ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Maestría en Seguridad Informática Aplicada

"IMPLEMENTACIÓN DE LET'S ENCRYPT PARA UN SERVIDOR WEB APACHE EN CENTOS 7"

EXAMEN DE GRADO (COMPLEXIVO)

Previa a la obtención del título de:

MAGISTER EN SEGURIDAD INFORMÁTICA APLICADA

ALBERTO JAVIER SANTOS FLORES

GUAYAQUIL – ECUADOR

AÑO: 2016

AGRADECIMIENTO

A mis padres, por todo el apoyo que me han brindado a lo largo de mi carrera profesional.

A mi novia, que me ha dado las fuerzas y el apoyo para levantarme y seguir adelante

A mis profesores, por el empeño y esfuerzo en impartir su conocimiento.

DEDICATORIA

A mis padres, a mi novia y amigos.

TRIBUNAL DE SUSTENTACIÓN

MSIG. LENIN FREIRE

DIRECTOR MSIA

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

MGS. JUAN CARLOS GARCIA

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

RESUMEN

El propósito de este proyecto es implementar Let’s Encrypt para un servidor web Apache en CentOS 7 que permita de manera sencilla generar un certificado SSL totalmente gratuito.

Con esta propuesta se podrán configurar servidores web que usen el protocolo https y que consten con un CA (autoridad certificadora) sin provocar costos adicionales, y se garantizará al usuario final que toda la comunicación realizada a través de ese servidor web se encuentra encriptada.

ÍNDICE GENERAL

ABREVIATURAS Y SIMBOLOGÍA

CA	Autoridad Certificadora
EPEL	Extra Packages for Enterprise Linux
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ISRG	Internet Security Research Group
RPM	Red Hat Package Manager
SSL	Secure Sockets Layer
TLS	Transport Layer Security

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Let’s Encrypt es una CA (autoridad certificadora) abierta y gratuita creada para el beneficio del público. Es un servicio que lo provee la ISRG (Internet Security Research Group) [1].

Los principales beneficios de Let’s Encrypt son:

- **Gratis:** Cualquiera que posea un dominio puede usarlo para obtener un certificado de confianza sin costo.
- **Automático:** El software que se ejecuta en un servidor web puede interactuar con Let’s Encrypt para obtener un certificado sin dolor, configurarlo seguramente y renovado automáticamente.
- **Seguro:** Let’s Encrypt servirá de plataforma para las mejores prácticas de seguridad avanzada TLS, tanto del lado del CA y ayudando a los operadores de sitios a asegurar adecuadamente sus servidores.
- **Transparente:** Todos los certificados emitidos o revocados son registrados públicamente y están disponibles para cualquier persona.
- **Abierto:** El protocolo de emisión y renovación automática se publicará como un estándar abierto que otros pueden adoptar.

- **Cooperativa:** Al igual que otros protocolos de internet subyacentes, Let’s Encrypt es un esfuerzo en conjunto para beneficio de la comunidad, más allá del control de cualquier organización.

La gran mayoría de servidores web manejan información que podría ser de importancia, por lo que es indispensable que la comunicación entre el cliente y el servidor se encuentre encriptada, para ello se hace uso del protocolo HTTPS y Let’s Encrypt nos ayudará a generar un certificado de forma gratuita [2].

CAPÍTULO 1

GENERALIDADES

1.1. Descripción del problema

Hoy en día la comunicación encriptada se ha vuelto una obligación en casi todas las implementaciones de servidores web, es por esto que se hace uso del protocolo HTTPS (protocolo seguro de transferencia de hipertexto). El protocolo HTTPS utiliza un cifrado basado en SSL/TLS. Para que el navegador pueda confiar en el servidor web, este debe contar con un certificado SSL, el cual es emitido por una CA (autoridad certificadora) [3]. Dicho esto, pueden surgir dos escenarios:

1. El servidor web no obtiene ningún certificado SSL.

2. El servidor web adquiere un certificado SSL pagando un valor a una de las autoridades certificadoras.

Si el administrador del servidor web opta por la opción 1, el navegador al no encontrar un certificado válido le emitirá al usuario una alerta (Ver Figura 1.1) la cual puede provocar temor en un usuario no experimentado y hará que este decida no entrar al sitio web.

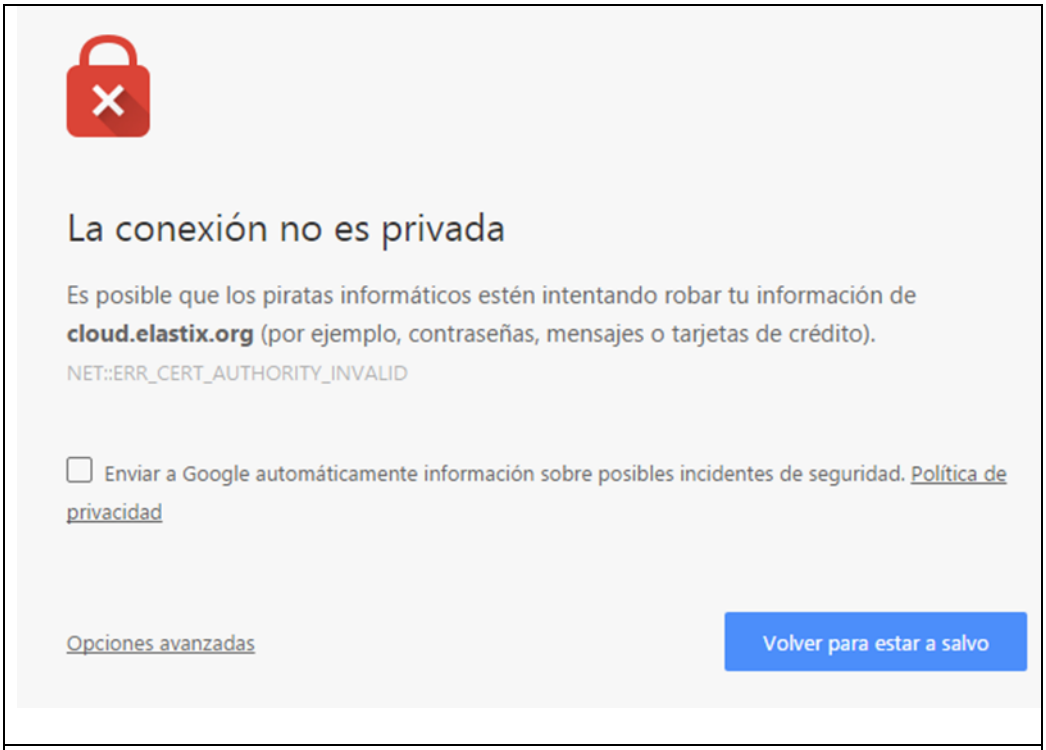


Figura 1.1
Conexión a servidor web sin certificado.
Fuente: https://cloud.elastix.org

Para la opción 2, no se presentará el escenario anterior, pero se debe cancelar un valor que en ciertas ocasiones la empresa no está dispuesta a pagar.

1.2. Solución propuesta

- Mantener la comunicación encriptada entre el cliente (navegador) y el servidor web a través del uso del protocolo HTTPS.
- La solución no debe generar costos para la obtención y renovación de certificados SSL.
- El usuario final debe poder ingresar al servidor web sin que el navegador le muestre una advertencia de que podría estar ingresando a un sitio inseguro.

BENEFICIO DE LA SOLUCIÓN:

- No requiere un gasto adicional para la empresa.
- El administrador del servidor no requiere de conocimientos adicionales sobre la herramienta Let’s Encrypt.
- Seguridad para el cliente de que la comunicación se encuentra encriptada.

CAPÍTULO 2

METODOLOGÍA DEL DESARROLLO DE LA SOLUCIÓN

2.1. Crear un Virtual Host para el dominio

Tal como se mencionó al inicio la implementación se la desarrollará para un webserver Apache en CentOS 7.

Asumiremos que el servidor CentOS 7 ya tiene instalado el paquete Apache que para esta distribución se conoce como httpd. El directorio de configuración de Apache se encuentra en /etc/httpd.

Comprobamos que el servidor sobre el que se hará la implementación es un CentOS 7 y tiene Apache instalado (httpd) (Ver Figura 2.1).

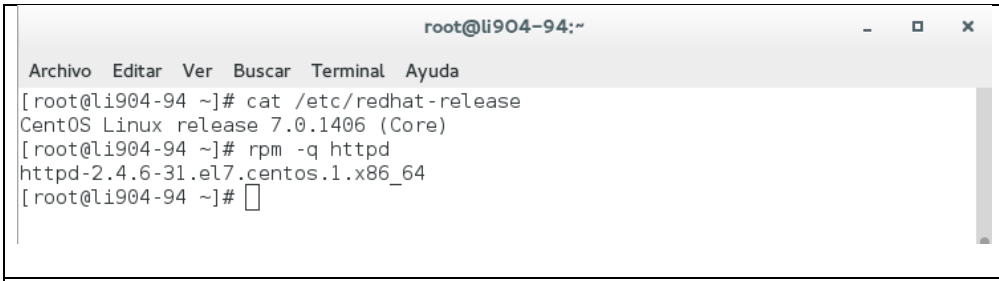


Figura 2.1
Versión de CentOS y paquete Apache instalado
Fuente: Ing. Alberto Santos Flores

Lo primero será crear un directorio que contenga todos los sitios disponibles en el servidor (Ver Figura 2.2).

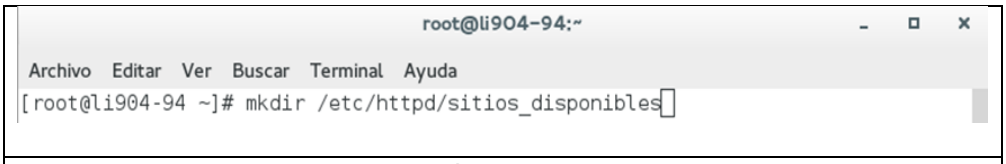


Figura 2.2
Se crea directorio con sitios disponibles en el servidor
Fuente: Ing. Alberto Santos Flores

Ahora crearemos el directorio que contendrá los hosts activos en el servidor. Este directorio sólo contendrá Links simbólicos a los virtual host localizados dentro de /etc/httpd/sitios_disponibles (Ver Figura 2.3).

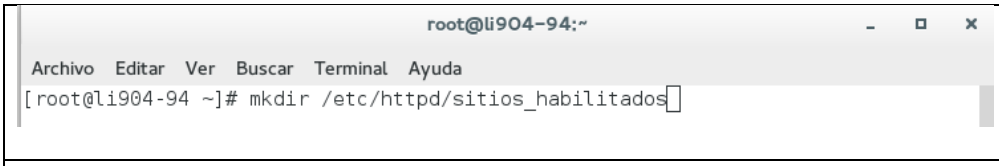


Figura 2.3
Se crea directorio con hosts activos en el servidor
Fuente: Ing. Alberto Santos Flores

Debemos indicarle a Apache cómo encontrar los archivos con los virtual host. Para ello debemos editar el archivo de configuración de Apache /etc/httpd/conf/httpd.conf y agregar la siguiente línea al final del archivo:

IncludeOptional sitios_habilitados/*.conf

El siguiente paso será crear el archivo de configuración del virtual host. Para ello debemos crear un archivo con cualquier nombre cuya extensión sea .conf debajo de la ruta /etc/httpd/sitios_disponibles (Ver Figura 2.4).

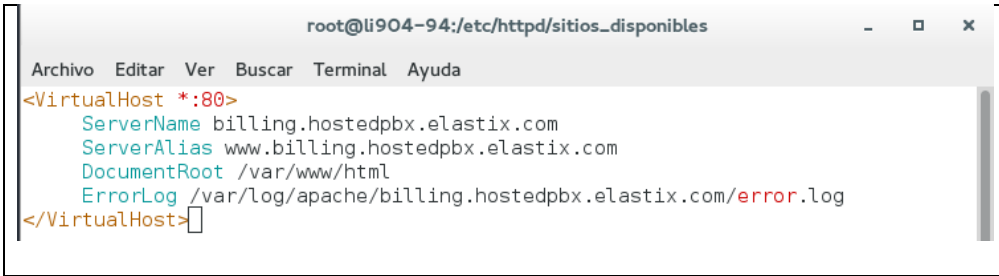


Figura 2.4
Se crea archivo de configuración del virtual host
Fuente: Ing. Alberto Santos Flores

A continuación se detalla brevemente los datos ingresados en el archivo de configuración del virtual host [4].

- **ServerName:** El nombre principal del dominio.
- **ServerAlias (Opcional):** Un alias para el dominio principal. Es una práctica común colocar como prefijo www al domino principal.
- **DocumentRoot:** La localización de los archivos del sitio web.
- **ErrorLog (Opcional):** Localización del log de errores para el virtual host. Recuerde que antes de iniciar el servicio, debe existir el archivo indicado aquí.

Ahora necesitamos indicarle a Apache que queremos que este sitio web sea habilitado. Para lograrlo debemos crear un link simbólico dentro de sitios_habilitados que apunte al nuevo archivo de configuración del virtual host. Por lo que se ejecutará el siguiente comando:

```
ln -s /etc/httpd/sitios_disponibles/billing.hostedpbx.elastix.com.conf
/etc/httpd/sitios_habilitados/billing.hostedpbx.elastix.com.conf
```

De esta forma si quisiéramos deshabilitar un virtual host simplemente debemos remover el link dentro de sitios_habilitados y reiniciar Apache.

Se debe abrir el archivo /etc/httpd/conf/httpd.conf y buscar por las directivas ServerName y ServerAlias. Si están configuradas con el mismo dominio que el virtual host entonces debemos comentar esa línea agregando el signo # al inicio de la línea.

Finalmente lo que queda es reiniciar el servicio de Apache (Ver Figura 2.5).

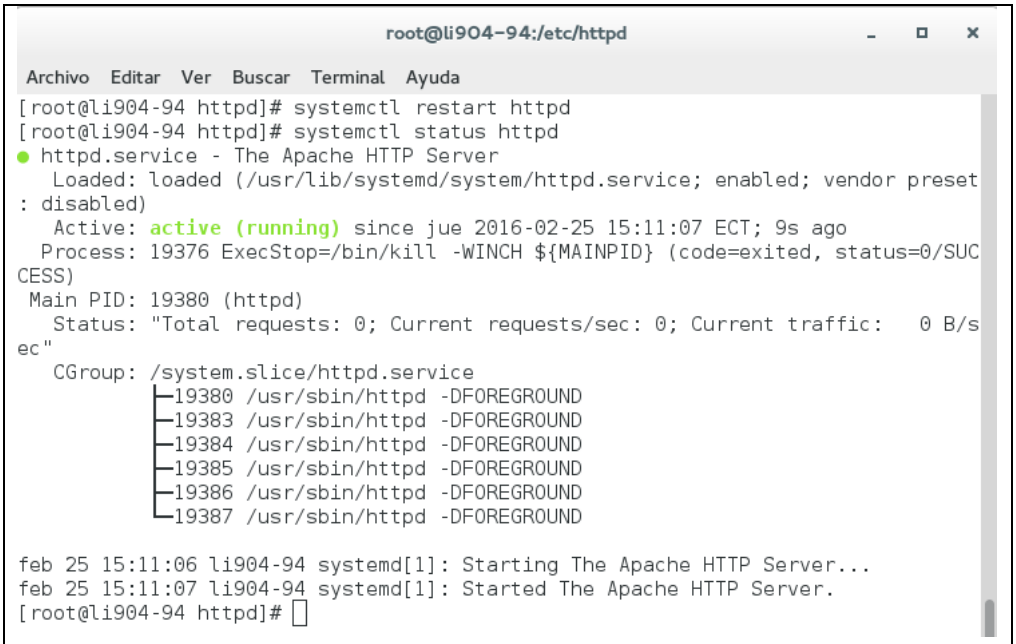


Figura 2.5
Se reinicia el servicio de Apache
Fuente: Ing. Alberto Santos Flores

2.2. Instalar dependencias en el servidor

Antes de poder instalar el cliente de Let's Encrypt en el servidor y generar el certificado SSL es necesario instalar unas cuantas dependencias en el servidor CentOS.

Primero se debe instalar el repositorio EPEL (Extra Packages for Enterprise Linux) (Ver Figura 2.6).

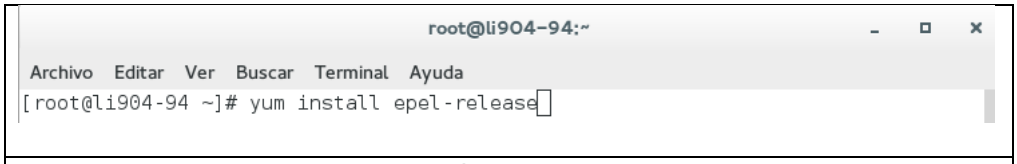


Figura 2.6
Se instala el repositorio EPEL
Fuente: Ing. Alberto Santos Flores

También se necesitará del paquete git para poder descargar el cliente de Let's Encrypt (Ver Figura 2.7).

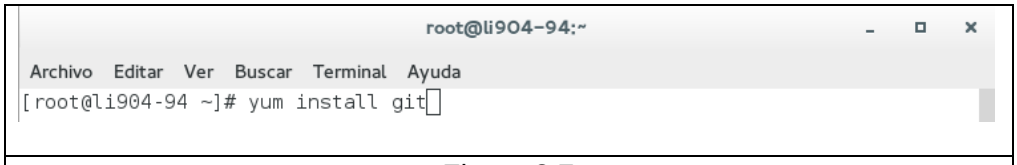


Figura 2.7
Se instala el paquete git
Fuente: Ing. Alberto Santos Flores

2.3. Descarga del cliente de Let's Encrypt

El siguiente paso es descargar el cliente de Let's Encrypt del repositorio oficial, colocando sus archivos en alguna ruta especial. Para esta implementación se lo instalará bajo la ruta /opt (ver Figura 2.8) [5].

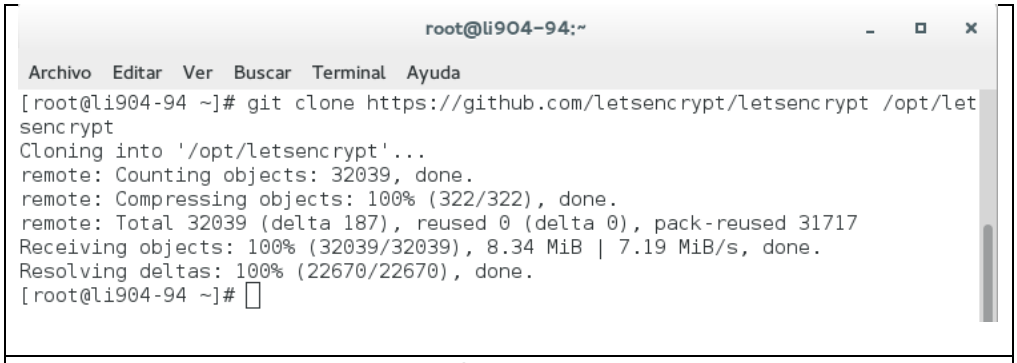


Figura 2.8
Descarga de Let's Encrypt del repositorio oficial
Fuente: Ing. Alberto Santos Flores

De esta forma se tendrá una copia local de Let's Encrypt en la ruta /opt/letsencrypt.

2.4. Generación del certificado SSL

Generar el certificado SSL para Apache usando Let's Encrypt es sencillo. El cliente obtendrá e instalará automáticamente un nuevo certificado SSL que sea válido para los dominios que son ingresados como parámetros.

Para ejecutar la instalación interactiva y obtener el certificado para un solo dominio se debe correr el comando letsencrypt-auto con:

./letsencrypt-auto --apache -d billing.hostedpbx.elastix.com

Recuerde que billing.hostedpbx.elastix.com es el dominio al que se le generará el certificado, este parámetro deberá cambiarlo si desea generar un certificado para otro dominio.

En caso de que desee generar un certificado para múltiples dominios o subdominios deberá ejecutar el siguiente comando:

./letsencrypt-auto --apache -d billing.hostedpbx.elastix.com -d www.billing.hostedpbx.elastix.com

El primer dominio en la lista de parámetros será el domino base usado por Let's Encrypt para crear el certificado.

Después de que las dependencias son instaladas, a usted se le mostrará una guía paso a paso para que personalice su certificado. Deberá ingresar su dirección de email para el caso de que pierda la llave de recuperación y también podrá escoger entre habilitar HTTP y HTTPS o forzar la redirección a HTTPS, para nuestro caso escogeremos la segunda opción.

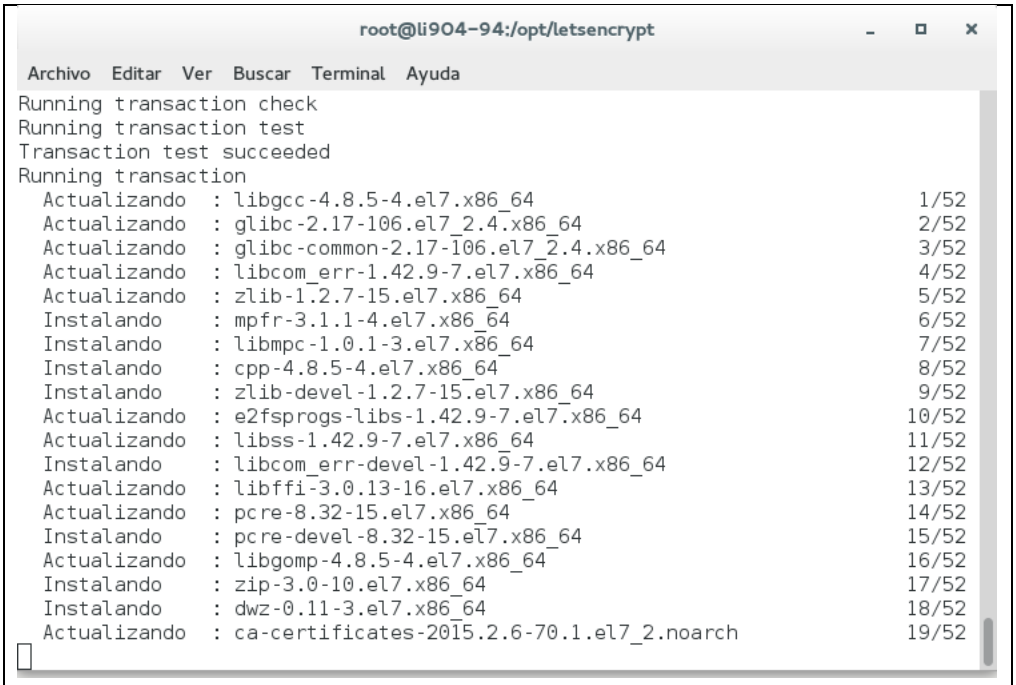


Figura 2.9
Descarga e instalación de dependencias de Let's Encrypt
Fuente: Ing. Alberto Santos Flores

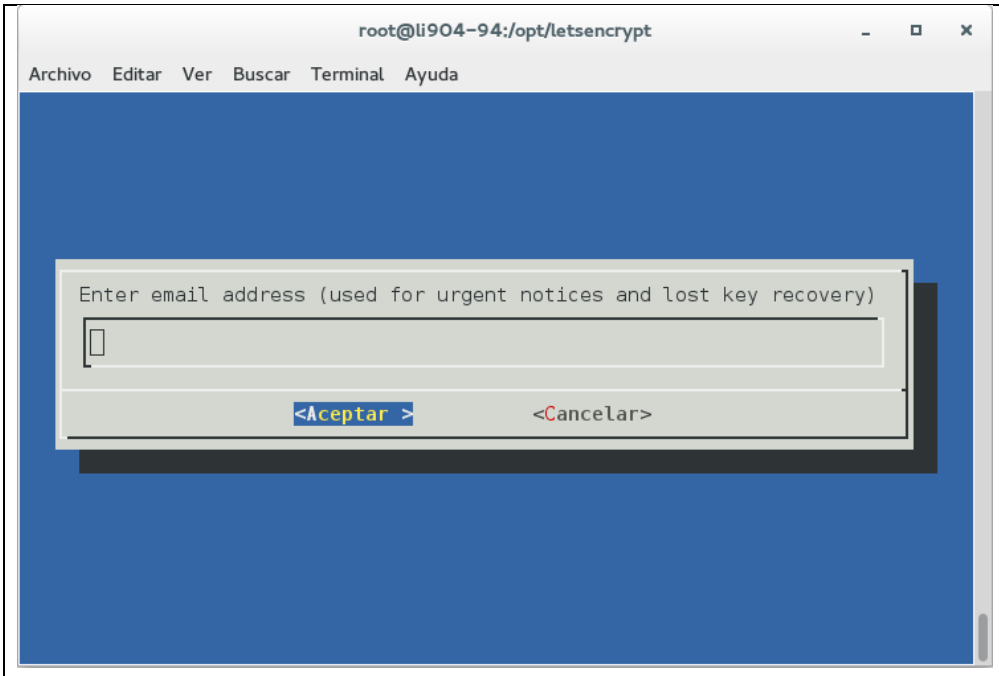


Figura 2.10
Ingreso de email para notificaciones
Fuente: Ing. Alberto Santos Flores

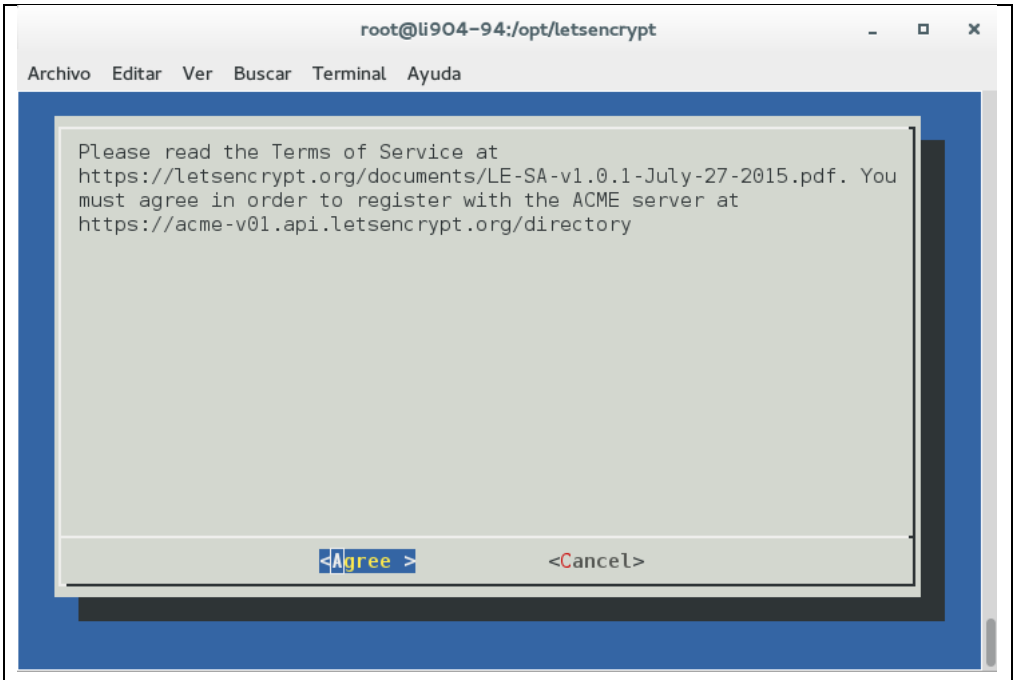


Figura 2.11
Aviso sobre términos y condiciones de Let's Encrypt
Fuente: Ing. Alberto Santos Flores

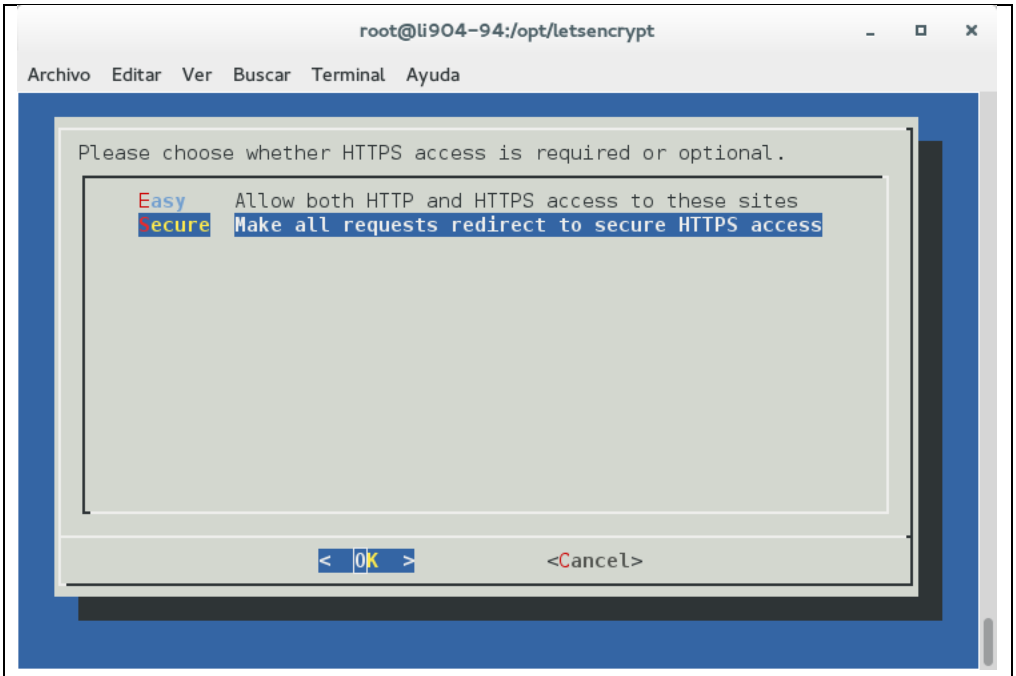


Figura 2.12
Escoger que las peticiones sean redireccionadas a HTTPS
Fuente: Ing. Alberto Santos Flores

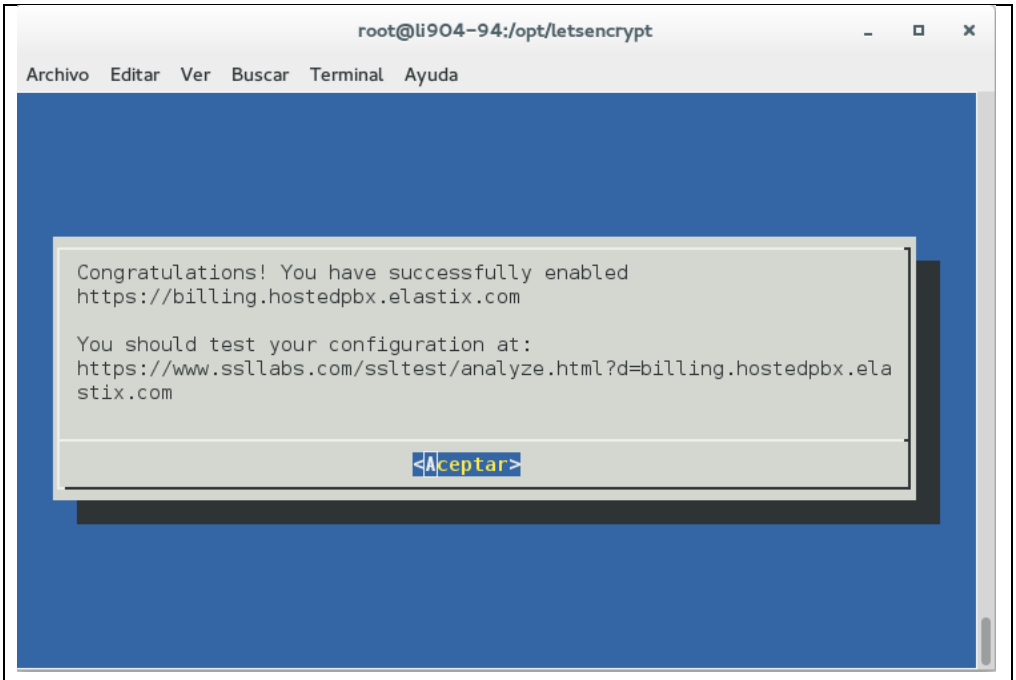


Figura 2.13
Finalización de la generación del certificado SSL
Fuente: Ing. Alberto Santos Flores

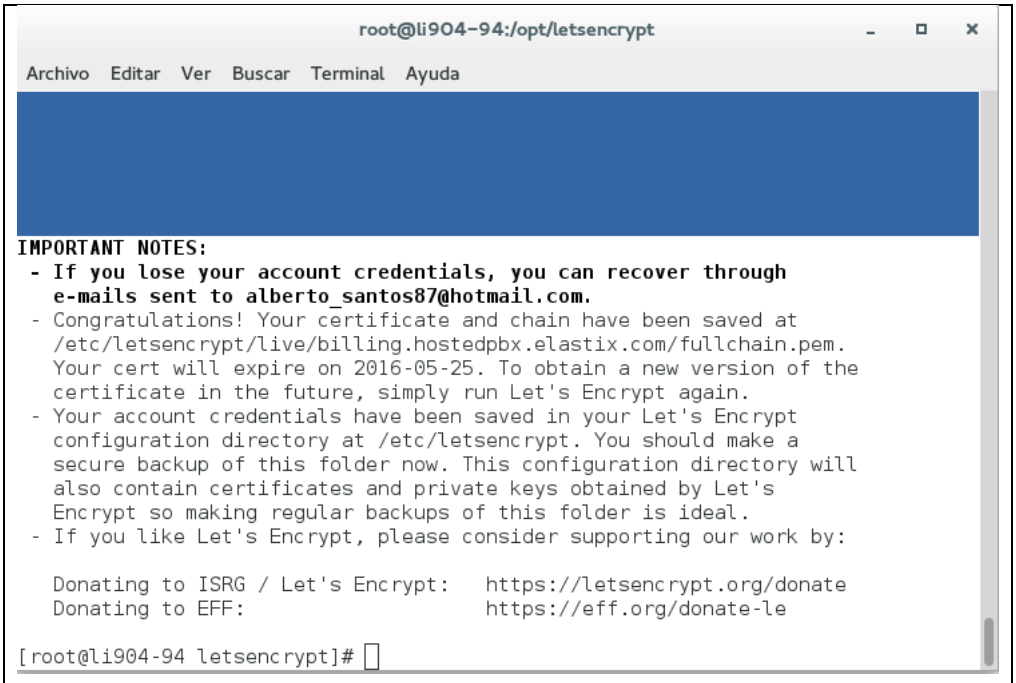


Figura 2.14
Notificación final con información sobre Let's Encrypt
Fuente: Ing. Alberto Santos Flores

Los archivos del certificado generado se encontrarán en la ruta /etc/letsencrypt/live.

Para mantener los archivos de los virtual host de una forma organizada, es una buena idea mover este nuevo archivo de virtual host al directorio sitios_disponibles y crear un enlace simbólico dentro de sitios_habilitados (Ver Figura 2.15).

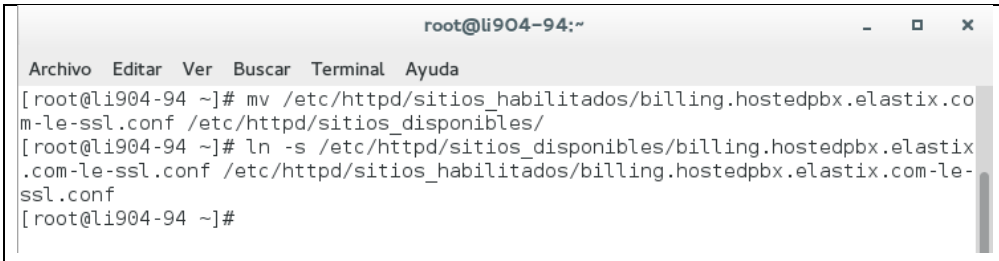


Figura 2.15
Creación de enlace simbólico hacia sitios_habilitados
Fuente: Ing. Alberto Santos Flores

El directorio sitios_habilitados debe lucir como la Figura 2.16.

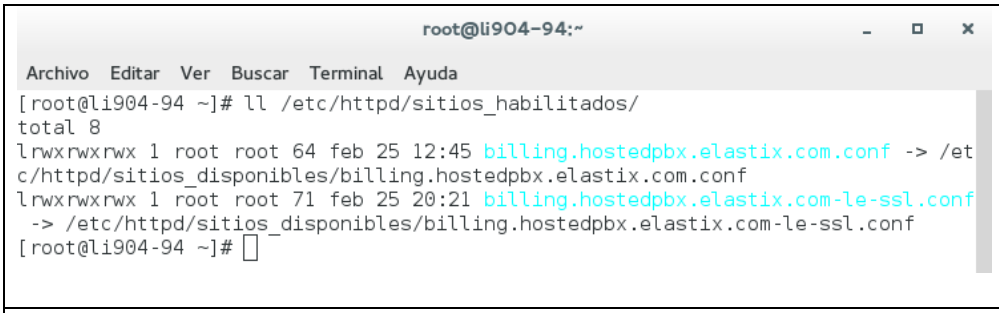


Figura 2.16
Contenido de directorio /etc/httpd/sitios_habilitados
Fuente: Ing. Alberto Santos Flores

Finalmente se debe reiniciar Apache con el comando:

```
systemctl restart httpd
```

2.5. Renovación automática del certificado SSL

Los certificados de Let's Encrypt tienen una duración de 90 días, por lo que es recomendable realizar la renovación del certificado antes de que se cumpla dicho tiempo, para permitir un margen de error considerable se aconseja realizar la renovación luego de pasados 60 días.

Para manualmente renovar un certificado de Let's Encrypt para Apache sin interacción en la línea de comandos se puede ejecutar el comando:

```
./letsencrypt-auto certonly --apache --renew-by-default -d
billing.hostedpbx.elastix.com
```

Para facilitar esta tarea se puede crear un cron que ejecute el comando anterior cada 60 días.

CAPÍTULO 3

ANÁLISIS DE RESULTADOS

3.1. Validez de certificado SSL

Antes de iniciar el proceso de generación del certificado SSL, el servidor donde se realizó la implementación era un CentOS 7 con un web server Apache pero sin haber adquirido el certificado SSL y con el dominio billing.hostedpbx.elastix.com. Por lo tanto al intentar ingresar al servidor web a través del protocolo HTTPS se obtiene lo mostrado en la Figura 3.1.

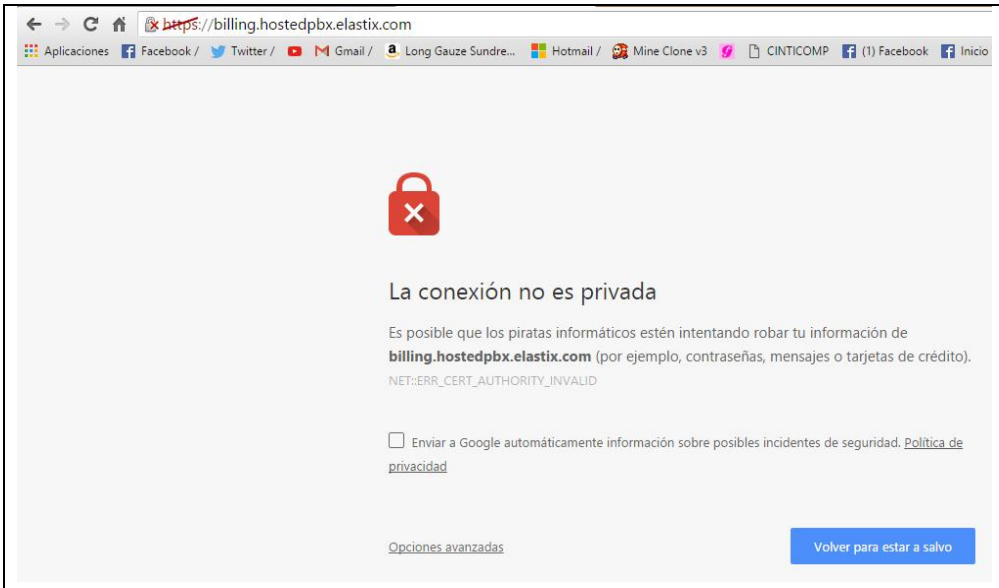


Figura 3.1
Ingreso al servidor billing.hostedpbx.elastix.com antes de generación del certificado SSL
Fuente: Ing. Alberto Santos Flores

Una vez realizada la implementación de la solución propuesta, podemos verificar la validez del certificado entrando a la dirección:

https://www.ssllabs.com/ssltest/analyze.html?d=billing.hostedpbx.elastix.com&latest

Al visitar la página podemos observar un reporte (Ver Figuras 3.2, 3.3 y 3.4).

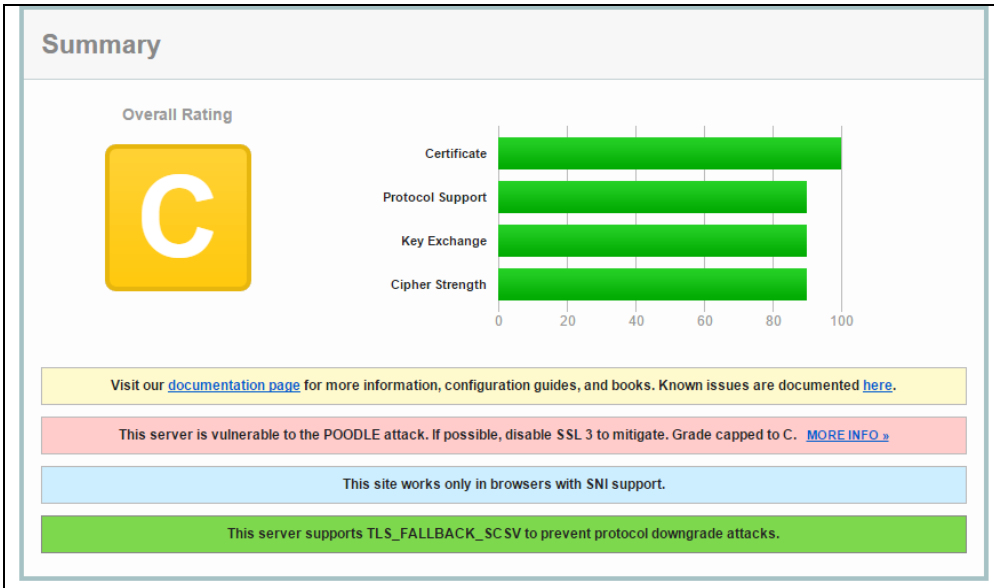
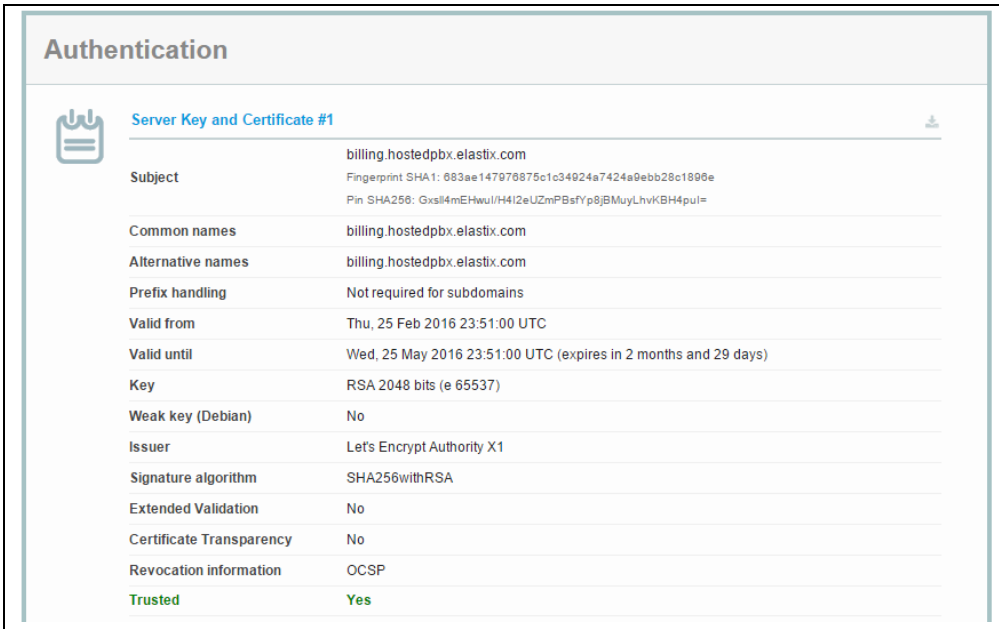


Figura 3.2
Reporte con calificación del servidor
Fuente:

https://www.ssllabs.com/ssltest/analyze.html?d=billing.hostedpbx.elastix.com



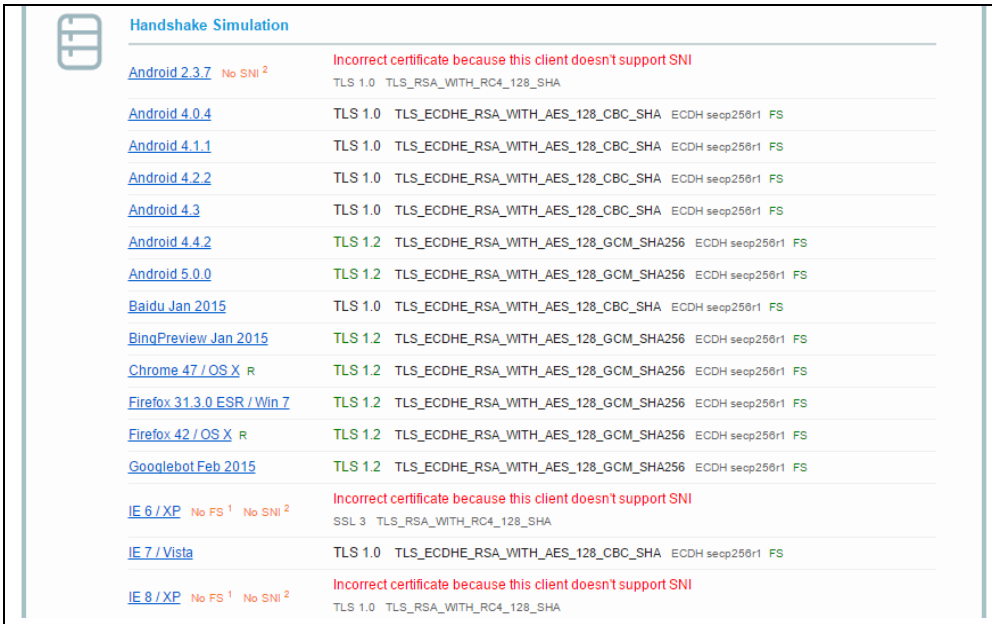
Authentication

Server Key and Certificate #1

Subject	billing.hostedpbx.elastix.com Fingerprint SHA1: 683ae147976875c1c34924a7424a9ebb28c1896e Pin SHA256: Gxsll4mEHwu//H4I2eUZmPBsfYp8jBMuyLhvKBH4pul=
Common names	billing.hostedpbx.elastix.com
Alternative names	billing.hostedpbx.elastix.com
Prefix handling	Not required for subdomains
Valid from	Thu, 25 Feb 2016 23:51:00 UTC
Valid until	Wed, 25 May 2016 23:51:00 UTC (expires in 2 months and 29 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	Let's Encrypt Authority X1
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	No
Revocation information	OCSP
Trusted	Yes

Figura 3.3
Reporte de autenticación del servidor
Fuente:

https://www.ssllabs.com/ssltest/analyze.html?d=billing.hostedpbx.elastix.com



Handshake Simulation

Android 2.3.7 No SNI [2]	Incorrect certificate because this client doesn't support SNI TLS 1.0 TLS_RSA_WITH_RC4_128_SHA
Android 4.0.4	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Android 4.1.1	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Android 4.2.2	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Android 4.3	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Android 4.4.2	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 5.0.0	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Baidu Jan 2015	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
BingPreview Jan 2015	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Chrome 47 / OS X R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 31.3.0 ESR / Win 7	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 42 / OS X R	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Googlebot Feb 2015	TLS 1.2 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
IE 6 / XP No FS [1] No SNI [2]	Incorrect certificate because this client doesn't support SNI SSL 3 TLS_RSA_WITH_RC4_128_SHA
IE 7 / Vista	TLS 1.0 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
IE 8 / XP No FS [1] No SNI [2]	Incorrect certificate because this client doesn't support SNI TLS 1.0 TLS_RSA_WITH_RC4_128_SHA

Figura 3.4
Reporte de simulación de handshake con diferentes versiones de navegadores
Fuente:

https://www.ssllabs.com/ssltest/analyze.html?d=billing.hostedpbx.elastix.com

Para verificar la validez del certificado SSL se procede a ingresar al web server https://billing.hostedpbx.elastix.com y se observa que el navegador ya no muestra una alerta indicando que el sitio es inseguro (Ver Figuras 3.5, 3.6 y 3.7).

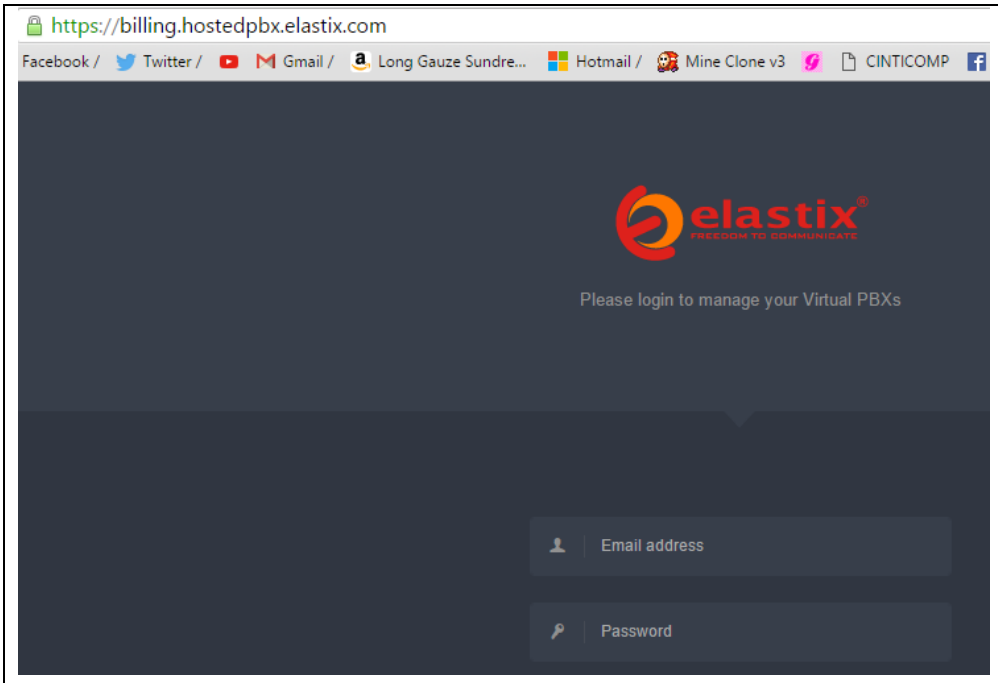


Figura 3.5
Ingreso al servidor billing.hostedpbx.elastix.com después de generación del certificado SSL
Fuente: Ing. Alberto Santos Flores

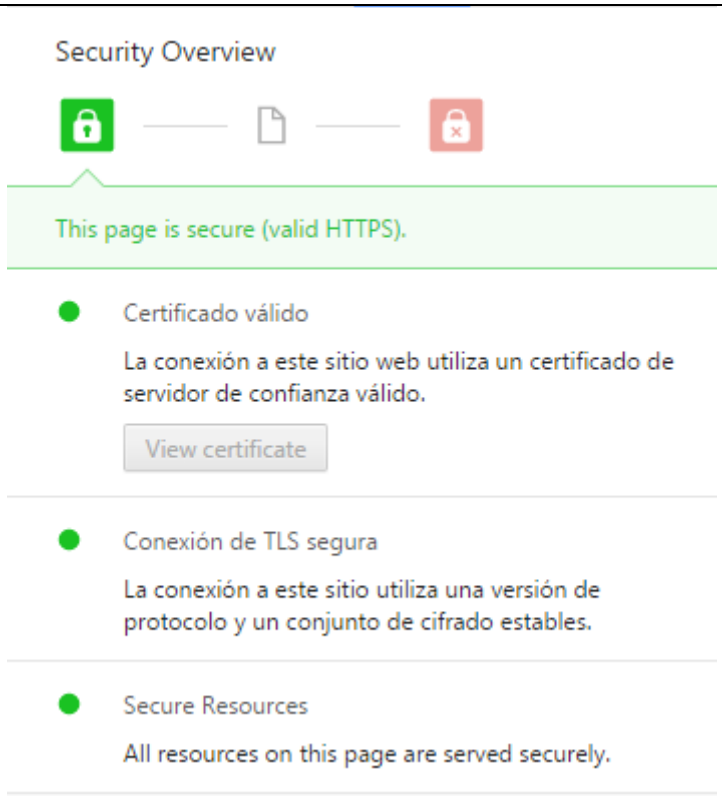


Figura 3.6
Información sobre la seguridad del servidor
Fuente: Ing. Alberto Santos Flores

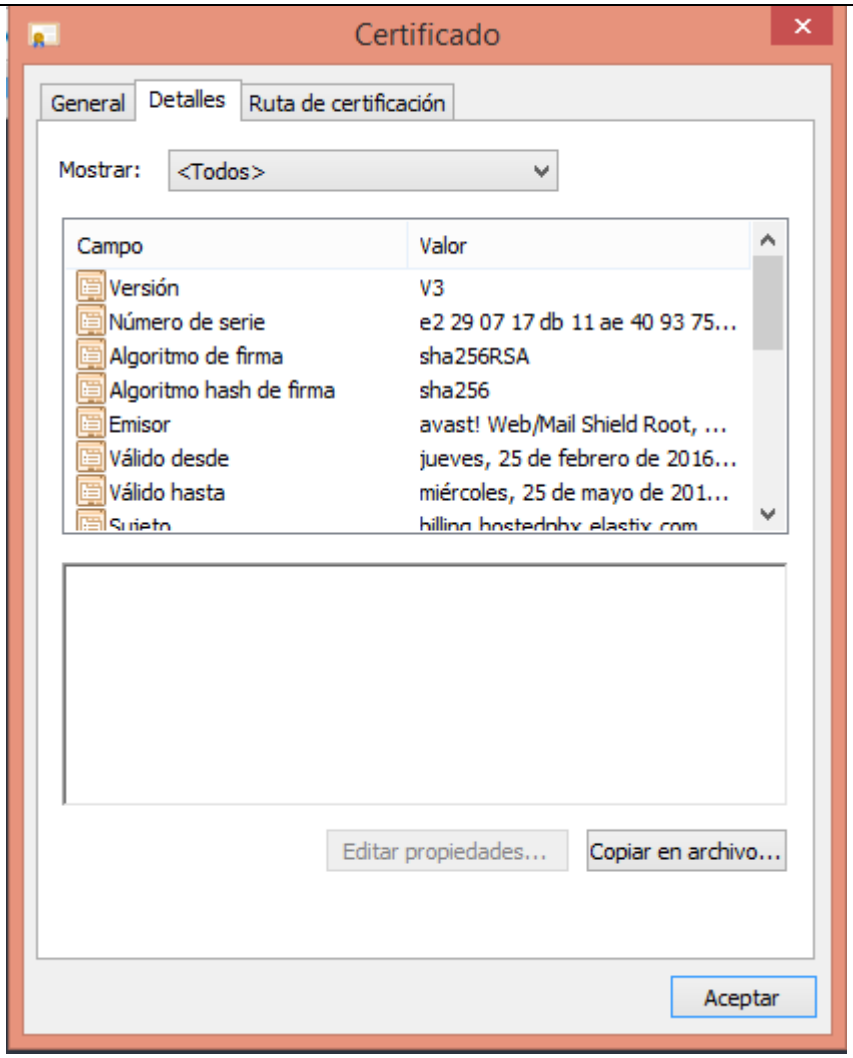


Figura 3.7
Detalles del certificado
Fuente: Ing. Alberto Santos Flores

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. El servidor web utiliza el protocolo HTTPS que garantiza que la comunicación entre el cliente y el servidor sea encriptada.
2. Let’s Encrypt genera el certificado SSL y lo publica para que sea aceptado por los navegadores.
3. No es necesario gastar en recursos adicionales para la generación ni renovación de los certificados SSL.
4. Qualys SSL Labs elabora un reporte con información del certificado, seguridad del servidor y da una calificación en base a esos parámetros.

RECOMENDACIONES

1. Se recomienda elaborar un paquete RPM que se encargue de instalar todas las dependencias y descargar el cliente Let's Encrypt para optimizar el tiempo.
2. Dado que el cliente Let's Encrypt está todavía en fase beta, es recomendable realizar constantemente actualizaciones para la corrección de bugs y nuevas funcionalidades.
3. Let's Encrypt es una autoridad certificadora, es decir garantiza que la comunicación por HTTPS está encriptada, pero no garantiza la autenticidad del servidor, por lo tanto los usuarios deben tener cuidado ya que el servidor podría ser usado para realizar ataques informáticos suplantando la identidad del mismo.

BIBLIOGRAFÍA

[1] Let's Encrypt, Página oficial de Let's Encrypt, https://letsencrypt.org/, fecha de consulta diciembre 2015

[2] Wikipedia, Let's Encrypt, https://en.wikipedia.org/wiki/Let%27s_Encrypt, fecha de consulta febrero 2016

[3] Wikipedia, Autoridad de certificación, https://es.wikipedia.org/wiki/Autoridad_de_certificaci%C3%B3n, fecha de consulta diciembre 2015

[4] DigitalOcean, Seguridad con Let's Encrypt, https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-centos-7, fecha de consulta enero 2016

[5] GitHub, Repositorio de Let's Encrypt, https://github.com/letsencrypt/letsencrypt, fecha de consulta febrero 2016

GLOSARIO

Let’s Encrypt	Autoridad certificadora abierta y gratuita.
CentOS	Sistema Operativo Linux basado en la distribución Red Hat.
Apache	Servidor web HTTP de código abierto.
Autoridad Certificadora	Entidad de confianza responsable de emitir y revocar los certificados digitales.