apache, ssl, certificados

Implementación de https

Recomendaciones de puertos

- Disponer del puerto estándar TCP/443, asociado al protocolo HTTPS, abierto.
- Disponer del puerto TCP/80 también abierto, pero configurar el servidor o aplicación web para que se lleve a cabo una redirección automática de tipo 301 (“301 Moved Permanently”) desde HTTP hacia HTTPS ante cualquier petición en el puerto TCP/80.

Generación de claves

* Hacer uso de claves RSA de 2.048 bits de longitud. • Alternativamente, o complementariamente, hacer uso de claves ECDSA de 256 bits de longitud, ya que ofrecen mayor seguridad y rendimiento que las claves RSA. • Se recomienda buscar el equilibrio entre seguridad y rendimiento, evitando excederse en introducir “demasiada” seguridad, por lo que actualmente es preferible hacer uso de claves ECDSA (256 bits) frente a RSA (2.048 bits). • Hacer uso simultáneo, si las tecnologías web empleadas lo soportan, de claves ECDSA y RSA. • Hacer uso de un buen generador de números aleatorios para generar las claves. • El almacenamiento de las claves debe ser protegido empleando una contraseña robusta (passphrase) y, alternativamente, mediante módulos de seguridad hardware específicos (HSM, Hardware Security Module). • Realizar una correcta gestión de las claves: mantener en secreto las claves privadas, disponer de copias de seguridad o backups seguros de las claves, renovar las claves periódicamente, emplear un proceso de borrado seguro, renovar las claves tras un incidente de seguridad, exponer las claves privadas al menor número posible de sistemas y personas, etc.

Gestión de los certificados digitales

Para los servidores web expuestos públicamente es necesario disponer de un certificado digital emitido por una CA pública de confianza y ampliamente reconocida, no siendo válido hacer uso de certificados digitales autofirmados, ni de CAs privadas. • Evaluar y seleccionar el tipo adecuado de certificado digital (DV, OV o EV) que se desea obtener para cada servidor o aplicación web. • Evaluar y seleccionar la CA que emitirá el certificado empleando múltiples criterios objetivos, como su reputación, la funcionalidad que ofrece, su adopción de nuevos estándares, los mecanismos de revocación de certificados que implementa, etc. • Seleccionar el periodo de renovación del certificado digital y renovar los certificados periódicamente y siempre antes de su vencimiento. • Hacer uso de certificados digitales firmados mediante SHA-256 (o superior). En ningún caso hacer uso de firmas basadas en SHA-1 o MD5. • Identificar y configurar adecuadamente (en el orden correcto) la cadena completa de certificación (o de certificados) asociada al certificado digital, desde la CA raíz, pasando por todas las CAs intermedias, hasta el certificado final. • Seleccionar el tipo adecuado de certificado digital respecto al nombre de la entidad representada por éste, es decir, referencia directa o única, lista de nombres, certificado comodín, o multi-dominio. • El certificado digital debería incluir tanto el nombre del servidor web representado (“www”, o la lista de servidores web, o el comodín) como de su dominio. • El nombre de la entidad debe estar reflejado en el certificado digital a través del campo “SAN” (Subject Alternative Name, extensión “subjectAltName”) en lugar de mediante el “CN” (Common Name). • Asegurar la validez del certificado digital en todo momento, incluyendo el nombre

de la entidad, periodo de validez, CA emisora del certificado, revocación del certificado y propósito. • Asociar el certificado digital a una CA que haga uso de Certificate Transparency (CT) y confirmar que el certificado ha sido registrado en los logs públicos de CT. • Monitorizar los logs o registros de CT para identificar la emisión de certificados digitales de dominios propios de manera ilegítima por parte de otras CAs, sin la autorización del propietario del dominio. Complementariamente, hacer uso de servicios de monitorización de CT para ser informado de cambios en los certificados digitales asociados a un dominio. • Proporcionar información de registro en CT (mediante SCT) en las respuestas HTTPS del servidor web hacia los clientes web mediante una extensión del certificado digital X.509v3, (o preferiblemente) mediante una extensión específica de TLS o empleando OCSP Stapling.

- Se recomienda valorar la utilización de la cabecera HTTP “Expect-CT” para declarar la necesidad de usar certificados digitales reconocidos en CT. En su defecto, hacer uso de las capacidades de notificación de “Expect-CT” para identificar el uso de certificados digitales no válidos desde el punto de vista de CT.

• Configurar en el servicio de resolución de nombres DNS los registros CAA correspondientes a la lista de CAs asociadas a los certificados digitales del dominio. • Asociar el certificado digital a una CA que haga uso de CAA (Certification Authority Authorization). • Hacer uso de las capacidades de notificación de CAA para identificar errores o peticiones de certificados no válidas. • Hacer uso de HPKP (HTTP Public Key Pinning) para establecer los certificados digitales reconocidos asociados al entorno, servidor o aplicación web. En su defecto, hacer uso de las capacidades de notificación de HPKP para identificar el uso de certificados digitales válidos, pero no legítimos o autorizados, contra el entorno web. • En el caso de hacer uso de HPKP, se recomienda emplear un valor de “max-age” de 5184000 (60 días o 2 meses), empezar haciendo un uso limitado de la cabecera HPKP en un recurso concreto, e ir incrementando progresivamente el valor de “max-age”. • Si se está haciendo uso de la directiva “report-uri” de HPKP, se debe emplear una referencia web HTTPS asociada a un servidor web distinto al que ha enviado la cabecera HPKP. • HPKP debería de ser implementado para la totalidad del dominio mediante la directiva “includeSubDomains”. • HPKP debe de incluir, como mínimo, dos pines: uno principal, asociado a uno de los certificados digitales de la cadena de certificación actual, y uno de backup, que no debe de estar en la cadena de certificación actual. • Evaluar y seleccionar el certificado digital de la cadena de certificación actual que será utilizado para la generación de los pines de HPKP, pudiendo elegir un escenario más restrictivo (servidor web final), intermedio (CAs intermedias) o más permisivo (CA raíz). • Seleccionar la CA vinculada a los pines de backup de HPKP y proteger convenientemente la clave privada asociada a estos pines.

Generar certificados autofirmados

Por ejemplo para generar unos certificados autofirmados para nextcloud

Si no tenemos el módulo ssl lo instalamos

```
yum install mod_ssl
```

Creamos un directorio específico para almacenar la clave privada

```
sudo mkdir /etc/ssl/private
sudo chmod 700 /etc/ssl/private/
```

```
sudo openssl req -x509 -days 365 -nodes -newkey rsa:2048 -keyout
/etc/ssl/private/nextcloud.key -out /etc/ssl/certs/nextcloud.crt
```

contestamos a las preguntas del asistente y se generaran nuestros certificados privados.

Hay que tener en cuenta que cuando nos pregunte el **Common Name** poner el nombre de dominio asociado a nuestro servidor o la ip en caso de no tener dominio

Para tener una negociación de certificados fuerte **Perfect Forward Secrecy** necesitamos crear un grupo DH (Diffie-Hellman group) y añadirlo a nuestro certificado. Para ello ejecutamos

```
sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

y procedemos a añadirlo a nuestro certificado autofirmado

```
cat /etc/ssl/certs/dhparam.pem | sudo tee -a /etc/ssl/certs/nextcloud.crt
```

Configurar Apache para SSL

Editamos el fichero de configuración SSL de nuestro servidor

Tendrás que editar/crear el fichero de configuración **/etc/httpd/conf.d/ssl.conf** o bien editar el fichero de configuración de tu Virtual Host.

```
##Fichero ejemplo de
https://www.sugeek.co/instalar-certificado-ssl-en-centos-7/
Listen 443
NameVirtualHost *:443
SSLPassPhraseDialog  builtin
SSLSessionCacheTimeout  300
SSLRandomSeed startup file:/dev/urandom  256
SSLRandomSeed connect builtin
SSLCryptoDevice builtin
SSLStrictSNIVHostCheck off

<VirtualHost *:443>
DocumentRoot /var/www/html/misitio
ServerName www.misitio.com
ServerAlias misitio.com
SSLEngine on
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
SSLCertificateFile /etc/letsencrypt/live/www.misitio.com/cert.pem
SSLCertificateKeyFile /etc/letsencrypt/live/www.misitio.com/privkey.pem
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
LogLevel warn
```

```
<Files ~ “\.(cgi|shtml|phtml|php3?)$”>
SSLOptions +StdEnvVars
</Files>
SetEnvIf User-Agent “.*MSIE.*” \
nokeepalive ssl-unclean-shutdown \
downgrade-1.0 force-response-1.0
CustomLog logs/ssl_request_log \
“%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \”%r\” %b”
</VirtualHost>
```

El comando “apache2ctl -t” te puede ayudar para detectar posibles errores de sintaxis en la configuración del archivo del virtualhost

Reiniciamos Apache para que se apliquen los cambios.

```
systemctl restart httpd
```

Redireccionar el tráfico http a https

Podrías redirigir todo el tráfico de http (normal) a https (seguro) mediante una configuración en el virtual host para el puerto 80. Para ello creamos un fichero de configuración

```
sudo vi /etc/httpd/conf.d/non-ssl.conf
```

Añadimos las siguientes línas al fichero que hemos creado para redireccionar el tráfico

```
<VirtualHost *:80>
        ServerName www.example.com
        Redirect "/" "https://www.example.com/"
</VirtualHost>
```

Referencias

- https://www.digitalocean.com/community/tutorials/how-to-create-an-ssl-certificate-on-apache-for-centos-7
- http://terminus.ignaciocano.com/k/2011/06/14/configurar-apache-para-servir-conexiones-seguras/
- http://terminus.ignaciocano.com/k/2012/05/10/forzar-el-uso-de-sslhttps-de-un-directorio-en-apache2-mediante-htaccess-y-mod_rewrite/

From:
https://intrusos.info/ - **LCWIKI**

Permanent link:
https://intrusos.info/doku.php?id=aplicaciones:apache:ssl&rev=1503473469

Last update: **2023/01/18 14:12**