Apple y "goto fail", un fallo de seguridad en SSL/TLS

SSL (Secure Sockets Layer) traducido al español significa Capa de Conexiones Seguras. Es un protocolo que hace uso de certificados digitales para establecer comunicaciones seguras a través de Internet. Recientemente ha sido sustituido por TLS (Transport Layer Security) el cual está basado en SSL y son totalmente compatibles. Permite confiar información personal a sitios web, ya que los datos se ocultan a través de métodos criptográficos. Es utilizado ampliamente en bancos, tiendas en línea y cualquier tipo de servicio que requiera el envío de datos personales o contraseñas.

Cuando se realiza una conexión a un banco nadie debe ver los datos que se le envían (cifrado), pero además hay que asegurar que se está hablando con el banco y no con un impostor (autenticación). Para cifrar los datos, el servidor y el navegador tienen que intercambiar una clave. Para ello, el servidor tiene que enviar algunos parámetros iniciales que van a ir firmados digitalmente. Esa firma hay que verificarla, y ahí es donde entra en juego el código de Apple. [Codigo]

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{

/* ......  */
    hashOut.length = SSL_SHA1_DIGEST_LEN;
    if ((err = SSLFreeBuffer(&hashCtx)) != 0)
        goto fail;

    if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
/* ......  */

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;

}
```

Seguramente a priori resulta extraño el uso de los **goto** (aunque en este caso son una buena práctica), pero allí no radica el problema. El error se encuentra en la siguiente línea:

```
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
```

Por error alguien duplicó la línea y por no encontrarse el segundo **goto** dentro del bloque **if** siempre se ejecuta y casualmente, el código que se salta es el encargado de verificar la firma. Esa línea duplicada se salta la verificación de la firma. Y como el código que devuelve la función es siempre 0, cualquier programa que use esa función pensará que al firma es siempre correcta.

¿Cómo evitar que esto nos ocurra?

Tiene varias aspectos criticables el código, pero en particular nos interesa hacer énfasis en que esta situación no hubiera ocasionado ningún inconveniente en el caso de seguir la recomendación de utilizar las llaves de apertura y cierre.

```
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
{
        goto fail;
        goto fail;
}
```