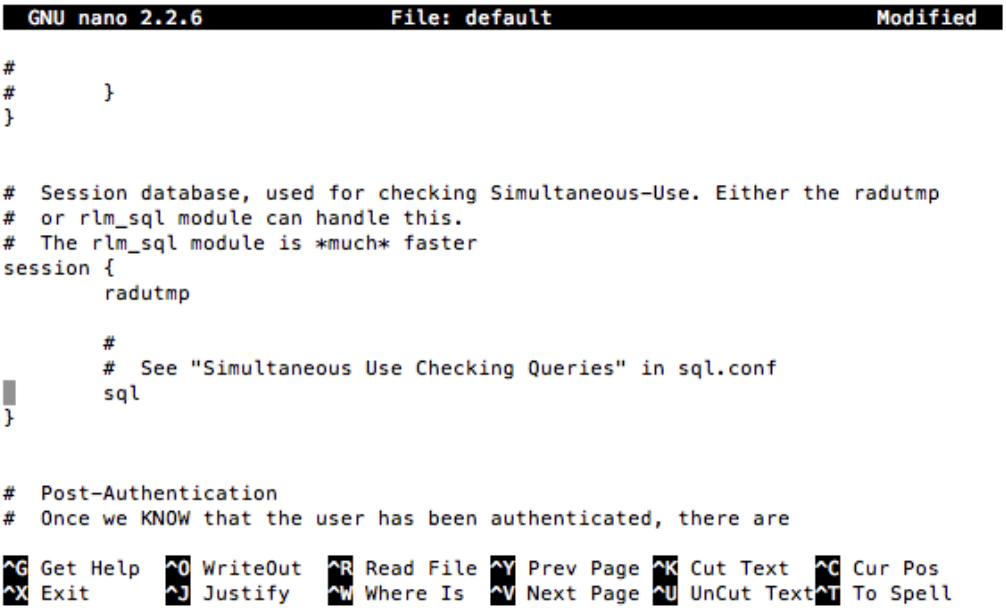


Figura. 39 Cambios en default en session

>> root@ubuntu:/etc/freeradius/sites-available# cd

>> root@ubuntu:~# cd /etc/freeradius/sql/mysql/

>> root@ubuntu:/etc/freeradius/sql/mysql# nano admin.sql

En el archivo *admin.sql*, se cambia el parámetro *PASSWORD* a *asttro14.*

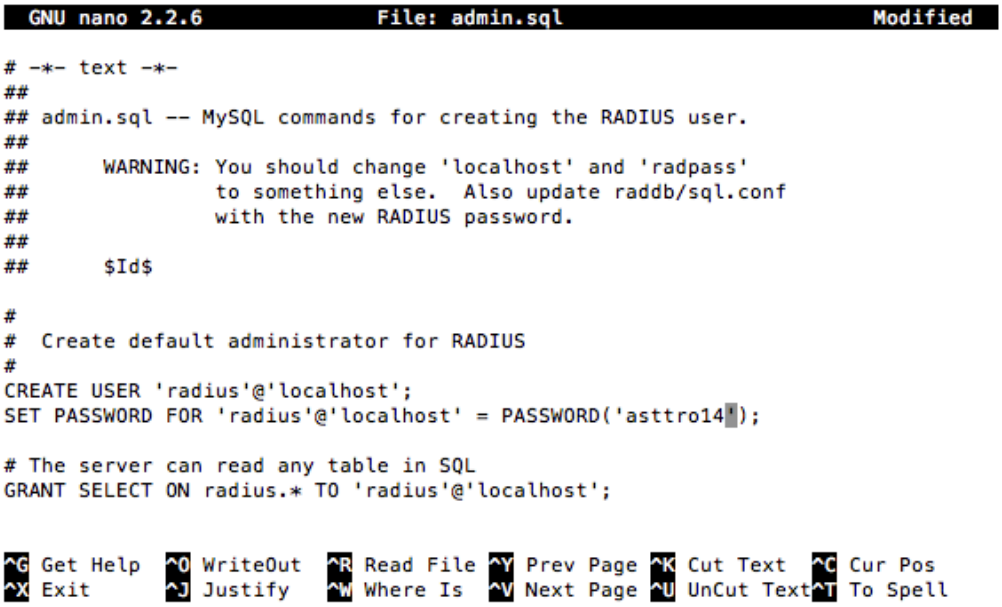


Figura. 40 Cambios en admin.sql

2.3.5 phpMyAdmin

phpMyAdmin es una herramienta de software libre escrito en PHP, tiene la intención o propósito de manejar la administración de MySQL a través de la Web.

phpMyAdmin es compatible con una amplia gama de operaciones en MySQL, MariaDB y Drizzle. Operaciones de uso frecuente (gestión de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc) que pueden realizarse mediante la interfaz del usuario, además se mantiene la capacidad de ejecutar directamente cualquier sentencia de *SQL*.

>> espe14@ubuntu:~$ sudo –i

[sudo] password for espe14: espetesis14

>> root@ubuntu:~# apt-get install apache2 phpmyadmin

Se selecciona las opciones que se muestran y <Ok>

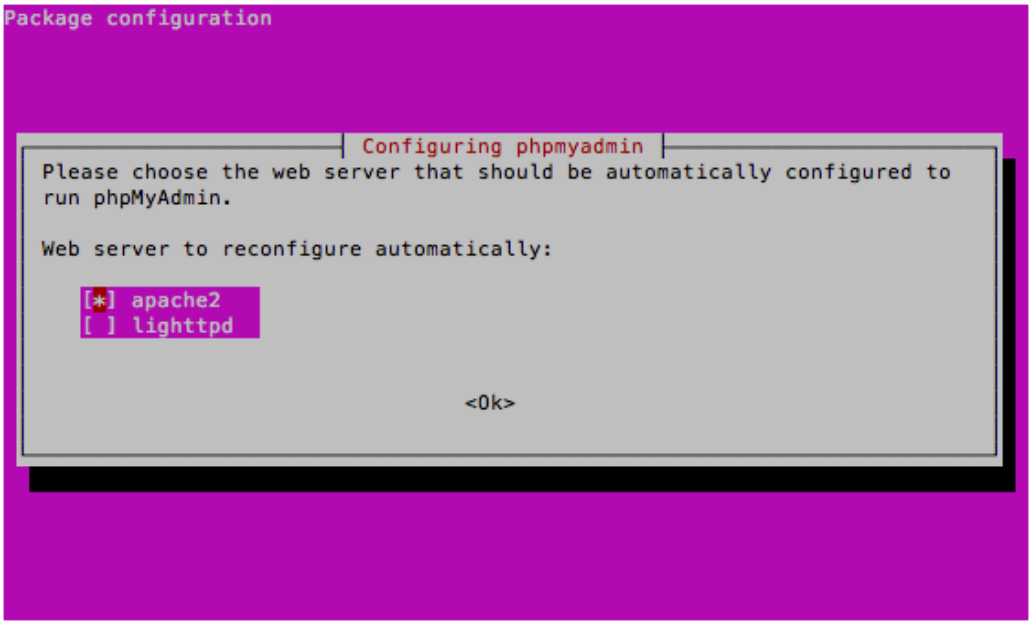


Figura. 41 Configuración phpmyadmin

Se despliega un mensaje sugiriendo la instalación de una base de dato, sin embargo como ya se han creado en la sección de MySQL se descarta y se selecciona <No>

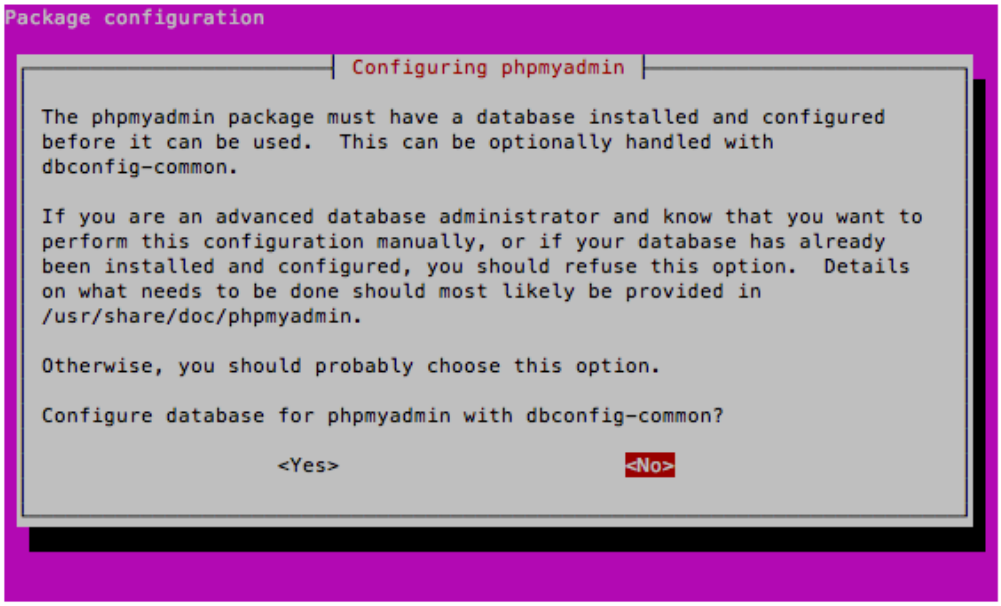


Figura. 42 Descartar la creación de DB

Se verifica la instalación de *phpMyAdmin* ingresando al navegador. Se pone: la dirección *IP* del servidor/phpmyadmin (192.168.1.12/phpmyadmin/) y se ingresa el *Username (root)* y *Password (espe14)*. Esto se muestra a continuación.

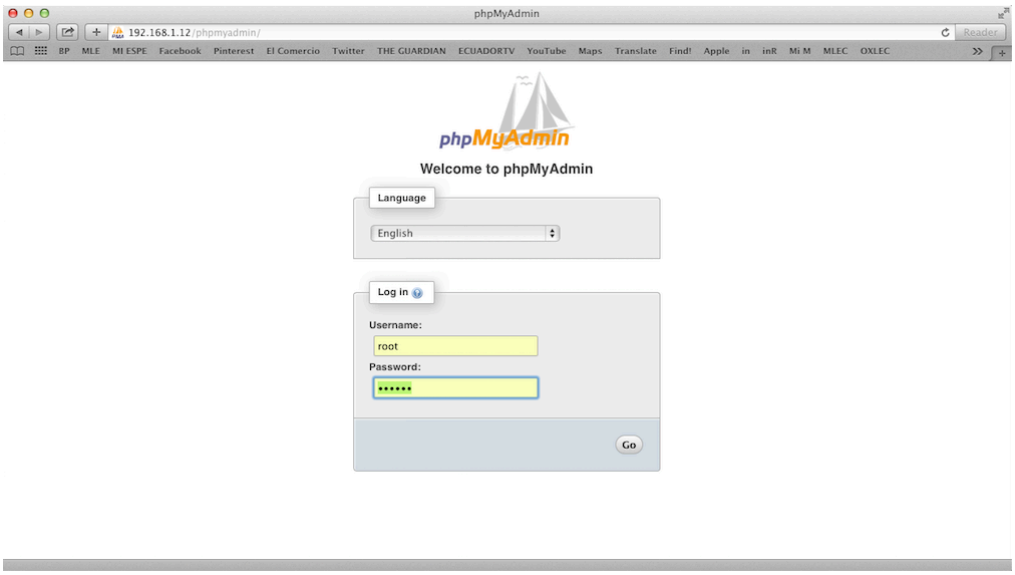


Figura. 43 Ingreso a phpmyadmin

La página de inicio muestra información la base de datos creada en el capítulo de MySQL, si todo se hizo correctamente, esta debería mostrarse

en el lado izquierdo con el nombre de *radius* junto con todas las tablas necesarias para administrar el servidor *RADIUS*.

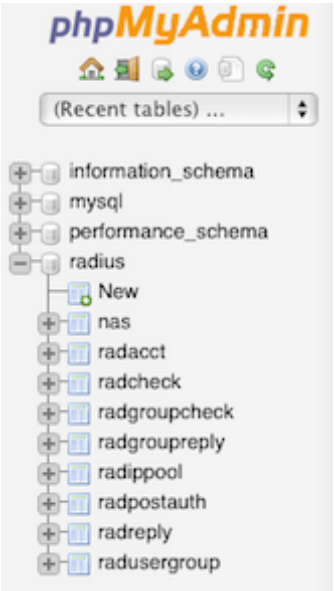


Figura. 44 tablas y DB radius

Adicional, se puede ver información útil sobre nuestra base de datos en el lado derecho, como por ejemplo: el tipo y versión instalada de la base de datos o la del servidor web, etc.

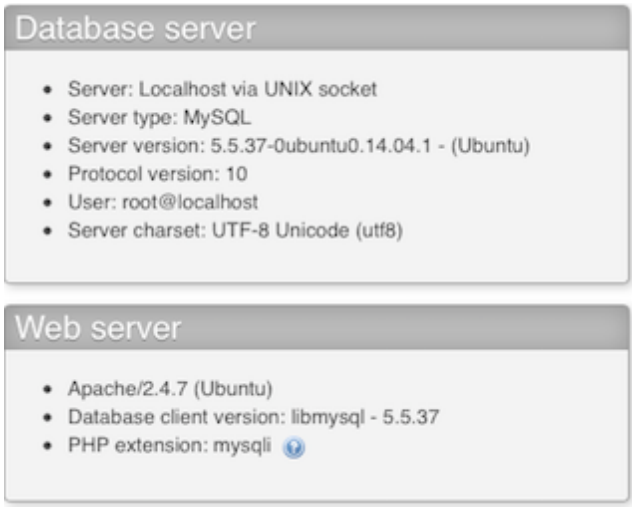


Figura. 45 Información del Servidor DB y web

Desde *phpMyAdmin* se va a administrar nuestra base de datos *MySQL* para gestionar el servidor de autenticación.

A continuación, se va a establecer comunicación con el equipo NAS *(Network Access Server). NAS* es un dispositivo que proporciona un cierto nivel de acceso a una red más grande. Un *NAS* utilizando una infraestructura *RADIUS* es también un cliente *RADIUS*, enviando solicitudes de conexión y mensajes de cuentas a un servidor *RADIUS* para la autenticación, autorización y contabilidad.

Los equipos cliente, como los ordenadores portátiles inalámbricos u otros equipos con sistemas operativos, no son clientes de *RADIUS*. Clientes de *RADIUS* son servidores de acceso a la red, tales como puntos de acceso inalámbrico *(AP)*, conmutadores compatibles con *802.1X*, redes privadas virtuales *(VPN)* y servidores *dial-up*, esto porque utilizan protocolos *RADIUS* para comunicarse con servidores *RADIUS*.

Después de aclarar lo que es un equipo *NAS* en una red, la información que se debe llenar para este escenario de prueba se muestra en la siguiente figura:

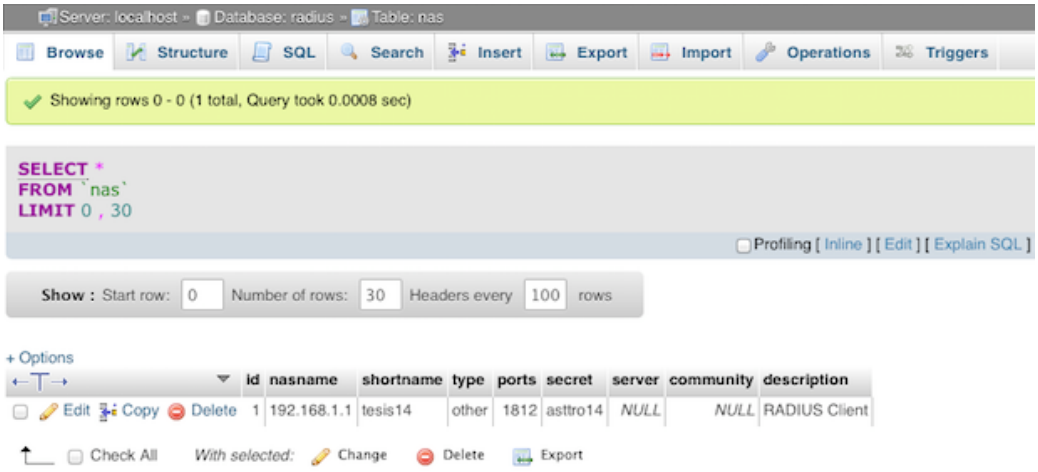


Figura. 46 Parámetros para NAS

Se selecciona la base de datos *radius*, se escoge la tabla *nas* y se puebla la información requerida como se muestra en la figura de *Parámetros para NAS*.

Se continúa con la tabla de *radcheck* en la base de datos *radius*. Es aquí donde se establecen los usuarios que tiene permitido el ingreso a nuestra red. Se muestra en la siguiente figura.

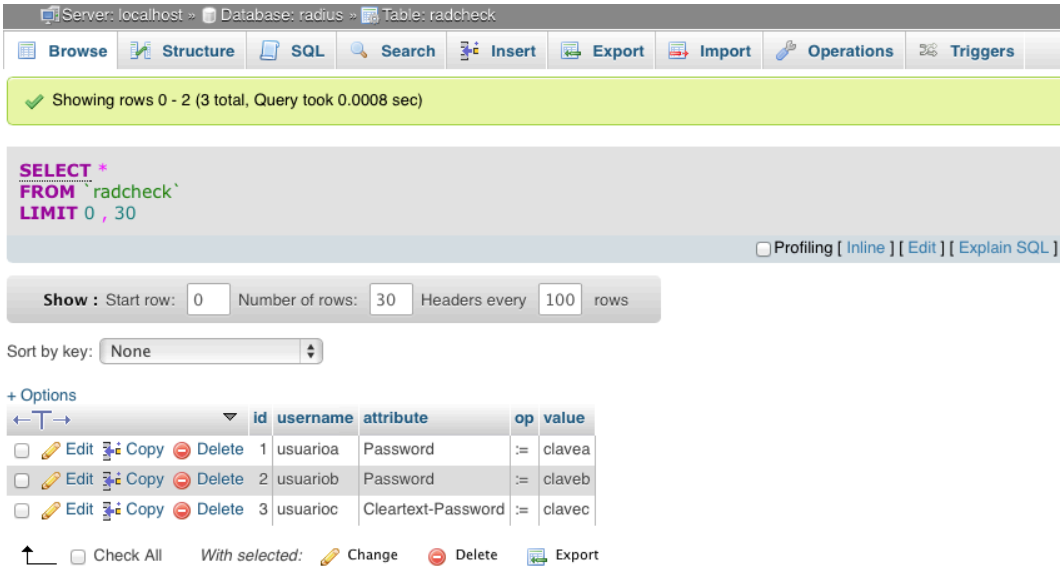


Figura. 47 Parámetros para radcheck

Se comprueba que los datos ingresados para en la tabla *radcheck* funcionan correctamente. Se utiliza como ejemplo, el *username: usuarioa* con *value:* clavea y *atributo: Password*

En la línea de comando se ingresa la siguiente instrucción:

>> espe14@ubuntu:~$ sudo –i
[sudo] password for espe14: espetesis14
>> root@ubuntu:~# /etc/init.d/freeradius start
>> root@ubuntu:~# radtest usuarioa clavea localhost 1812 testing123

Si la autenticación de este usuario es exitosa *(Access-Accept)*, entonces se tiene la siguiente figura que demuestra que todo está funcionando correctamente:

```
Sending Access-Request of id 62 to 127.0.0.1 port 1812
        User-Name = "usuarioa"
        User-Password = "clavea"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1812
        Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=62, length=20
```

Figura. 48 Comprobación de acceso con radtest

2.4 AP (Real) / OpenWrt

El punto de acceso *(AP)* de nuestra red, es fundamental para cumplir con el estándar *802.11i*. Este es, como se ha estudiado, el autenticador. Para el escenario de prueba, se ha propuesto la utilización de la tarjeta *Alix2D2* para este fin.

Esta tarjeta, tiene instalado un sistema operativo llamado OpenWrt de distribución libre, que está basado en Linux para hacer el enrutamiento del tráfico en una red, este OS *(Operative System)* ha sido optimizado para ser lo más liviano posible con la finalidad de que pueda funcionar en espacios de memoria limitados, como en este caso el *AP*.

La características más atractivas del OpenWrt, es que nos permite la libertad de instalar más de 3500 paquetes de software alternativos a través del sistema de administración de paquetes *(opkg).* Como se puede apreciar, esto nos concederá más adelante, la posibilidad de instalar un software analizador de paquetes, detector de redes e intrusos para redes *LAN* inalámbricas *802.11* en la capa de enlace, este software se llama Kismet.

Figura. 49 Tarjeta Alix2d2 con sus elementos

Los siguientes comandos son recomendados para el óptimo funcionamiento de las tarjeta *wireless*, además de la actualización del software OpenWrt que está instalado en el *AP*.

Primero, se ingresa al punto de acceso por medio del protocolo de *ssh.*

>> ssh root@192.168.1.1

root: es el nombre del usuario del AP.
192.168.1.1: es la dirección IP del *access point*

Se solicitará la clave (*password*) del *AP*, se ingresa la misma y se da *enter*.

```
Last login: Sun Jun 22 12:50:41 on ttys000
Andres-MacBook-Pro:~ andrestroya$ ssh root@192.168.1.1
root@192.168.1.1's password:
```

Figura. 50 SSH al AP

Al ingresar, se tiene la siguiente figura:

```
BusyBox v1.15.3 (2011-07-14 17:03:04 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 Backfire (10.03.1-RC5, r27608) ----------------------------
  * 1/3 shot Kahlua     In a shot glass, layer Kahlua
  * 1/3 shot Bailey's   on the bottom, then Bailey's,
  * 1/3 shot Vodka      then Vodka.
 -----------------------------------------------------
root@OpenWrt:~#
```

Figura. 51 AP login vía terminal

>>root@OpenWrt:~# opkg update

>>root@OpenWrt:~# opkg install kmod-madwifi

>>root@OpenWrt:~#rm –f /etc/config/wireless; wifi detect > /etc/config/wireless

El esquema a instalar en esta dispositivo es el siguiente:

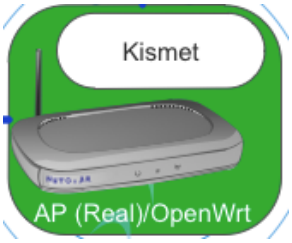


Figura. 52 AP (Real)/OpenWrt

2.4.1 Configuración del AP

Ingresando al navegador se pone: la dirección *IP* del *AP* (192.168.1.1) y se ingresa el *Username (root)* y *Password (espetesis2012)* como se muestra continuación.

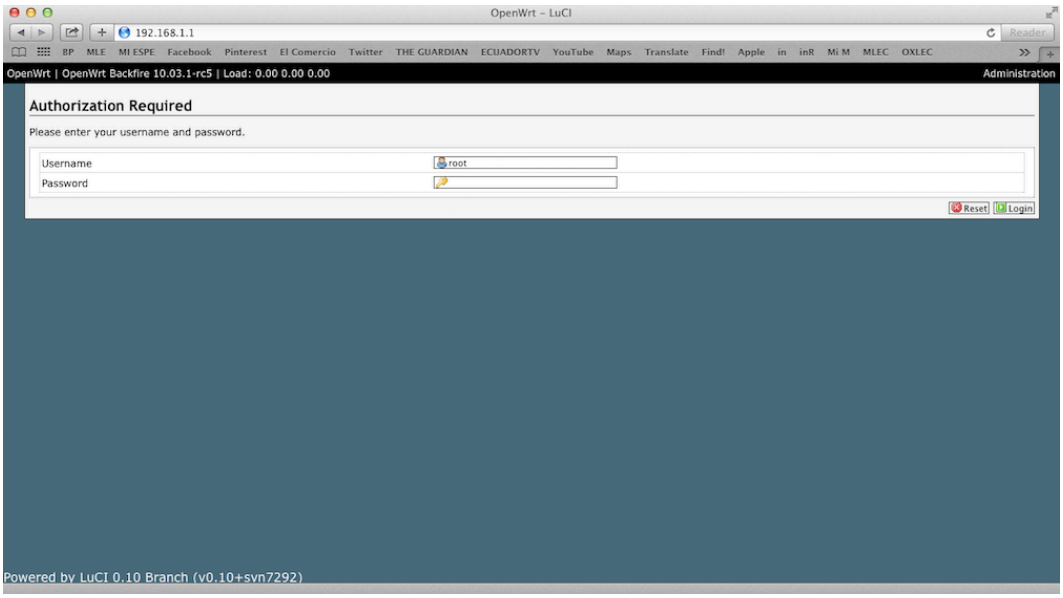


Figura. 53 Ingreso a OpenWrt via web

Configuración de la interfaz *WAN (eth1)*:

Protocol: DHCP

DNS-Server: 8.8.8.8

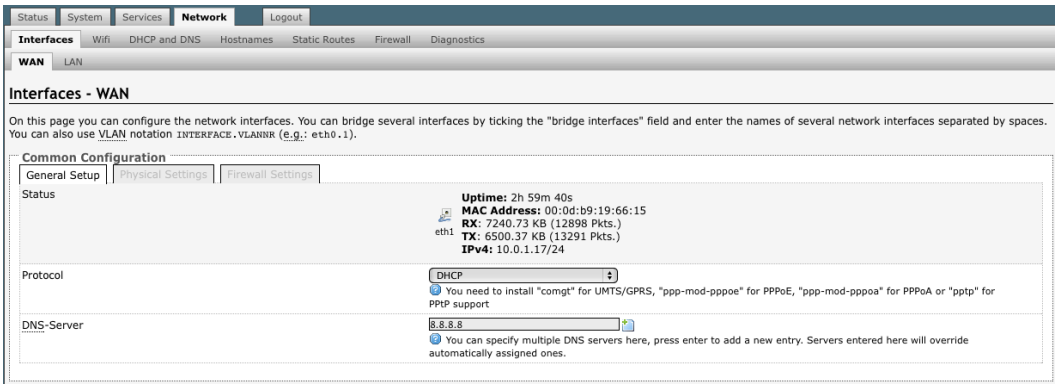


Figura. 54 Interfaz-WAN

Configuración de la interfaz *LAN (br-lan)*:

Protocol: static

IPv4-Address: 192.168.1.1

IPv4-Netmask: 255.255.255.0

DNS-Server: 8.8.8.8

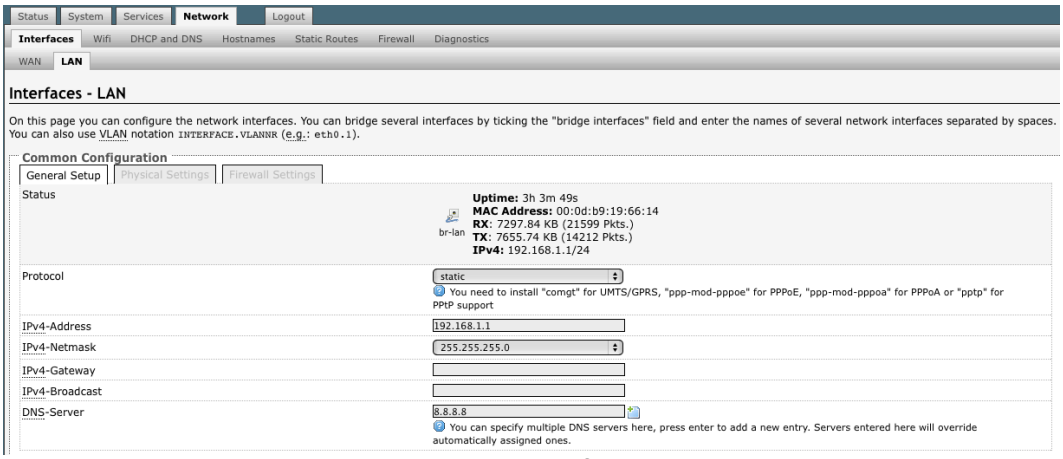


Figura. 55 Interfaz-LAN

Configuración de la interfaz Wireless Master (wifi0) :

ESSID: tesis14

Mode: Access Point

Network: lan

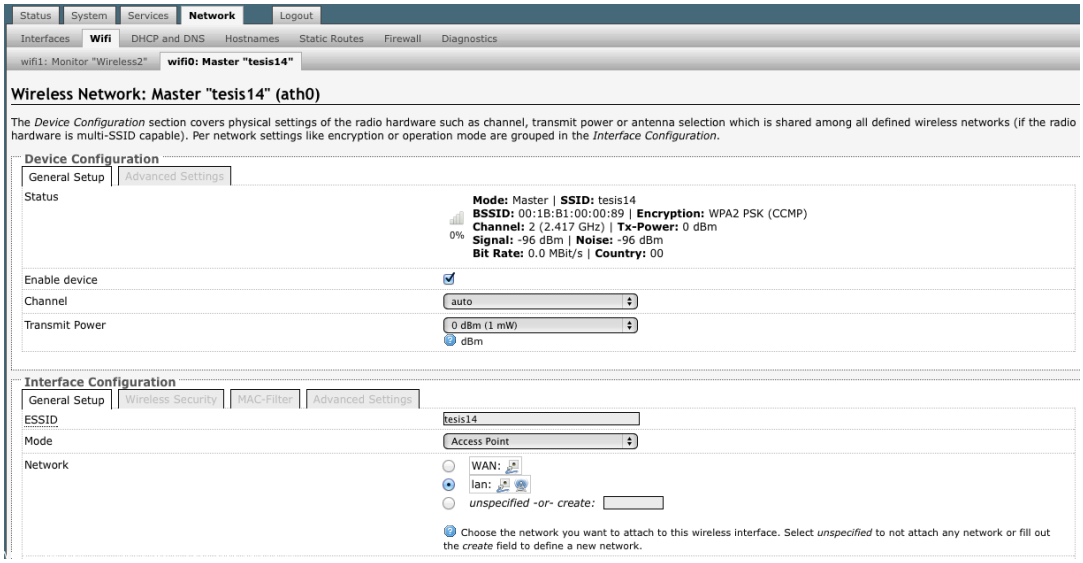


Figura. 56 Interfaz-Wireless Master-General Setup

Encryption: WPA2-EAP

Radius-Server: 192.168.1.12

Radius-Port: 1812

Key: asttro14

NAS ID: 1812

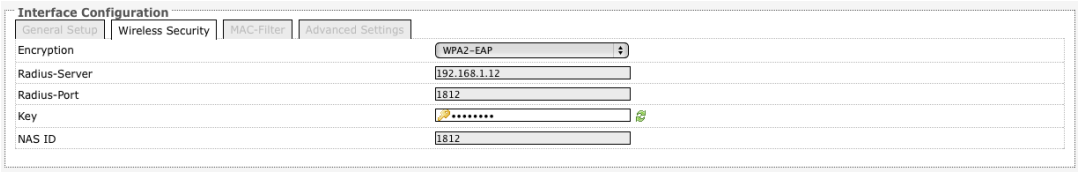


Figura. 57 Interfaz - Wireless Master – Wireless Security

Configuración de la interfaz *Wireless Monitor (wifi1)* :

ESSID: wireless2
Mode: Monitor
Network: unspecified-or-create

Esta interfaz en modo monitor nos servirá para la captura de tramas y paquetes que serán analizadas mediante el software *Kismet-Drone.*

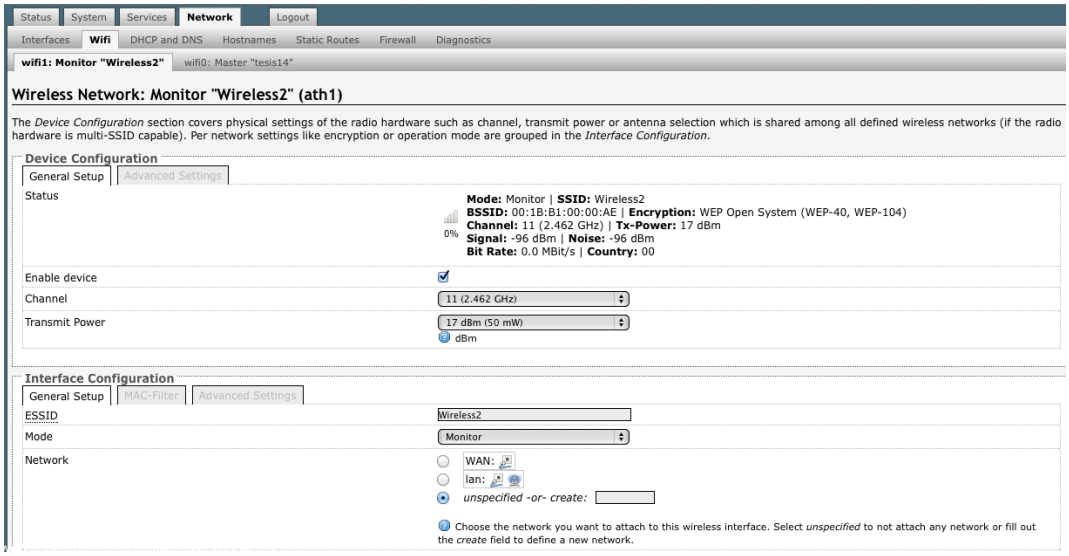


Figura. 58 Interfaz-Wireless Monitor-General Set up

Configuración del firewall para habilitar servicios de internet a la red:

Name: Internet
Input: accept
Output: accept
Forward: accept
Covered networks: WAN, lan

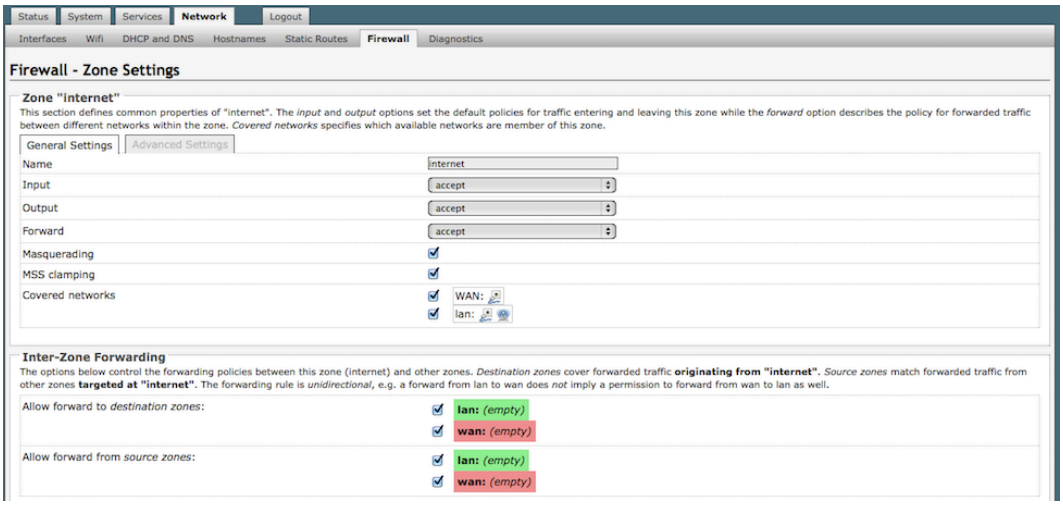


Figura. 59 Configuración Firewall

2.4.2 Kismet

Kismet se diferencia de otros detectores de red inalámbrica en el trabajo de forma pasiva. Es decir, sin enviar ningún paquete de autenticación, es capaz de detectar la presencia de *AP* y clientes *wireless*, además de la capacidad de asociarlos entre ellos. También, es la herramienta más utilizada para el monitoreo inalámbrico de código abierto.

Este software tiene la capacidad de registrar todos los paquetes de redes inalámbricas disponibles, capturarlos y guardarlos en un formato de archivo compatible con Wireshark o Airsnort. A más de esto, permite determinar el nivel de encriptación usado por los *AP* detectados.

Kismet se puede utilizar de tres maneras: La primera, como sensor *Kismet-Drone* para recoger los paquetes y enviarlos hacia un servidor para su análisis; La segunda, *Kismet-Server* puede ser utilizado en conjunto con *Kismet-Drone*, esto para el análisis de paquetes capturados y almacenados de las redes inalámbricas y Finalmente, *Kismet-Client* que se comunica con el servidor para visualizar por medio de una interfaz amigable la información que el servidor recoge.

La instalación de *Kismet-Drone* en el OpenWrt se realiza con los siguientes comandos:

>>root@OpenWrt:~# opkg update

>>root@OpenWrt:~# opkg install kismet-drone

>>root@OpenWrt:~# vi kismet_drone.conf

versión=newcore.1

servername=drone1

dronelisten=tcp://192.168.1.1:2502

droneallowedhosts=192.168.1.12

dronemaxclients=1

droneringlen=65535

gps=false

ncsource=wifi1:type=Madwifi

channelvelocity=5

channellist=IEEE80211b:1:3,6:3,11:3,2,7,3,8,4,9,5,10

```
# Kismet drone config file

version=newcore.1

# Name of drone server (informational)
servername=drone1

# Drone configuration
# Protocol, interface, and port to listen on
dronelisten=tcp://192.168.1.1:2502
# Hosts allowed to connect, comma separated.  May include netmasks.
# allowedhosts=127.0.0.1,10.10.10.0/255.255.255.0
droneallowedhosts=192.168.1.12
# Maximum number of drone clients
dronemaxclients=1
droneringlen=65535

# Do we have a GPS?
gps=false
```

Figura. 60 kismet_drone.conf

>>root@OpenWrt:~# kismet_drone

```
root@OpenWrt:~# kismet_drone
ERROR: Kismet was started as root, NOT launching external control binary.  This
       is NOT the preferred method of starting Kismet as Kismet will continue
       to run as root the entire time.  Please read the README file section
       about Installation & Security and be sure this is what you want to do.
INFO: Reading from config file /etc/kismet/kismet_drone.conf
INFO: Plugin system disabled by Kismet configuration file or command line
INFO: Setting drone connection buffer to 65535 bytes
INFO: Kismet will attempt to hop channels at 5 channels per second unless
      overridden by source-specific options
INFO: No specific sources named on the command line, sources will be read from
      kismet.conf
INFO: Using default channel list 'IEEE80211b' on source 'wifi1'
INFO: Source 'wifi1' will attempt to create and use a monitor-only VAP instead
      of reconfiguring the main interface
ERROR: Packetsource::MadWifi - Unknown source type 'Madwifi'.  Will treat it as
       auto radio type
INFO: Created source wifi1 with UUID 479754be-3ae3-11df-bbb7-a104921ee001
INFO: Will attempt to reopen on source 'wifi1' if there are errors
INFO: Created TCP listener on port 2502
INFO: Starting GPS components...
INFO: GPS support disabled in kismet.conf
INFO: Kismet drone starting to gather packets
INFO: Madwifi source wifi1 created monitor-mode VAP wifi1::kis0.
ERROR: Source 'kis0' doesn't have mac80211 support, disabling VAP creation of
       default monitor mode VAP
INFO: Interface 'kis0' is already marked as being in monitor mode, leaving it
      as it is.
INFO: Started source 'wifi1'
```

Figura. 61 Ejecución de kismet_drone

La instalación de *Kismet-Server + Client* en el servidor *RADIUS* se realiza con los siguientes comandos:

>> espe14@ubuntu:~$ sudo –i

[sudo] password for espe14: espetesis14

>> root@ubuntu:~# apt-get install kismet

Se selecciona <*Ok*> en la ventana que se muestra a continuación:

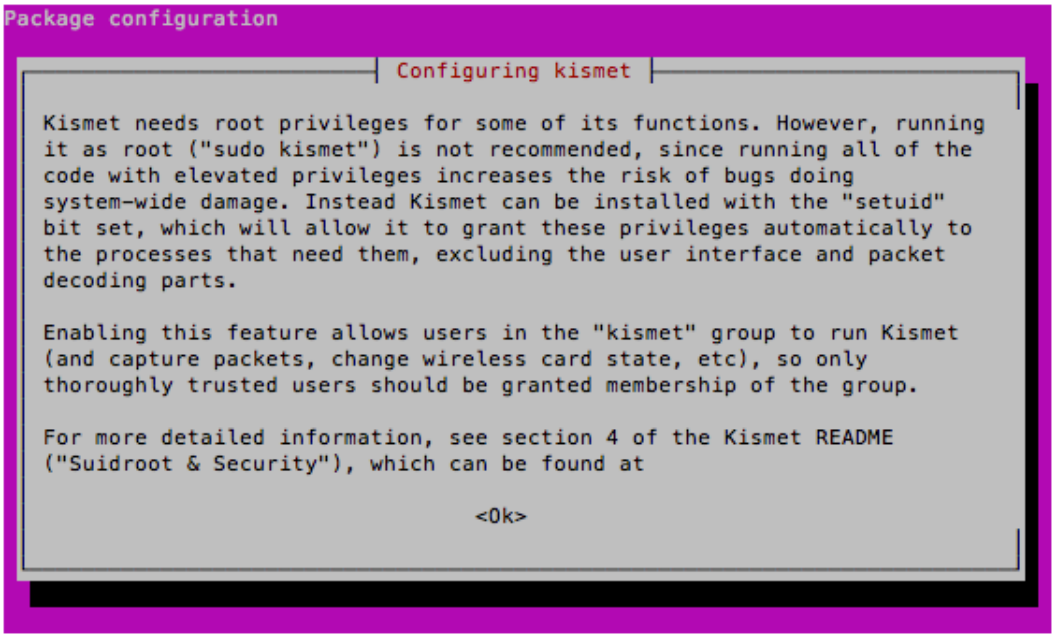


Figura. 62 Configuración Kismet-Server

Se habilita el usuario al grupo Kismet para su utilización:

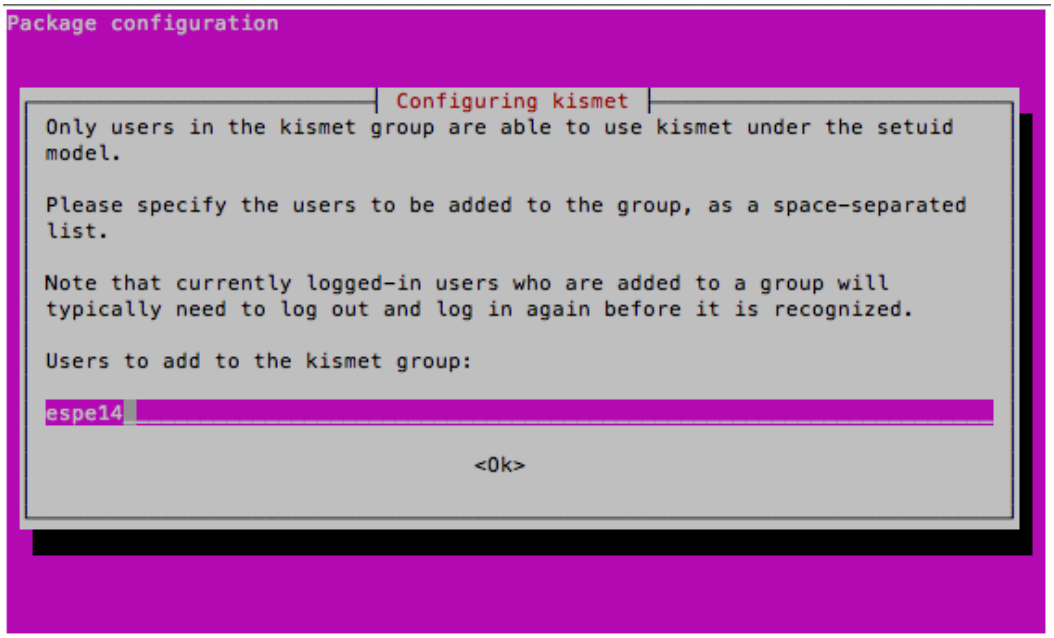


Figura. 63 Configuración Kismet-Server usuario

Se realiza las configuraciones necesarias para establecer la comunicación entre *Kismet – Server* y *Kismet-Drone:*

>> root@ubuntu:~# cd /etc/kismet/

>> root@ubuntu: /etc/kismet# nano kismet_drone.conf

versión=newcore.1

servername=drone1

dronelisten=tcp://192.168.1.1:2502

droneallowedhosts=192.168.1.12

dronemaxclients=1

droneringlen=65535

gps=false

ncsource=eth0

channelvelocity=5

channellist=IEEE80211b:1:3,6:3,11:3,2,7,3,8,4,9,5,10

```
  GNU nano 2.2.6          File: kismet_drone.conf

# Kismet drone config file

version=newcore.1

# Name of drone server (informational)
servername=drone1

# Drone configuration
# Protocol, interface, and port to listen on
dronelisten=tcp://192.168.1.1:2502
# Hosts allowed to connect, comma separated.  May include netmasks.
# allowedhosts=127.0.0.1,10.10.10.0/255.255.255.0
droneallowedhosts=192.168.1.12
# Maximum number of drone clients
dronemaxclients=1
droneringlen=65535

# Do we have a GPS?
gps=false

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Figura. 64 Configuración kismet_drone.conf

>> root@ubuntu: /etc/kismet# nano kismet.conf

servername=Server

allowplugins=false

ncsource=drone:host=192.168.1.1,port=2502

listen=tcp://192.168.1.12:2501

allowedhosts=192.168.1.0/24

gps=false

writeinterval=60

```
  GNU nano 2.2.6               File: kismet.conf                      Modified

# Kismet config file
# Most of the "static" configs have been moved to here -- the command line
# config was getting way too crowded and cryptic.  We want functionality,
# not continually reading --help!

# Version of Kismet config
version=2009-newcore

# Name of server (Purely for organizational purposes)
# If commented out, defaults to host name of system
servername=Server

# Prefix of where we log (as used in the logtemplate later)
# logprefix=/some/path/to/logs

# Do we process the contents of data frames?  If this is enabled, data
# frames will be truncated to the headers only immediately after frame type
# detection.  This will disable IP detection, etc, however it is likely
# safer (and definitely more polite) if monitoring networks you do not own.

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Figura. 65 Configuración kismet.conf

>> root@ubuntu: /etc/kismet# cd

>> root@ubuntu:~# kismet

```
~ Kismet Sort View Windows
   Name                T C  Ch  Pkts  Size                          Kismet
                                                                     Not
[ --- No networks seen --- ]                                         Connected
  MAC               Type     Freq  Pkts  Size Manuf

[ --- No clients seen --- ]

No GPS info (GPS not connected)
 0          ┌─Start Kismet Server──────────────────────────────────┐
            Automatically start Kismet server?
            Launch Kismet server and connect to it automatically.
            If you use a Kismet server started elsewhere, choose
            No and change the Startup preferences.
 0 ─────────          [ No ]                    [ Yes ]

                                                    Data
INFO: Welcome to the Kismet Newcore Client... Press '`' or '~' to ac
ERROR: Could not connect to Kismet server 'localhost:2501'
       (Connection refused) will attempt to reconnect in 5 seconds.
ERROR: Could not connect to Kismet server 'localhost:2501'
       (Connection refused) will attempt to reconnect in 5 seconds.
```

Figura. 66 Ejecución de Kismet Server [yes]

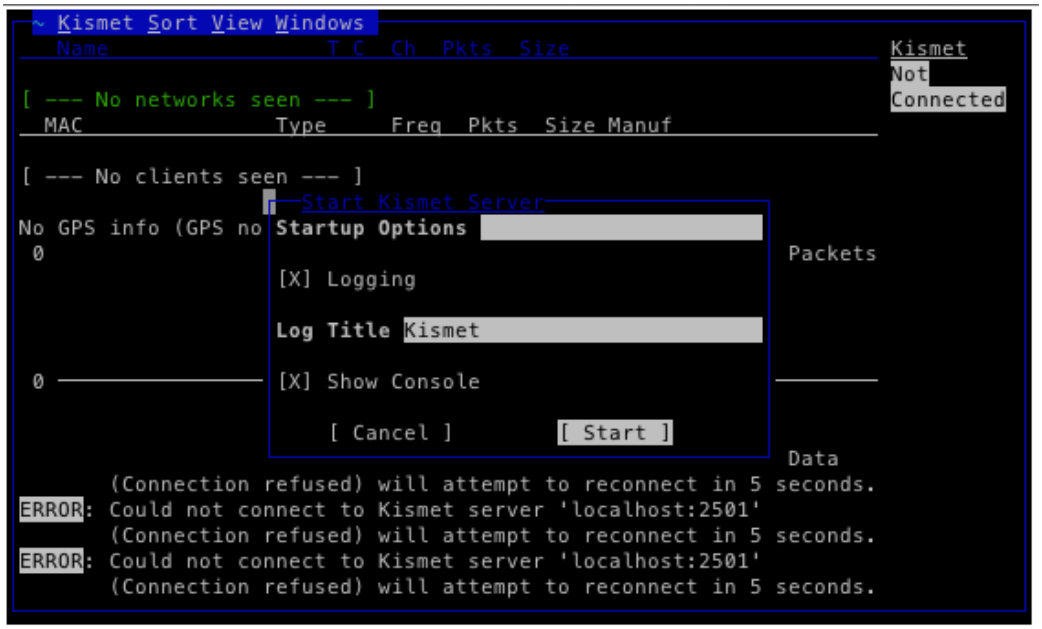


Figura. 67 Ejecución de Kismet Server [Start]

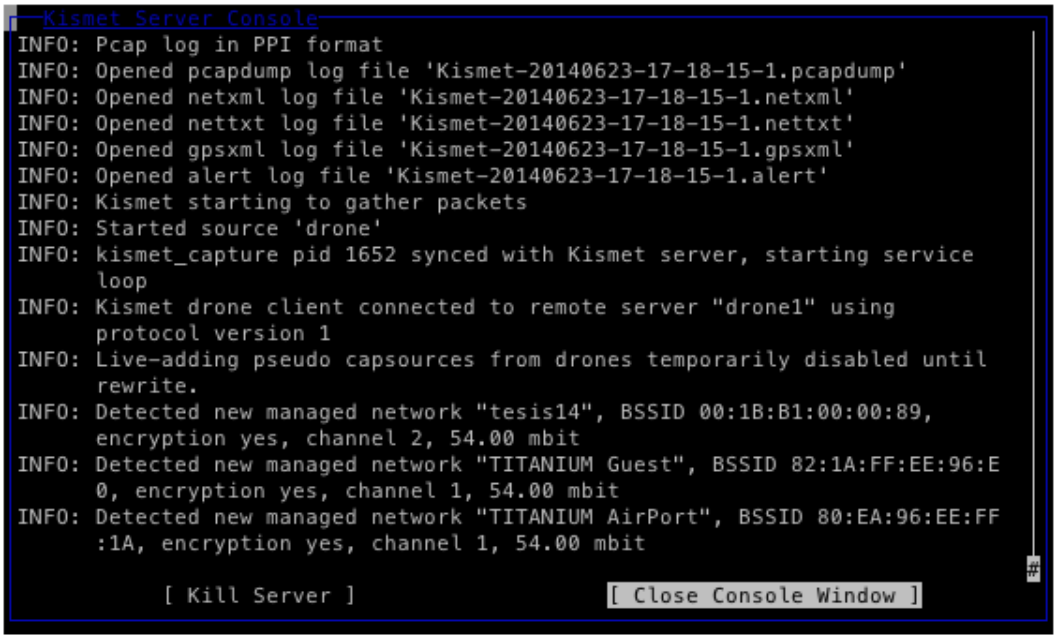


Figura. 68 Ejecución de Kismet Server [Close Console Window]

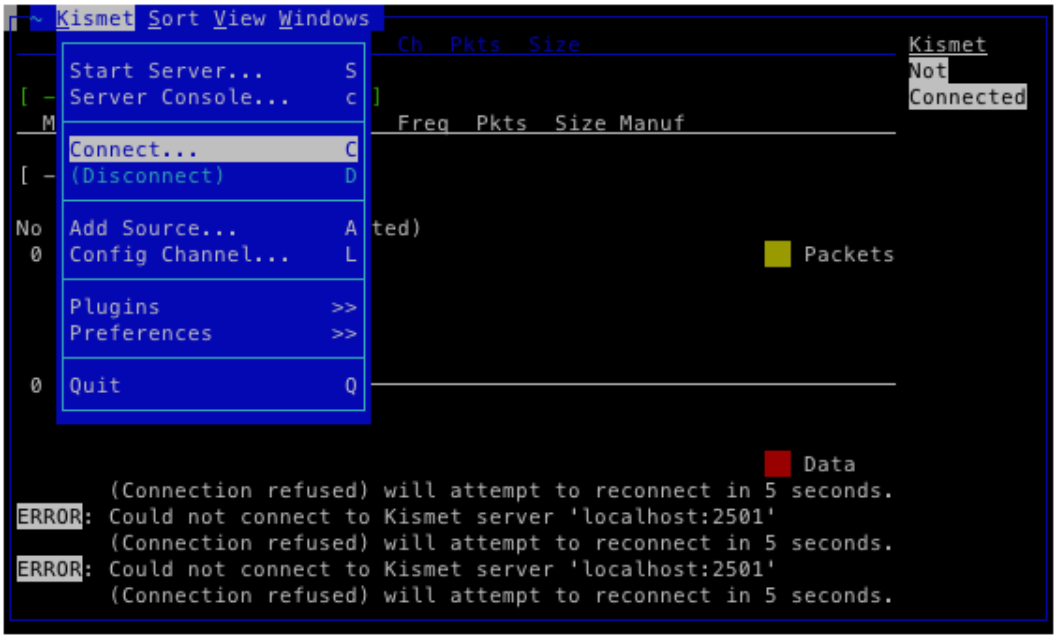


Figura. 69 Ejecución de Kismet Server Connect

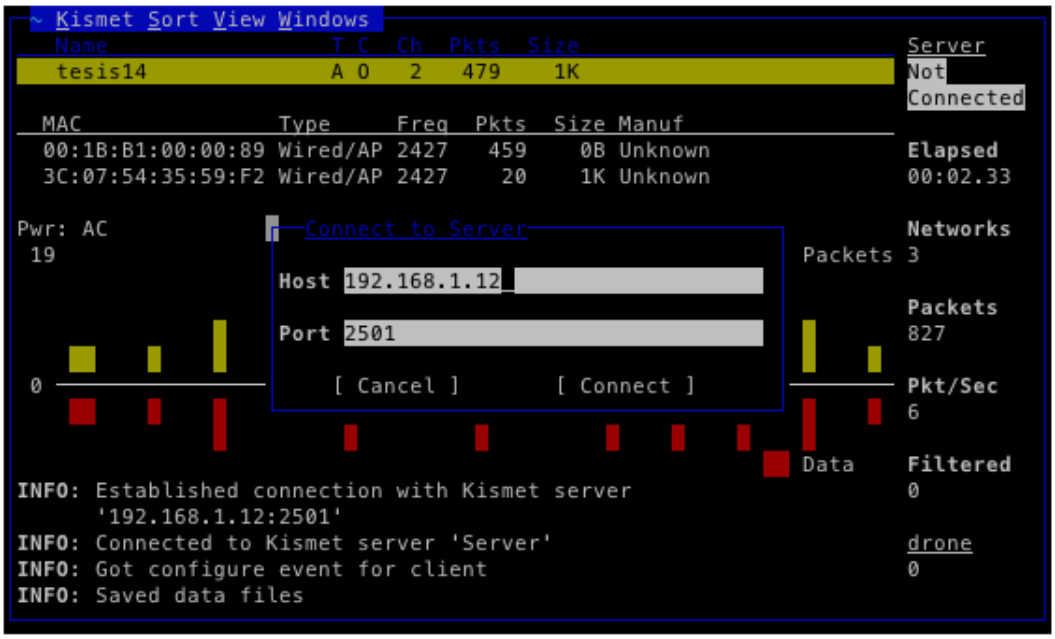


Figura. 70 Ejecución de Kismet Server Connect to Server [Connect]

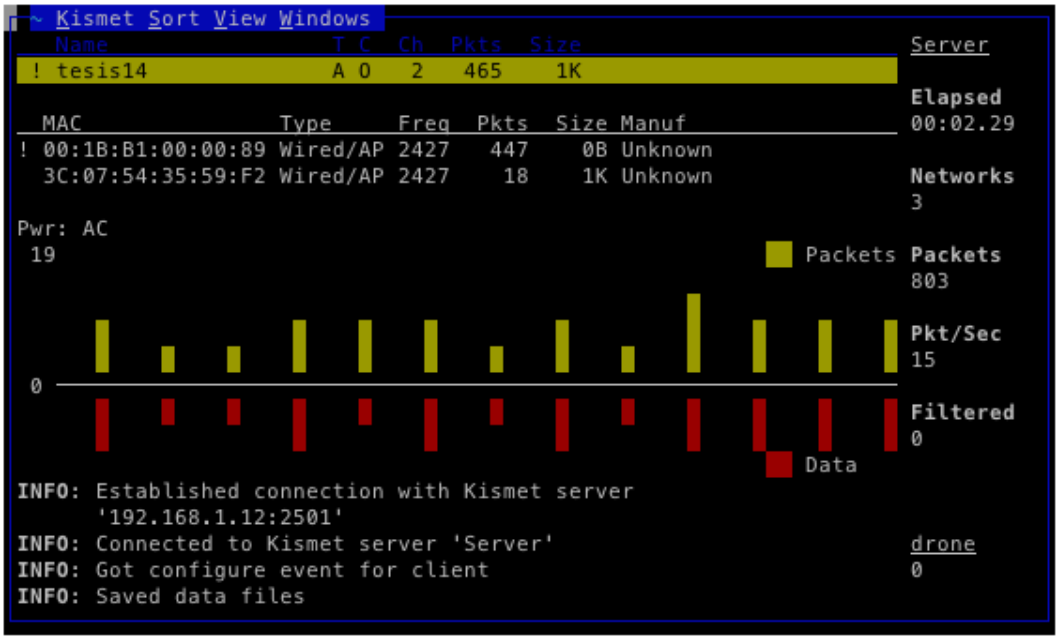


Figura. 71 Conexión Kismet Server establecida

Tabla. 3

Detalle de los componentes en la ventana Kismet

Name	Significado	T	Significado	C	Significado	Ch	Pkts	Size
[!]	Actividad en los últimos 3 segundos	[A]	Punto de acceso	[Y]	Encriptación WEP	Canal de operación de la red	Numero de paquetes capturados	Tamaño de los paquetes capturados
[.]	Actividad en los últimos 6 segundos	[H]	Modo ad-hoc	[N]	Sin encriptación			
[ ]	No hay actividad	[G]	Grupo de redes wireless	[O]	Otro tipo de encriptación			
	Encriptada	[P]	Dispositivo en modo "probe request"					
	Sin encriptación							
	Propio gateway							

2.5 BT5r3–Hacker–VM2

BackTrack es una distribución de *GNU/Linux*, pensada y diseñada para la auditoría relacionada con la seguridad informática en general. Esta herramienta nos permite analizar el comportamiento de la red en el caso de un escenario de prueba normal y además, realizar los ataques para probar y vulnerar la seguridad del estándar *802.11i* en el caso del escenario de prueba intrusivo.

Para esta máquina virtual, se instala la versión de BackTrack (BT5r3) distribuida por Ubuntu (10.04.3 LTS) de 32 bits de nombre código: *lucid.* Una

vez instalada, se debe ingresar los siguientes comandos para ingresar a BT5:

bt login: root
Password: toor
>>root@bt:~# startx

Figura. 72 Ventana de inicio BT5

El esquema a preparar en esta máquina virtual es el siguiente:

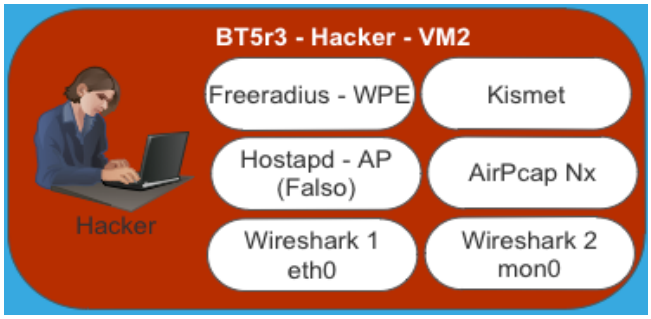


Figura. 73 BT5r3-Hacker-VM2

BackTrack proporciona a los usuarios un fácil acceso a una colección completa y amplia de herramientas relacionadas con la seguridad, estas se encuentran pre instaladas como por ejemplo: Wireshark y Kismet. Adicional,

BT5 es totalmente compatible con el chipset y driver de las tarjetas *AirPcap Nx,* tal como se mencionó en el capítulo de: Tarjetas *AirPcap Nx Adapter*. Por esta facilidad, en este capítulo el enfoque es exclusivamente en la instalación y configuración de *Freeradius-WPE* y *Hostapd*.

2.5.1 FreeRADIUS-WPE

FreeRADIUS-WPE (Wireless Pwnage Edition) es un parche para el popular servidor de código abierto *FreeRADIUS* para demostrar la vulnerabilidad de suplantación en *RADIUS,* este parche añade las siguientes funcionalidades:

- Simplifica la configuración de *FreeRADIUS* añadiendo todas las direcciones *RFC1918* como dispositivos NAS aceptables.

- Facilita la configuración de la autenticación *EAP* mediante la inclusión de todos los tipos *EAP* soportados por *FreeRADIUS*.

- Agrega credenciales de autenticación para los múltiples tipos de *EAP,* incluyendo: *PEAP, TTLS, TLS, LEAP, EAP-MD5, EAP-MSCHAPv2, PAP, CHAP,* entre otros.

El proceso de instalación es el siguiente:

>>root@bt:~# wget ftp://ftp.freeradius.org/pub/radius/old/freeradius-server-2.1.12.tar.bz2

>>root@bt:~# wget https://raw.github.com/brad-anton/freeradius-wpe/master/freeradius-wpe.patch

>>root@bt:~# tar -jxvf freeradius-server-2.1.12.tar.bz2

>>root@bt:~# cd freeradius-server-2.1.12

>>root@bt:~/freeradius-server-2.1.12# patch -p1 < /root/freeradius-wpe.patch

>>root@bt: ~/freeradius-server-2.1.12# ./configure

>>root@bt: ~/freeradius-server-2.1.12# make && make install

>>root@bt: ~/freeradius-server-2.1.12# ldconfig

Si todo fue instalado correctamente, se debería ver lo que se muestra a continuación:

>>root@bt: ~/freeradius-server-2.1.12# radiusd –v

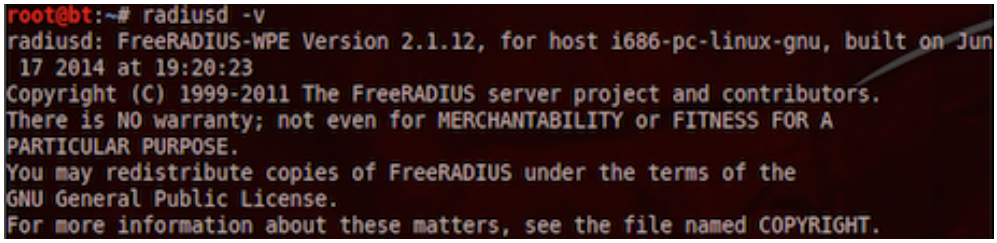


Figura. 74 FreeRADIUS-WPE instalado

>>root@bt: ~/freeradius-server-2.1.12# cd

>>root@bt:~# radiusd -X

Figura. 75 FreeRADIUS-WPE Ready to process requests

2.5.2 Hostapd – AP (Falso)

Es un *daemon* que sirve para crear puntos de acceso *(AP)* y servidores de autenticación. Aplica el estándar IEEE 802.11 para la gestión del punto de acceso, *IEEE 802.1X/WPA/WPA2/EAP* para el protocolo de seguridad, cliente y servidor de autenticación *RADIUS*. La versión actual soporta Linux *(Host AP, MadWiFi)*, entre otros.

La instalación se muestra a continuación:

>>root@bt:~# wget http://hostap.epitest.fi/releases/hostapd-2.0.tar.gz
>>root@bt:~# tar -zxvf hostapd-2.0.tar.gz
>>root@bt:~# cd hostapd-2.0/
>>root@bt:~/ hostapd-2.0# cd hostapd/
>>root@bt:~/ hostapd-2.0/hostapd# cp defconfig .config
>>root@bt:~/ hostapd-2.0/hostapd# apt-get install libnl-dev
>>root@bt:~/ hostapd-2.0/hostapd# apt-get update
>>root@bt:~/ hostapd-2.0/hostapd# apt-get install libssl-dev
>>root@bt:~/ hostapd-2.0/hostapd# make && make install

Se continúa con la configuración del *AP* (Falso). Este *access point* se utilizará cuando se realice el estudio en el escenario de prueba intrusivo.

>>root@bt:~/hostapd-2.0/hostapd# nano hostapd.conf

interface=wlan0
driver=n180211
ssid=tesis14
logger_stdout=-1
looger_stdout_level=0
dump_file=/tmp/hostapd.dump
ieee8021x=1

eapol_key_index_workaround=0

own_ip_addr=127.0.0.1

auth_server_addr=127.0.0.1

auth_server_port=1812

auth_server_shared_secret=testing123

wpa=2

wpa_key_mgmt=WPA-EAP

channel=11

wpa_pairwise=TKIP CCMP

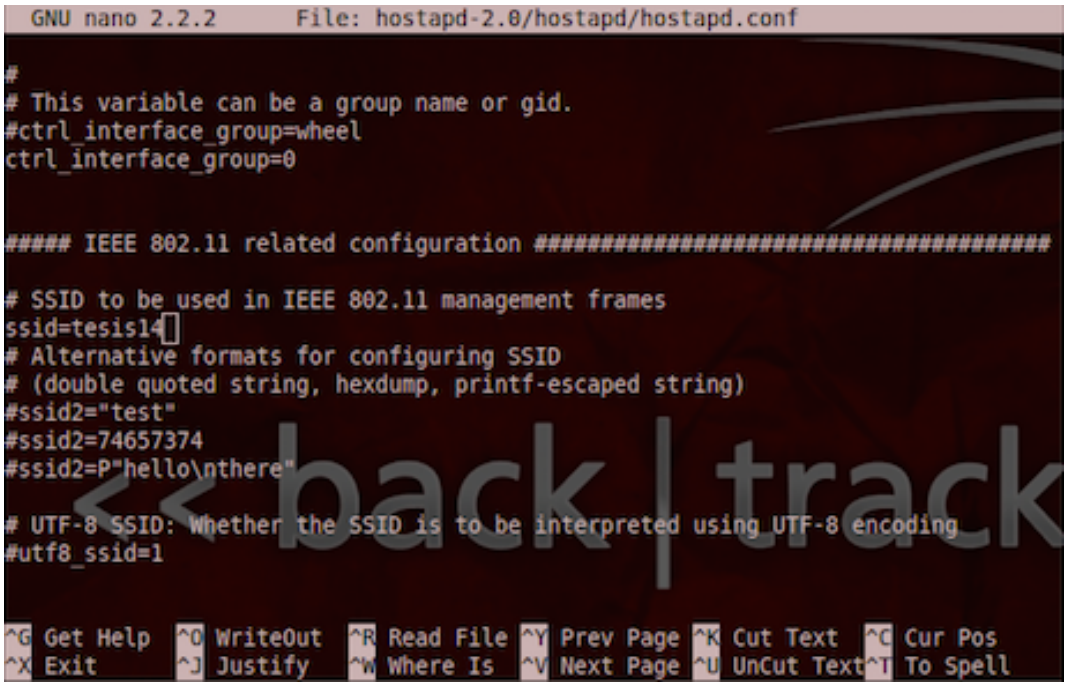


Figura. 76 hostapd.conf

Si los parámetros se han llenado correctamente, se debería ver lo que se muestra a continuación:

>>root@bt:~/hostapd-2.0/hostapd# cd

>>root@bt:~# hostapd hostapd-2.0/hostapd/hostapd.conf

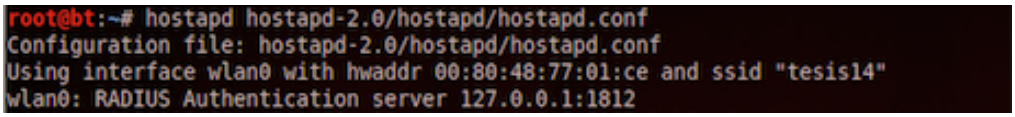


Figura. 77 Ejecutar hostapd

2.6 Instalación de los certificados en los suplicantes

Como se explicó en capítulos anteriores, fueron creados los certificados en el servidor mediante la utilización de OpenSSL, los mismos que se encuentran almacenados en el directorio *PKI*, y que deben ser extraídos para ser instalados en cada uno de los suplicantes

Para poder seguir con este procedimiento, primero se debe montar un dispositivo extraíble donde se copiarán los certificados, para esto es necesario realizar el siguiente procedimiento:

>> espe14@ubuntu:~$ sudo –i
[sudo] password for espe14: espetesis14
>> root@ubuntu:~# cd /media
>> root@ubuntu:/media# mkdir usb

Creado el directorio se procede a montar el dispositivo extraíble con el siguiente comando:

>> root@ubuntu:/media# mount -t vfat /dev/sdxX /media/usb

Para comprobar que el disco se encuentra listo para ser utilizado, se accede al mismo con el siguiente comando, de ser correcto se podrá observar la información que contiene el mismo:

>> root@ubuntu:/media # cd /usb
>> root@ubuntu:/media/usb# ls –l

Los certificados se encuentran dentro del directorio */etc/ssl/PKI*, y se accede mediante el siguiente comando:

>> root@ubuntu:~# cd /etc/ssl/PKI

>> root@ubuntu:/etc/ssl/PKI# cp cacert.pem /etc/freeradius/certs/cacert.pem

>> root@ubuntu:/etc/ssl/PKI# cp client_cert.pem /etc/freeradius/certs/client_cert.pem

>> root@ubuntu:/etc/ssl/PKI# cp client_key.pem /etc/freeradius/certs/client_key.pem

>> root@ubuntu:/etc/ssl/PKI# cp client_cert.p12/etc/freeradius/certs/client_cert.p12

Una vez finalizado el proceso de copiar los certificados dentro del dispositivo extraíble, se procede a comprobar si se encuentran dentro, y finalmente se desmonta el dispositivo:

>> root@ubuntu:~# cd /media/usb

>> root@ubuntu:/media/usb# ls –l

>> root@ubuntu: /media/usb# umount /media/usb

2.6.1 Instalación de certificados en Windows

A continuación se presenta el procedimiento para la instalación de certificados para un suplicante Windows, en este caso se realizó sobre un sistema operativo Windows 7, que en general es el más utilizado.

Se copian los cuatro archivos dentro del *disco C*, como se muestra en la figura:

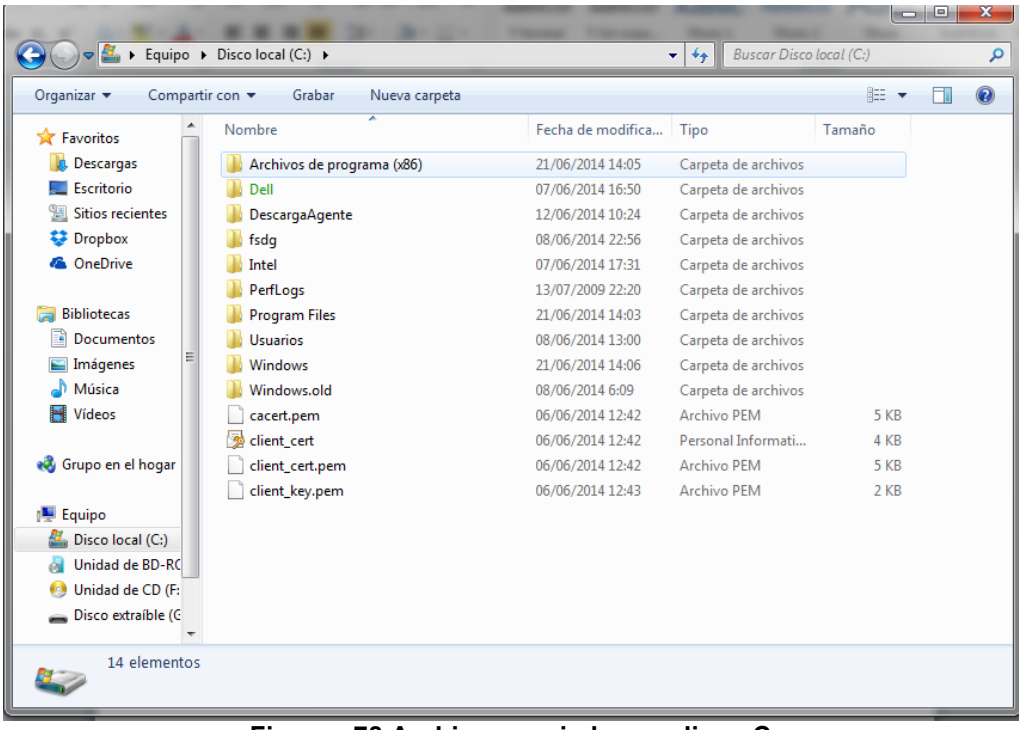


Figura. 78 Archivos copiados en disco C

Se ubica el archivo con el nombre *client_cert.p12,* se da clic derecho, en la ventana que se presenta, se selecciona *Instalar PFX*, como se muestra a continuación:

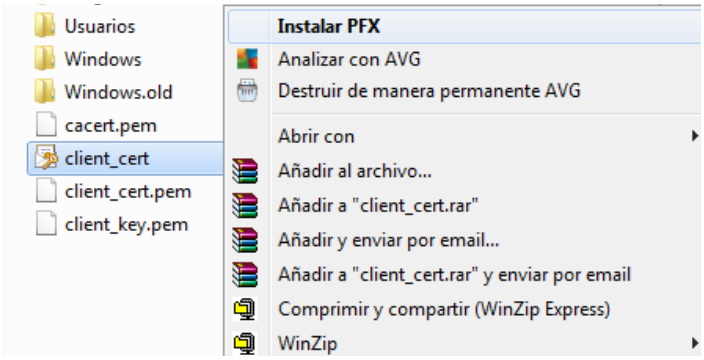


Figura. 79 Instalación de certificado

Una vez seleccionada la opción, se presentará la siguiente ventana de asistente para la instalación de certificado en la que se presiona el botón siguiente.

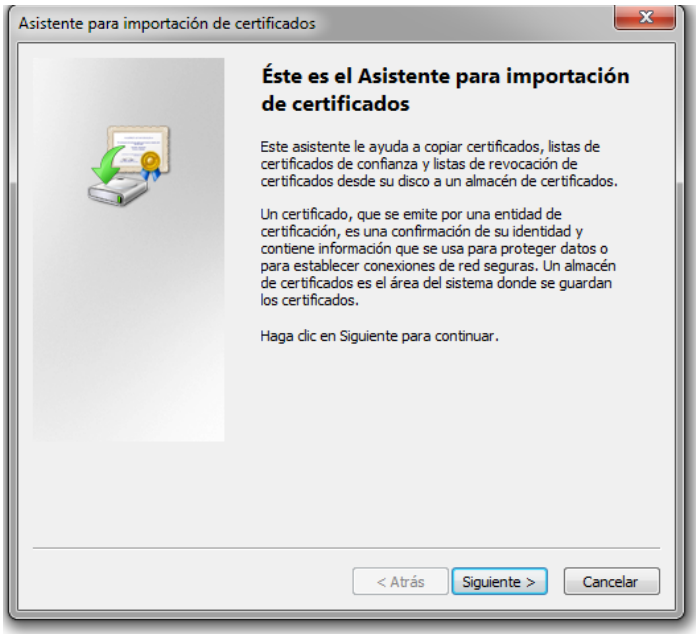


Figura. 80 Ventana de asistente para instalación de certificados

Para confirmar la ubicación y el tipo de certificado a instalar, se hace clic en siguiente como se observa en la siguiente figura:

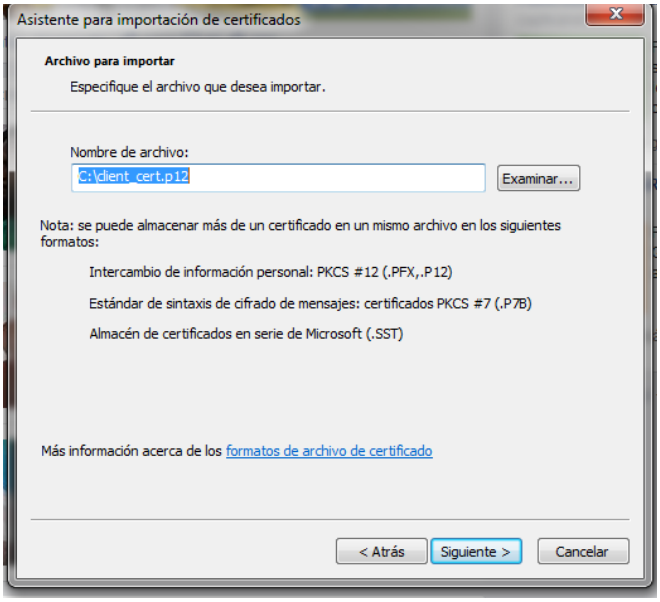


Figura. 81 Ubicación y tipo de archivo a instalar

Una vez confirmada la ubicación del archivo, el asistente solicitará la clave para instalar el certificado, en este caso la clave será: *espe14* y se selecciona las tres opciones que se encuentran por debajo del campo de texto y se presiona siguiente:

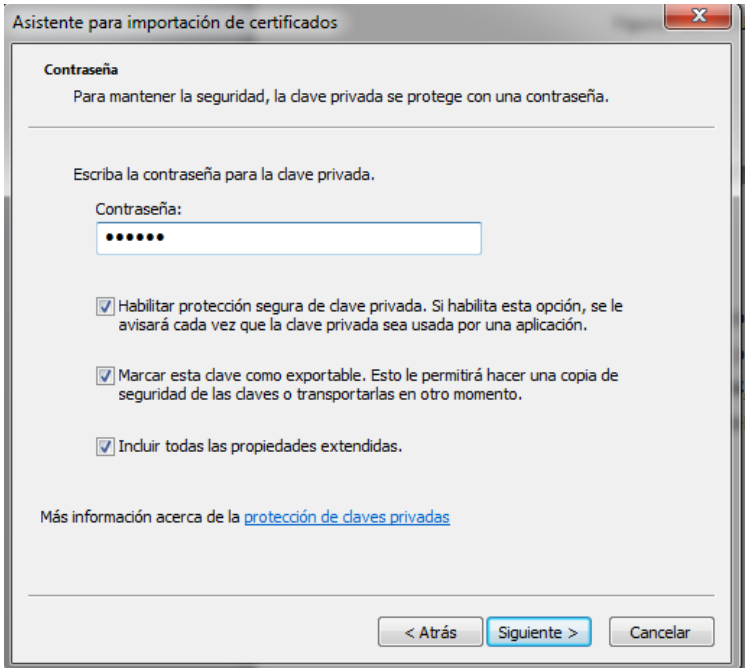


Figura. 82 Contraseña de certificado

El siguiente paso es seleccionar la carpeta en el que el certificado se almacenará, es decir que el usuario selecciona a que raíz pertenece el certificado, es recomendable que se deje en la opción que se encuentra por defecto.

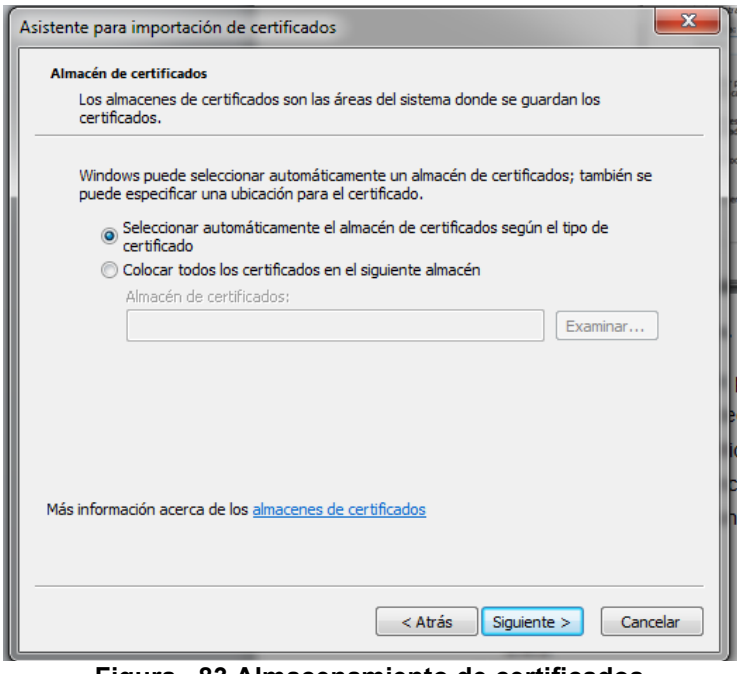


Figura. 83 Almacenamiento de certificados

Para finalizar el asistente, se presiona el botón finalizar como se muestra en la figura:

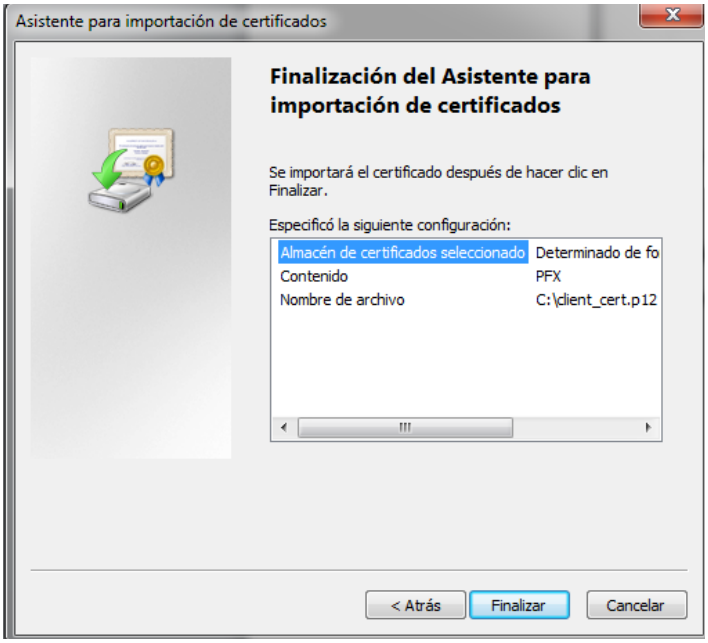


Figura. 84 Finalización del asisten de instalación

La instalación genera una nueva clave para el intercambio de información, para los que se debe seguir con los siguientes pasos:

En la ventana que se presenta, se presiona el botón de nivel de seguridad, selecciona nivel alto y *siguiente*, se escribe la contraseña de encriptación, para este caso: *espe14,* el proceso pedirá confirmación de la instalación del certificado, se confirma presionando *si*:

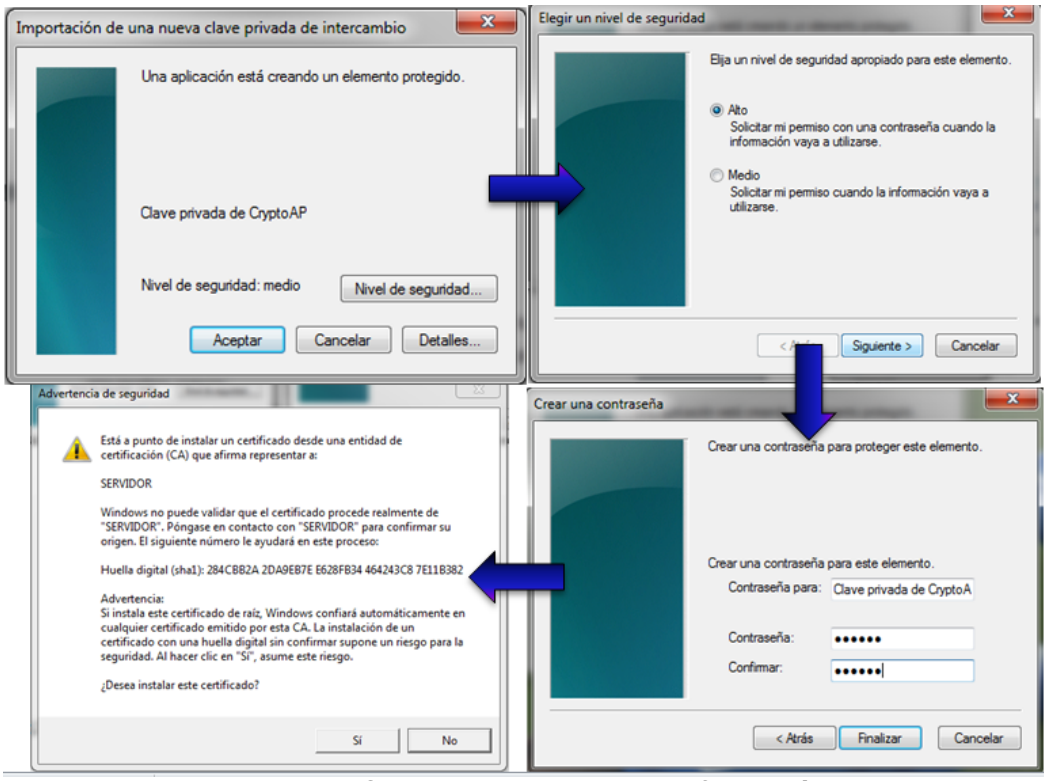


Figura. 85 Clave de intercambio de información

Una vez instalado el certificado, se debe confirmar que es un certificado de confianza para que poder ser utilizado en la autenticación del suplicante, para ello, se da clic en el botón de inicio y se ingresa el siguiente comando:

certutil – f –enterprise –addstore NTAuth AD.crt

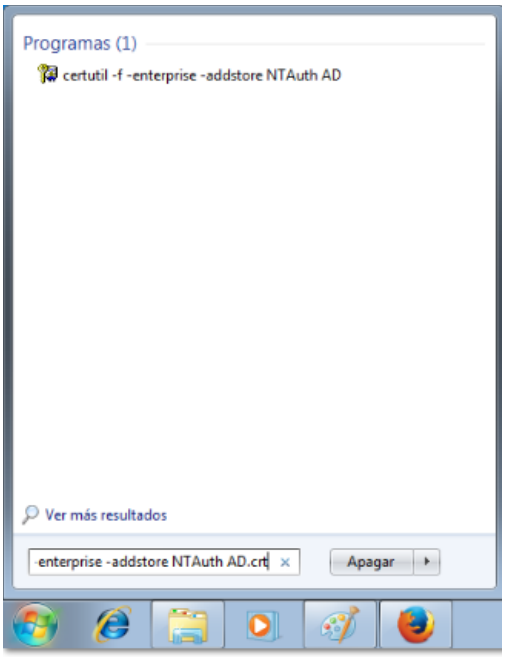


Figura. 86 Clave de intercambio de información

Instalados los certificados y validados, se procederá a configurar los parámetros de la red inalámbrica, para lo cual se siguen los siguientes pasos:

Se ingresa al centro de redes y recursos compartidos de Windows 7, de la ventana se selecciona *configurar una nueva conexión o red ,* como se observa en la siguiente figura:

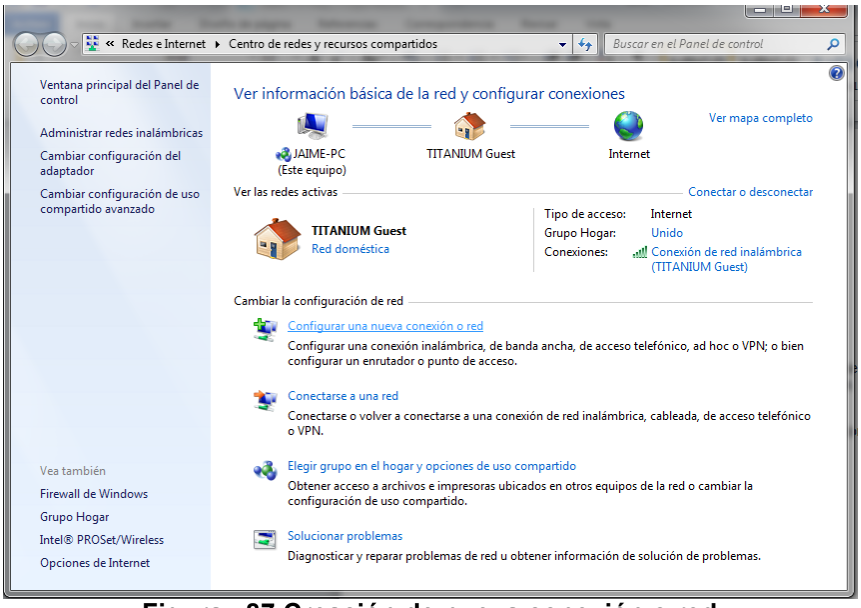


Figura. 87 Creación de nueva conexión o red

Se debe seleccionar la opción de C*onectar manualmente a una red inalámbrica.*

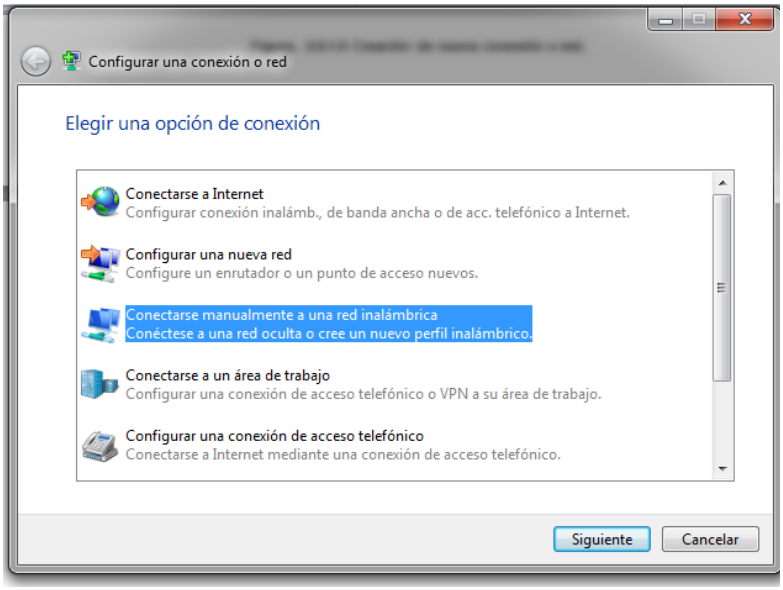


Figura. 88 Tipos de conexión

Se configuran los parámetros de la red al que se va a conectar:

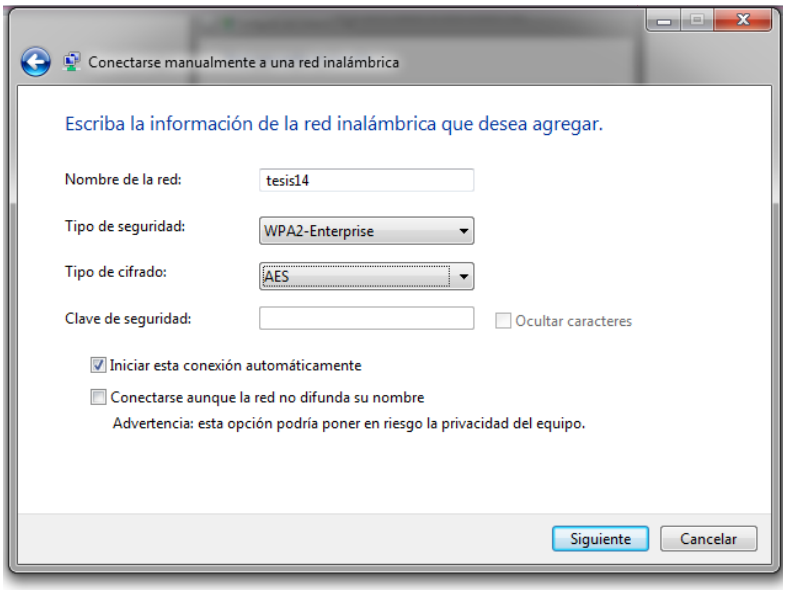


Figura. 89 Parámetros de red

A continuación como se muestra en las imágenes se configura los parámetros de seguridad.

Se selecciona la opción de *Cambiar la configuración de conexión,* y en la ventana que se presenta seleccionar la pestaña de *seguridad*, depuse se presiona el botón de *Configuración.*

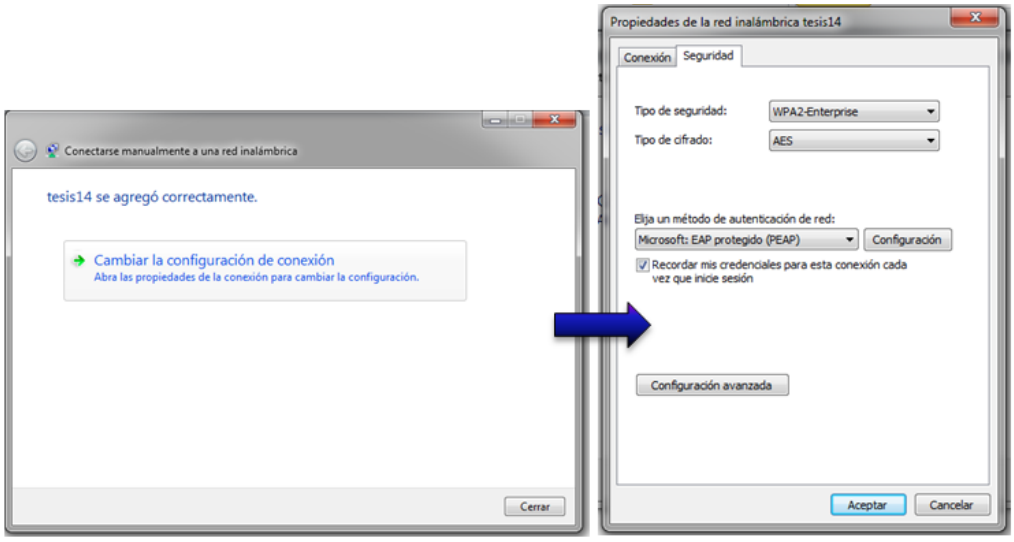


Figura. 90 Configuración de parámetros

En la siguiente ventana se muestra las configuraciones de seguridad, como la selección del certificado que ya fue instalado en pasos anteriores,

se configura la dirección *IP* del servidor al que se va y se presiona le botón *Configurar…* y se desmarca el visto en el dialogo, se acepta y se cierran las ventanas.

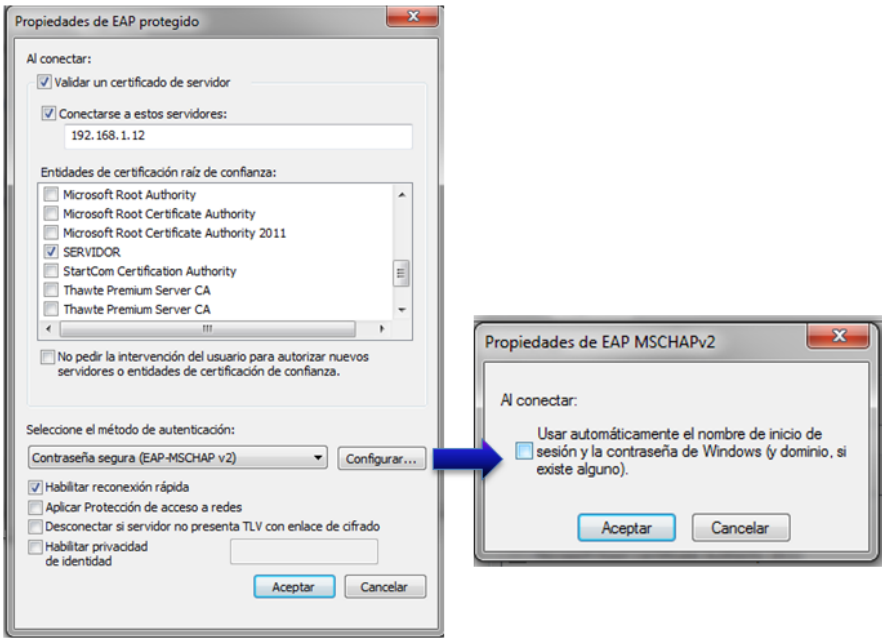


Figura. 91 Parámetros de seguridad

Se selecciona la red a conectar en este caso *tesis14*, posterior se solicitará las credenciales, las mismas que fueron creadas en la base de datos en pasos anteriores, para el ejemplo se utilizará *usuarioa, clavea.* Como se indica en la siguiente figura:

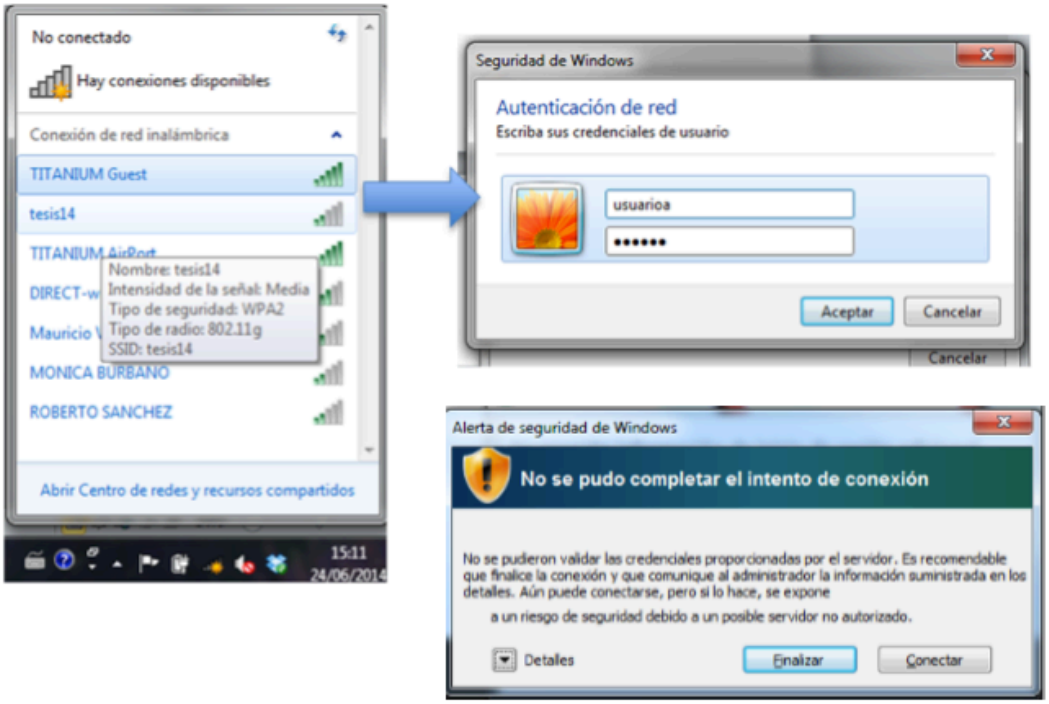


Figura. 92 Autenticación

2.6.2 Instalación de certificados en Android

A continuación el procedimiento a seguir para instalar los certificados y conectar un teléfono con sistema operativo Android.

Para poder instalar los certificados dentro del teléfono se necesita copiar los 4 archivos necesarios para el suplicante en la raíz del dispositivo, como se muestra en la siguiente imagen:

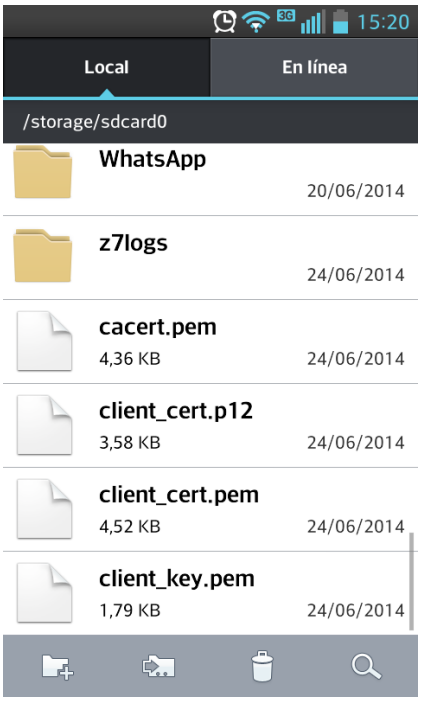


Figura. 93 Archivos ubicados en la raíz

Para instalar los certificados se ingresa *Ajustes* > *Seguridad* >*Instalar desde el almacenamiento,* para poder instalar el certificado se solicitará la contraseña del certificado (*espe14*), finalmente confirma si el certificado a instalarse es el correcto como indica la siguiente figura.

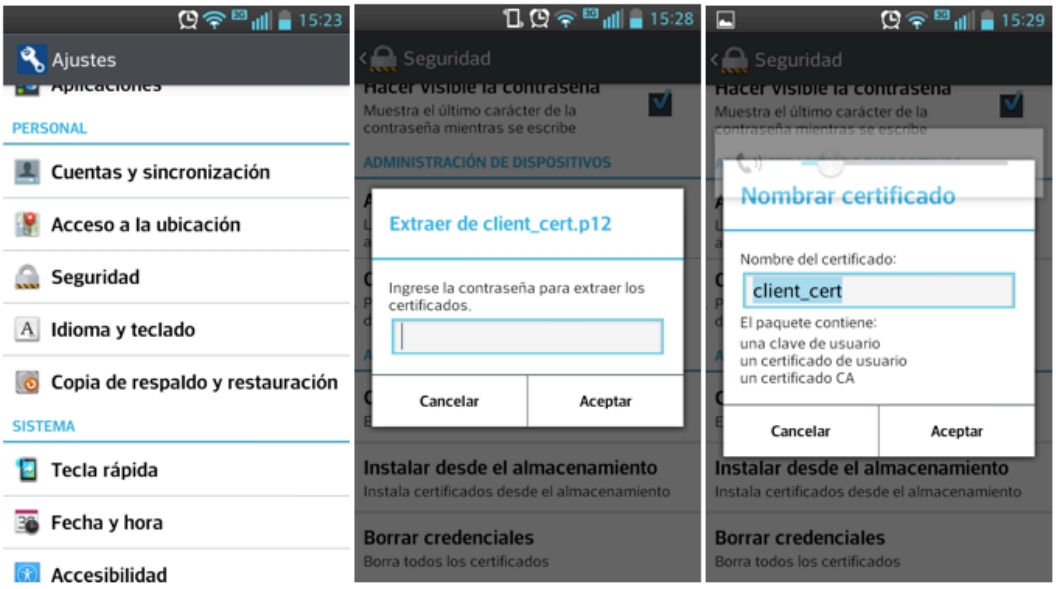


Figura. 94 Archivos ubicados en la raíz

Si se desea comprobar que el certificado ha sido instalado de forma correcta, se ingresa: *Ajustes* > *Seguridad* > *Credenciales de confianza* > *PERSONAL* donde se podrá comprobar el certificado instalado y los datos que este contiene.

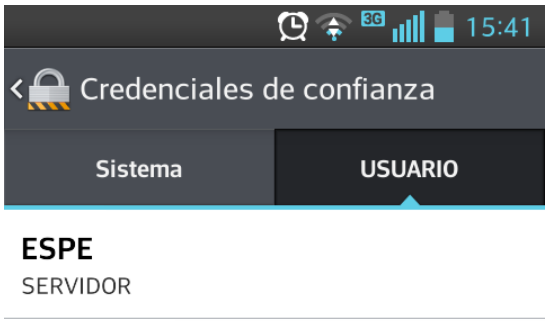


Figura. 95 Certificado instalado

Para conectarse a la red se ingresa a: *Ajustes* > *Wifi,* se selecciona la red *(tesis14*), donde se solicitará toda la información necesaria para que el suplicante en este caso el teléfono *Android* se conecte.

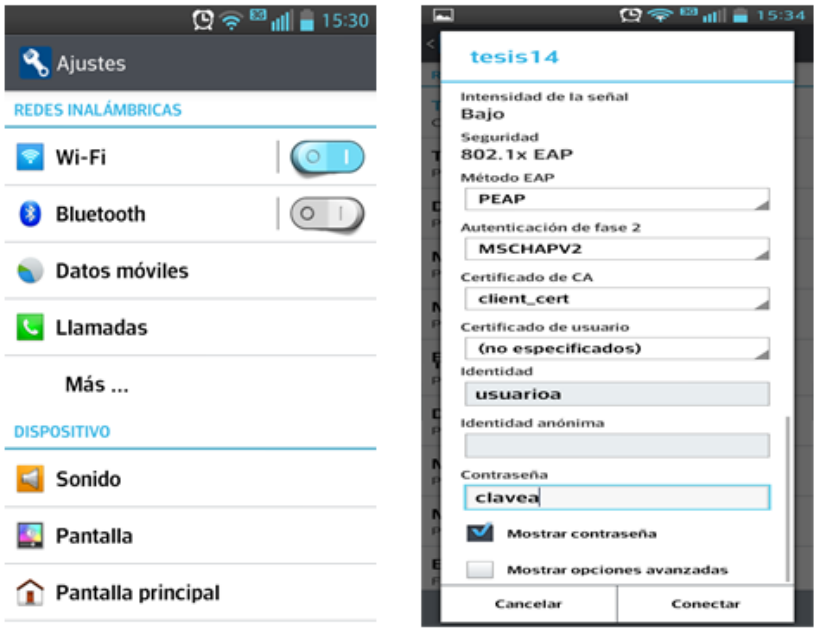


Figura. 96 Conexión a red inalámbrica

2.6.3 Instalación de certificados en OS X

Para instalar los certificados en OS X, se hace doble clic en el certificado a instalar:

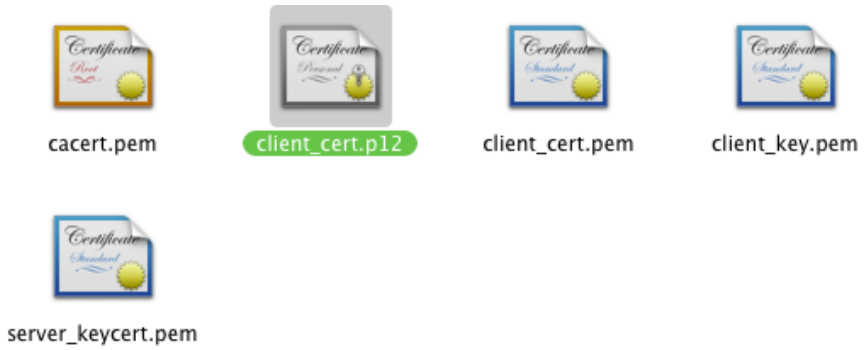


Figura. 97 Certificado cliente_cert.p12

Aparecerá una ventana para autorizar el certificado en el llavero *(keychain)*, se pone la clave del certificado *(espe14)* y este es agregado como se puede apreciar en la siguiente figura:

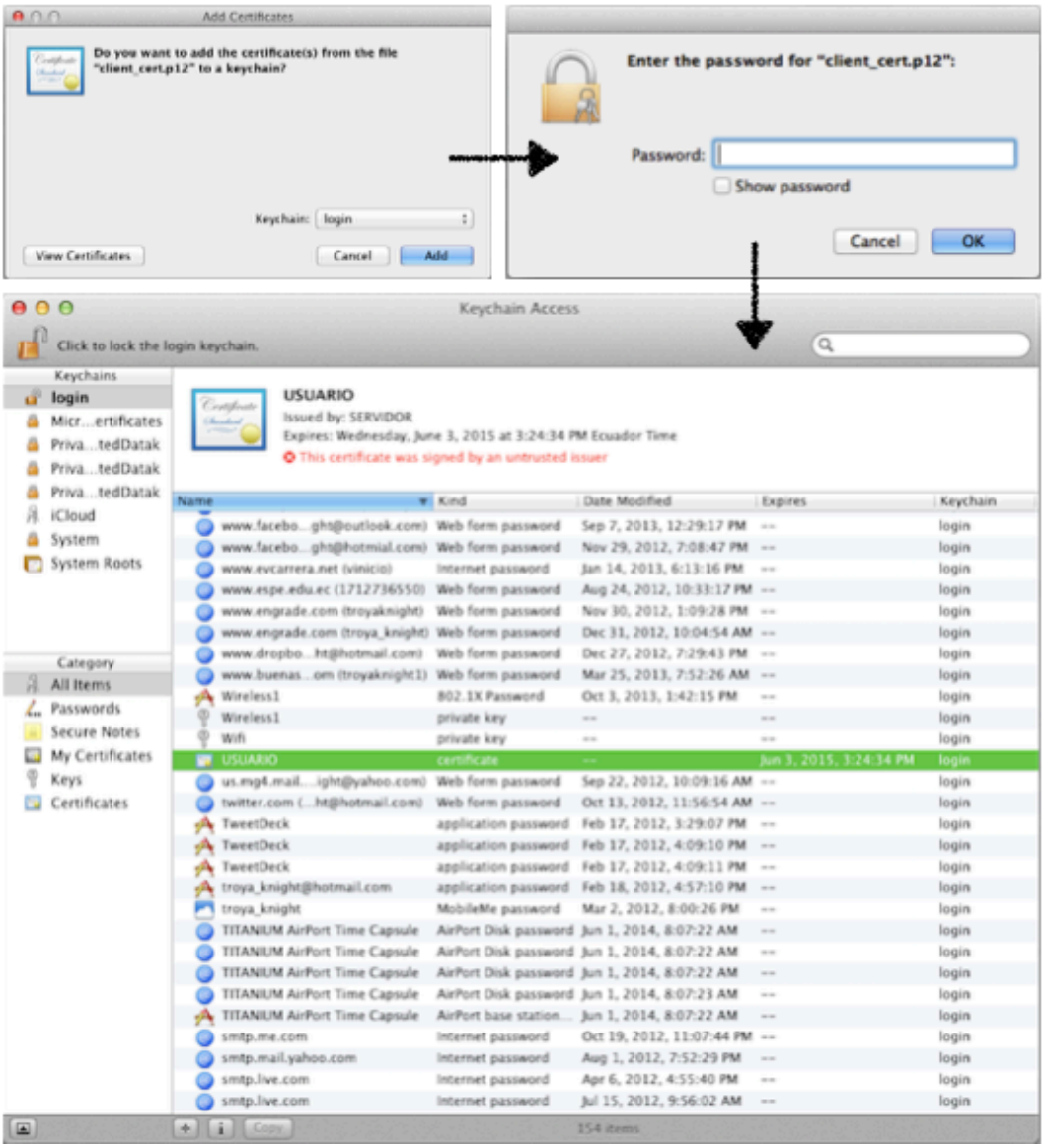


Figura. 98 Certificado USUARIO agregado al keychain

Al certificado instalado, se tiene que dar privilegios de confianza. Se hace doble clic en el certificado *USUARIO* y lo hace de confianza en todos sus parámetros como se muestra:

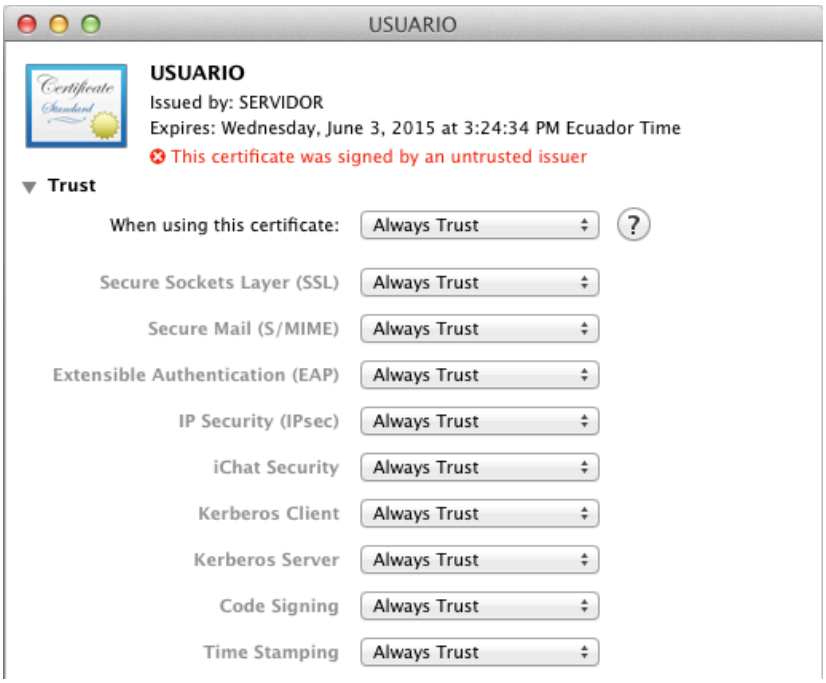


Figura. 99 Certificado USUARIO

Se selecciona nuestra red *wireless (tesis14)*:

Figura. 100 Certificado USUARIO siempre de confianza

Se despliega una ventana donde se pide que se llene los capos de *Mode (EAP-TLS)* e *Identity* (se selecciona el certificado instalado *USUARIO*) además, se pide el *Username (usuarioa)* registrado en el servidor *RADIUS*.

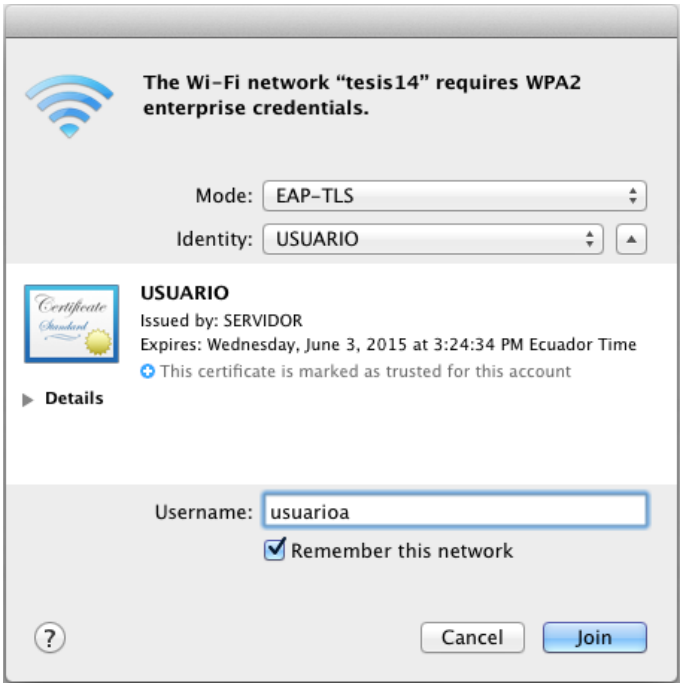


Figura. 101 Credenciales Enterprise

Se verifica el certificado y se selecciona siempre confianza con el certificado emitido por el *SERVIDOR*.

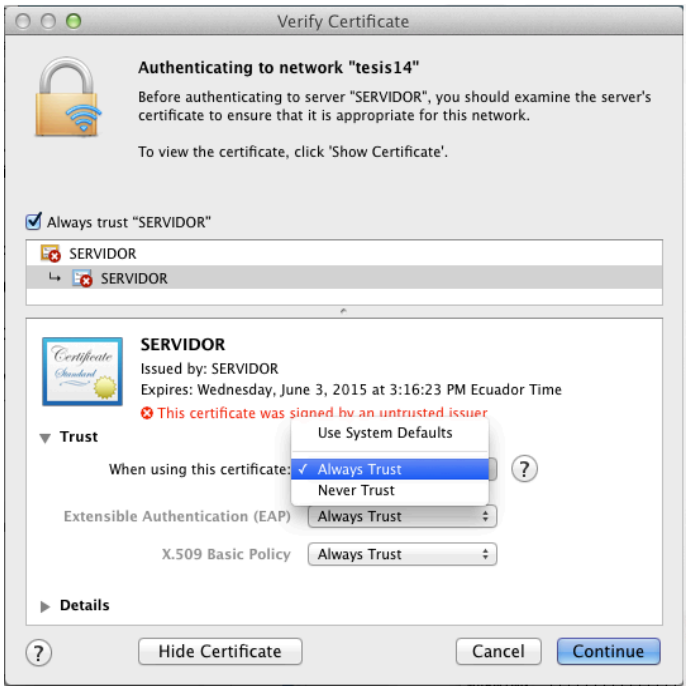


Figura. 102 Certificado SERVIDOR

Como se puede verificar en las preferencias de red inalámbrica, nuestra conexión con *tesis14* esta autenticada vía *EAP-TLS*.

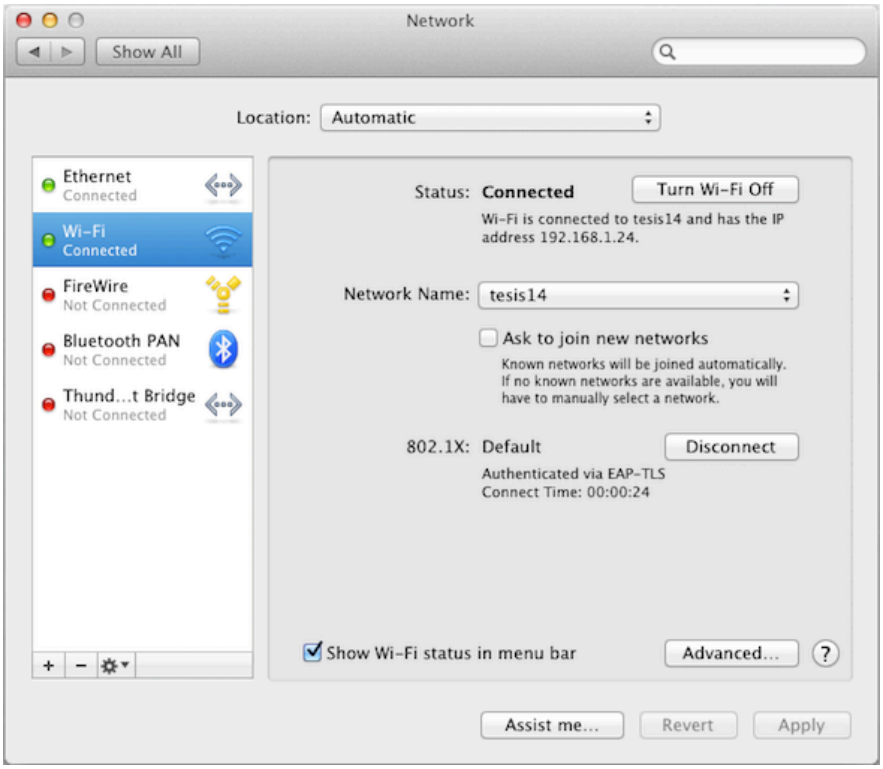


Figura. 103 Conexión establecida vía EAP-TLS

2.6.4 Instalación de certificados en iOS

Para la instalación de los certificados en *iOS*, es necesaria la instalación de una aplicación *(App)*, la misma es gratuita y se descarga en *OS X* a través del *App Store*. Esta aplicación se llama *Apple Configurator.*

Se conecta el dispositivo con iOS vía USB y se ejecuta *Apple Configurator*. Clic en *Prepare* > *Setup* y en *Settings* se selecciona la opción *Certificates.*

Figura. 104 Apple Configurator

Se asigna un nombre al certificado *(USUARIO)*, se selecciona el certificado *(client_cert.p12)* y se pone el *Password (espe14)* del certificado y se guarda.

Figura. 105 Instalación certificados iOS

Seguido, se configura la red *Wi-Fi*. Se llenan los campos de: *Service Set Identifier (tesis14)* y *Security Type (WPA/WPA2 ENTERPRISE)*. En *Accepted EAP Types* se selecciona *TLS* y se indica la identidad del certificado *(Identity Certificate)*, finalmente se selecciona *USUARIO* y se guarda.

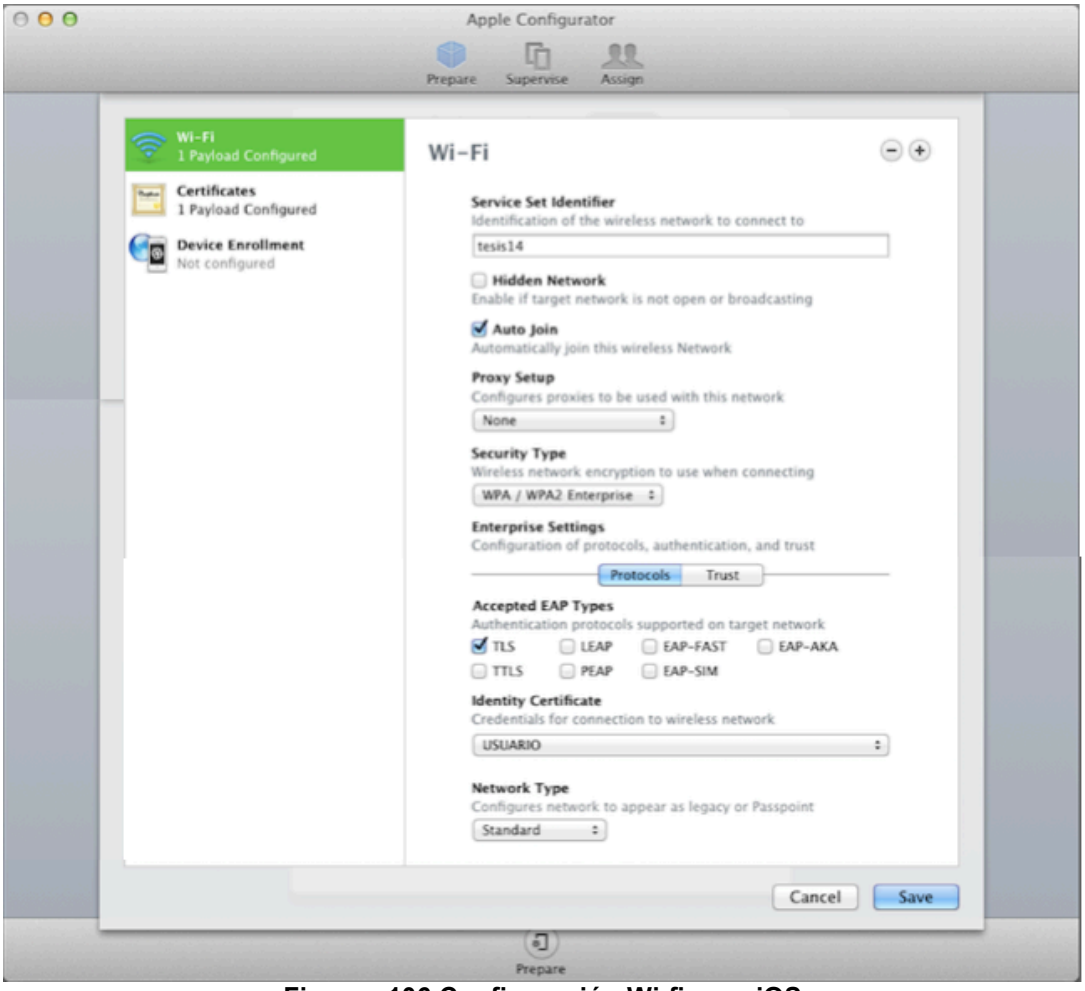


Figura. 106 Configuración Wi-fi para iOS

Completado este paso, se hace clic en *Prepare* (icono en la parte inferior de la ventana) con lo que el/los equipos que estén conectados por medio de *USB* serán actualizados con esta configuración.

Para comprobar la conexión a nuestra red *tesis14*, en el dispositivo *iOS* se hace clic en dicha red y se comprueba la conexión sin inconvenientes.

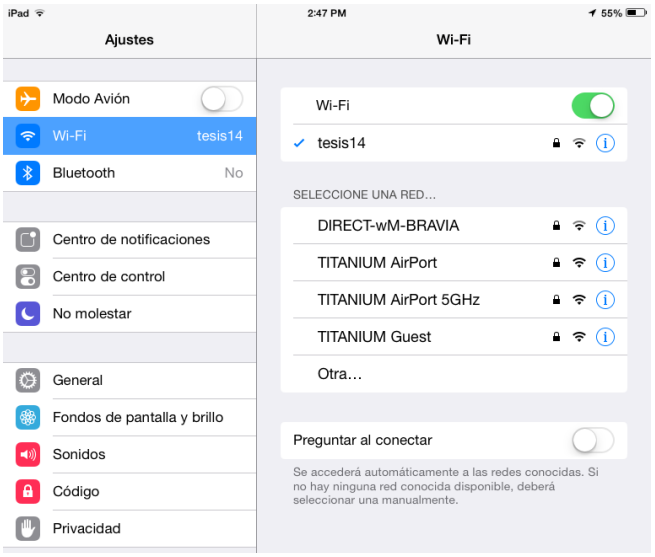


Figura. 107 iOS conectado a tesis14

CAPÍTULO III

ESCENARIO DE PRUEBA NORMAL

En el protocolo *802.11i* se establecen procedimientos de autenticación *802.1X* y protocolos de gestión de claves. Como se ha revisado anteriormente, son tres las entidades que participan: el suplicante, el autenticador *(AP)* y el servidor de autenticación. Cuando una autenticación es correcta, se supone que el suplicante y el autenticador han verificado su identidad mutuamente y generan una clave compartida, esta servirá a su tiempo para satisfacer llaves subsecuentes. Una vez se haya establecido un túnel seguro, el servidor de autenticación verificará las credenciales del suplicante en su base de datos. De ser exitosa, se realiza un apretón de manos *(handshake)* de cuatro vías que permitirá el tráfico de los paquetes o información protegida. Todo este proceso se detallará en las 5 etapas que se muestran en la siguiente figura.

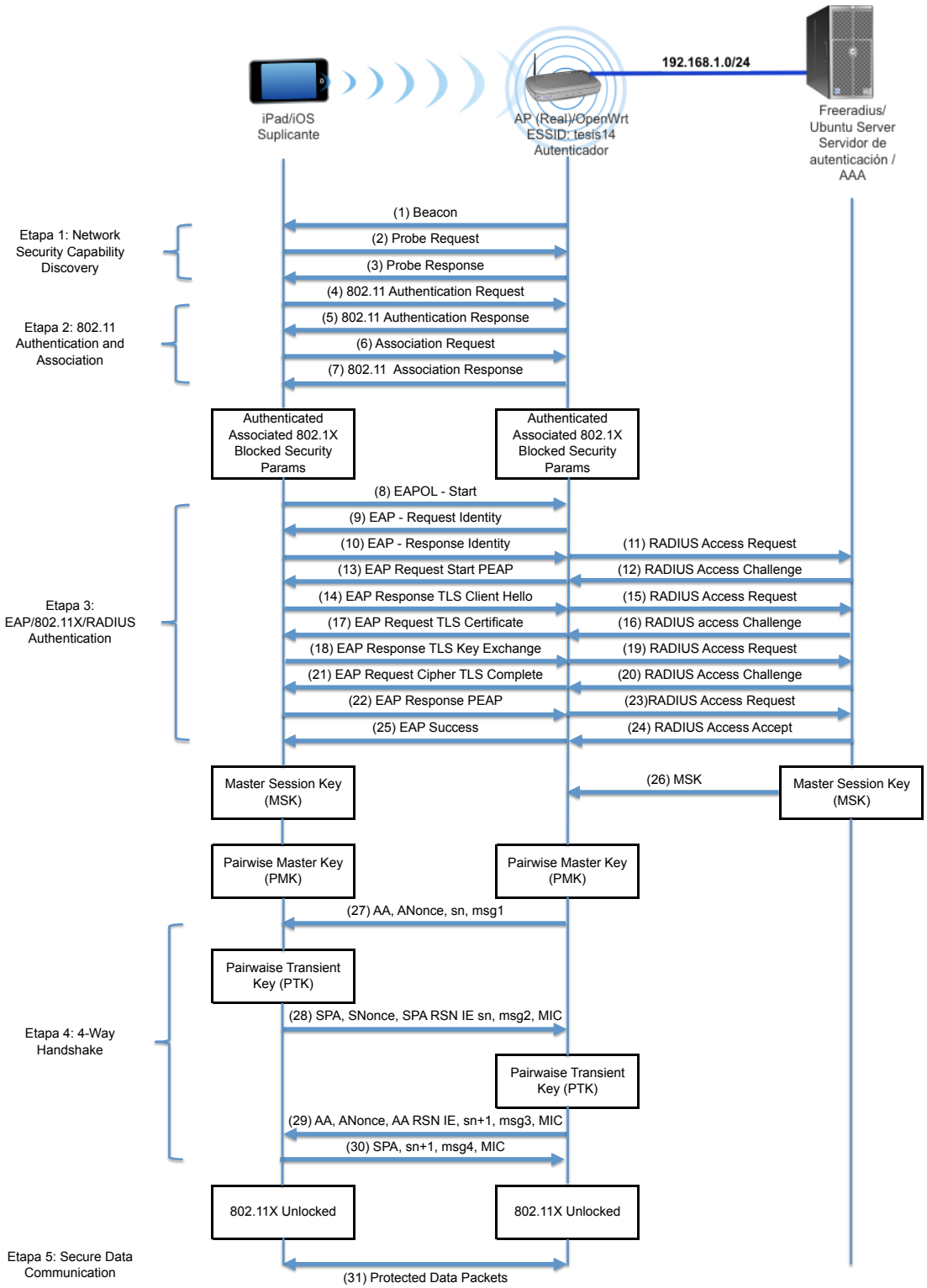


Figura. 108 EAP/802.11X/RADIUS Authentication

Con el fin de facilitar la búsqueda y filtrar las tramas de Administración y Control en Wireshark que resultan de la captura que se realizó en los escenarios, se presenta la siguiente tabla que resume los comandos para este fin:

Tabla. 4

Filtros en Wireshark

Tramas de Administración (Management)		
wlan.fc.type==0		
#	**Subtype description**	**Wireshark display filter**
1	Association Request	wlan.fc.type_subtype==0x00
2	Association Response	wlan.fc.type_subtype==0x01
3	Reassociation Request	wlan.fc.type_subtype==0x02
4	Reassociation Response	wlan.fc.type_subtype==0x03
5	Probe Request	wlan.fc.type_subtype==0x04
6	Probe Response	wlan.fc.type_subtype==0x05
7	Beacon	wlan.fc.type_subtype==0x08
8	ATIM	wlan.fc.type_subtype==0x09
9	Disassociation	wlan.fc.type_subtype==0x0A
10	Authentication	wlan.fc.type_subtype==0x0B
11	Deauthentication	wlan.fc.type_subtype==0x0C
12	Action	wlan.fc.type_subtype==0x0D
Tramas de Control		
wlan.fc.type==1		
13	Block ACK Request	wlan.fc.type_subtype==0x18

Continua ➔

14	Block ACK	wlan.fc.type_subtype==0x19
15	Power-Save Poll	wlan.fc.type_subtype==0x1A
16	RTS	wlan.fc.type_subtype==0x1B
17	CTS	wlan.fc.type_subtype==0x1C
18	ACK	wlan.fc.type_subtype==0x1D
19	CF-end	wlan.fc.type_subtype==0x1E
20	CF-end + CF-ack	wlan.fc.type_subtype==0x1F
	Trama de Datos	
	wlan.fc.type==2	

3.1 Etapa 1: Network Security Capability Discovery

Esta etapa consiste de los mensajes *(1)* al *(3)*. El *Access Point (AP)* emite periódicamente a manera de *broadcast* sus parámetros de seguridad *RSN (Robust Security Network)* que están establecidos dentro del protocolo *802.11*. Esto se lo hace a través de un canal *(channel)* específico por medio de una trama llamada *Beacon.* Cuando un cliente se activa, envía una trama conocida como *Probe Request* y en respuesta, el *AP* enviará una trama *Probe Response* la cual es información referente a su capacidad, velocidad de datos, etc. Las tramas son:

3.1.1 Beacon

Es una trama de tipo de administración *(management frame).* Los *Access Points* continuamente envían estas tramas para anunciar su presencia en una red *WLAN*. Esto incluye el *SSID (service set identification)* e información que permiten sincronizar estaciones con *BSSID (basic service set identification)*.

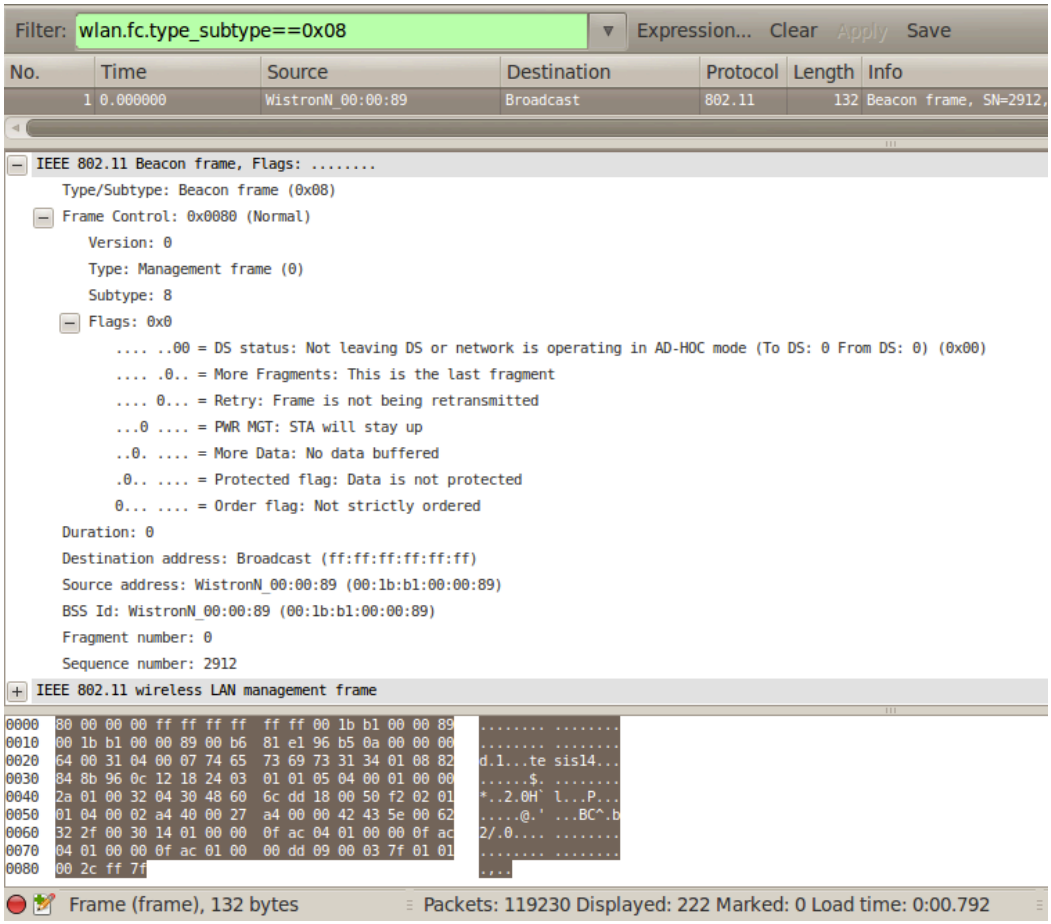


Figura. 109 Beacon Frame

3.1.2 Probe Request

Es una trama de tipo de administración *(management frame)* que aparece en el momento en que un cliente se activa o una tarjeta *NIC (Network Interface Controller)* se habilita con el objetivo de obtener información de cualquier AP en rango en los diferentes canales que soporten sus características para asociarse.

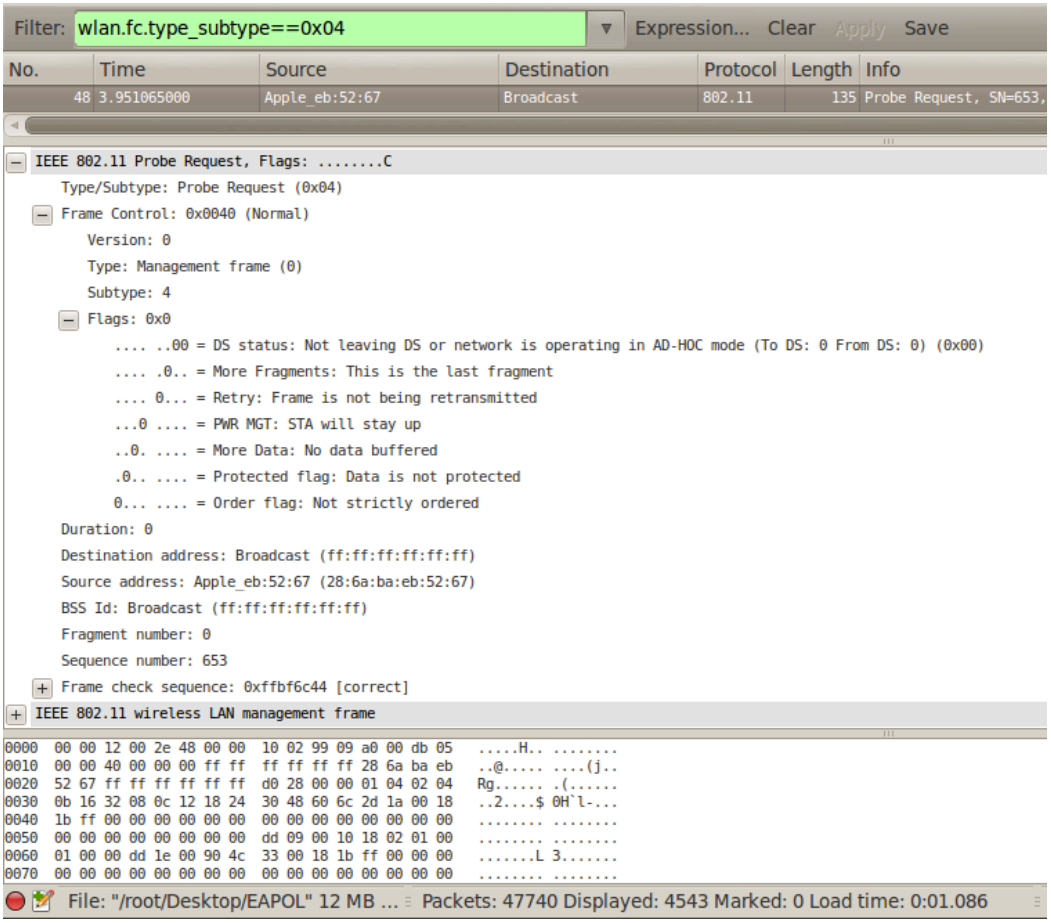


Figura. 110 Probe Request Frame

3.1.3 Probe Response:

Es una trama de tipo de administración *(management frame)* que responde a la trama *probe request* para facilitarle información sobre sincronización.

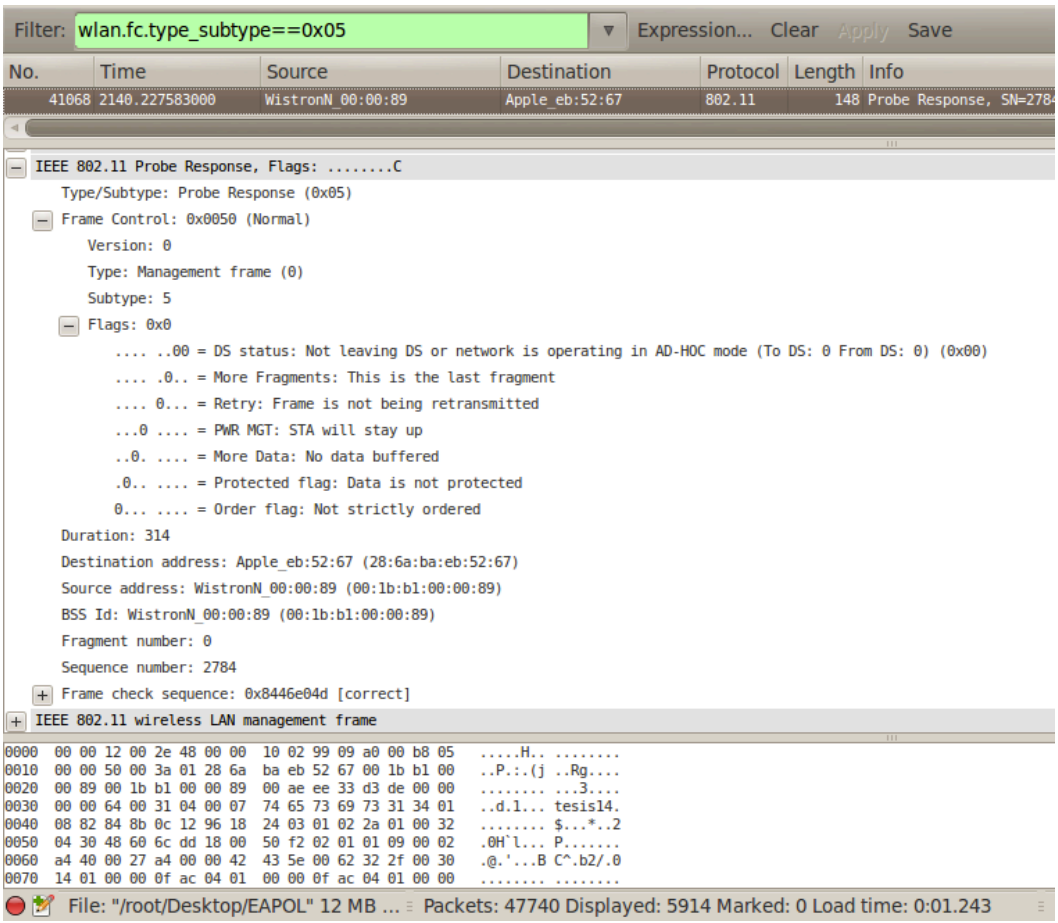


Figura. 111 Probe Response Frame

3.2 Etapa 2: 802.11 Authentication and Association

Se abarca los mensajes del *(4)* al *(7).* El cliente escoge de entre los *AP* que respondieron al *probe response* e inicia la autenticación y asociación con el mismo. Al final de esta etapa, los puertos de protocolo *802.11X* permaneces bloqueados y ningún paquete con información puede ser intercambiado. Los pasos son:

3.2.1 802.11 Authentication Request/Response

Es una trama de tipo de administración *(management frame).* Es un proceso en el que la NIC del cliente envía sólo una trama de autenticación que contiene su identidad hacia el AP. El punto de acceso responde con una trama de autenticación como respuesta indicando aceptación o rechazo.

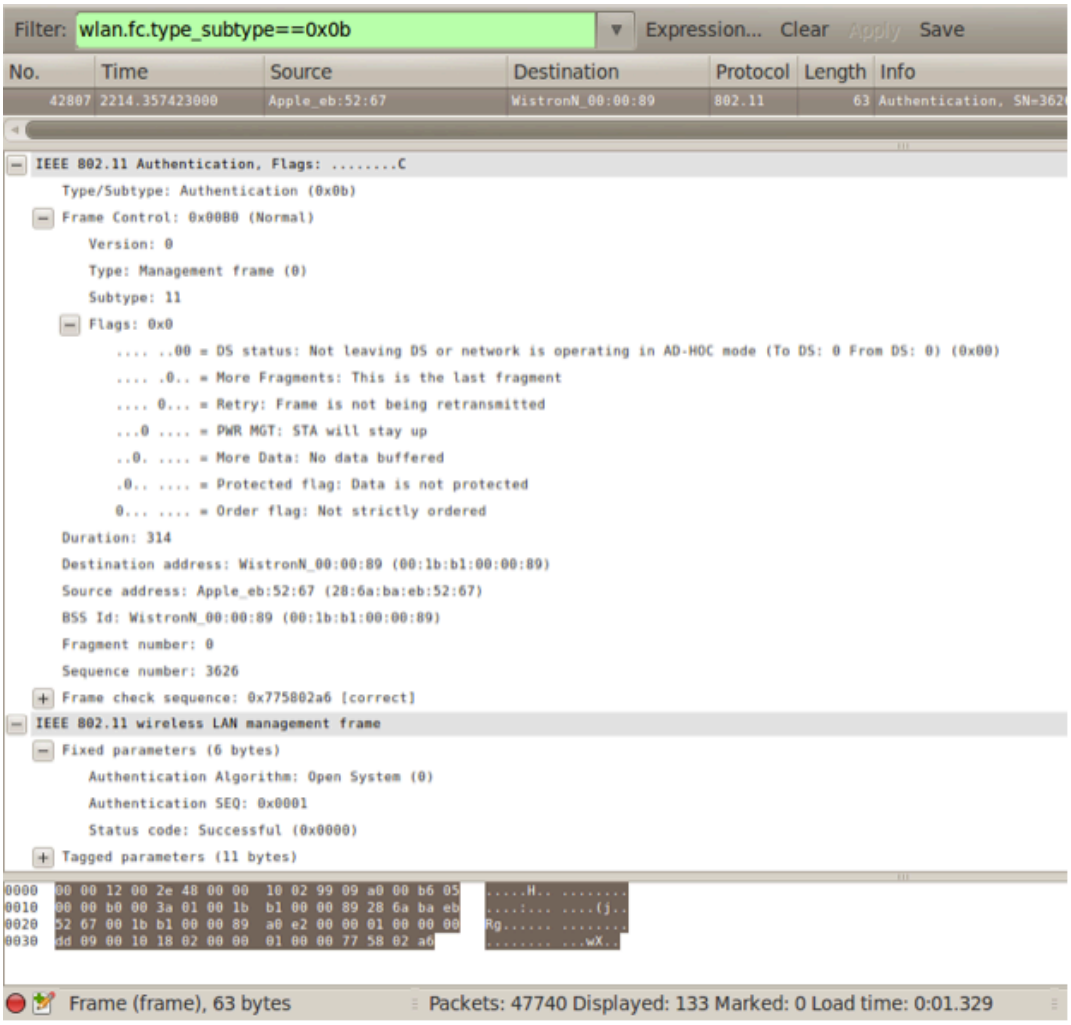


Figura. 112 802.11 Authentication Request Frame

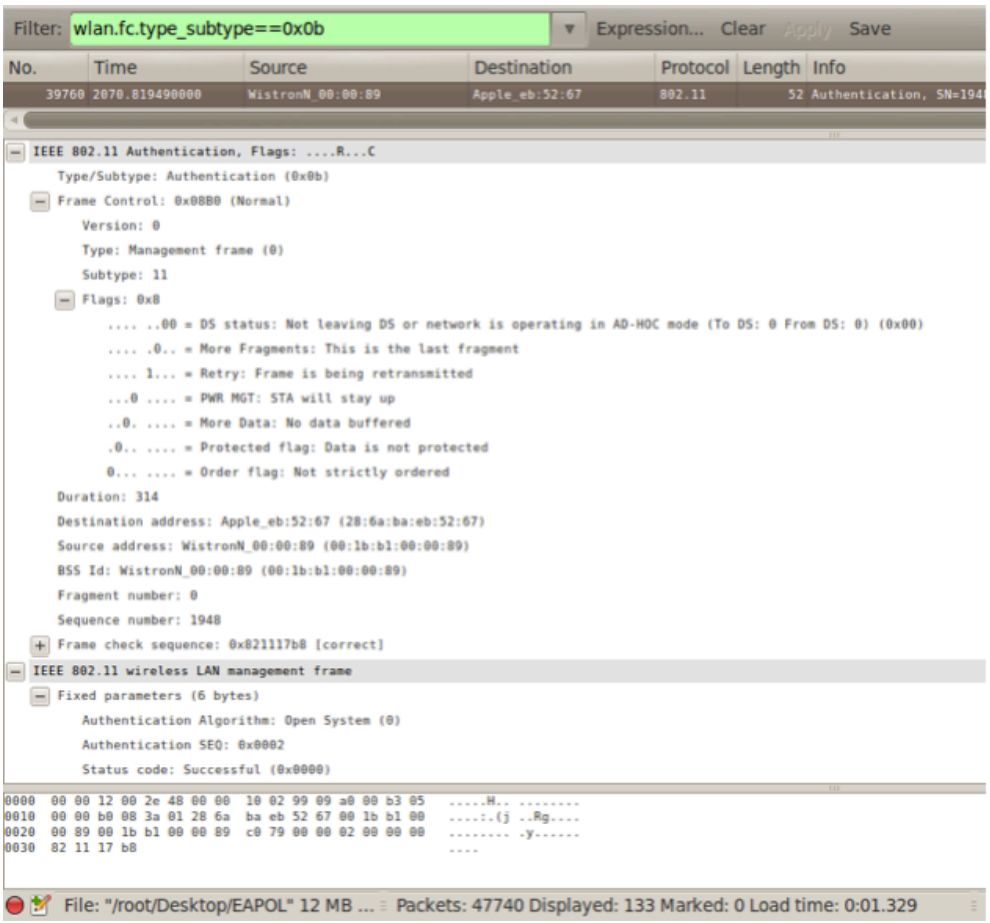


Figura. 113 802.11 Authentication Response Frame

3.2.2 Association Request

Es una trama de tipo de administración *(management frame)* que permite al *Access Point* distribuir los recursos necesarios para sincronizar con una *NIC*. El controlador de interface de red comienza el proceso al enviar una trama de petición de asociación al punto de acceso. Esta trama lleva información sobre la *NIC* y el *SSID* de la red con la que desea conectarse. Luego de recibir esta petición de asociación, el *AP* considera la asociación con la *NIC* y reserva espacio en la memoria junto con una *ID (Identification)* de asociación para la *NIC*.

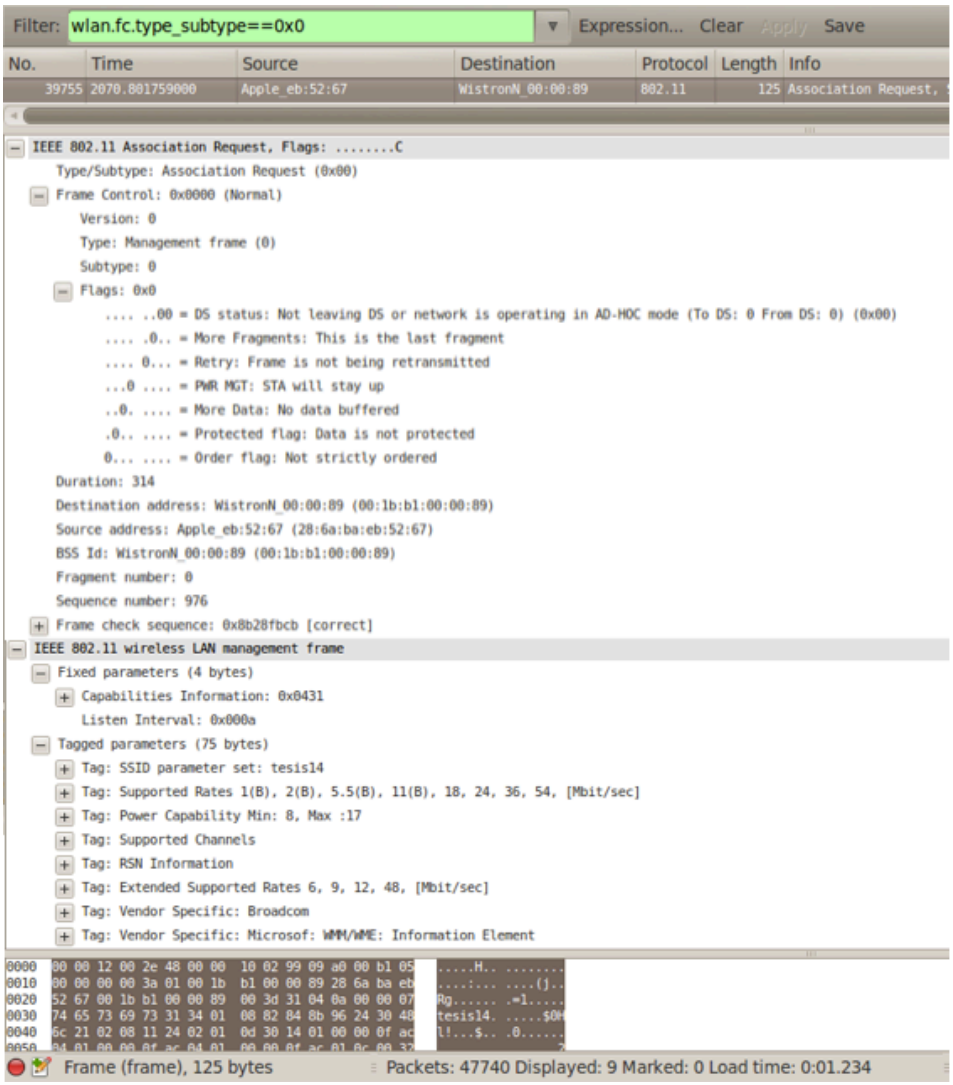


Figura. 114 Association Request Frame

3.2.3 Association Response

Es una trama de tipo de administración *(management frame)* donde el punto de acceso responde con el aviso de aceptación (o rechazo) a la *NIC* junto con un *ID* de asociación. En el caso en que el *AP* acepte a la tarjeta *NIC,* la trama incluirá información como velocidades de datos soportados, etc. Si la asociación es exitosa, la *NIC* puede utilizar el *Access Point* para comunicarse con sistemas de la distribución lateral del punto de acceso.

Figura. 115 Association Response Frame

3.3 Etapa 3: EAP/802.11X/RADIUS Authentication

La etapa siguiente consiste en el intercambio de varios mensajes entre los equipos suplicante, autenticador y el servidor *RADIUS*, como se puede observar en la figura 3.1.1, donde procedimiento se realiza con el fin se establecer un protocolo de autenticación en este caso *EAP-TLS*, donde el autenticador actúa como un distribuidor para el intercambio de tramas, adicional como medio de protección durante esta etapa se genera un túnel para el intercambio seguro de información entre el suplicante y el servidor *RADIUS*. Posterior a haber validado la información y haber autenticado, se

genera una llave de sesión maestra entre ellos *(MSK)*, la que el suplicante utiliza para generar una nueva llave maestra de emparejamiento *(PMK)*, el servidor por su parte envía la clave de forma segura hacia el autenticador por medio del *MSK*, que permite al autenticador generar un *PMK* igual al generado por el suplicante, esto permitirá a los equipos continuar con las siguientes etapas de conexión. A continuación se muestran las tramas de los mensajes generados en la etapa 3.

3.3.1 EAPOL START

El primer mensaje en ser enviado por parte del suplicante, por medio del cual se solicita iniciar la conexión al equipo de acceso en este caso el Access Point. En la siguiente figura se puede observar el mensaje enviado en el escenario de pruebas implementado.

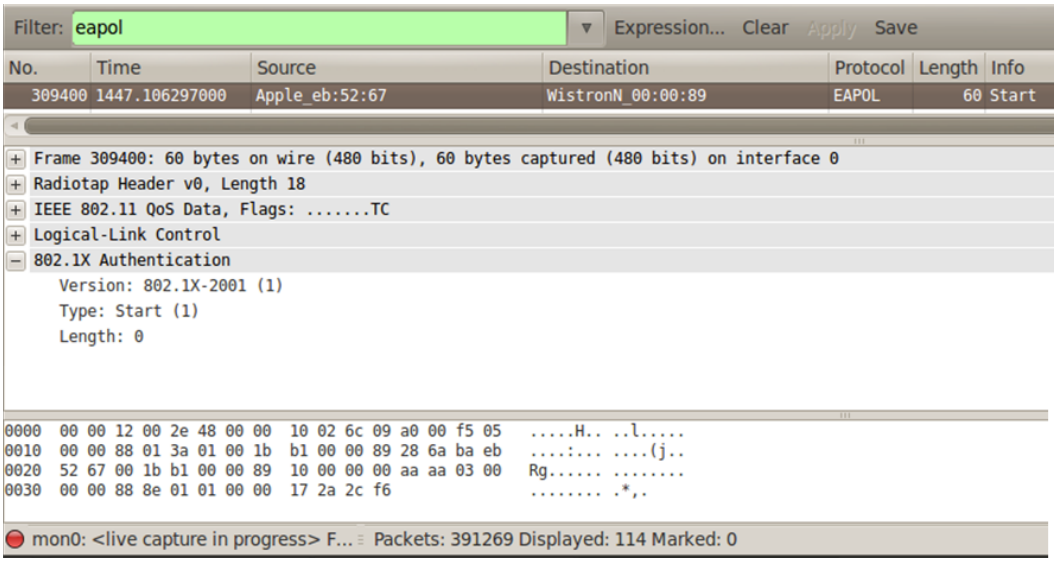


Figura. 116 Mensaje EAPOL START

3.3.2 EAP – Request identity

Este mensaje es enviado desde el autenticador hacia el suplicante, en el cual se solicita la identidad del usuario.

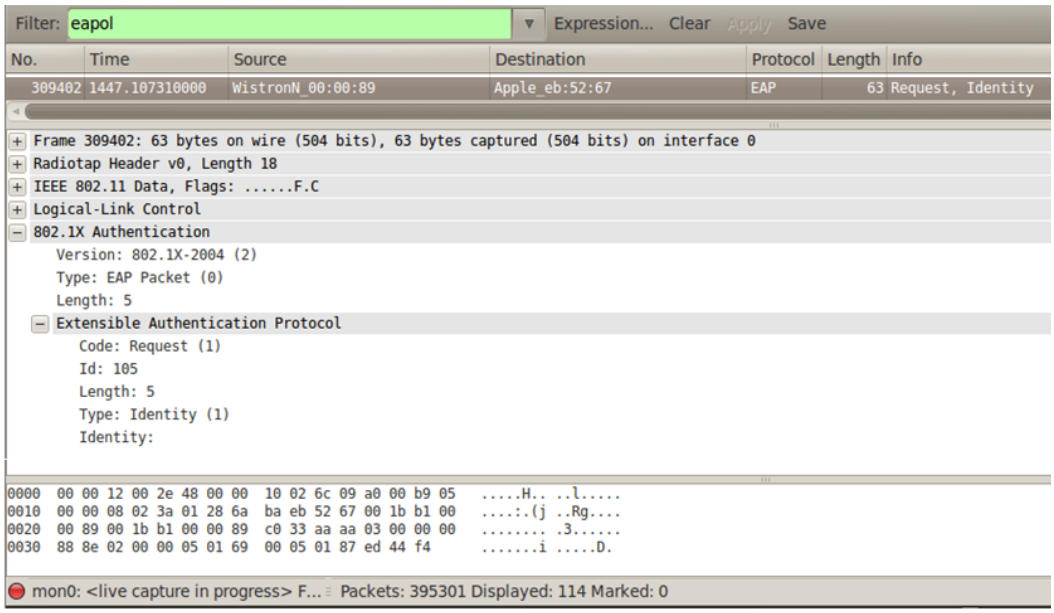


Figura. 117 Mensaje EAP – Request Identity

3.3.3 EAP – Response Identity

Como respuesta al mensaje anterior, el suplicante envía un mensaje en donde se especifica su identidad, es decir su nombre de usuario.

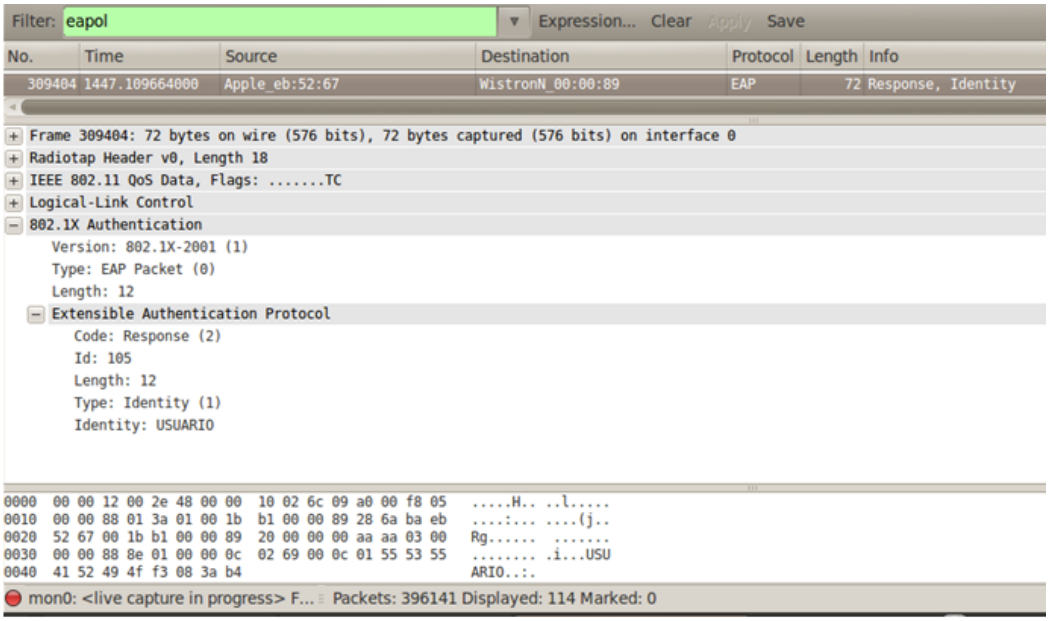


Figura. 118 Mensaje EAP – Response Identity

Enviado el mensaje de *Response identity*, el autenticador lo encapsula en una trama *RADIUS Access Request*, y lo envía hacia el servidor, el

mismo que toma la identidad dentro del mensaje y la consulta dentro de la base de datos, de ser correcta la identidad enviada, el servidor *RADIUS* envía un mensaje de *RADIUS Access Challenge*, para iniciar la negociación del tipo de canal *EAP*, para el establecimiento de un túnel para el intercambio de información.

3.3.4 EAP Request Start PEAP

El autenticador requiere establecer el canal de intercambio de información *EAP* de tipo *PEAP*, por lo que envía al suplicante el siguiente mensaje.

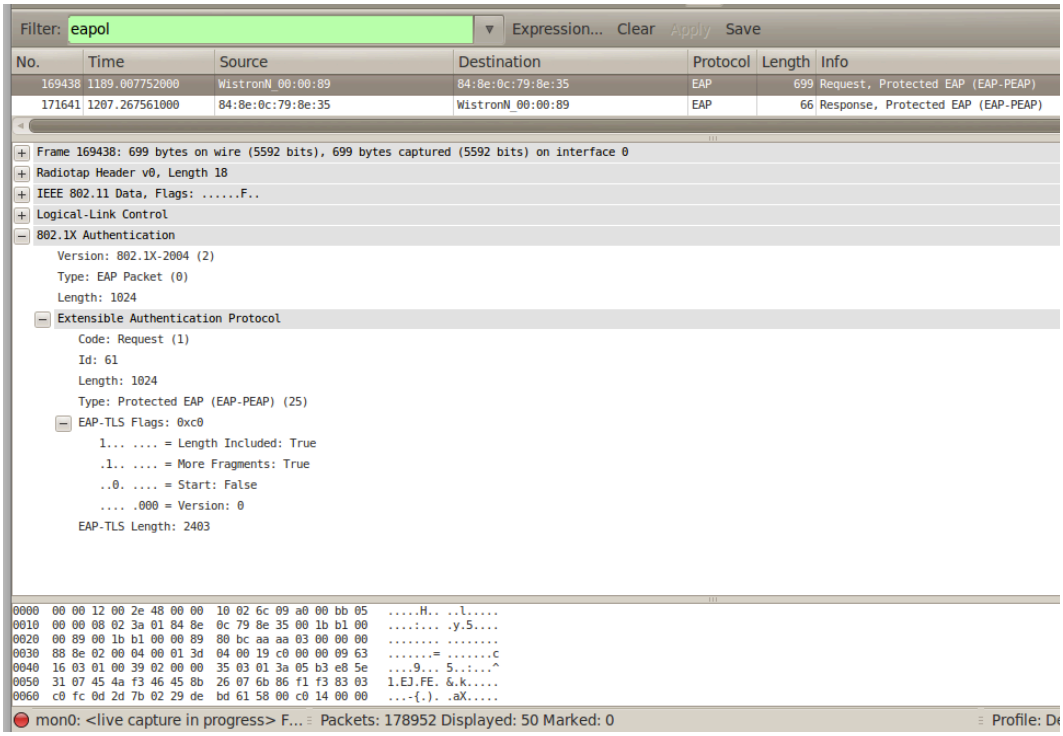


Figura. 119 Mensaje EAP Request PEAP

3.3.5 EAP Response TLS Client Hello

Mediante el mensaje anterior y el presente mensaje inicia la negociación del tipo de canal, donde el suplicante envía el mensaje *TLS Client Hello*, para el establecimiento de canal *TLS*.

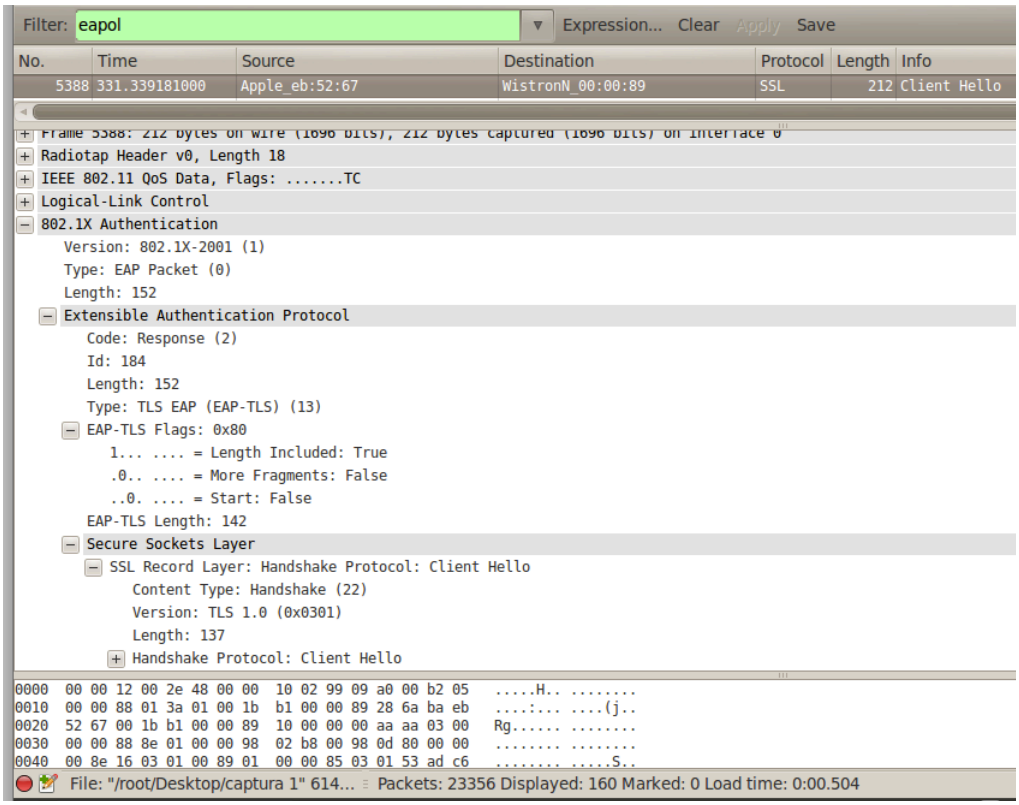


Figura. 120 Mensaje EAP response TLS Client Hello

El mensaje es encapsulado en una trama *RADIUS request*, por parte del autenticador y es enviado hacia el servidor, el mismo que verifica el mensaje y responde enviando el certificado mediante un mensaje *RADIUS Challenge*, este mensaje contiene varios parámetros como : *server hello*, *server certificate*, *server done*.

3.3.6 EAP Request TLS Certificate

El autenticador recibe el mensaje *RADIUS Challenge* enviado por el servidor, y lo envía hacia el suplicante, en donde se incluye el certificado.

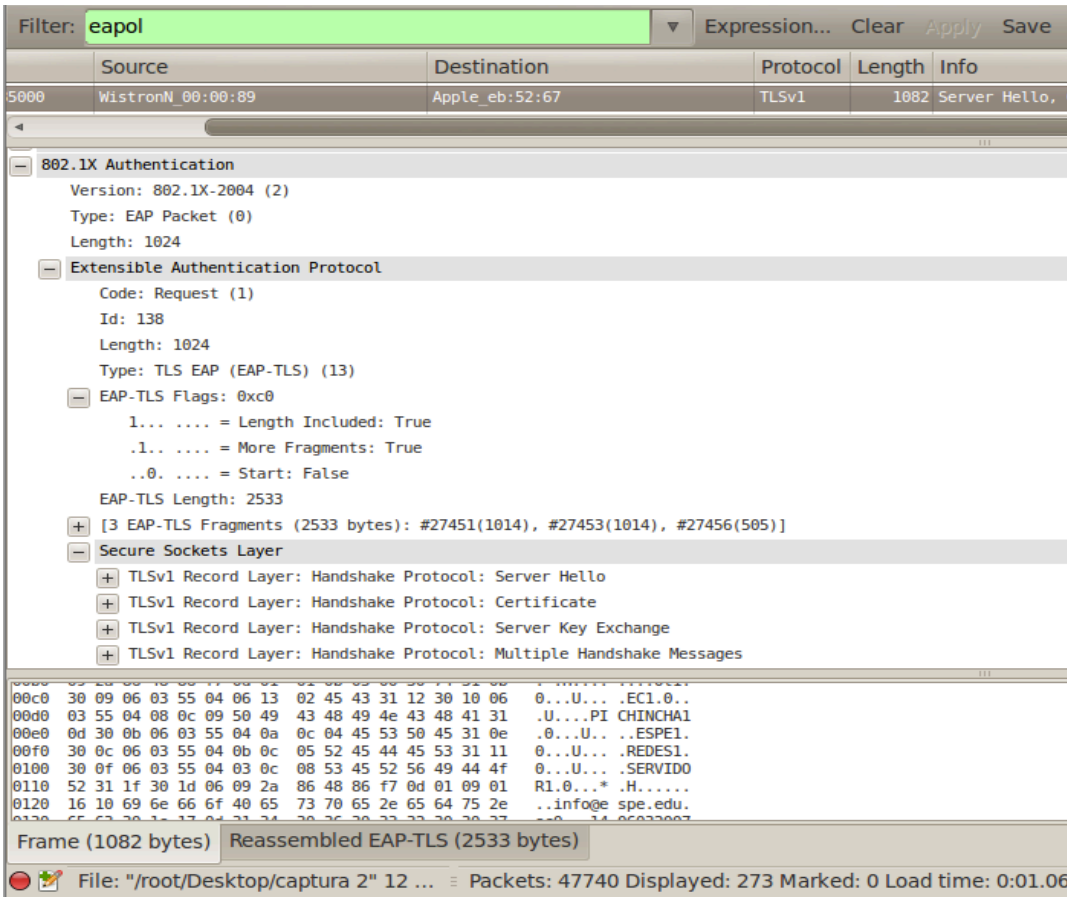


Figura. 121 Mensaje EAP Request TLS Certificate

3.3.7 EAP Response TLS Key Exchange

En suplicante envía hacia el servidor un mensaje, que contiene la clave del usuario por medio del canal cifrado establecido.

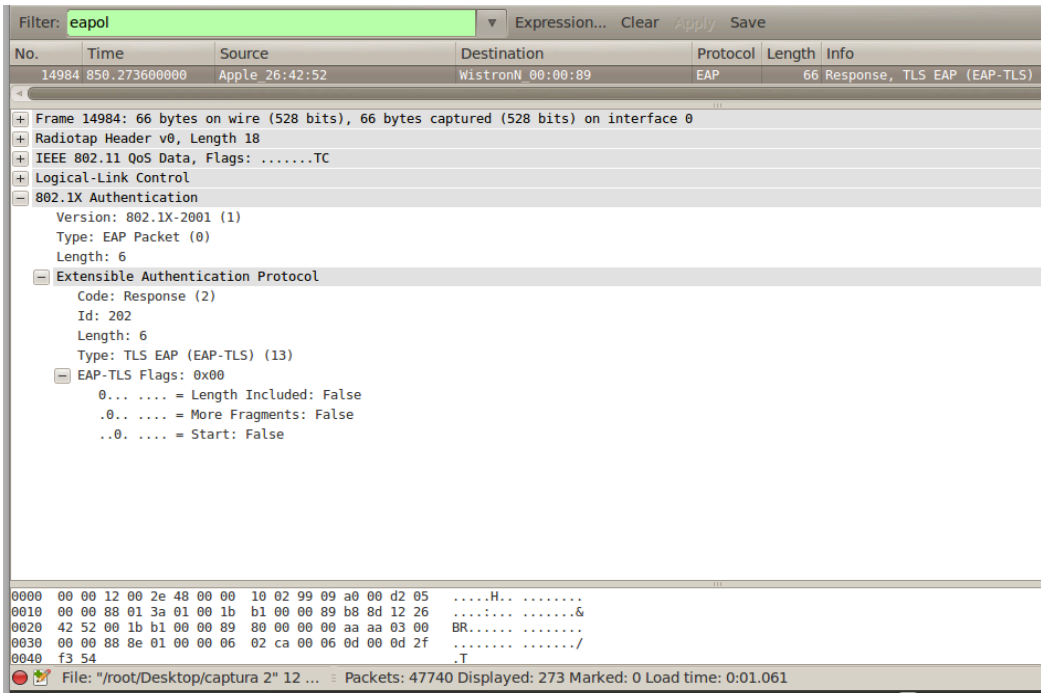


Figura. 122 Mensaje EAP Response TLS Key Exchange

La trama es encapsulada en una nueva trama *RADIUS Request*, la misma que contiene las credenciales de usuario, y es enviada hacia el servidor para que sean consultadas en la Base de Datos.

- De comprobar que las credenciales son correctas el servidor responde con un mensaje *RADIUS challenge*, estableciendo el canal seguro de tipo *TLS*.

- De no ser correctas las credenciales el servidor responderá con un mensaje *RADIUS Reject*, finalizando el proceso para establecer el canal.

3.3.8 EAP Request Cipher TLS Complete

El autenticador recibe el mensaje del servidor *RADIUS*, y lo envía hacia el suplicante, para finalizar el establecimiento del canal.

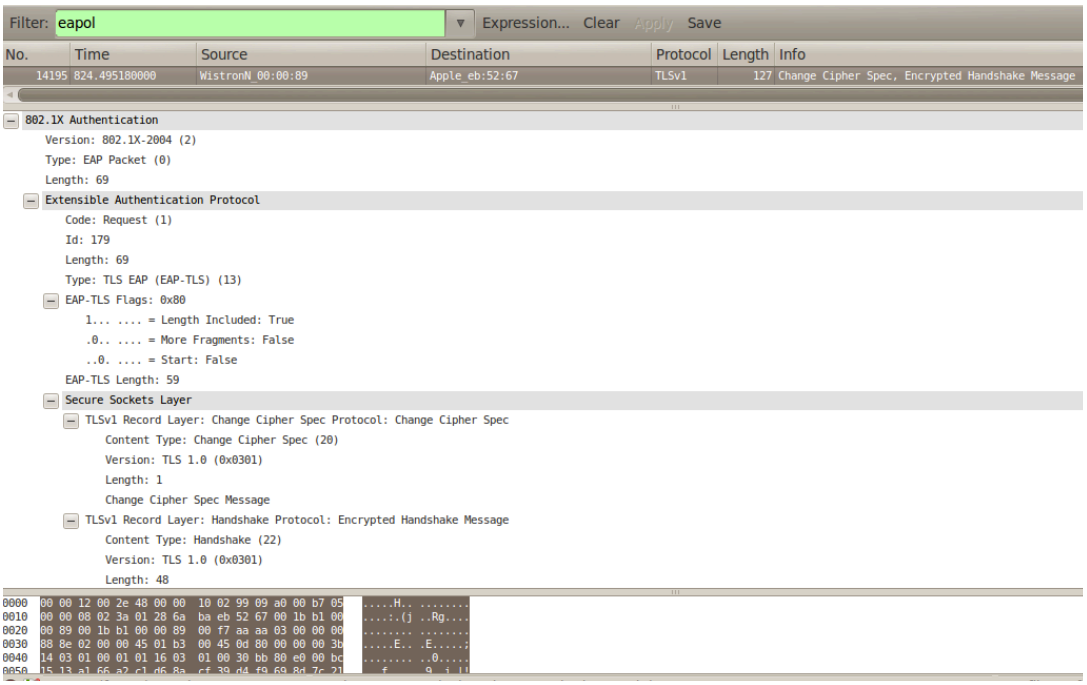


Figura. 123 EAP Request Cipher TLS Complete

3.3.9 EAP Response PEAP

Este mensaje es enviado por parte del suplicante hacia el autenticador, para solicitar el acceso a la red.

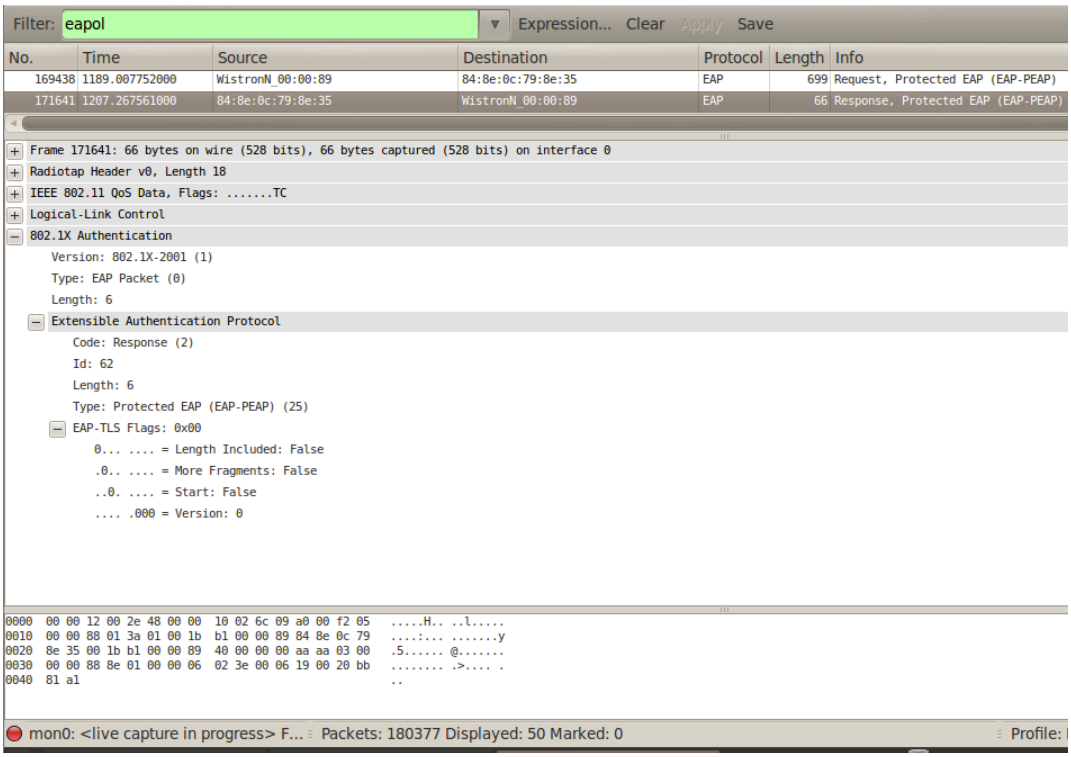


Figura. 124 EAP Response PEAP

El autenticador recibe el mensaje, lo encapsula en un mensaje *RADIUS Request*, y responde enviando un mensaje *RADIUS Accept*, hacia el autenticador.

3.3.10 EAP SUCCES

Finalmente el autenticador envía una el mensaje hacia el suplicante, el mismo que se encuentra habilitado para hacer uso de la recursos de la red.

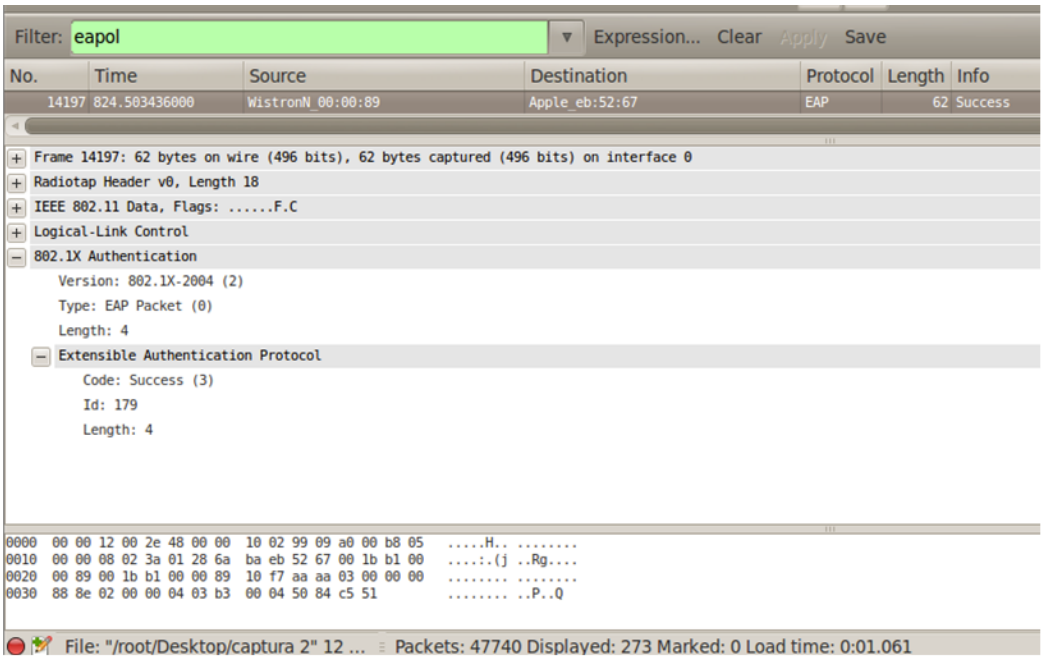


Figura. 125 EAP Success

3.3.11 Mensajes RADIUS

Los mensajes que se ven en la imagen a continuación, son la que fueron descritas en cada uno de los pasos anteriores, si bien son importantes en el proceso de validación, cabe señalar que se intercambian entre el autenticador y el servidor por medio de *Ethernet*.

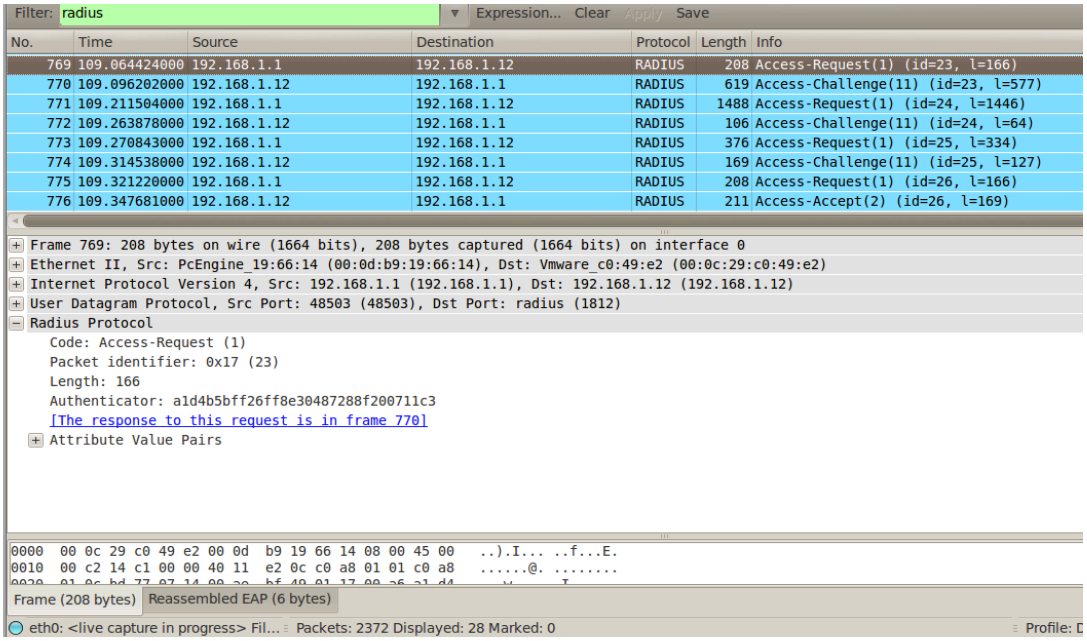


Figura. 126 EAP Success

3.4 Etapa 4: 4-Way Handshake

Se explicará los mensajes del *(27)* al *(30).* El Access Point necesita autenticarse con el suplicante *(STA)* y las llaves para encriptar el tráfico necesitan establecerse. En intercambio de *EAP* analizado anteriormente, ha compartido una llave secreta *PMK (Pairwise Mater Key)*. Esta llave es diseñada para durar toda la sesión y debe ser expuesta lo mínimo posible. Por lo tanto, el apretón de manos de 4 vías *(4-Way Handshake)* es usada para establecer otra llave llamada *PTK (Pairwaise Transient Key)*. La *PTK* es generada al concatenar los atributos de: *PMK, AP nonce (ANonce), STA nonce (SNonce), AP y STA MAC address*. Esta combinación es sometida a *PBKDF2 (Password-Based Key Derivation Function 2)* como la funciona criptográfica. Al completarse esta etapa la *PTK* es compartida entre el autenticador y el suplicante, entonces los puertos *802.11X* son desbloqueados para los paquetes de datos. Este proceso se detalla a continuación:

3.4.1 Mensaje 1

El *AP* envía el msg1 al suplicante, este contiene los parámetros: dirección *MAC* del *Access Point* (*AA*), un número aleatorio escogido por el autenticador (*ANonce*) y un contador para prevenir un ataque de repetición (*sn*).

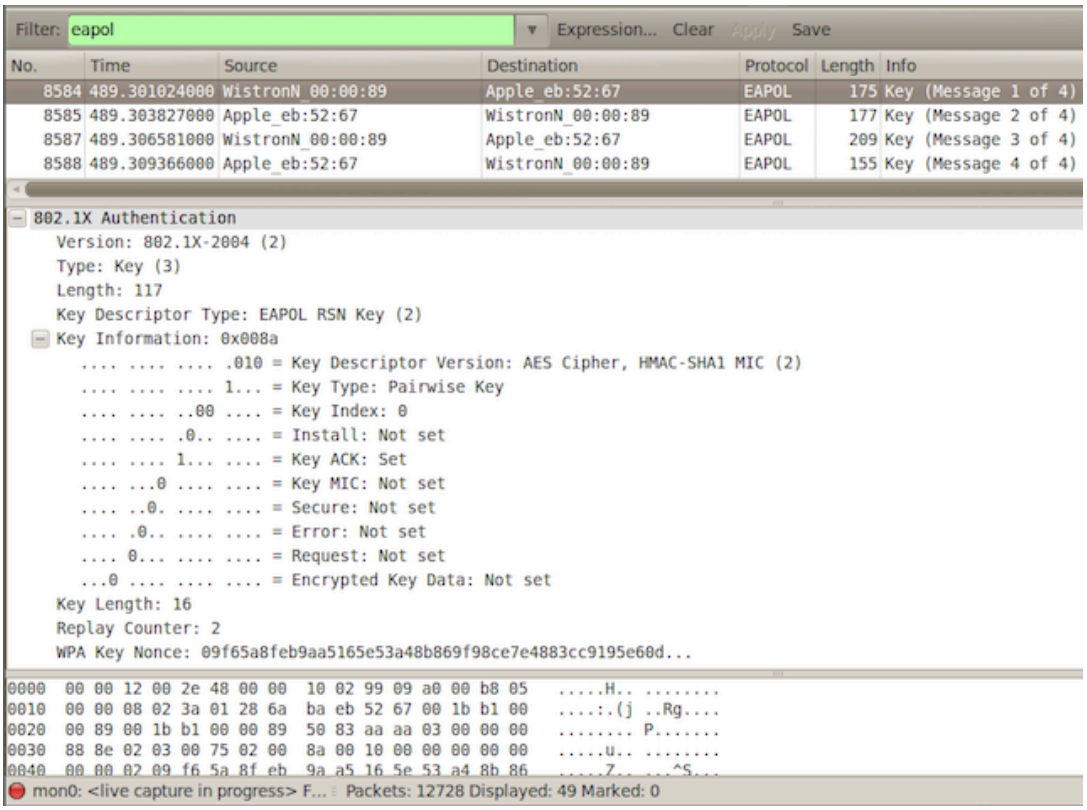


Figura. 127 Handshake Mensaje 1

3.4.2 Mensaje 2

El suplicante, genera dos parámetros adicionales: la dirección *MAC* del cliente (*SPA*) y un número aleatorio escogido por el cliente (*SNonce*). El suplicante deduce el *PTK* a partir de cinco parámetros: *AA, ANonce, SPA, SNonce* y el *PMK*. Se debe considerar que el *PTK* no se deduce entre el cliente y el *AP* en el primer mensaje, esto solo sucede después de que se recibe el msg 2 usando el *MIC (Message Integrity Code)*.

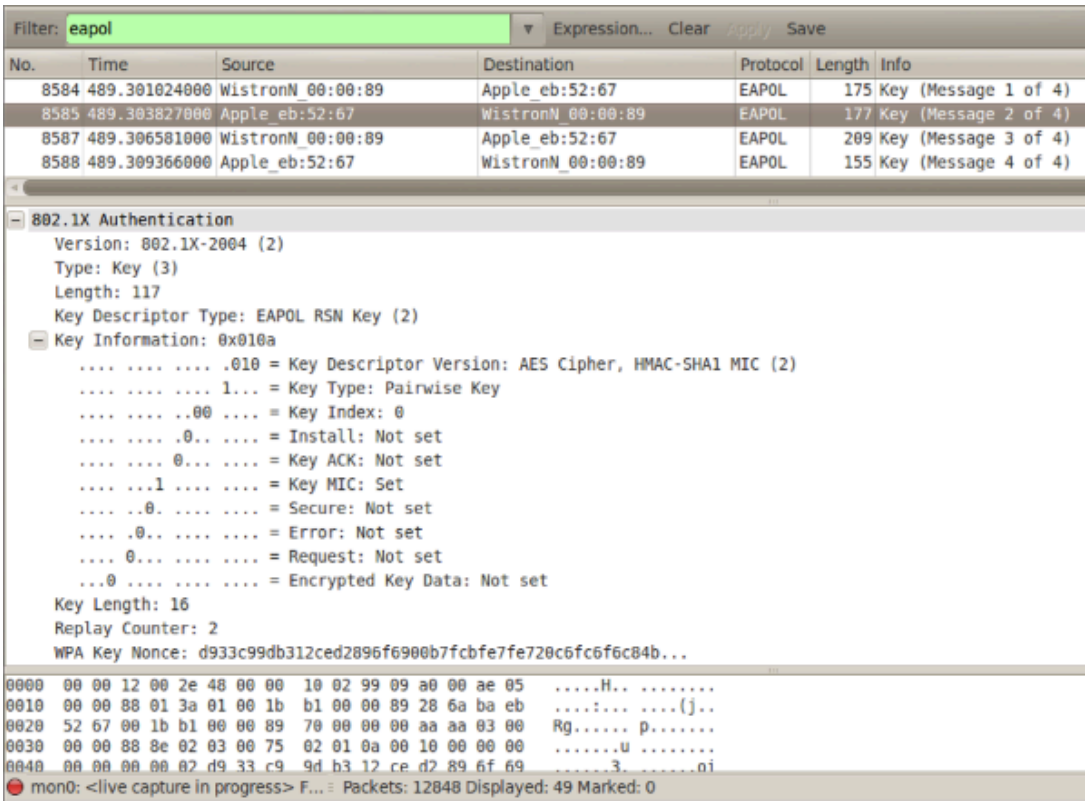


Figura. 128 Handshake Mensaje 2

3.4.3 Mensaje 3

El *AP* envía un número de secuencia o contador *(sn+1)* junto con otro *MIC*. Este número de secuencia se usa en la siguiente trama de difusión, de modo que el suplicante puede realizar la detección básica de repetición. En este paso el *AP* verifica que el suplicante tenga el mismo *PTK*.

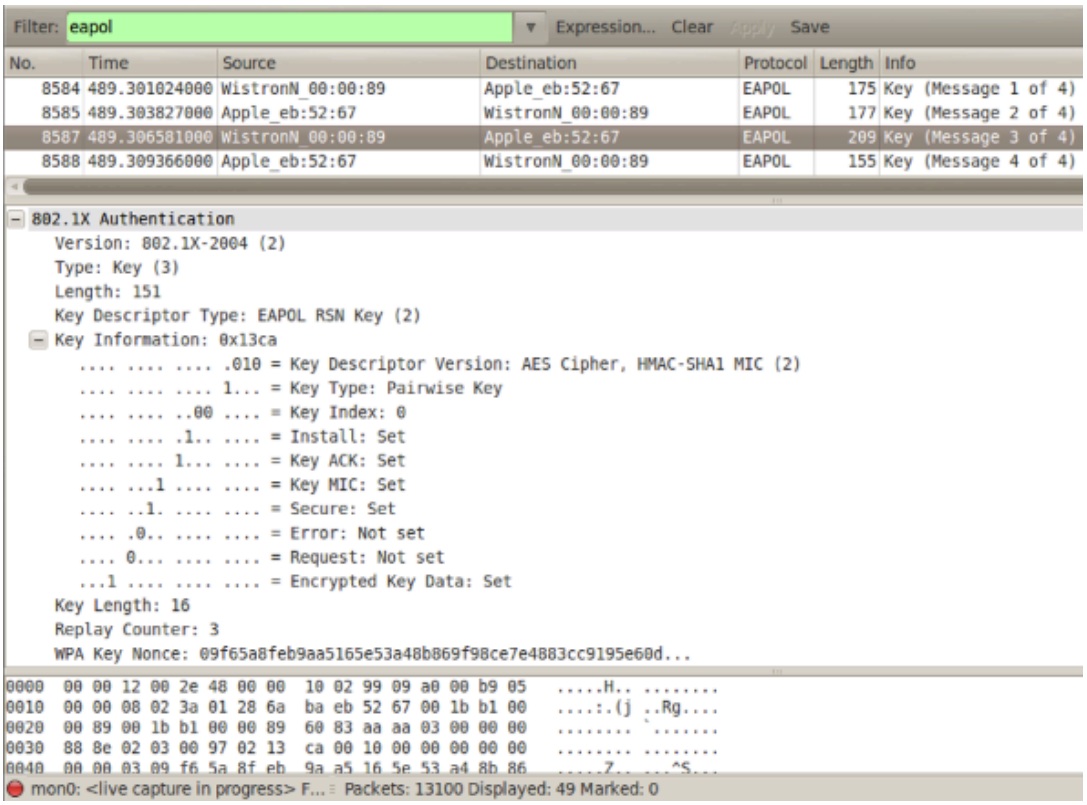


Figura. 129 Handshake Mensaje 3

3.4.4 Mensaje 4

El suplicante envía una confirmación con el mismo contador *(sn+1)* al autenticador sobre la recepción del *msg3.* Todas los mensajes se envían como tramas bajo el protocolo *EAPoL (Extensible Authentication Protocol over LAN).*

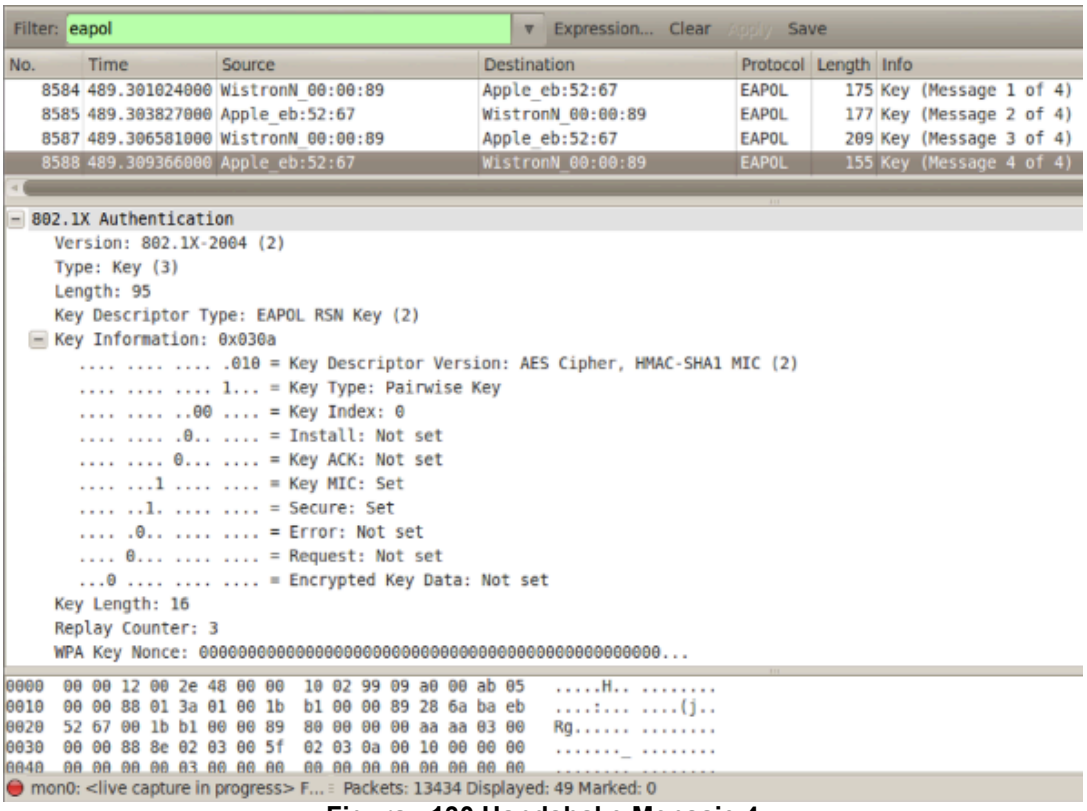


Figura. 130 Handshake Mensaje 4

3.5 Etapa 5: Tramas de Datos / Secure Data Communication

La trama de datos *(31)* se establece una vez que la comunicación se ha establecido entre el suplicante y el autenticador al completar las etapas anteriores. Esta trama transporta información de capas superiores como por ejemplo páginas web, etc.

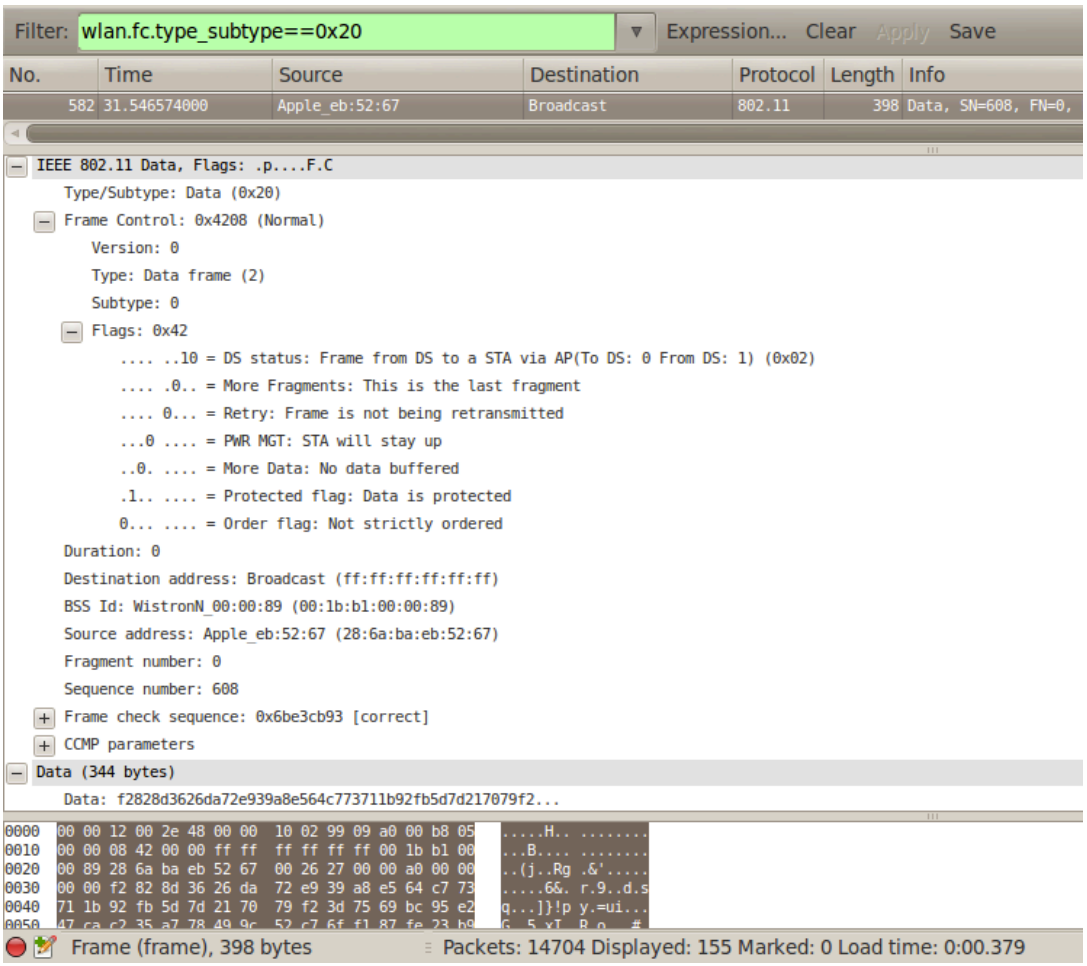


Figura. 131 Data Frame

3.6 Tramas de Control

Las trama de control ayudan en el intercambio de tramas de Administración y Datos. Estas tramas, administran el acceso al medio inalámbrico y proporcionan funciones de fiabilidad a nivel de la capa de enlace.

3.6.1 Request-to-Send (RTS)

La función *RTS* junto con la *CTS (Clear-to-Send)* reducen las colisiones de las tramas presentes cuando existe una conexión con un mismo *AP*. Una estación envía una trama *RTS / CTS* a otra como la primera fase de un enlace de dos vías necesario antes de enviar cualquier trama de datos.

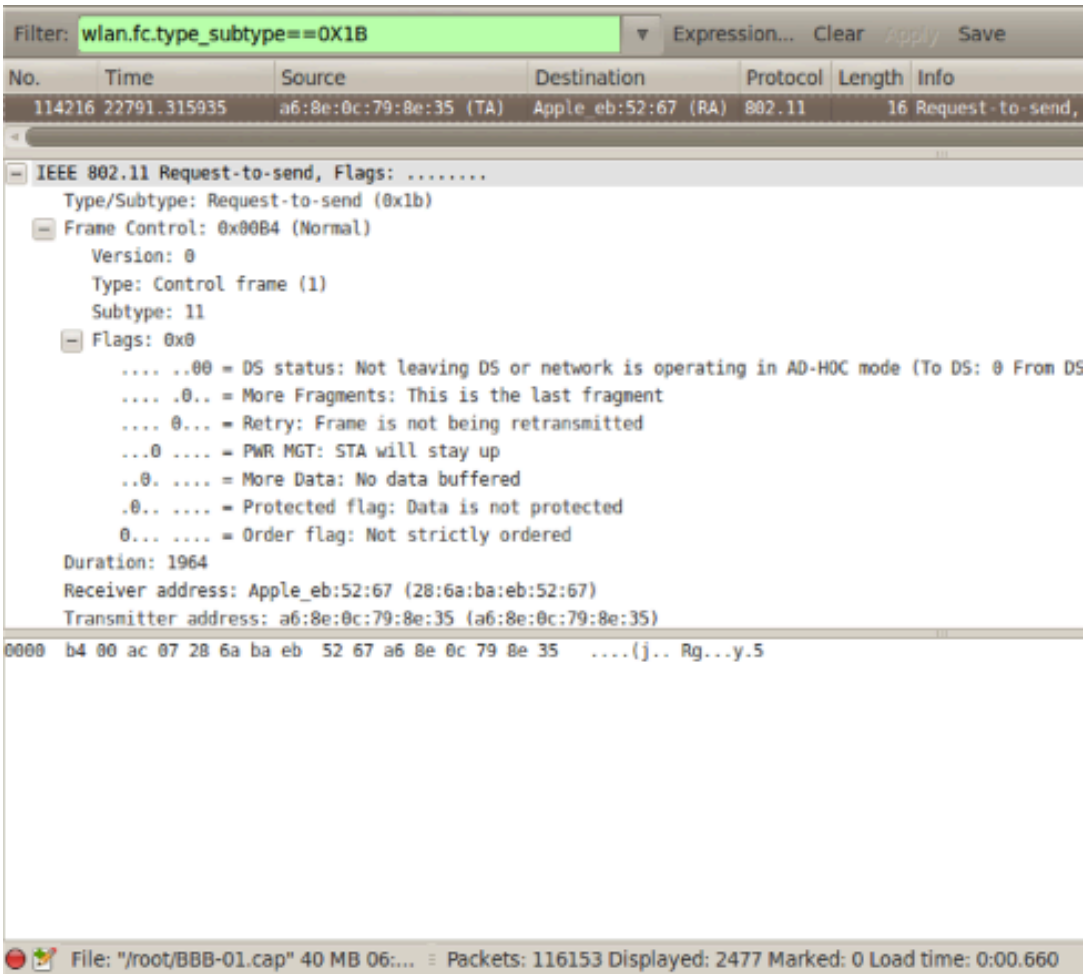


Figura. 132 Request to Send Frame

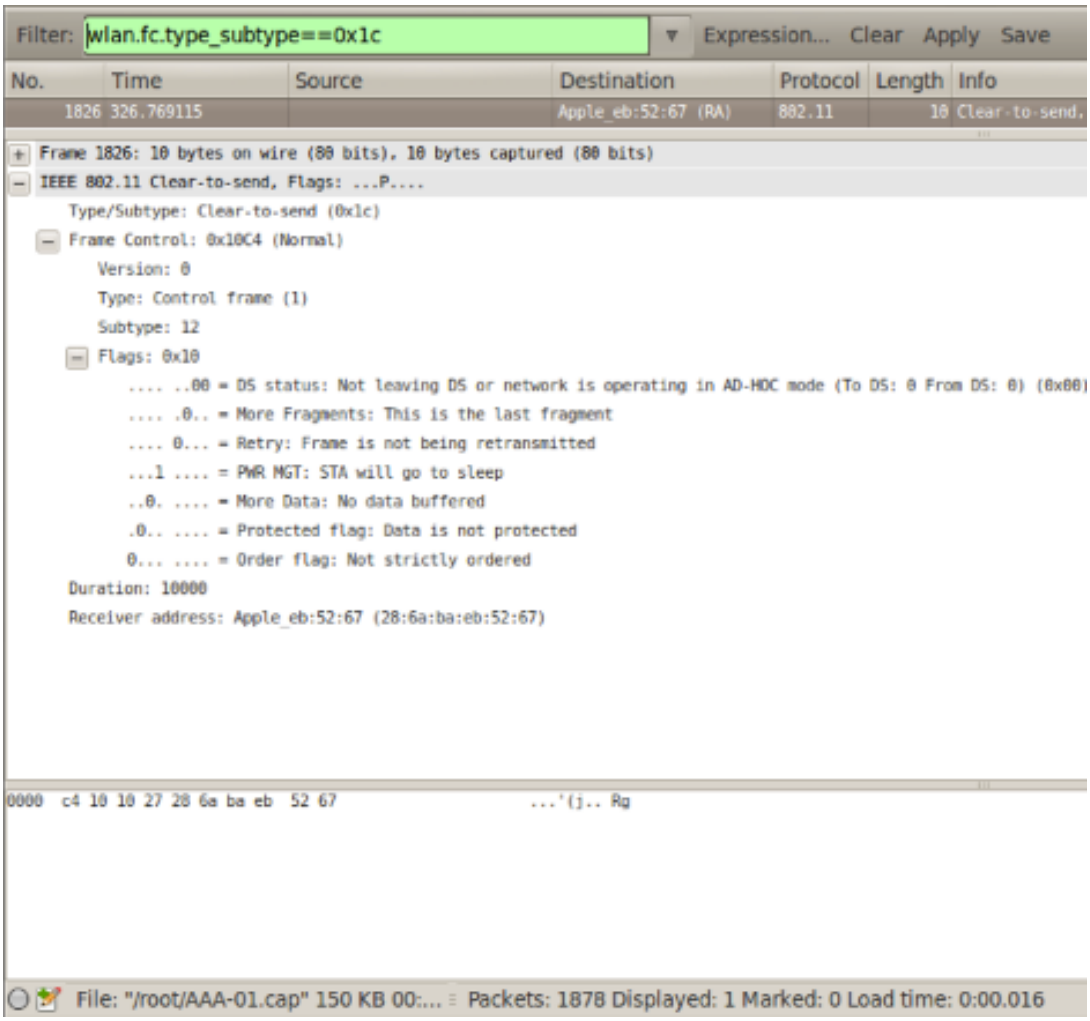


Figura. 133 Clear to Send Frame

3.6.2 Acknowledgment (ACK)

Después de recibir una trama de datos, la estación receptora utilizará una comprobación de errores. La estación receptora entonces envía una trama *ACK* a la estación emisora si no se encuentra ningún error. En caso de que existiera un error, no se envía la trama *ACK*, con lo que después de un periodo de tiempo, la estación de envió retrasmitirá esta trama.

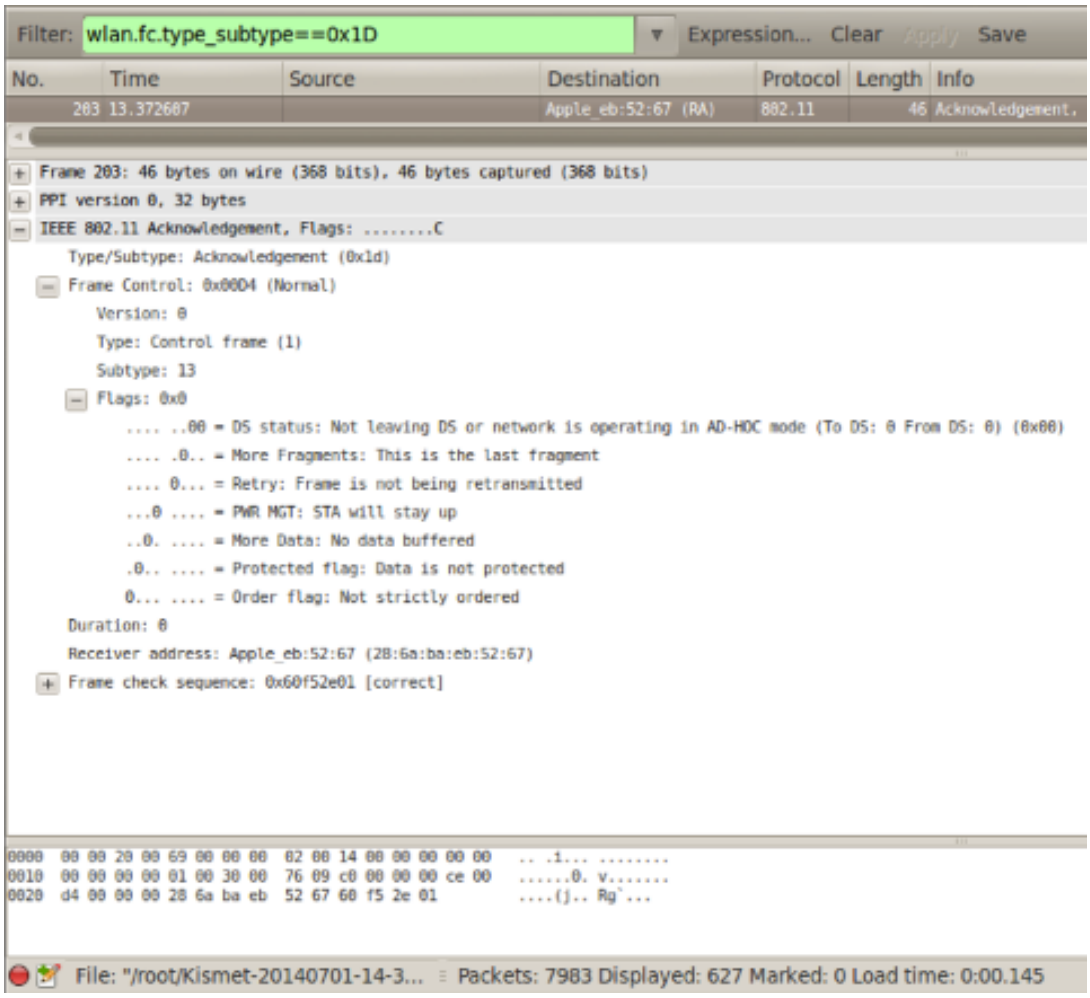


Figura. 134 Acknowledgement Frame

CAPÍTULO IV

ESCENARIO DE PRUEBA INTRUSIVO

Con el fin de realizar el estudio de vulnerabilidades es importante analizar las posibles capacidades de cualquier hacker o adversario. A partir de la capa de enlace de una *WLAN*, existen tres posibles tipos de tramas que ya han sido analizadas en el escenario de prueba normal, estas son: Administración, Control y Datos. Como se demuestra en este capítulo, cualquier manipulación de estas tramas puede comprometer la confidencialidad, integridad y autenticación.

Los diferentes ataques en el presente capítulo fueron realizados sobre el escenario que se ha implementado y cuyo comportamiento ha sido descrito en capítulos anteriores. Estos escenarios de prueba se encuentran en un modelo de red de infraestructura bajo el estándar *802.11i* y un mecanismo de autenticación *EAP-TLS*, considerado el más seguro y robusto en comparación a otros mecanismos *EAP* existentes. *EAP-TLS* se encuentra analizado a profundidad y no ha podido ser demostrado o evidenciado deficiencias graves dentro del protocolo tales como vulnerabilidades a ataques *MITM*, esto gracias a su característica principal que, como norma para realizar la autenticación y el establecimiento del túnel cifrado, debe existir un intercambio de certificados digitales, es decir que, el servidor envía su certificado hacia el cliente y de forma recíproca, el cliente envía el certificado hacia el servidor. Cabe señalar que el certificado del cliente debe ser instalado previamente y debe estar protegido por una contraseña. Cada uno de los certificados debió haber sido firmado por la misma entidad, de no ser así, serán rechazados y la conexión no será establecida. Si bien el proceso de autenticación de *EAP-TLS* lo define como el protocolo más

seguro en la protección de confidencialidad y autenticación, sigue siendo vulnerable a ataques de denegación de servicios *(DoS)*.

El protocolo de autenticación *TLS* presenta una desventaja para la implementación ya que para redes de media o gran escala la instalación y distribución de los certificados se torna en un proceso tedioso y complicado.

4.1 Amenaza 1: Análisis de tráfico Pasivo

Debido a la características de comunicación inalámbrica, un adversario puede fácilmente esnifar la red con herramientas como Wireshark y obtener información que puede ser almacenada y posteriormente analizada para revelar claves de cifrado o información sobre la red *WLAN*. Los ataques pasivos son difíciles de detectar, esto por cuanto no se realiza ninguna modificación de los datos o paquetes trasmitidos pero aporta valiosa información de la red. Cuando se realiza intercambio de información, tanto el emisor como receptor no son conscientes del monitoreo de un tercero en su intercambio. La información que se captura al realizar una esnifada son las tramas que se han venido estudiando: Control, Administración y Datos.

Utilizando BT5 se puede visualizar esta información de la siguiente manera:

>>root@bt:~#airmon-ng start wlan0
>>root@bt:~#airodump-ng mon0

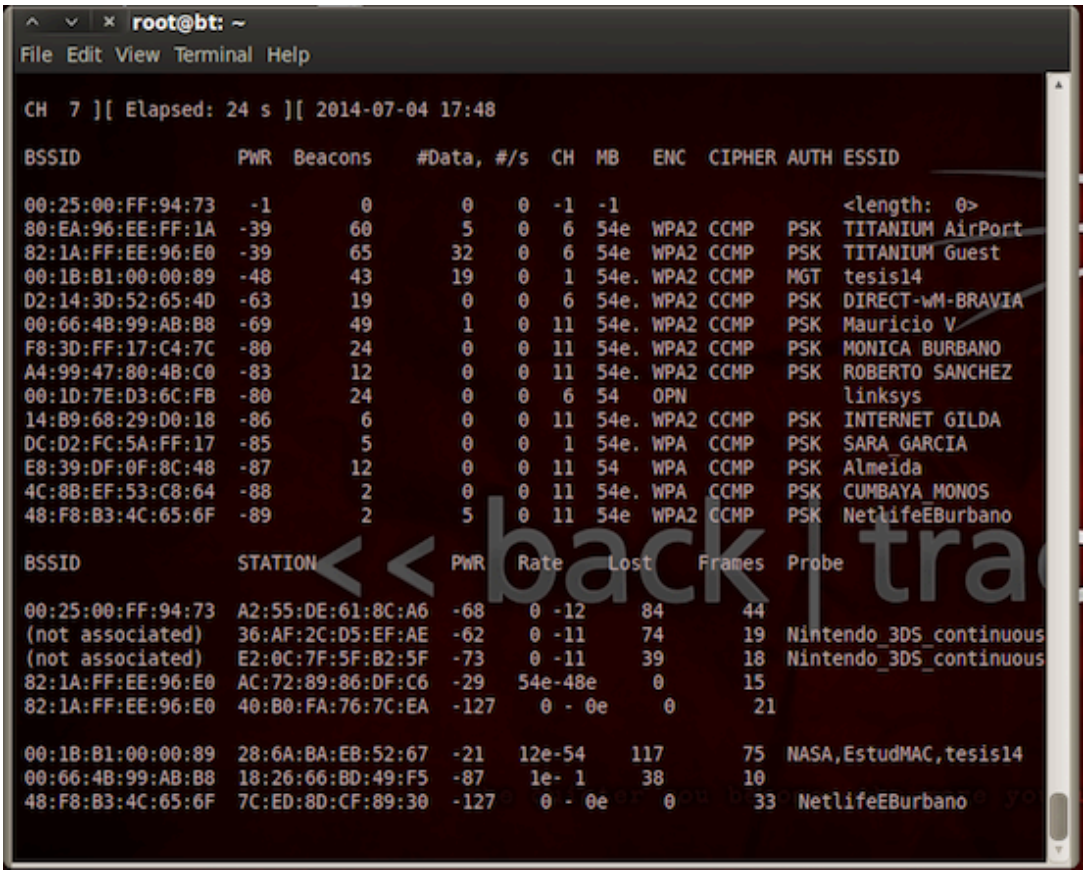


Figura. 135 airodump-ng mon0

Para almacenar la información que se captura en un archivo formato *.cap* de un *AP* especifico, se pone el siguiente comando:

>>root@bt:~#airodump-ng -c (CH) -w (nombre del archivo) --bssid (BSSID) mon0

4.2 Amenaza 2: Denegación de Servicios (DoS)

El estándar *802.11i* con *EAP-TLS* es vulnerable a los ataques *DoS (Denial-of-Service)*. El atacante de denegación de servicio es capaz de interrumpir la conexión legítima de suplicantes con el autenticador por medio de las características propias de las redes inalámbricas, un atacante por ejemplo puede denegar servicios mediante la falsificación de tramas de Administración no protegidas como las de deautenticación. En este capítulo,

se consideran ataques de *DoS* de uso frecuente que requieren un esfuerzo razonable y no elaborado por parte del adversario para su análisis.

4.2.1 Inundación EAPoL-Start

El protocolo *802.1x* como se ha estudiado anteriormente, comienza con una trama *EAPOL-Start* que es enviada por el cliente para iniciar la autenticación y continua con la respuesta del *AP* al enviar una trama *EAP Request Identity.* Un atacante interrumpirá un Access Point al inundarlo con tramas *EAPoL – Start* para agotar los recursos internos del punto de acceso. Este ataque se puede lograr en BT5 al utilizar el siguiente comando:

>>root@bt:~# mdk3 mon0 x 0 –t (BSSID) –n (ESSID) –s (Speed-Paquetes/s) –c (CH)

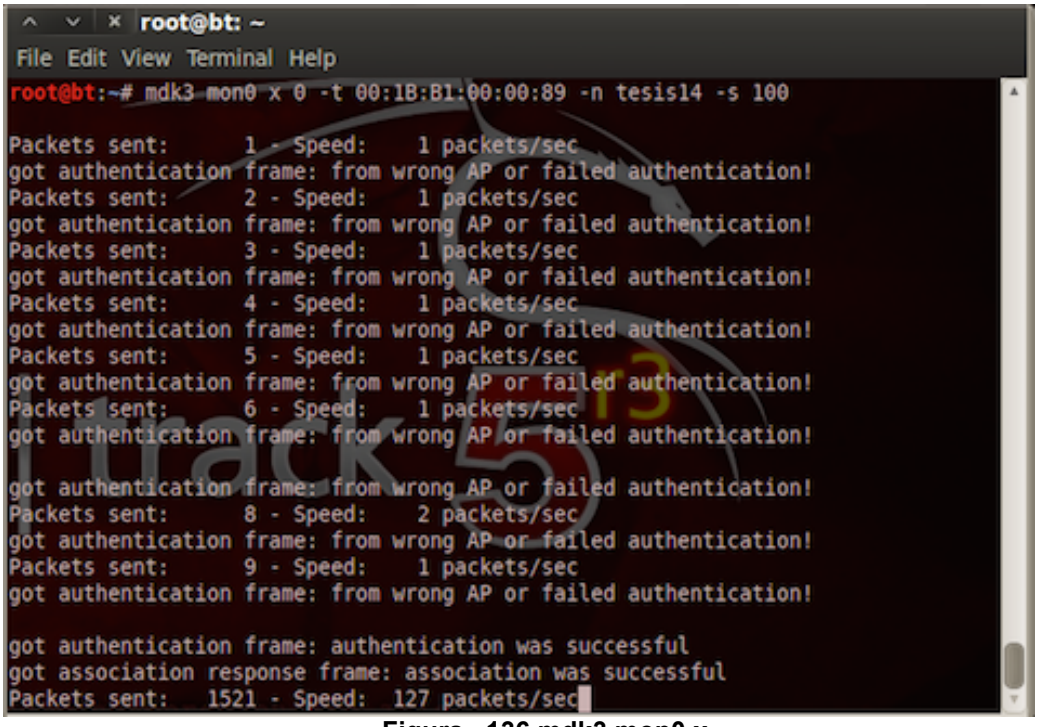


Figura. 136 mdk3 mon0 x

A continuación se ve la captura de las tramas tipo *EAPOL – Start* generadas a través de Wireshark con el objetivo de agotar los recursos del *AP* y obligar a este equipo su reseteo. Se evidencia en el *Source,* la creación

de direcciones *MAC* aleatorias de suplicantes realizando peticiones de tramas *EAPOL – Start* al punto de acceso.

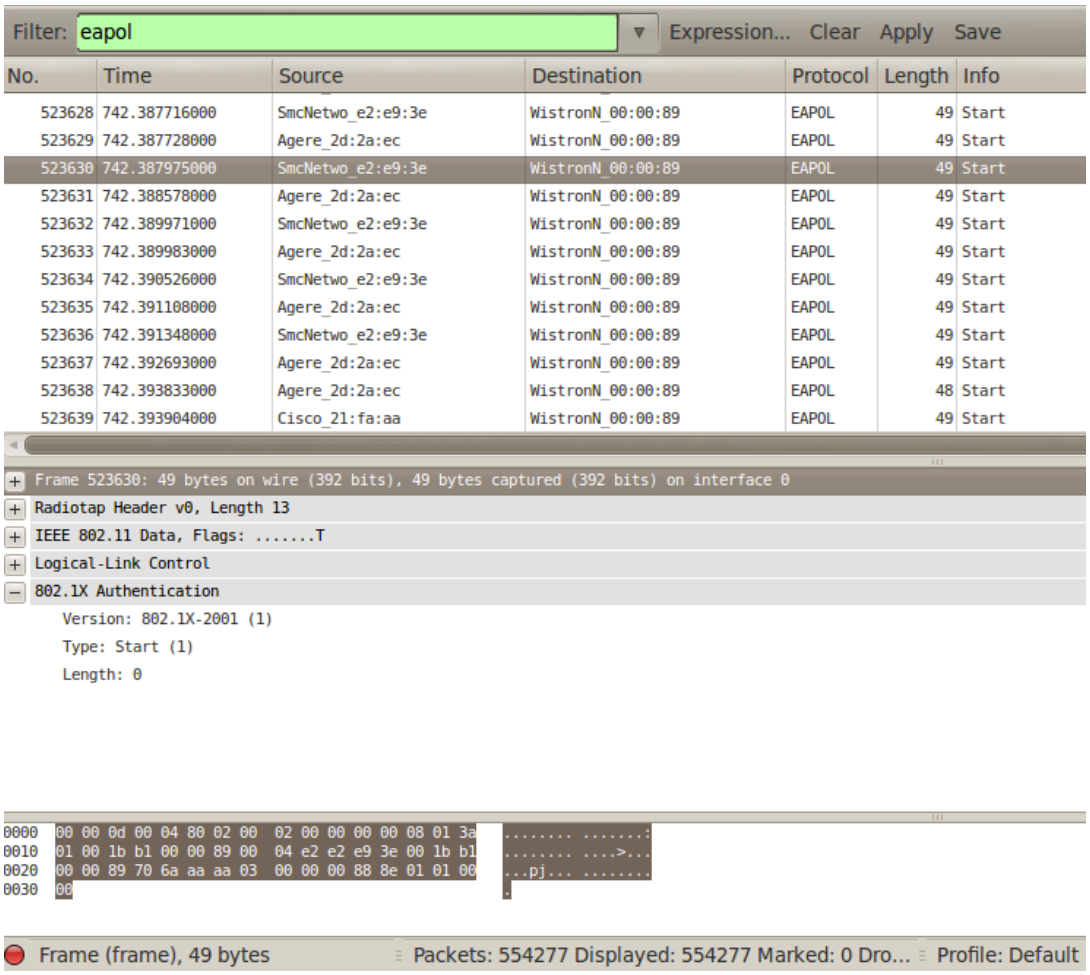


Figura. 137 Wireshark: Inundación EAPOL – START

4.2.2 Inundación Authentication

En el *Access Point*, cada cliente tiene su estado almacenado en la tabla de asociación. Este estado, tiene un límite de tamaño. Una forma de ataque *DoS* es inundar esta tabla de asociación al crear de forma aleatoria varias tramas de *Authentication Request* desde *MAC* ficticias hacia el *AP*. El punto de acceso al no poder verificar todas estas solicitudes alcanzará su límite y al hacerlo, no podrá autenticar clientes legítimos en el *AP*. Este ataque se lo realiza con el siguiente comando:

>>root@bt:~# *mdk3 mon0 a 0 –t (BSSID) –n (ESSID) –s (Speed-Paquetes/s) –c (CH)*

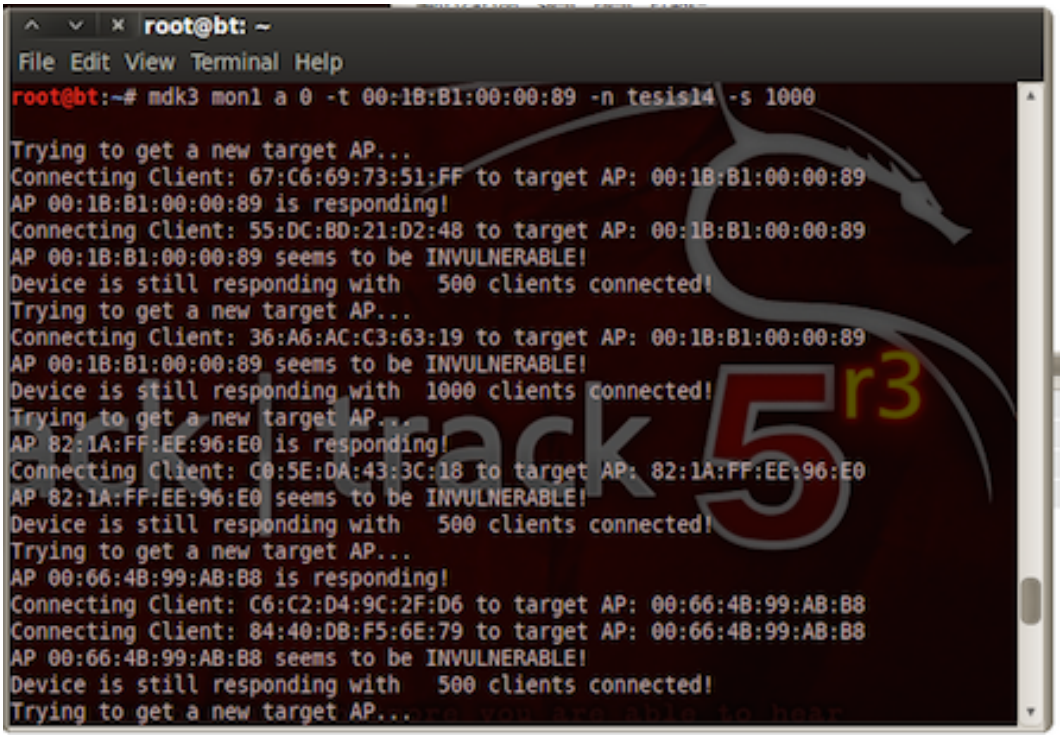


Figura. 138 mdk3 mon1 a

En Wireshark se aprecia la captura de las tramas de clientes con *MAC* aleatorias solicitando un *Authentication Request* al *AP*. Nuevos intentos de autenticación con usuarios legítimos al punto de acceso no fueron exitosos después de este ataque.

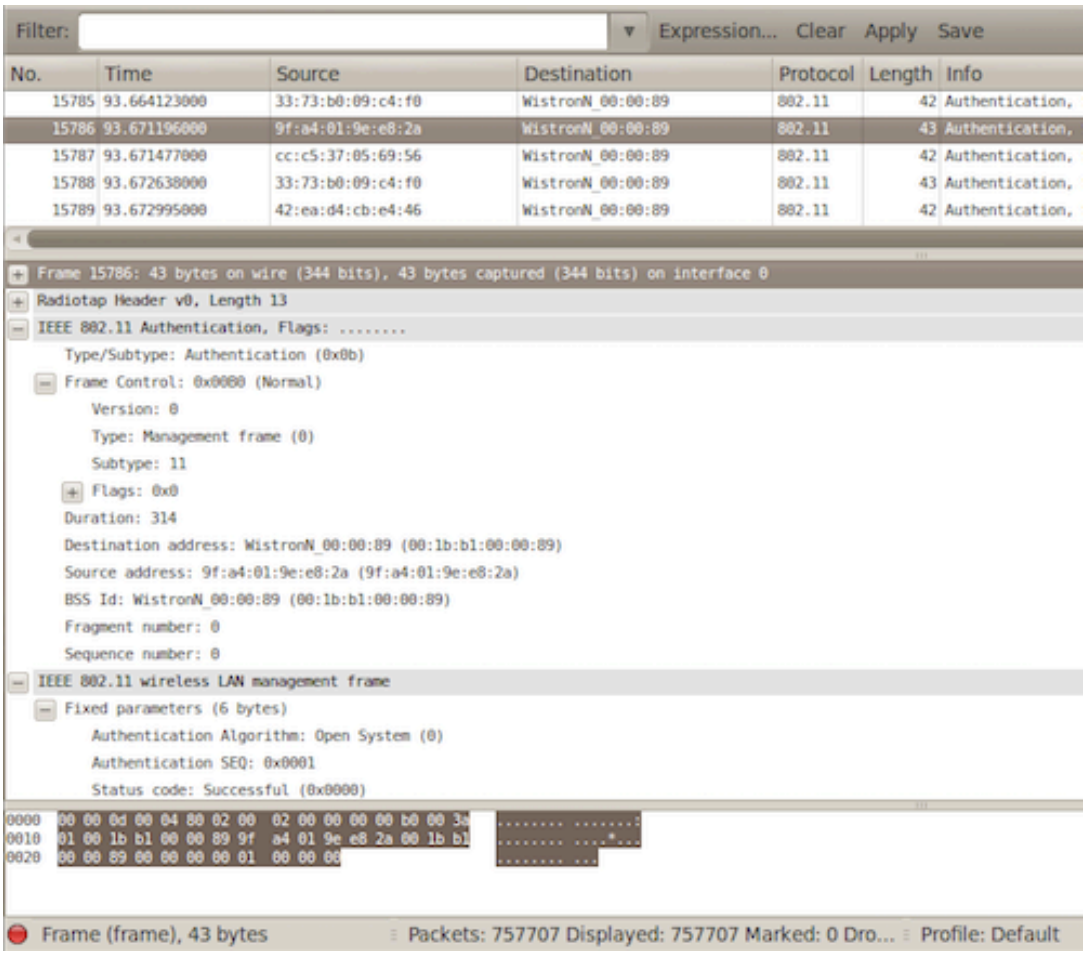


Figura. 139 Wireshark: Inundación Authentication

Los ataques de inundación *EAPoL-Start* y *Authentication* son ejemplos de denegación de servicio contra el autenticador. Además, la inundación de *Authentication* es manipulación de la trama de administración.

4.2.3 Inundación CTS / RTS

El estándar 802.11 establece tramas de Control como *Request to Send* y *Clear to Send* para minimizar la posibilidad de colisiones al existir envió de tramas extensas. En la siguiente figura se puede ver cómo funciona.

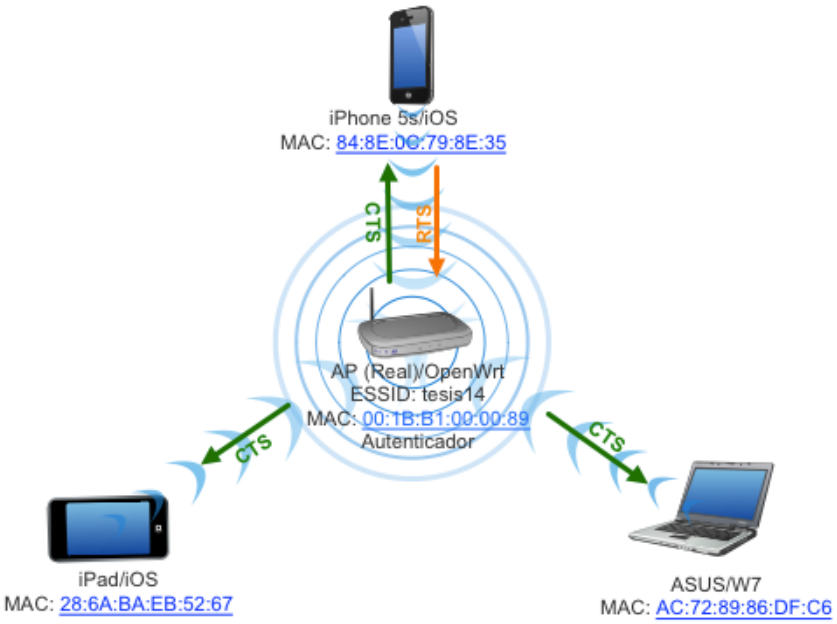


Figura. 140 CTS / RTS

En el caso en que el cliente (iPhone 5s) quiera realizar él envió al *AP* de una trama extensa primero envía una petición de *RTS* para reservar el canal con el fin de evitar así las colisiones. Entonces el punto de acceso responde con una trama *CTS* que reserva el canal por el tiempo que dure el envío. La trama *CTS* a su vez, se enviará a los demás clientes *(iPad y ASUS)* dejándolos saber que solo el *iPhone 5s* puede trasmitir por ese tiempo.

Teniendo este concepto claro, la inundación de *CTS* por parte de un atacante impulsa a otros dispositivos inalámbricos que comparten la red *WiFi* a frenar su trasmisión hasta que el adversario deje de transmitir las tramas *CTS*.

Para realizar este ataque, en BT5 se descarga la herramienta necesaria y se la instala de la siguiente manera:

>>root@bt:~# wget http://matej.sustr.sk/code/framespam/framespam-0.2.tar.gz

>>root@bt:~# tar –xzvf framespam-0.2.tar.gz

>>root@bt:~# cd framespam-0.2

>>root@bt:/framespam-0.2# make && make install

luego seguimos con el ingreso de los comandos:

>>root@bt:~# arimon-ng start wlan0
>>root@bt:~# framespam –i mon0 < file

El “file” es un archivo *.txt (trama CTS.txt)* donde se ingresa la trama *CTS* que se va a enviar. La trama *Clear to Send* es:

Figura. 141 trama CTS

En el archivo *.txt* únicamente la información de la trama: \0304\0\0\0175\01\02\03\04\05\06

Se ignora el *FCS (Frame Check Sequence)* debido a que se calcula por el hardware antes de enviarse.

Figura. 142 framespam –i mon0

Este ataque se ve en Wireshark como se muestra en la siguiente figura:

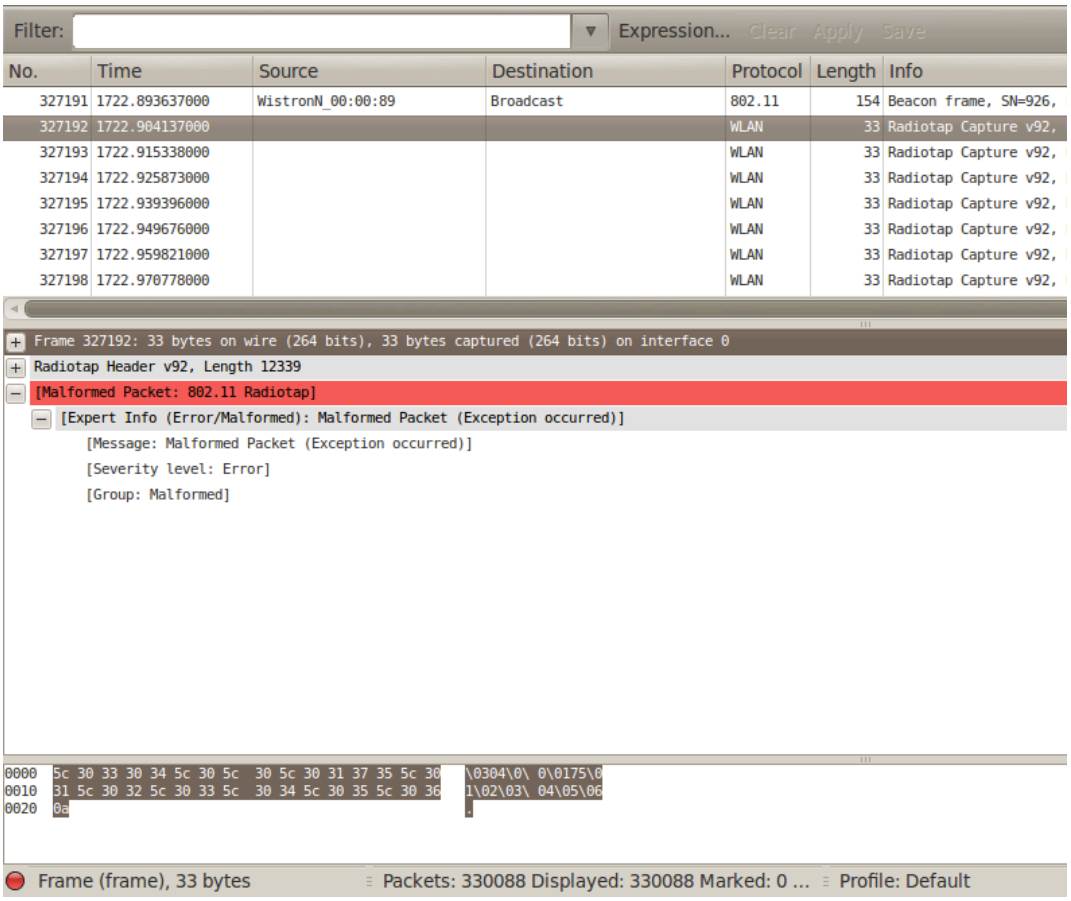


Figura. 143 Wireshark: Inundación CTS

Esta inundación deja la red atacada totalmente sin respuesta, tanto el cliente como el *AP* dejan de trasmitir.

4.2.4 Ataque de interferencia

Un atacante puede manipular la potencia *(Tx-Power)* de radio frecuencia *(RF)* en el espectro de 2.4 GHz o 5 GHz al modificar la ganancia de una antena. Para poder realizar este ataque, seguimos los siguientes pasos:

Se reinicia nuestra máquina virtual *VM2* y no se conecta la tarjeta *AirPecap Nx*. Se escribe el siguiente comando:

>>root@bt:~# tail –f –n 0 /var/log/messages

Ahora se conecta la tarjeta *AirPecap Nx* y se obtiene la siguiente figura que muestra los ajustes regulatorios aplicados en la tarjeta para Ecuador *(EC)*.

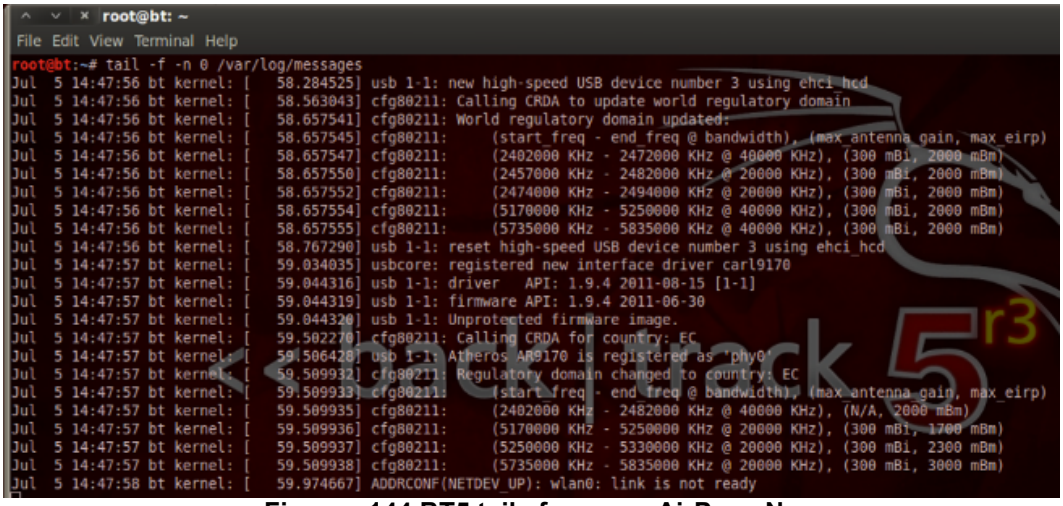


Figura. 144 BT5 tail –f –n para AirPcap Nx

Se pude ver que, para el espectro de 2.4 GHz se permite máximo 20 dBm debido a las regulaciones del país. Sin embargo, este valor se puede modificar para ventaja del atacante. La tarjetas *AirPcap Nx* promociona una potencia de 1 Watt (30 dBm) por lo que se la llevará a ese nivel siguiendo estos pasos:

Se escoge un país donde las regulaciones del país permitan transmitir potencias de 1 Watt. En este caso se escoge Bolivia (BO) para configurar nuestra tarjeta *AirPcap Nx*:

>>root@bt:~# iw reg set BO

Como se evidencia en la siguiente figura, se ha cambiado los niveles de potencia máximo para el espectro de 5 GHz a 30 dBm.

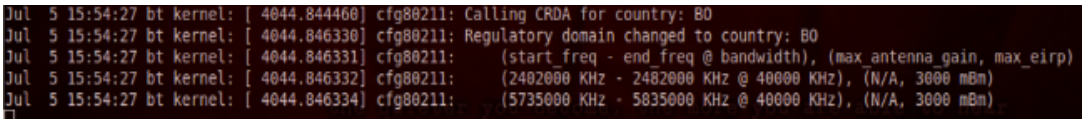


Figura. 145 Cambio de niveles máximos de potencia a 30 dBm

Para establecer el nivel de potencia *(Tx-Power)* a 30 dbm, escribimos el siguiente comando:

>>*root@bt:~#* iwconfig wlan0 txpower 30

Al implementar un *AP* falso, el mejorar los niveles de potencia nos da una ventaja sobre el *AP* legítimo, esto por cuanto se amplía el área de cobertura y atrae la conexión o asociación de clientes legítimos con el *AP (Falso).*

Se inicia el *AP* falso como se vio en capítulos anteriores al iniciar el *hostapd* en nuestra máquina virtual *VM2.*

>>*root@bt:~# hostapd hostapd-2.0/hostapd/hostapd.conf*

El archivo *hostapd.conf* ya está pre configurado como esta detallado en capítulos anteriores. Este archivo contiene toda la información que se obtuvo de esnifar la red para simular ser el *AP* real, es decir, mismo *ESSID* (tesis14), etc. Como el *AP* falso tiene mejores niveles de potencia. Esto llevará a los clientes o suplicantes a conectarse con este equipo, logrando así un ataque exitoso de *DoS*.

En la siguiente figura se evidencia los clientes legítimos y *AP (Real)* conectados.

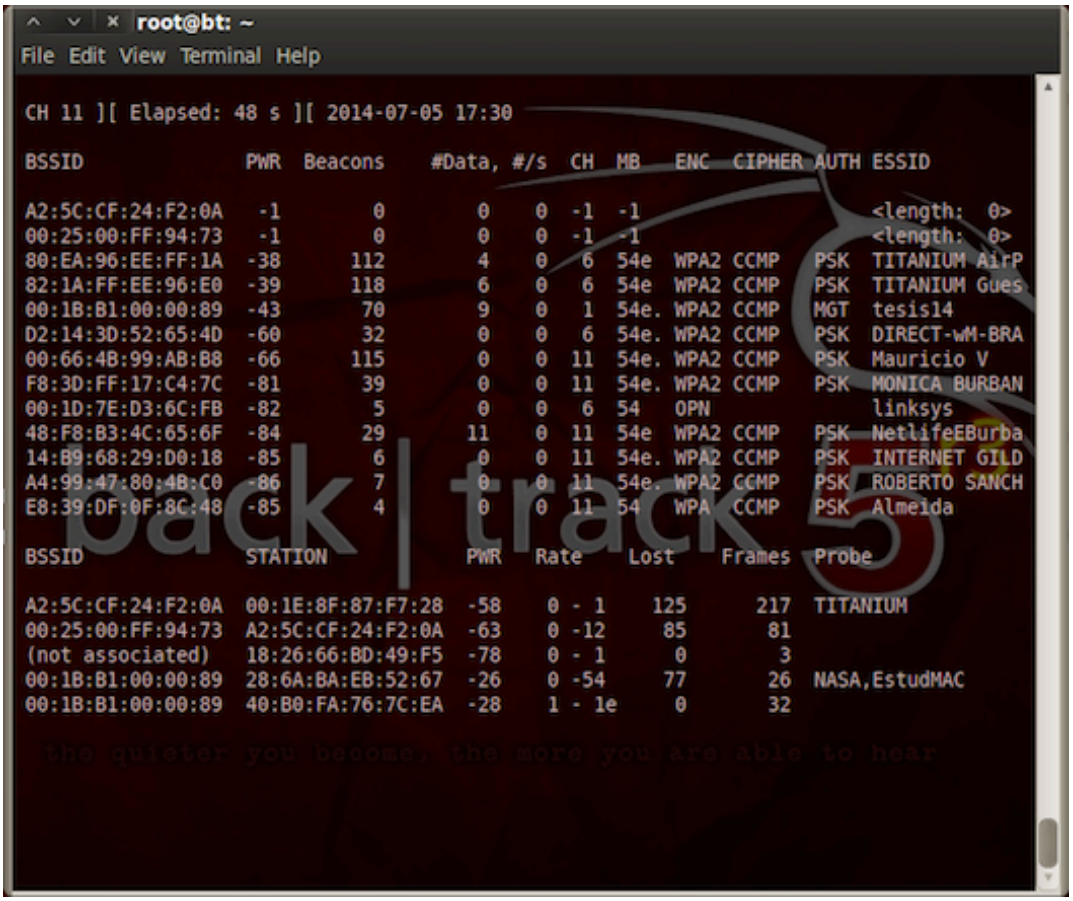


Figura. 146 AP (Real) y clientes legítimos asociados

Ahora se muestra en la figura los clientes legítimos que se asocian en el *AP (falso)* debido al ataque de interferencia.

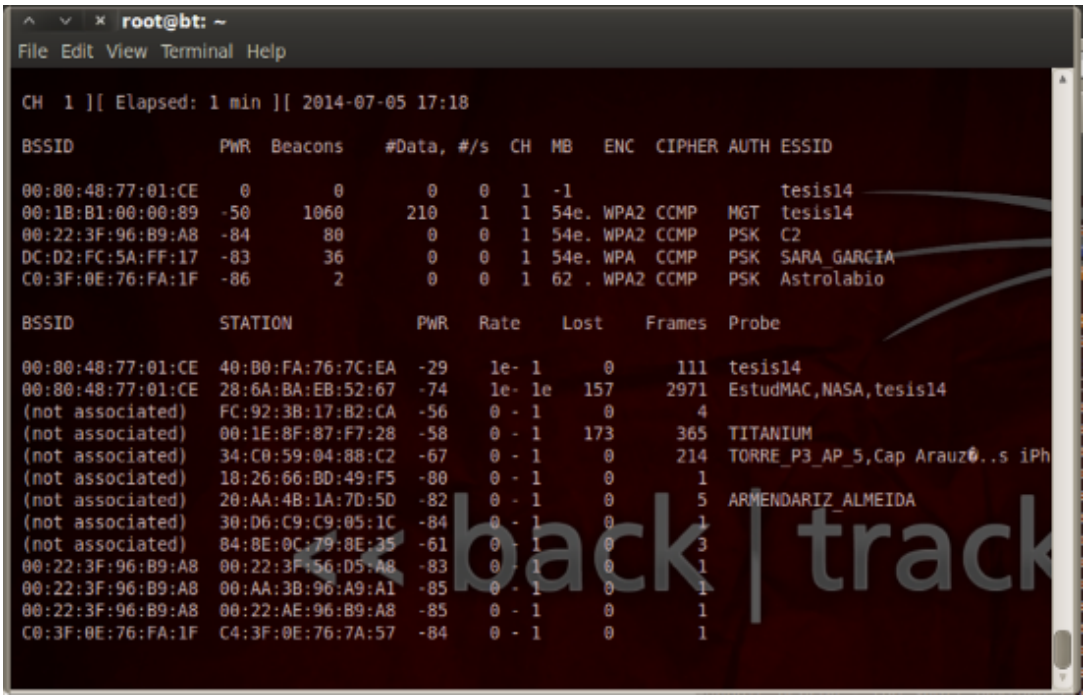


Figura. 147 AP (Falso) y clientes legítimos asociados

Los ataques de Inundación *CTS / RTS* e interferencia son ejemplos de denegación de servicio contra la infraestructura. Además, la inundación de *CTS / RTS* es manipulación de la trama de control.

4.2.5 Inundación de Desautenticación (Deauthentication)

Este ataque tiene como objetivo desvincular al cliente del punto de acceso. Este ataque se vulnera la red con los siguientes comandos:

Para crear un *DoS* generalizado que des autentique todos los usuarios del AP se pone:

>>root@bt:~# aireply-ng -0 5 –a (BSSID) mon0

Esto se muestra en la siguiente figura:

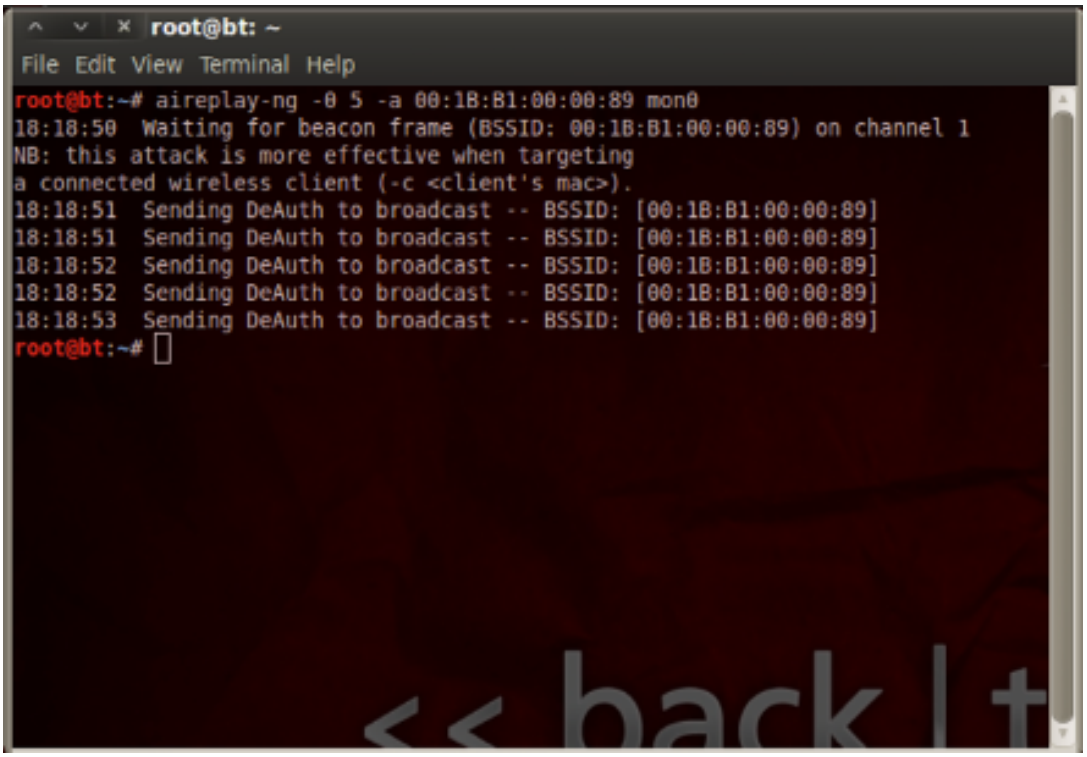


Figura. 148 Deauthentication todos los usuarios

En Wireshark *se* puede ver las tramas suplantadas con la dirección *MAC* del *Access Point* para des autenticar todos los clientes.

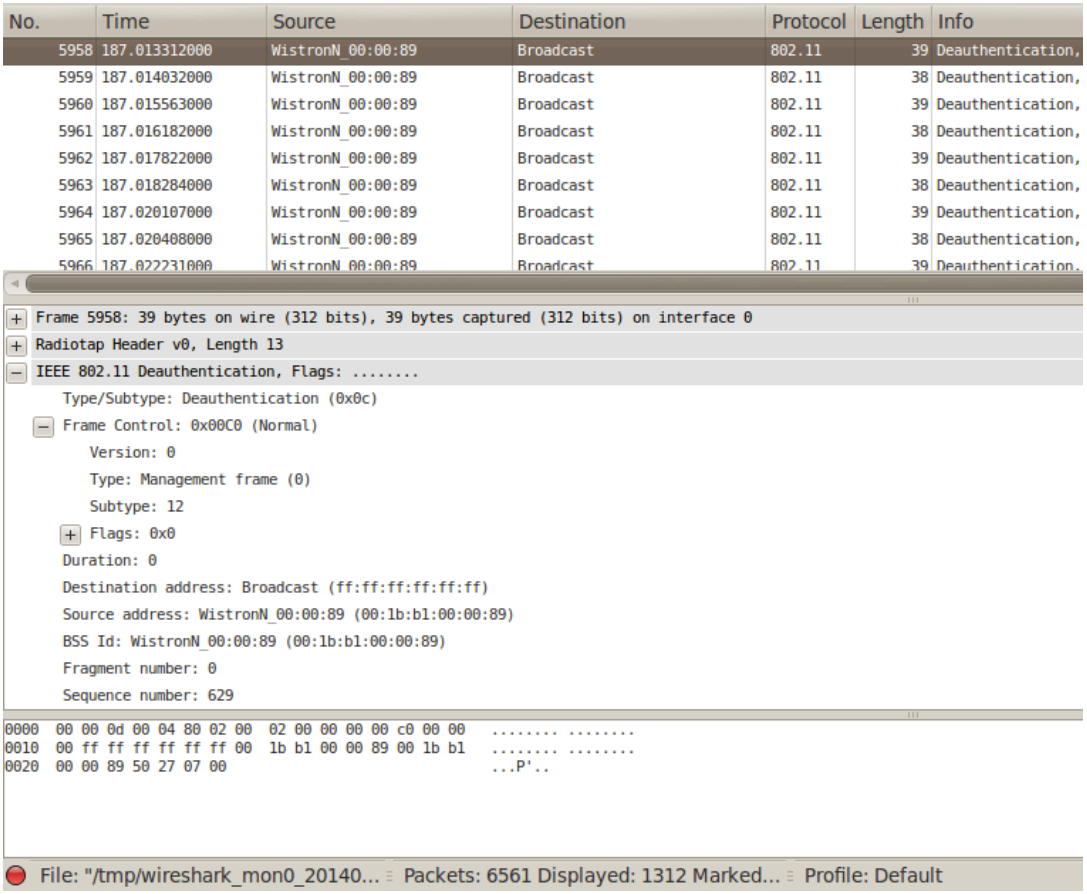


Figura. 149 Wireshark: Inundación Deauthentication

En el caso en que se pretenda desautenticar un solo cliente del *AP* con el fin de realizar un ataque *DoS* se hace uso del siguiente comando:

>>root@bt:~# aireply-ng -0 5 –a (BSSID) –c (MAC STATON) mon0

En la siguiente figura se puede ver la ejecución de este comando.

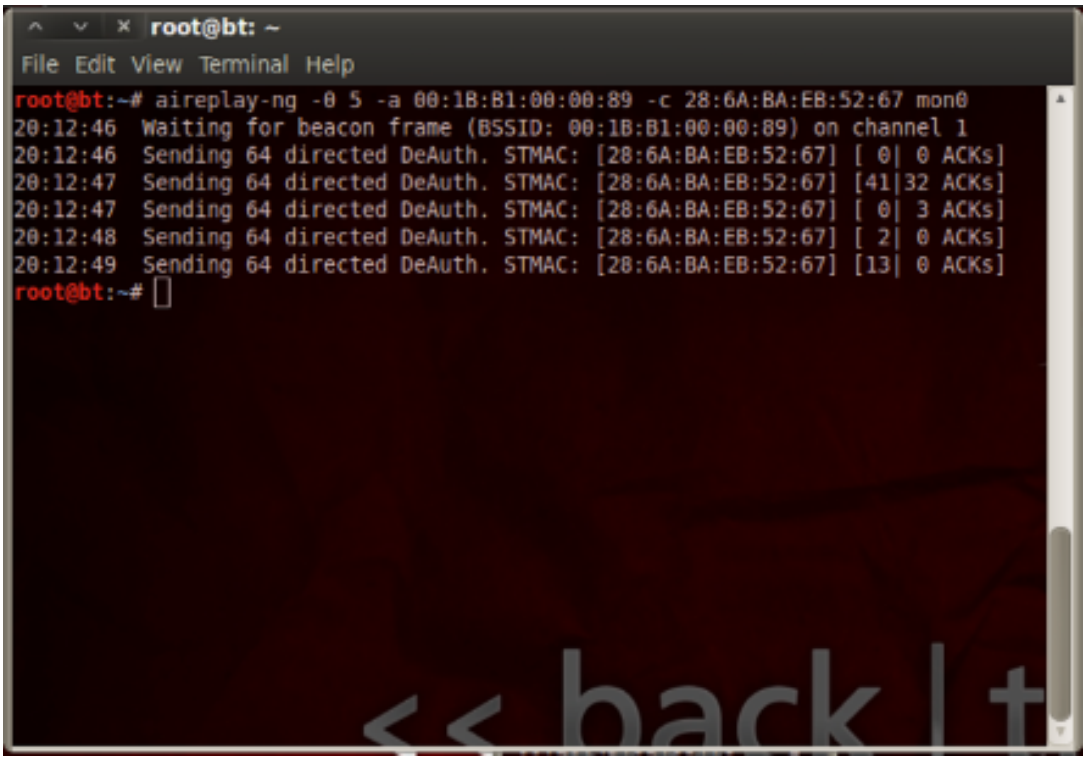


Figura. 150 Deauthentication usuario especifico

A continuación se puede visualizar la des autenticación de un único usuario en Wireshark*:*

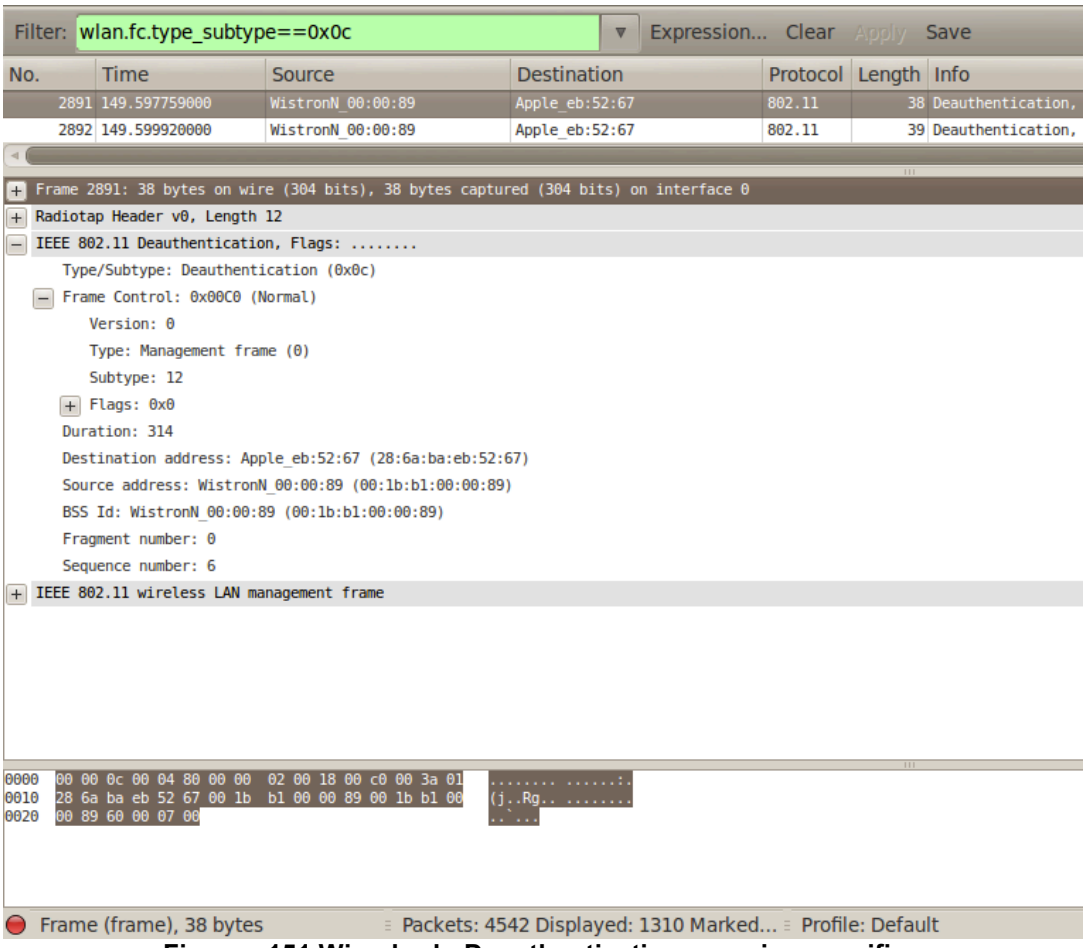


Figura. 151 Wireshark: Deauthentication usuario especifico

El ataques de des autenticación es un ejemplo de denegación de servicio contra el cliente manipulando la trama de administración.

4.3 Amenaza 3: Implementación incorrecta del método EAP-TLS

Como se ha explicado anteriormente, el estándar *802.11i* con *EAP-TLS* es uno de los protocolos más seguros en la actualidad. Sin embargo, este estándar puede ser vulnerado por una mala práctica por parte del usuario o el administrador de red. Esta fragilidad se da cuando el usuario desconoce el modo adecuado para autenticar sus equipos terminales en la red. En el caso del administrador, el principal factor que pone en riesgo a la seguridad de la red es debido a que para establecer el túnel *EAP-TLS* se debe instalar los certificados en cada uno de los equipos terminales y el administrador puede obviar este paso por comodidad o falta de recursos.

Al realizar el análisis del escenario de prueba normal, se evidencia la existencia de un mensaje *(EAP Request Start PEAP)* que puede ser protagonista de una vulnerabilidad si no se implementa de forma correcta el túnel *EAP-TLS*. Este mensaje negocia el tipo de túnel *EAP* que se va a establecer, por defecto se establece *PEAP* si no existe otro método.

Como se ha comparado, el *PEAP* es un método menos seguro que *TLS*, esto por cuanto solo necesita tener certificado instalados del lado del servidor más no del lado del cliente para establecer el túnel a lo contrario de *TLS*, el cual requiere que se instalen los certificados en ambos lados. Es por esto que *PEAP* resulta ser vulnerable a ataques de fuerza bruta con diccionario para la obtención de las credenciales (usuario y contraseña) del usuario.

Al tener implementada una red 802.11i con *EAP-TLS*, existe la posibilidad de que un usuario intente conectar sus equipos inalámbricos. En este punto, el usuario no tendría instalado el certificado e intentará conectarse a la red, se establecerá una negociación *EAP* y resultará en un túnel *PEAP*.

En *PEAP*, el usuario selecciona la red inalámbrica a la que desea conectarse (tesis14) e ingresa sus credenciales (usuario y contraseña). Seguido, el servidor envía su certificado y si el usuario acepta este certificado y se establece la conexión.

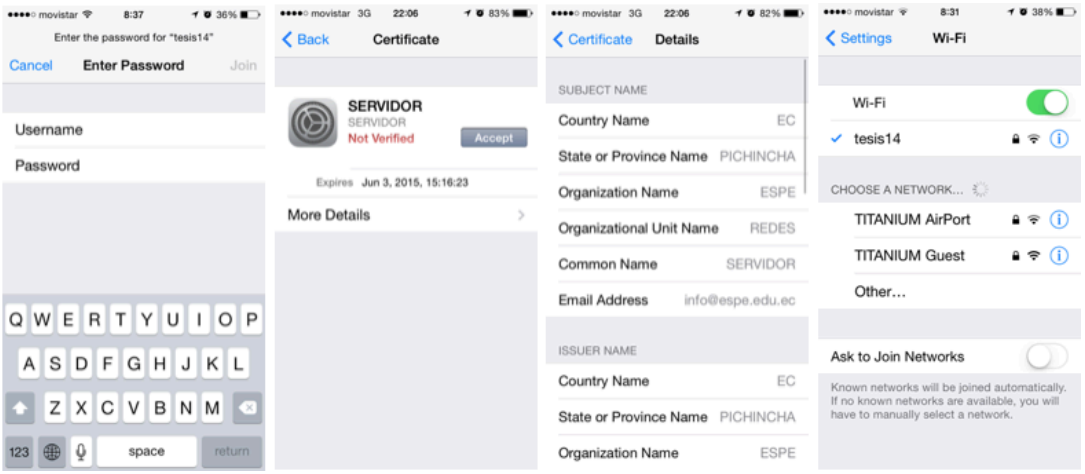


Figura. 152 Establecimiento de conexión PEAP

Ahora se comprobará como el atacante puede realizar el *hackeo* por medio de fuerza bruta para conseguir las claves del usuario. A pesar de que es una red definida para funcionar con *EAP-TLS* y ser segura, al no establecerse normas y políticas de buenas prácticas de seguridad dentro de una empresa esta puede ser vulnerada. Como se ha analizado, si el certificado no es instalado del lado del usuario para establecer el túnel *TLS*, este por defecto establece *PEAP*. Entonces a continuación se muestra el ataque para este método.

El atacante inicializa el servidor *FreeRADIUS – WPE* que se explicó e implementó en el módulo de BT5 – *Hacker – VM2*. Se ingresa el comando:

>>root@bt:~# radiusd –X

A continuación, se ejecuta *hostapd* para inicializar el *AP (Falso)* que contiene las mismas características que el punto de acceso real.

>>root@bt:~# hostapd hostapd-2.0/hostapd/hostapd.conf

Ahora, se genera un ataque de des autenticación para cancelar la conexión entre el *AP (Real)* y cliente ingresando:

>>root@bt:~# aireplay-ng -0 5 –a 00:1B:B1:00:00:89 –c 28:6A:BA:EB:52:67 mon0

Esto con el fin de que el cliente intente conectarse con el *AP (Falso)*, ahora el servidor *FreeRADIUS – WPE* enviará un certificado falso al usuario, el usuario al no ser capacitado sobre esta vulnerabilidad aceptará dicho certificado que, como se comprobará en la siguiente figura es completamente diferente al certificado original.

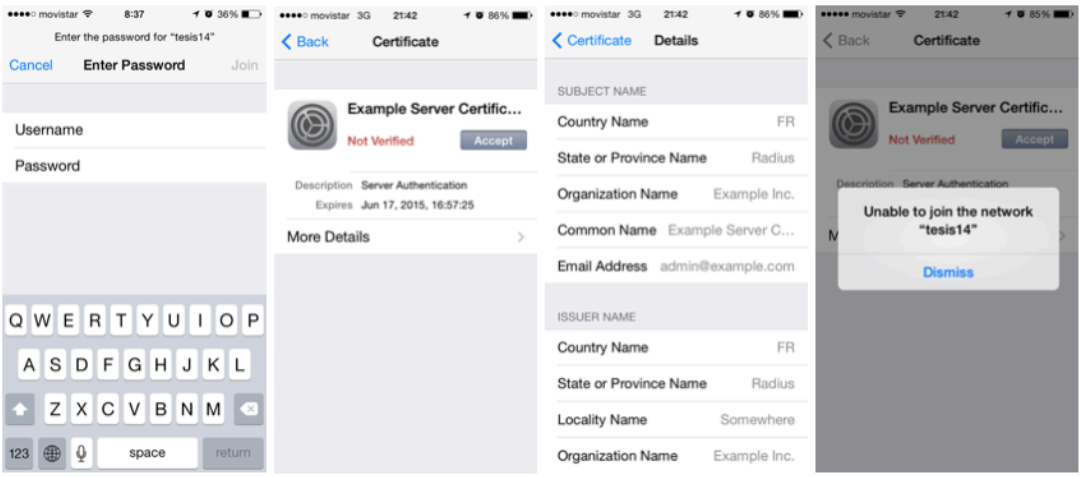


Figura. 153 Envío de certificado falso

Lógicamente, la conexión será rechazada debido a que el servidor *FreeRADIUS – WPE* no puede autenticar el usuario con su tabla de usuarios. Lo que nos interesa es que el usuario acepte este certificado debido a que con esto se crea un *log (freeradius-server-wpe.log)* que contiene información sobre el *challenge* y *response* de este intento de establecer conexión, en la figura se muestra este proceso:

>>root@bt:~# tail /usr/local/var/log/radius/freeradius-server.log

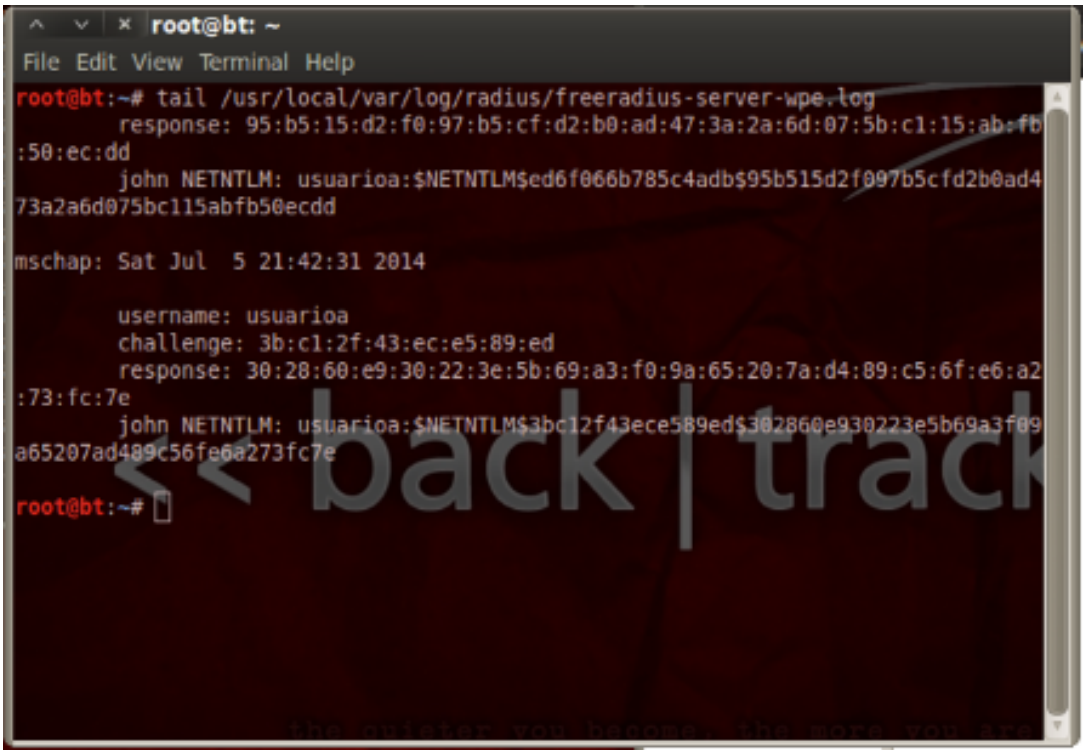


Figura. 154 Ejecución de tail para obtener el challenge y response

Con esta información, se realizará un ataque de fuerza bruta con diccionario para obtener la clave del usuario. Esto se lo realiza con el siguiente comando:

>>root@bt:~# asleap –C (challenge) –R (response) –W (Dirección del diccionario)

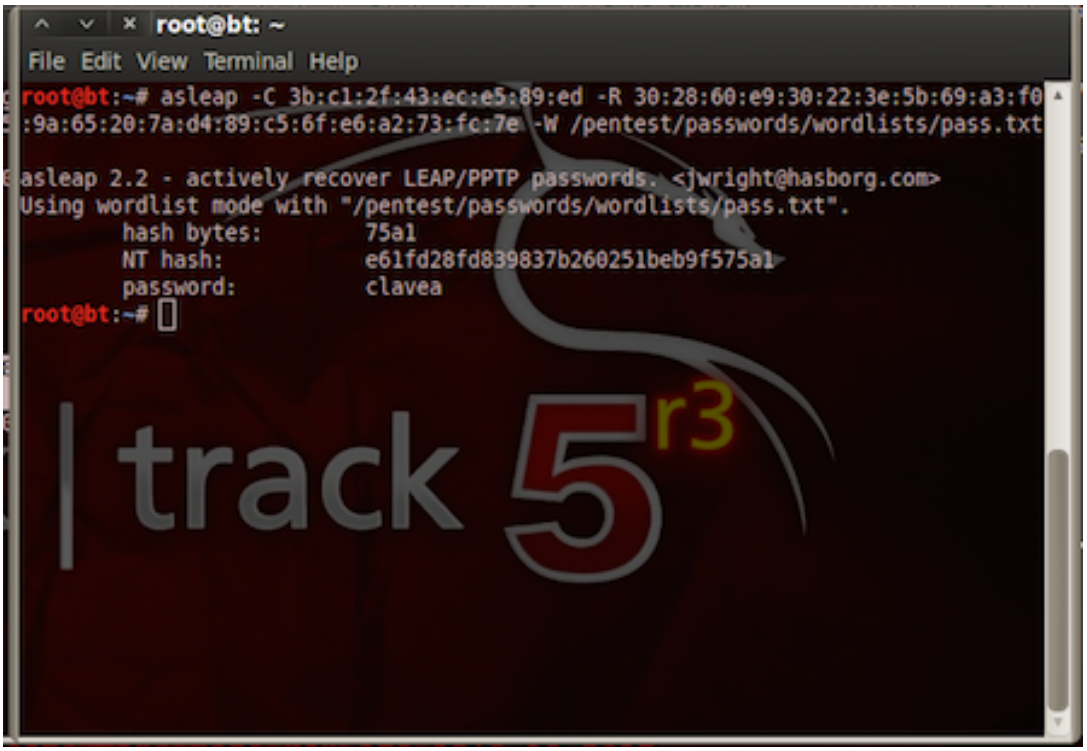


Figura. 155 Ejecución asleap

Finalmente, se obtiene la información que se deseaba del usuario. Con el comando *tail* se obtiene el nombre del usuario *(usuarioa)* y al ejecutar el comando *asleap* se conoce la clave del usuario *(clavea)*.

CAPÍTULO V

ANÁLISIS

La detección de vulnerabilidades en redes inalámbricas *802.11i*, mediante el análisis de tráfico de la capa de enlace comienza estableciendo un escenario de prueba. El diseño del escenario contempla la implementación del protocolo *EAP-TLS* para definir el mecanismo de autenticación para los dispositivos que requieran conectarse a la *WLAN,* considerado el más seguro desarrollado para protección de este tipo de redes.

El mecanismo de autenticación *802.1X* involucra tres elementos: el suplicante, autenticador y el servidor de autenticación. El suplicante es un cliente con un dispositivo inalámbrico que requiere conectarse a la red inalámbrica. El autenticador es un dispositivo de red que en nuestro caso es un AP que actúa como punto de acceso a una red protegida, este impide el acceso del suplicante hasta que su identidad sea validada y autorizada por medio del envió de credenciales (usuario y contraseña) o certificados digitales que serán reenviadas al servidor para su verificación. El servidor de autenticación en un *host* que utiliza *RADIUS* y los protocolos *EAP*, este determina si las credenciales son válidas y permite al cliente, en caso de ser validado, acceder a recursos dentro del lado de la red protegida.

Establecido el escenario de prueba, se realiza la captura de los paquetes por medio de un *sniffer* llamado Kismet para el análisis de las tramas de Control y Administración en un ambiente normal de funcionamiento. Luego, utilizando el adaptador *AirPcap Nx* se realiza la inyección de trafico de tramas modificadas de Administración o Control para realizar ataques de

denegación de servicio *(DoS).* Cabe señalar que el análisis se basa sobre este tipo de tramas ya que al momento de realizar la captura de tráfico se evidenció que son las más propensas a ser violentadas por un ataque, ya que son enviadas por el medio no guiado sin encriptación y son transmitidas por difusión a todas las estaciones, donde un usuario con las herramientas suficientes, puede obtener y extraer información valiosa acerca del punto de acceso y los parámetros de la red. Al igual que las tramas de administración las tramas de control se envían sin ningún tipo de encriptación y viajan en las cabeceras de las tramas a diferencia de las tramas de control cuya captura u obtención presentan un grado de complicación mayor, ya que son las encargadas del control de acceso al medio y se directamente entre envían entre el *AP* y el equipo suplicante para reservar el canal de comunicación, existen varios tipos de tramas de control. Las dos tramas Administración y Control serán analizadas a continuación.

Como se puede observar en capítulos anteriores existen varios tipos de tramas y subtipos de cada una de ellas, que pueden ser identificadas mediante el valor que este dentro de los campos en la cabecera que se encuentra dentro de todas las tramas que se envían a través de la capa de enlace. En la siguiente figura se puede observar cómo se estructura la trama de la capa

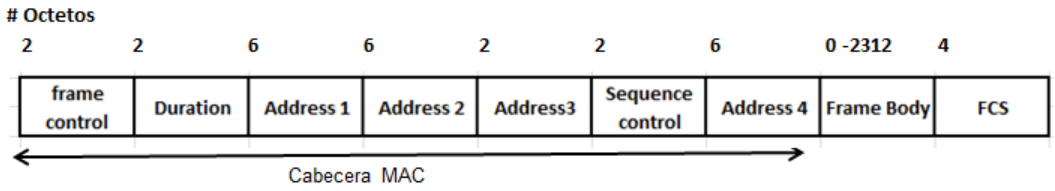


Figura. 156 Estructura trama MAC

Como se explicó, los campos de interés son aquellos dentro de la cabecera de la trama, ya que son aquellos que contiene información relevante sobre la conexión. A continuación en la imagen se puede observar los tramos descritos anteriormente, dentro del campo *Frame control.*

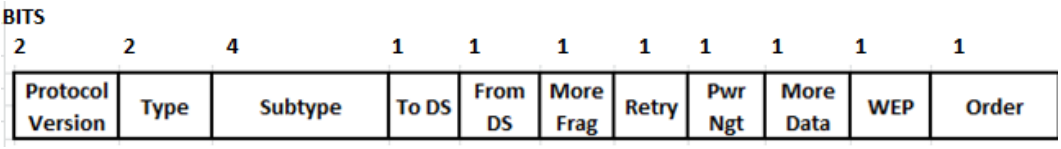


Figura. 157 Estructura trama de Control

Los bits que se encuentran en los campos de *Type*, definirán si la trama es de Administración, control, o datos, de igual manera los bits dentro del campo *Subtype,* indica el tipo de subtipo de trama es decir en el caso de una trama de Administración los bits del primer campo serán 00*,* y un valor de 08 en el siguiente campo indicara que se trata de una trama *Beacon*, este información es de suma importancia, considerando que conocer los valores que toman los campos de la cabera de la trama, permitirá definir el tipo de trama y a su vez la información que cada una contiene, ya que como fue descrito en capítulos anteriores cada trama contiene valores o datos importantes acerca de las redes inalámbricas.

Adicional al momento de realizar un análisis y estudio de tramas, se tiene que tomar en cuenta los parámetros *ToDS* y *FromDS*, los mismos que determinan que equipos están involucrados en el intercambio de información, en la tabla que se muestra a continuación se puede aprecia los valores que toman cada uno de los campos y como las combinaciones definen que tipo de equipos están involucrados en el intercambio de información.

Tabla. 5

Parámetros DS

To DS	From DS	Addr 1	Addr 2	Addr 3	Addr 4
0	0	DA	SA	BSSIS	N/A
1	0	DA	BSSID	SA	N/A
0	1	BSSID	SA	DA	N/A
1	1	RA	TA	DA	SA

- *DS: Distribution system.*
- *DA: Destination address.*

- *SA: Source address.*
- *TA: Transmitter address.*
- *RA: Receiver address.*
- *BSSID: Basic service set identifier.*

Un campo tomado en cuenta para el análisis fue el de *Power Management,* es un campo de un solo bit que indica el estado de energía de la estación posterior a completar el intercambio de tramas. De igual forma el campo *WEP* determina si el cuerpo de la trama tiene encriptación *WEP*.

En las siguientes imágenes se puede apreciar un ejemplo de una trama de administración *(Beacon)*, y control (*RTS*), junto con los valores que toman los campos que componen la cabecera.

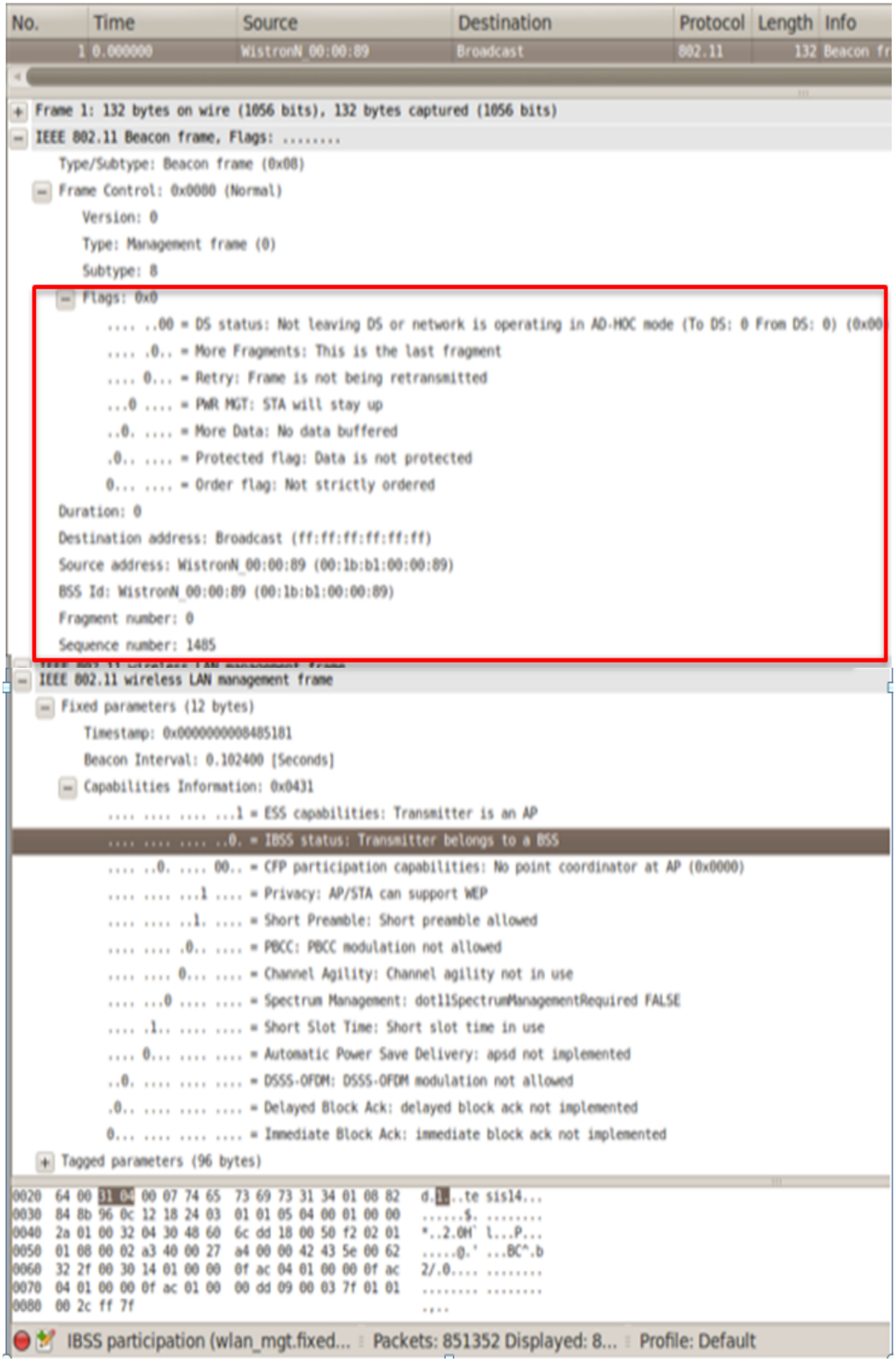


Figura. 158 BEACON FRAME

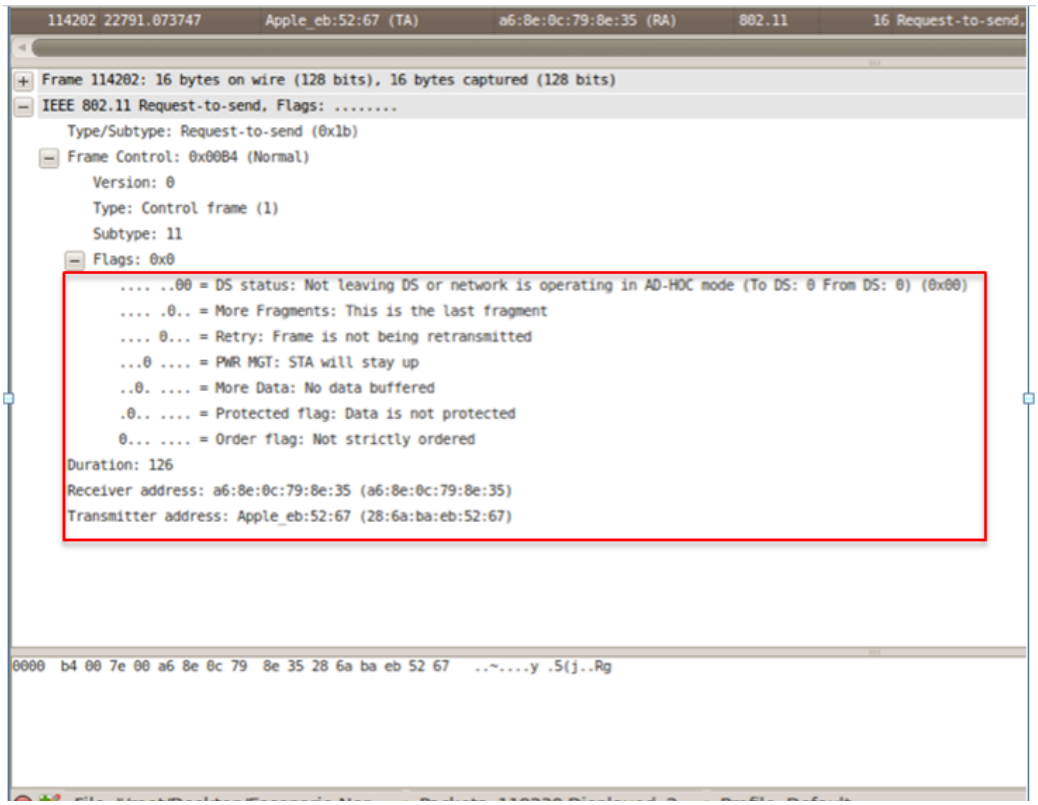


Figura. 159 RTS FRAME

Como se puede observar las dos imágenes presentan diferencias notables en los parámetros que contienen, siendo la trama de administración la que proporciona mayor información acerca de la red, de igual manera se puede apreciar la similitud en los parámetros de *Frame Control* que fueron descritos.

Previo a realizar ataques al estándar se debe conocer el funcionamiento del mismo, es decir identificar las etapas, los mensajes, la información que se intercambia en el proceso por el cual un usuario es autenticado, es así que se establece tres pasos fundamentales para la autenticación, estos son: inicialización, Iniciación - Negociación y autenticación.

Para el primer paso de inicialización el *AP* bloquea los puertos para el intercambio de información, permitiendo únicamente el intercambio de trafico *802.1X*. en esta etapa se puede destacar el intercambio de mensajes y tramas de administración descritas en páginas anteriores, dentro de estas

tramas podemos rescatar información sensible sobre los equipos como: las direcciones *MAC* de los clientes, direcciones *MAC* de *AP's* y sus *ESSID*. Además podemos aprender sobre las asociaciones entre los distintos equipos, sus niveles de potencia de transmisión, el canal de trasmisión, el tipo de encriptación y el protocolo de autenticación. Esta información se puede identificar son facilidad ya que como fue explicado las tramas son enviadas no están encriptadas o protegidas.

En la siguiente etapa de iniciación, el suplicante inicia o reinicia la autenticación al enviar tramas *EAPOL-Start* al autenticador, a partir de este mensaje empieza un continuo intercambio de mensajes *EAP* entre el suplicante y el servidor utilizando como distribuidor al autenticador que a este punto a permitido el intercambio de nuevas tramas la mayoría de ellas basadas en mensajes *EAP*, como se ha explicado en capítulos anteriores, en esta etapa se realiza el proceso de negociación del túnel seguro *EAP-TLS* y generación de llaves entre el suplicante y el autenticador que establecen una conexión segura para continuar con el proceso de *hand-shake.* De todas las tramas y mensajes que se intercambian en esta etapa se puede obtener información relevante de conexión como son la identidad del usuario, la información que se encuentra dentro del certificado, considerando que el resto de información viaja por el medio encriptada. Este es lo que destaca y define como al estándar *802.11i* con autenticación *EAP-TLS* como el sistema más seguro para una red inalámbrica. Pese a eliminar la mayoría de ataques conocidos como, *MITM*, *AP* MALICIOSO, continua siendo vulnerable a la más grande amenaza que representa el usuario o la negligencia del administrador de red que al no realizar una configuración e instalación correcta y responsable de los certificados, creará un espacio para generar ataques que facilitarán al usuario malicioso la obtención de nuevos parámetros como la contraseña que junto con la identidad permitirá establecer la conexión y aprovechar los recursos de la red.

La última etapa es la autenticación, en donde el suplicante y el autenticador ya poseen un *PMK (Pairwise Master Key)* idéntico, que permitirá generar llaves de emparejamiento por medio el intercambio de mensaje en el proceso de *hand-shake* de 4 vías, cabe señalar que ya en esta etapa no se puede realizar ataques, ni capturar información relevante ya que todo lo que se intercambia viaja encriptado. Finalizada la autenticación, de ser exitosa, se desbloquean los puertos, el tráfico empieza a fluir con normalidad y se tiene acceso a todos los recursos de al red. En caso de una disociación de un cliente, los puertos vuelven a un estado de bloqueo de todo el tráfico que no sea *EAP.*

Habiendo analizado el funcionamiento de la red, y el proceso que realiza cada usuario que desee conectarse, de igual forma en capítulos anteriores se presentaron los mensajes que se intercambian en todo el transcurso.

Ahora se podría decir que el sistema implementado en el escenario de pruebas es completamente seguro ya que elimina una cantidad de ataques conocidos, pero por fallas inherentes del protocolo *802.11*, como la falta de encriptación en tramas de Administración y Control, considerando la cantidad de información que contienen estas tramas, además la falta de control a los usuarios siendo ellos los que representan el punto débil de la seguridad en una red. Bajo esta premisa los ataques a los que el estándar se encuentra expuesto son los de Denegación de servicio *(DoS)*, que son ataques que afectan a la disponibilidad de la red, existen tres tipos de ataques: al cliente, a la infraestructura, y al *AP*, los mismos que ya fueron expuestos en capítulos anteriores,

En la búsqueda de las vulnerabilidades, se hace uso de herramientas como BT5. Este software permite realizar la manipulación de las tramas de administración como por ejemplo *Authentication* o *Deauthentication Requests*.

Lo que se hace es modificar los parámetros de las direcciones *MAC* para apuntar hacia equipos que se desee atacar. Una vez realizado el procedimiento, se produce la denegación de servicios *(DoS)* al inundar la red *WLAN* con estas nuevas tramas que no han sido generadas en un funcionamiento normal de la red sino más bien por un usuario malicioso ajeno a la red. Esto genera confusión en la red y provoca disociaciones de los usuarios con los *AP*. Los ataques no están limitados únicamente a la manipulación de las tramas de Administración, sino que también se lo hace con las tramas de Control, por ejemplo tenemos ataques de inundación de tramas *CTS/RTS,* en el que se inunda con tramas de este tipo que afecta al control de acceso al medio de los usuarios, restringiendo la transmisión de tramas de los mismo ya que mantiene reservado el canal para una dirección *MAC* falsa generada con el ataque.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La implementación del escenario de prueba permite conocer en profundidad todos los componentes que conforman el estándar *802.11i*. Esto crea un conjunto de condiciones reales para analizar y determinar las vulnerabilidades que se presenten en el estudio de la red inalámbrica.

- Kismet es un analizador de tráfico de red que captura paquetes por medio de una tarjeta *wireless* compatible. A diferencia de los adaptadores AirPcap, Kismet no puede inyectar tráfico en la red inalámbrica.

- El protocolo de autenticación *EAP-TLS* crea un túnel seguro entre el servidor de autenticación y el suplicante, esto por cuanto se requiere la instalación de certificados validos en ambos partes para establecer confianza con la autoridad de certificación *(CA)*. Esto garantiza la inefectividad de ataques como *MITM (men in the middle)* o enmascaramiento de *AP* maliciosos para la obtención de las credenciales por medio de fuerza bruta utilizando ataques con diccionarios.

- Por las grandes ventajas que ofrecen las redes inalámbricas, se ha evidenciado un crecimiento en la implementación dentro del campo empresarial, generando un mayor volumen de información en el espectro radioeléctrico, medio en el que se encuentra expuesta y vulnerable a ataques donde se puede violentar su integridad y confidencialidad.

- Existen varios protocolos de protección para una red inalámbrica, los mismo que han ido evolucionando, considerando la necesidad de aumentar la seguridad en la red, empezando desde *WEP* con una encriptación de los datos básica, hasta *WPA2 Enterprise*, siendo el más robusto por su característica principal que se basa en una red de infraestructura para realizar la validación y autenticación de clientes para el acceso a la red, dispone de varios métodos de autenticación *EAP*, cada uno con características propias y ventajas para la validación de usuarios.

- El análisis de tráfico sobre la capa de enlace del escenario de pruebas, en el que se ha implementado el protocolo *802.11i* con autenticación *EAP-TLS*, ha permitido conocer el funcionamiento, y comportamiento, que desarrollan el protocolo y el método de autenticación para el control de acceso de usuarios a la red inalámbrica, es decir ha permitido reconocer el comportamiento de la red en estado normal reconociendo las tramas de administración, control y datos, que se envían las cuales contienen información relevante de la red.

- El estándar *802.11i* con autenticación *EAP–TLS* es considerado el sistema más seguro en contra de intrusiones o ataques, ya que por sus características elimina ataques conocidos como *MITM (men in the middle),* enmascaramiento de *AP* maliciosos, ataques de diccionario, pero no es un sistema completamente seguro ya que ese encuentra expuesto a ataques de *DoS* (denegación de Servicios) por vulnerabilidades inherentes en estándar como por ejemplo, las tramas de la capa de enlace, y puntualmente en las tramas de administración y control las mismas que son enviadas a las estaciones sin ningún tipo de encriptación, permitiendo conocer información valiosa de la red, que puede ser utilizada por un usuario malicioso para generar ataques.

- Las tramas de administración son las que poseen mayor cantidad de información de la red y las más numerosas en comparación a las de control,

es decir contienen parámetros como *ESSID*, direcciones *MAC*, potencias de transmisión, y son enviadas en difusión a todas las estaciones que se encuentren dentro del alcance de la misma, presentando uno de los puntos de vulnerabilidad del estándar *802.11i*, ya que este puede ocasionar la realización de ataques de disociación y desautenicación de usuarios para posterior al realizar un ataque de *AP* malicioso, que ocasionará que los usuarios se conecten al *AP* falso y enviarán información importante como credenciales, y datos de autenticación, utilizadas por le atacante para conectarse a la red.

- Las tramas de control con las encargadas de controlar el acceso al medio en donde, dentro el protocolo *CSMA/CA* sincronizan el envió de tramas de las estaciones hacia el *AP*, este procedimiento puede ser violentado ya que al reconocer las tramas y las direcciones *MAC* de los equipos se pueden generar ataques de Dos, inundado con tramas *RTS/CTS* reservando el canal de trasmisión esto produce que ningún usuario tengo acceso al mismo que se encuentra reservado por tramas enviadas por el atacante, para dejar sin servicio el *AP*.

- El método de *EAP-TLS* presenta una vulnerabilidad en el caso en que no se instale los certificados tanto en el servidor como en el cliente. Esto lo se detectó en dispositivos iOS, sucede cuando el cliente no tiene instalado el certificado y se requiere establecer la conexión por medio del protocolo de autenticación *EAP*. En la negociación del túnel en el paso 13 de la Figura. 108, se realiza la negociación y el equipo al no contar con los certificados instalados opta por el método *PEAP*, el mismo que es vulnerable a ataques de *MITM*, enmascaramiento de *AP* malicioso y ataques de fuerza bruta con diccionario, que permitirá obtener las credenciales del usuario, y posterior el ingreso a la red.

5.2 Recomendaciones

- Todo el análisis de vulnerabilidades de seguridad en la capa de enlace realizado en este documento para el estándar *802.11i*, abarca las tramas de Administración y Control, por tanto puede ser utilizado como sustento en el desarrollo de futuros estudios o en la creación de software para la detección de intrusiones en la red.

- En el estudio del escenario de prueba, se detecta como amenaza potencial de la red inalámbrica al usuario, por lo que es de suma importancia que en un ambiente corporativo, este sea capacitado sobre las vulnerabilidades en el protocolo de autenticación *EAP-TLS* junto con normas de seguridad, ya que si este no es establecido de forma adecuada puede ser propenso a un ataque.

- Se debe analizar el método de autenticación a implementar en una red inalámbrica, es decir considerar los paramentos como número de usuarios, escalabilidad de la red, se necesitará o no un administrador de red, para poder plantear una solución de seguridad eficiente, ya que de tomar una decisión incorrecta, se podrá causar vulnerabilidades en la red, molestias a los usuarios, pérdida de control sobre los usuarios a integrar a la red.

- A medida de lo posible se debe implementar un sistema integral, para el control de la seguridad, ataques, análisis de la red, es decir concentrar en un mismo equipo la mayor cantidad de funciones, como ejemplo el escenario planteado en este estudio, donde, el *AP* ofrece la función de autenticador, y capturar el tráfico que para por la red, esto por su característica de integrar dos tarjetas *wireless* y de permitir la instalación de sistema operativo (OpenWrt), basado en Linux, que ofrece la posibilidad de tener varios paquetes para el control de tráfico.

- Un ataque de *DoS* a un sistema seguro como lo es el estándar *802.11i* con autenticación *EAP-TLS,* genera una cantidad de tráfico excesivo en la red, al incrementar la cantidad de tráfico para ser procesado, obligara al *AP* a incrementar el procesamiento, esto producirá una alteración dentro de varios parámetros del AP como: aumento de calor, aumento de procesamiento, y finalmente la disociación de los equipos, estos parámetros pueden ser identificados por software de administración y gestión de red, siendo estos una gran herramienta para un administrador de red al momento de reconocer un ataque ya que ofrecen la posibilidad del monitoreo continuo de los equipos por medio del protocolo *SNMP.*

- Se debe seguir el procedimiento correcto, sin omitir ningún paso al momento de instalar los certificados en los equipos, ya que al no establecer bien el método de autenticación generará un espacio para que se puedan producir ataques, que permitan vulnerar la seguridad de la red.

BIBLIOGRAFÍA

W., & Support, E. A. (s.f.). *AirPcap Nx*. Obtenido de http://www.cacetech.com/documents/AirPcap Nx Datasheet.pdf

Alix2d2 System board. (s.f.). Obtenido de http://www.pcengines.ch/alix2d2.htm

Download Kismet. (s.f.). Obtenido de https://www.kismetwireless.net/download.shtml

Download Wireshark. (s.f.). Obtenido de http://www.wireshark.org/download.html

He, C., & Ca, S. (2004). *Security Analysis and Improvements for IEEE 802 . 11i*.

How To Get TUONET-PEAP Up And Working On Windows 7. (s.f.). Obtenido de http://idoc.vsb.cz/en/okruhy/cit/tuonet/sluzby/wifi/tuonet-peap/windows-7-peap/

Jonathan, H. (2002). *RADIUS* (p. 206). O'Reilly.

Ramachandra, V. (2011). *BackTrack 5 Wireless Penetration Testing*. Packt Publishing Ltd. Obtenido de www.packtpub.com

Sébastien, W. (s.f.). *WPA2 + FreeRADIUS + EAP-TLS*. Obtenido de http://blog.wains.be/2009/09/13/wpa2-freeradius-eap-tls/

Walt, D. van der. (2011). *FreeRADIUS*. Packt Publishing Ltd. Obtenido de www.packtpub.com

What is 802.1x Security Authentication for Wireless Networks? (s.f.). Obtenido de http://kb.netgear.com/app/answers/detail/a_id/1209/~/what-is-802.1x-security-authentication-for-wireless-networks?

Wireless Pentesting on the Cheap (Kali + TL-WN722N) - WPA-Enterprise - Part II. (s.f.). Obtenido de http://securitysynapse.blogspot.com/2014/03/wireless-pentesting-on-cheap-kali-WPAEntPartII.html

Wireless Pentesting on the Cheap (Kali + TL-WN722N) - WPA-PSK. (s.f.). Obtenido de http://securitysynapse.blogspot.com/2014/01/wireless-pentesting-on-cheap-kali-tl.html

WIRESHARK'S MOST USEFUL DISPLAY FILTERS. (s.f.). Obtenido de http://www.firstdigest.com/2009/05/wiresharks-most-useful-display-filters/

ANEXOS

C A C E T E C H N O L O G I E S

AirPcap Nx

AirPcap Nx Pricing

(APCNX) AirPcap Nx single USB WLAN Adapter ..... $698 USD

(APCNX3) AirPcap Nx 3-Pack (includes 4-port USB mini-hub) ..... $2,094 USD

CACE Pilot / AirPcap Nx Bundles Pricing

(P12NX) CACE Pilot with 12 mo. Update Subscription and single AirPcap Nx adapter ..... $1,923 USD

(P12NX3) CACE Pilot with 12 mo. Update Subscription and AirPcap Nx 3-Pack..... $3,180 USD

Pricing for all CACE products can be found on our catalog page at www.cacetech.com/products/catalog.

Upgrading to AirPcap Nx

Owners of any functioning version of AirPcap can upgrade to AirPcap Nx for the difference in price of the adapters plus the associated shipping costs. Please contact sales@cacetech.com for upgrade instructions.

AirPcap Nx Host System Requirements

Pentium compatible or better

Available USB 2.0 or 1.1 Slot

Microsoft® Windows 2000
Microsoft® Windows XP
Microsoft® Windows 2003 Server
Microsoft® Windows Vista
Microsoft® Windows 7

AirPcap Nx Product Specifications

Network Topology:
IEEE 802.11a/b/g/n

Operating Frequencies:
2.412-2484 GHz (b/g/n)
4.920-5.825 GHz (a/n)

802.11n Supported Modes:
- Legacy Mode
- HT20 mixed mode
- HT40 mixed mode. In this mode, the adapter captures HT40, HT20 and legacy mode frames simultaneously.

802.11n Aggregation:
Capture of A-MPDU and A-MSDU frames

Modulations:
- CCK
- OFDM
- HT-OFDM
  - MCS 0-15
  - Up to 2 spatial streams
  - Detection of Short and Long Guard Interval
  - 2x2 MIMO

Data Rates:
Up to 54Mbps for 802.11a/b/g
Up to 130Mbps (20MHz channels) and 300Mbps (40MHz channels) for 802.11n

Antennas:
Two built-in Antennas
Two integrated MC-Card connectors for optional external antennas

Decryption:
WEP
WPA PSK (in Wireshark)
WPA2 PSK (in Wireshark)

Power:
USB Bus (no external power)
5VDC
300mA (max) or 1.5 Watt (max)

Temperature:
Operating: 0C to 55C
Storage: -20C to 70C

Humidity:
Operating: 10%-70% (Non-condensing)
Storage: 5%-95% (Non-condensing)

Certifications:
RoHS compliant

About CACE Technologies, Inc.

Founded in 2005 with the vision of offering superior tools to the networking community, CACE Technologies, Inc. has since become a leader in the network analysis industry. Our innovative product portfolio includes the AirPcap family of wireless packet capture adapters for Windows, the TurboCap™ full-rate dual-port Gigabit Ethernet capture and injection solution, and CACE Pilot™, a powerful and intuitive network analysis, visualization and reporting tool. CACE also provides support, training, and development for two of the most popular and highly acclaimed open-source networking tools: WinPcap and Wireshark. All CACE products are fully integrated with Wireshark and are designed to enhance the Wireshark user experience. For information on our tools and services, visit www.cacetech.com.

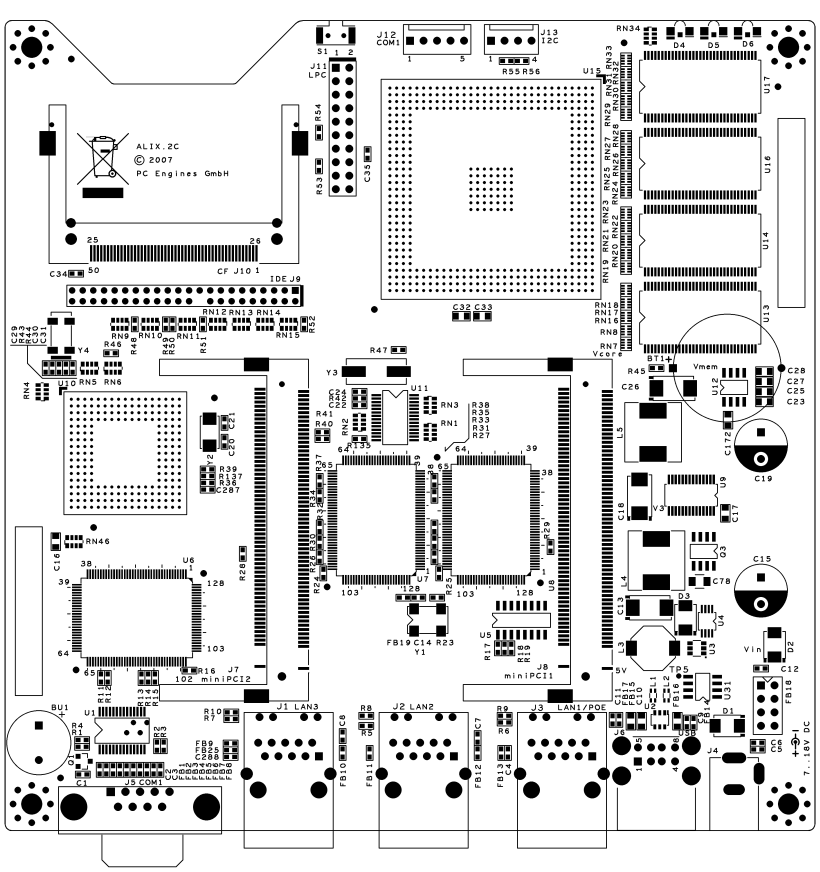
CACE TECHNOLOGIES

Creative
Advanced
Communication
Engineering

CACE Technologies
1949 5th Street, Suite 103
Davis, CA 95616
www.cacetech.com
tel: 530.758.2790
fax: 530.758.2781
email: info@cacetech.com

ALIX.2 series

Configuration	2 LAN / 2 miniPCI, or 3 LAN / 1 miniPCI
Power supply	7 to 20V DC, about 3 to 4W at Linux idle, peak about 6W without miniPCI cards and USB devices. Suggest a 18V / 15W supply. Center pin = positive, sleeve = ground, 2.1 mm diameter.
Temperature range	0 to 50°C.
Dimensions	6 x 6" (152.4 x 152.4 mm)



FECHA DE ENTREGA:

En la ciudad de Sangolquí, firman en constancia de la entrega del presente proyecto de Grado titulado **"DETECCIÓN DE VULNERABILIDADES EN REDES INALÁMBRICAS 802.11i, MEDIANTE EL ANÁLISIS DE TRÁFICO DE LA CAPA DE ENLACE"**, en calidad de Autores al Sr. Jaime Javier Astudillo Cabrera y al Sr. Andrés Sebastián Troya Estrella, estudiantes de la carrera de Ingeniería Electrónica en Redes y Comunicación de Datos, y recibe por parte del Departamento de Eléctrica y Electrónica el Director de Carrera de Redes y Comunicación de Datos, el Señor Dr. Nikolai Espinosa.

Jaime Javier Astudillo Cabrera	Andrés Sebastián Troya Estrella
140043161-3	171273655-0

Dr. Nikolai Espinosa
DIRECTOR DE LA CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS