Configurando SSL/TLS

Hacia la seguridad real…

Miguel Macías Enguídanos
miguel.macias@upv.es

RedIRIS

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

XXXIII Grupos de Trabajo
Cáceres, 06/06/2012

Índice

- Introducción
- Sedes electrónicas analizadas
- Herramientas
- Elementos
  - certificados digitales
  - entidad raíz
  - jerarquía de certificados
  - suites de seguridad
  - contenido web
- Referencias

Motivación

- Todos sabemos que es imprescindible utilizar SSL/TLS para garantizar la seguridad de las comunicaciones
- Pero ¿basta con tener SSL?
- Una web que utiliza el protocolo HTTPS ¿es segura?

- La tecnología está madura (estándares: 1995-2008) y es sencillo implementar un canal SSL/TLS
- Pero ¿lo estamos configurando correctamente?
- ¿Y si pensamos que tenemos un canal seguro pero es factible descifrar el tráfico?

Objetivos

- Los sitios web (en general) y las sedes electrónicas (en particular) deben garantizar un nivel alto de seguridad, independientemente del cliente
- Han de evitar, también, todos los ataques conocidos que puedan darse
  - por difíciles o improbables que puedan parecer
- Y, sobre todo, deben prestar la máxima atención para que el usuario no se encuentre nunca con ningún tipo de advertencia sobre la seguridad
- En este proyecto vamos a analizar el estado real de la seguridad web tomando como ejemplos las sedes electrónicas de instituciones afiliadas a RedIRIS

Buscando las sedes electrónicas…

- Instituciones afiliadas a RedIRIS
  - 462 dominios DNS
- Algunas instituciones aparecen más de una vez
  - ej.: xtec.cat, xtec.es
- Algunas instituciones están *relacionadas*
  - ej.: IMPIVA (impiva.gva.es), Generalitat (gva.es)
- ¿Se publican las sedes electrónicas en **sede.xxx.xx**?
  - 41 dominios existentes prefijando con sede.

Ni son todas las que están ni están todas las que son

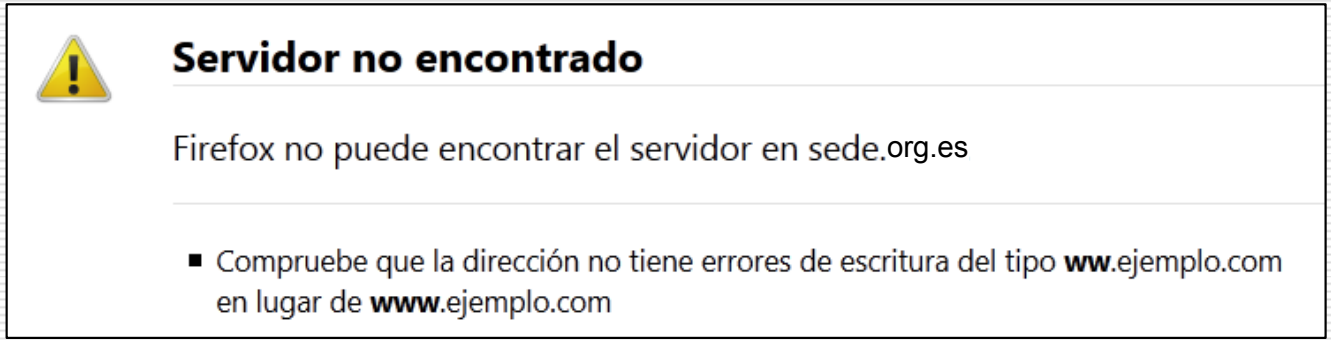
Ejemplo

- Centro Tecnológico de la Energía y del Medio Ambiente (CETENMA)
  - intentamos http://sede.ctmedioambiente.es

- ¿Debería revisar RedIRIS sus instituciones afiliadas y los dominios DNS asociados? ;-)

Ejemplo

- El usuario puede intentar lo que ya conoce…
  - intentamos http://sede.*organización.es*



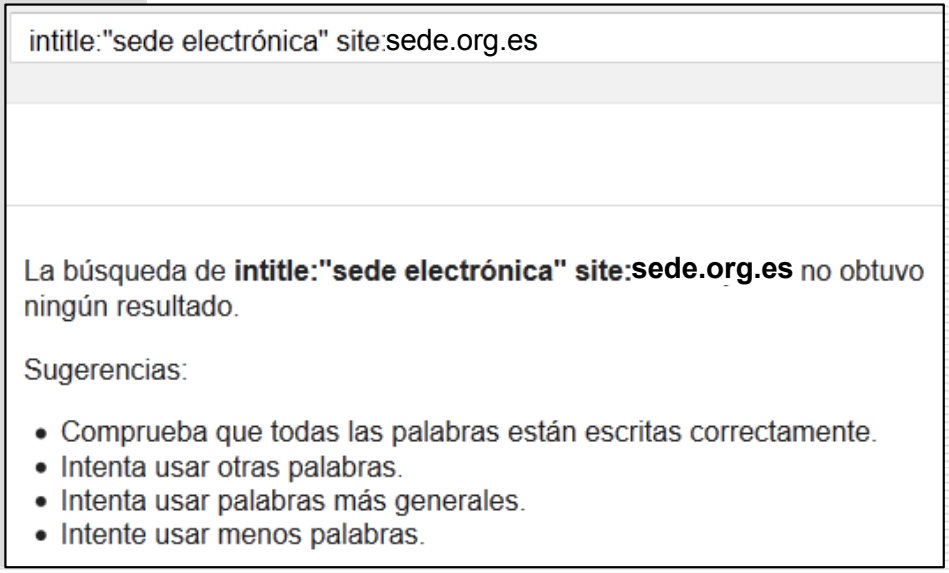
  - URL real: http://*otraCosa.organización.es* (portada)

Ejemplo real

- ¿Tiene sentido no publicar bajo **sede.xxx.xx**? ¿Ni siquiera establecer un alias?

¿Podemos buscar las sedes?

- El usuario intenta buscar la sede…



- ¿Tiene sentido que ninguna página de la sede contenga en el título "**sede electrónica**"?

Muestra

- Nos quedamos con 14 sedes electrónicas
  - esperemos que sea una muestra representativa

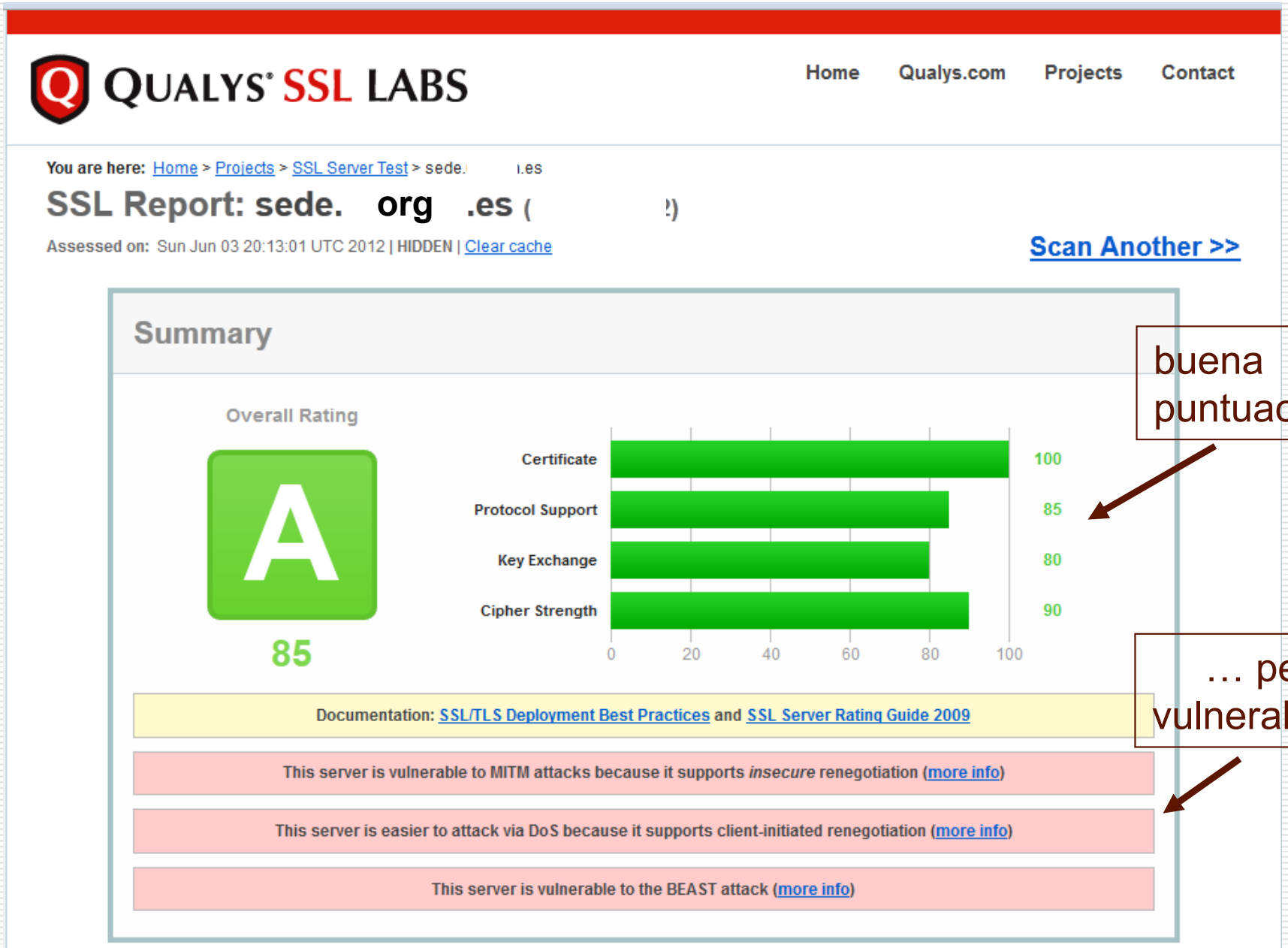
sede.xxxx01.es	sede.xxxx02.es*
sede.xxxx03.es*	sede.xxxx04.es
sede.xxxx05.es	sede.xxxx06.es
sede.xxxx07.es	sede.xxxx08.es
sede.xxxx09.es*	sede.xxxx10.es
sede.xxxx11.es	sede.xxxx12.es
sede.xxxx13.es	sede.xxxx14.es

* los dominios marcados realizan una redirección 302 a un dominio distinto

Herramientas: SSL Server Test

- SSL Server Test
  - https://www.ssllabs.com/ssltest/
- Herramienta gráfica sencilla, completa y vistosa
- Ventajas:
  - establece una puntuación total
  - ofrece información para corregir las vulnerabilidades
  - permanentemente actualizada
- Inconvenientes:
  - sólo valida servicios Web
  - no se pueden ampliar las entidades raíz

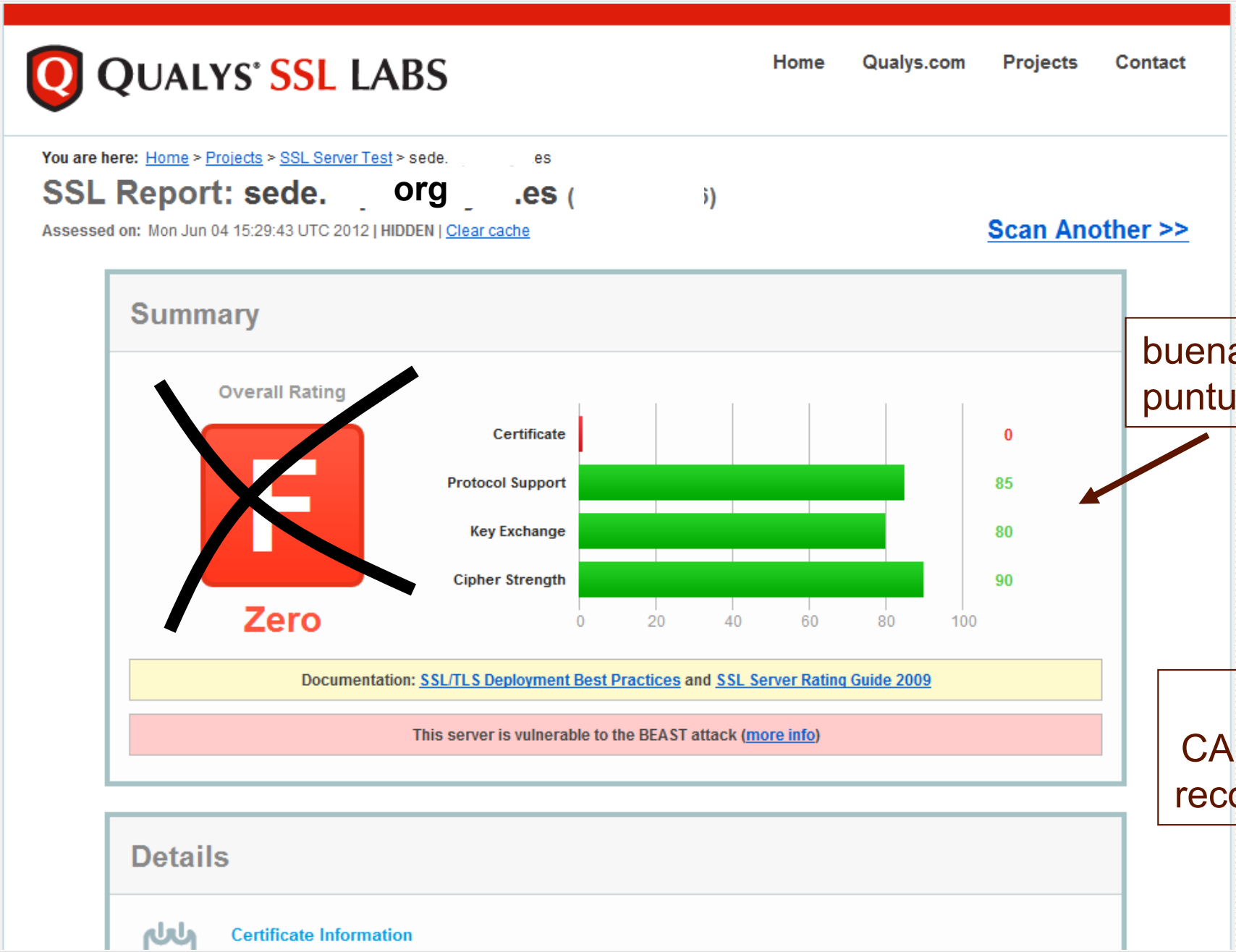
Herramientas: SSL Server Test



buena puntuación…

… pero vulnerable

Herramientas: SSL Server Test



buena puntuación…

… pero CA raíz no reconocida

Otras herramientas

- **OpenSSL**: herramienta completísima que permite su ejecución por línea de comandos o como librería
  - realmente es la única herramienta necesaria
  - la obtención de resultados requiere cierto esfuerzo
- **ScanSSL**: automatiza los análisis utilizando OpenSSL para interactuar con el destino
- **TLSSLed**: utiliza las dos anteriores para generar los informes sobre características y vulnerabilidades
- **SSLAudit**: no utiliza OpenSSL y verifica todas las posibles suites de seguridad

http://openssl.org/

http://sourceforge.net/projects/sslscan/

http://www.taddong.com/en/lab.html

http://www.g-sec.lu/tools.html

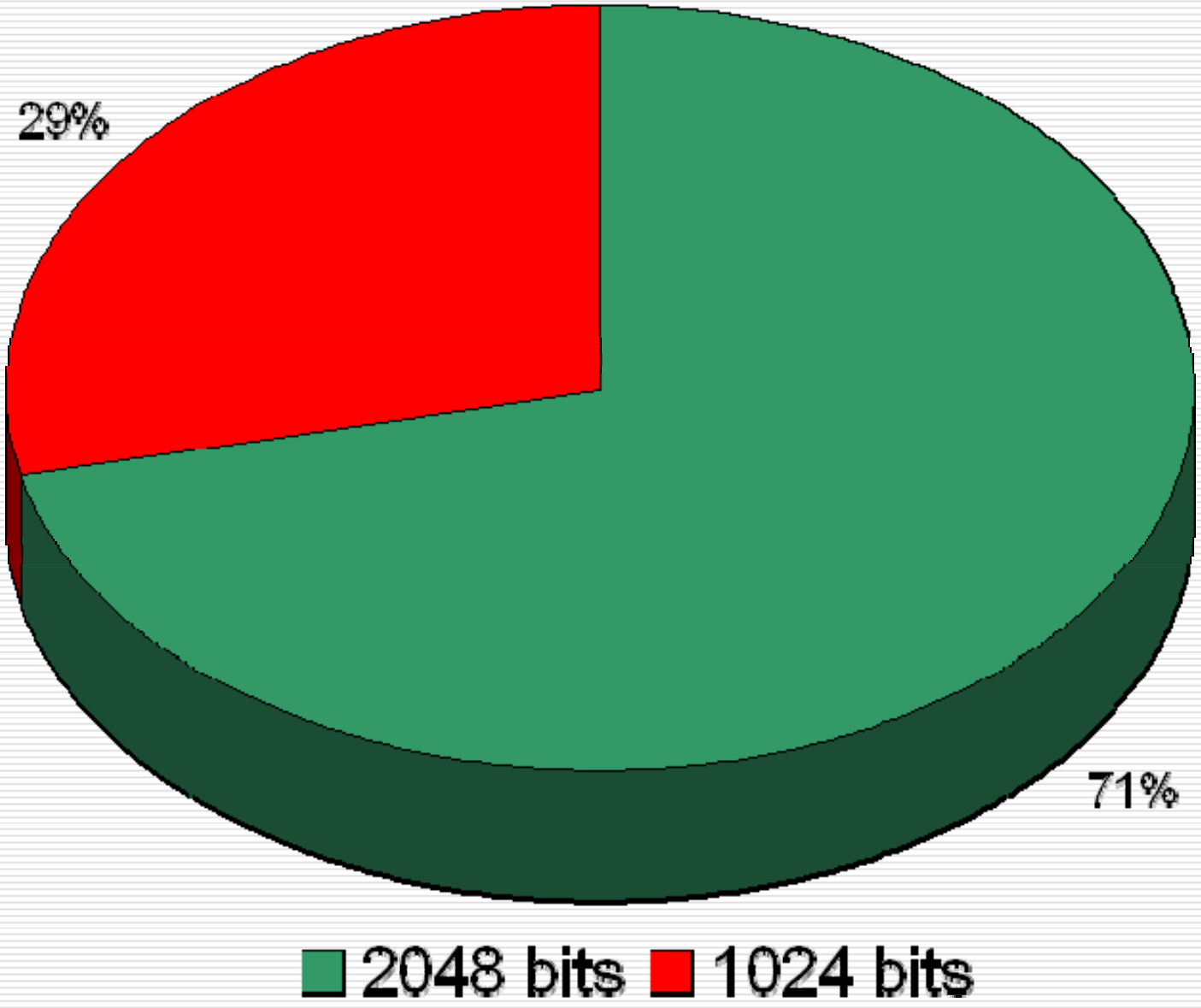
Elementos a tener en cuenta

- El certificado digital del servidor
  - nombres, tamaño de clave, vigencia
- La Entidad Raíz emisora del certificado y la jerarquía de entidades intermedias
  - soporte por parte de los navegadores, 'validez legal'
- El protocolo utilizado para el canal
  - versiones obsoletas, versiones problemáticas, soporte actual en clientes y servidores
- El mecanismo de cifrado
  - tamaño de clave, robustez
- El servicio web
  - cabeceras HTTP, cookies, contenido

Certificados digitales

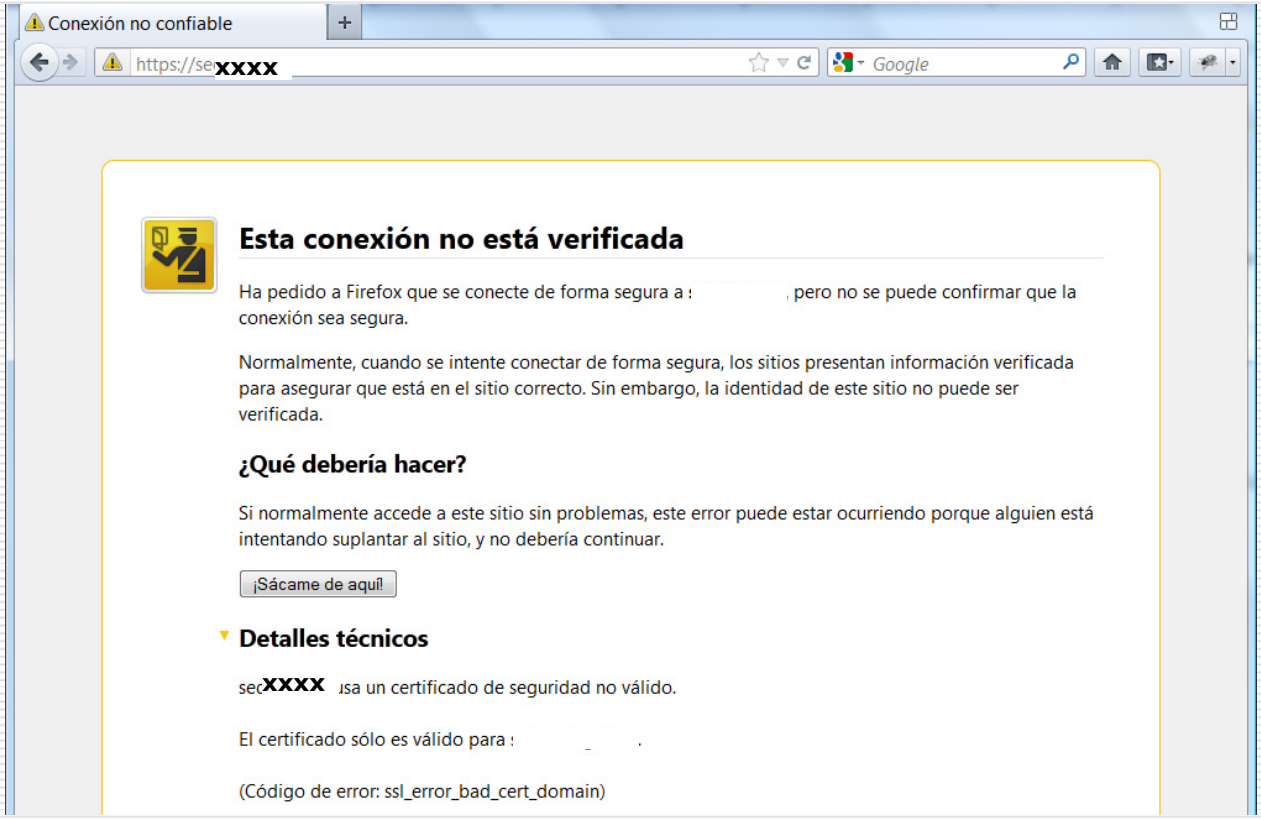
- El tamaño de clave recomendado es **2048 bits**
  - todavía se consideran seguras las claves de 1024 bits, pero los expertos recomiendan no solicitar nuevos certificados con este tamaño de clave
- El certificado ha de contener todos los nombres DNS que permitan acceder al servicio
  - en el mismo certificado todos los alias y en el mismo o distinto certificado los nombres que redirigen al servicio
- Protección adecuada de la clave privada
  - tanto en el servidor donde se está utilizando como en las copias de seguridad
- Se aconseja una renovación anual con claves nuevas

Tamaño de claves: estadísticas



Nombres en los certificados: ejemplo

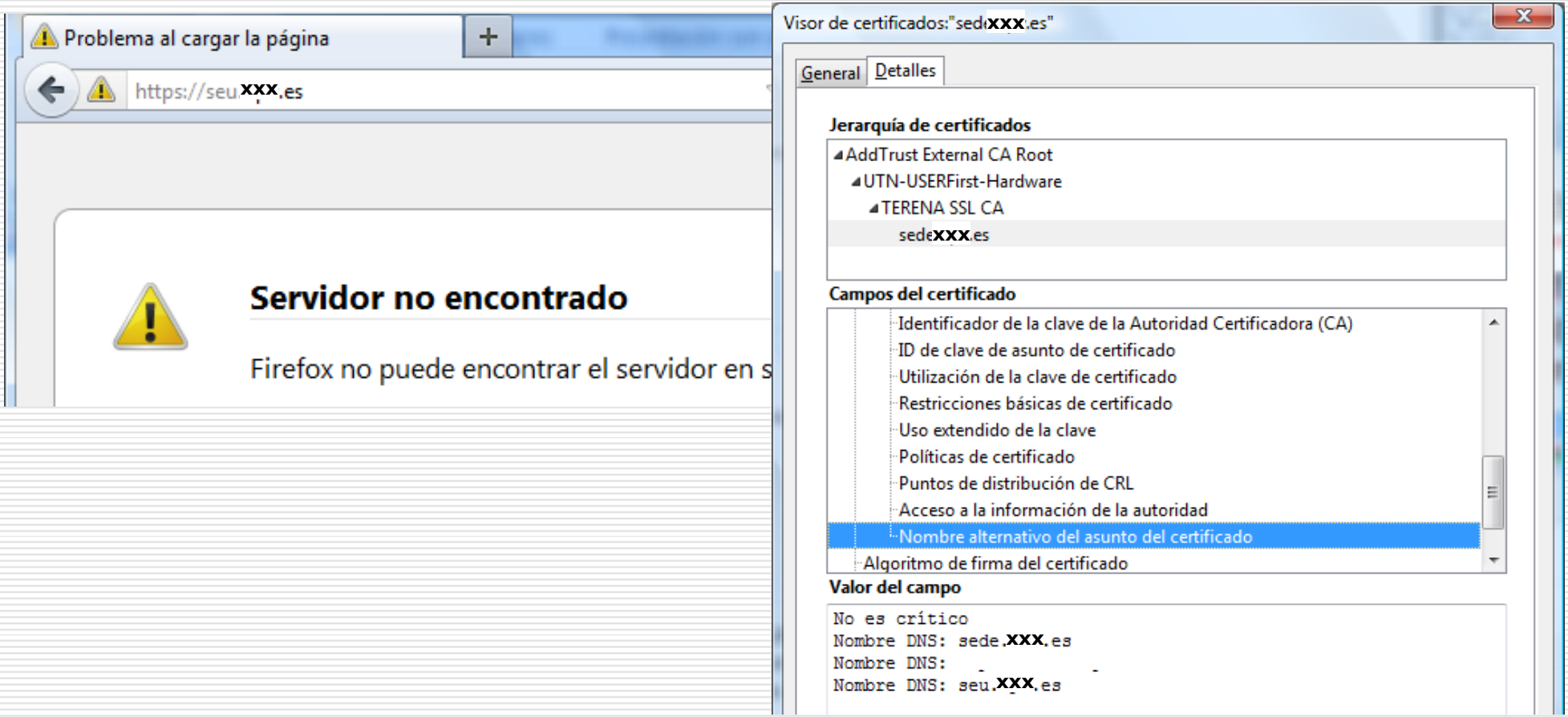
- Una institución que publica bajo sede…
  - HTTP: admite xxx.es, www.xxx.es, sede.xxx.es
  - HTTPS: ¡no admite ninguno de los anteriores!



  - http://sede.xxx.es redirige a https://sede.yyyy.es
  - ¿por qué no se ha incluido el nombre alternativo?

Nombres en los certificados: ejemplo

- Una institución que publica bajo sede…
  - el certificado admite como nombres DNS **sede.xxx.es** (en castellano) y **seu.xxx.es** (en otra lengua oficial)

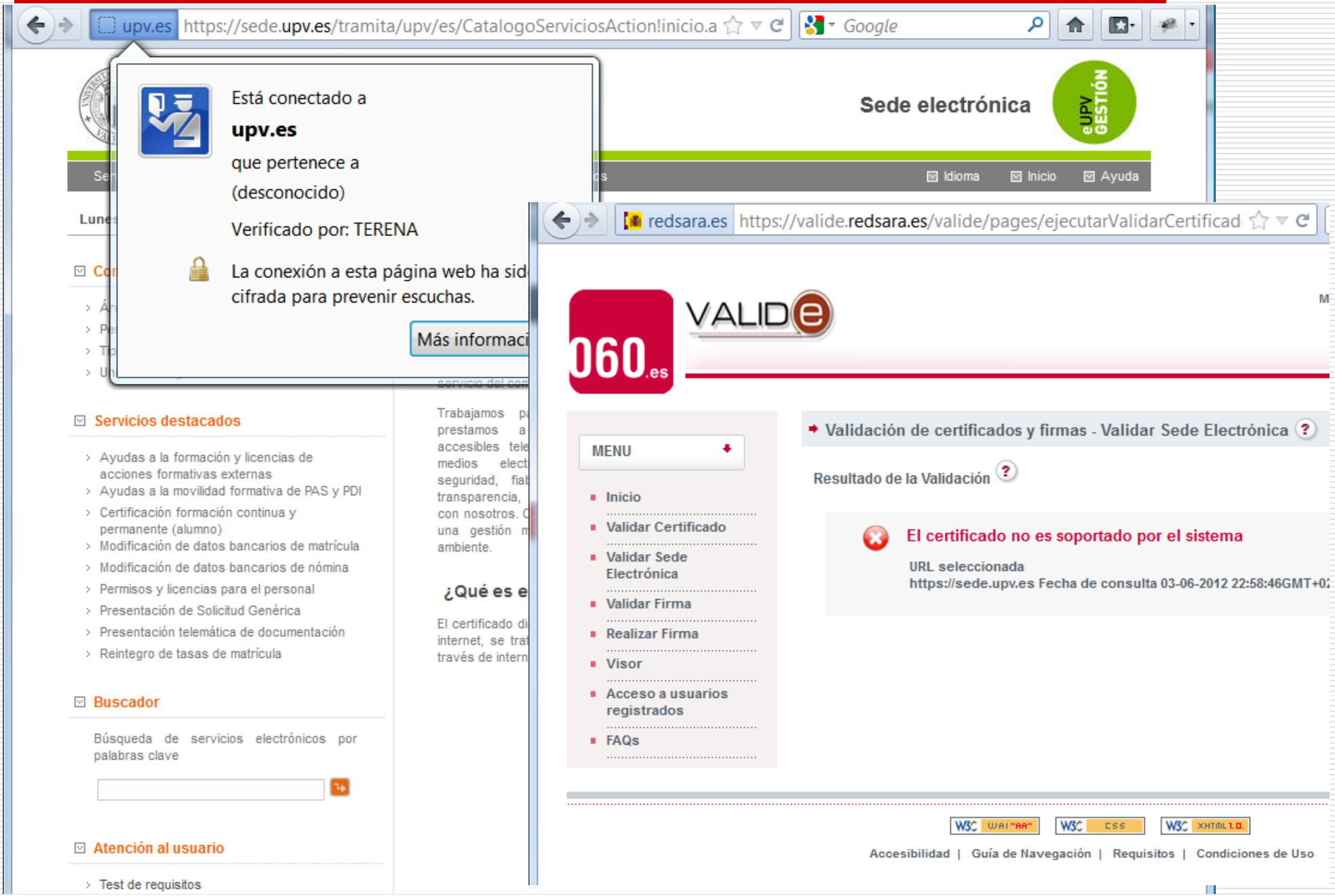


  - ¿no sería conveniente crear la entrada DNS para **seu**?

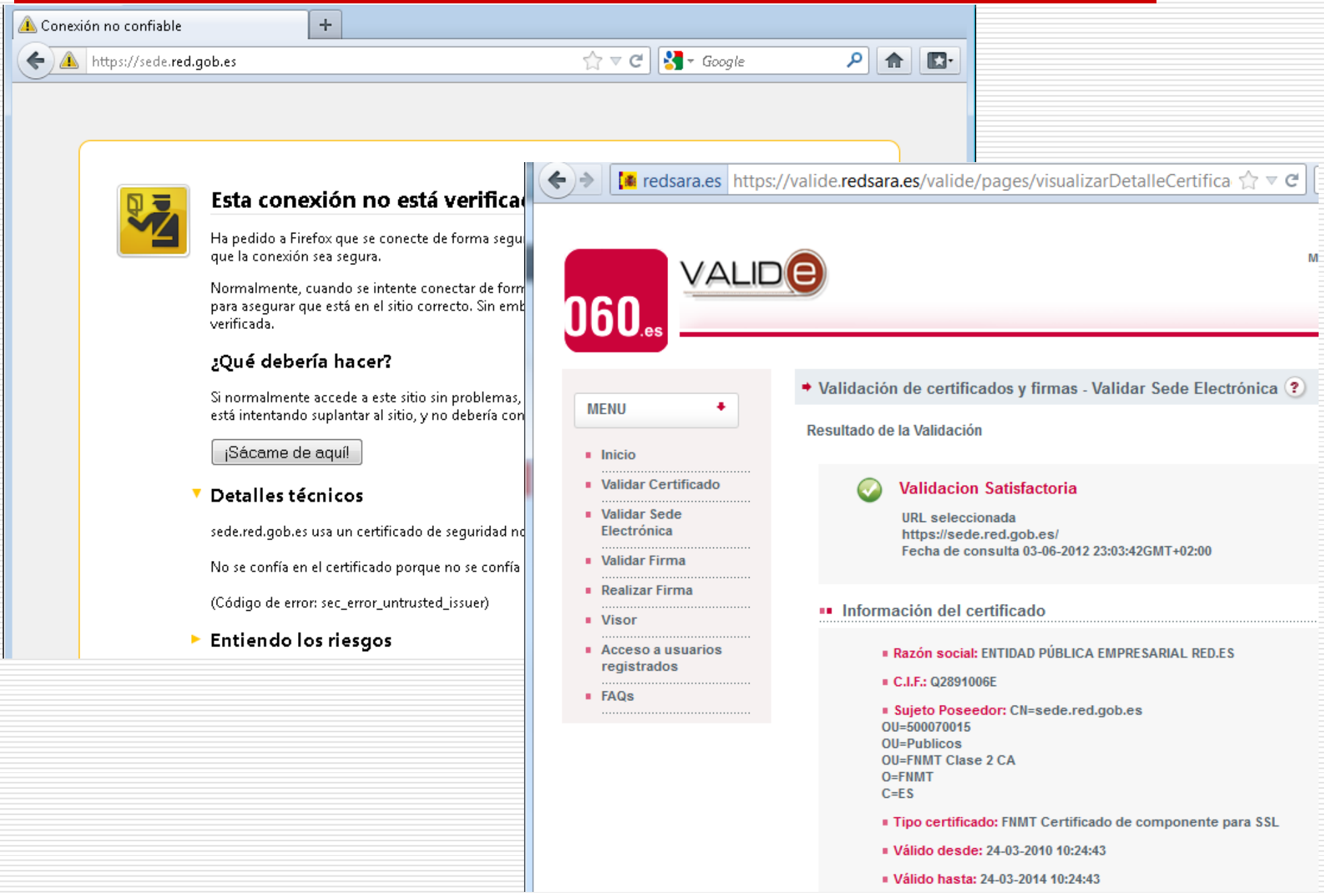
Entidades raíz

- La ley 11/2007 no establece ningún requerimiento sobre los certificados utilizados por las sedes
  - El RD 1671/2009 impone unas características en los certificados para la Administración General del Estado
- Básicamente tenemos 2 opciones:
  - Certificados obtenidos a través de SCS (Comodo)
    - gratuitos y ampliamente distribuidos
    - no aceptados por la herramienta Valide: https://valide.redsara.es/
  - Certificados de entidades inscritas en el registro del Ministerio de Industria
    - la herramienta Valide los da por buenos
    - generalmente tienen una distribución muy limitada (ej.: FNMT)
    - aunque algunos ya son reconocidos por Firefox (ej.: ACCV, a partir de la versión 8)

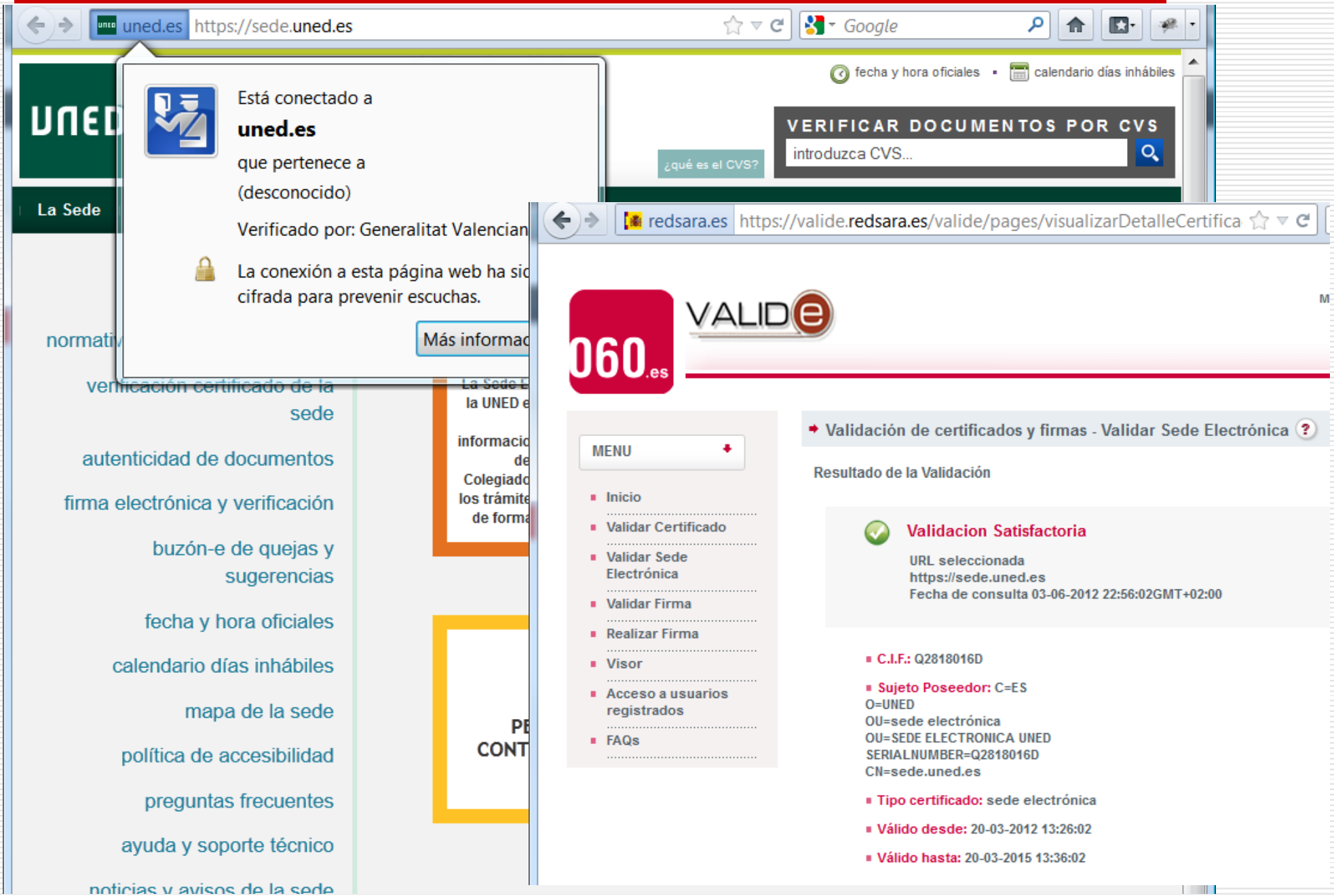
Entidades raíz: ejemplo 1 (SCS)



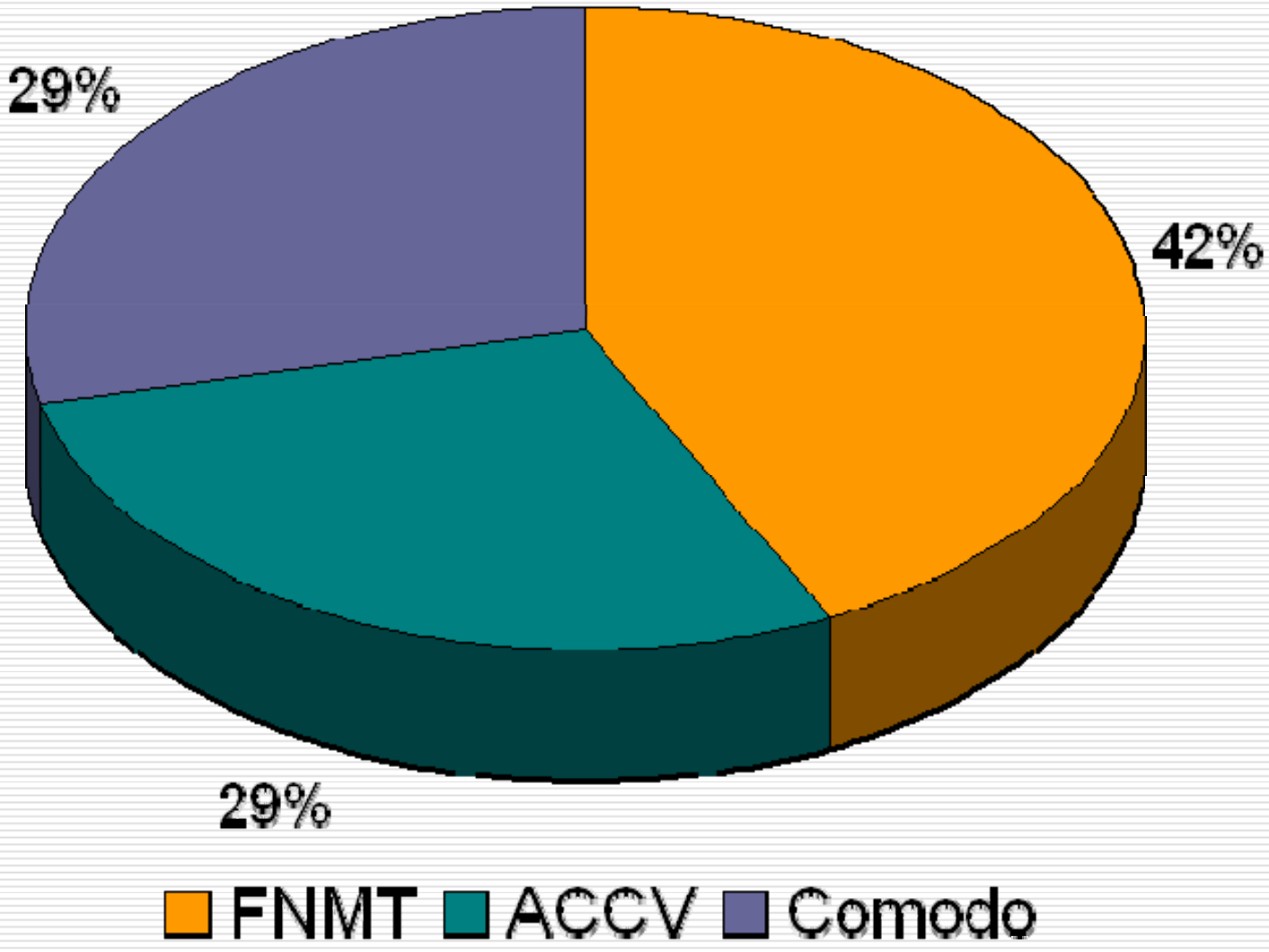
Entidades raíz: ejemplo 2 (FNMT)



Entidades raíz: ejemplo 3 (ACCV)

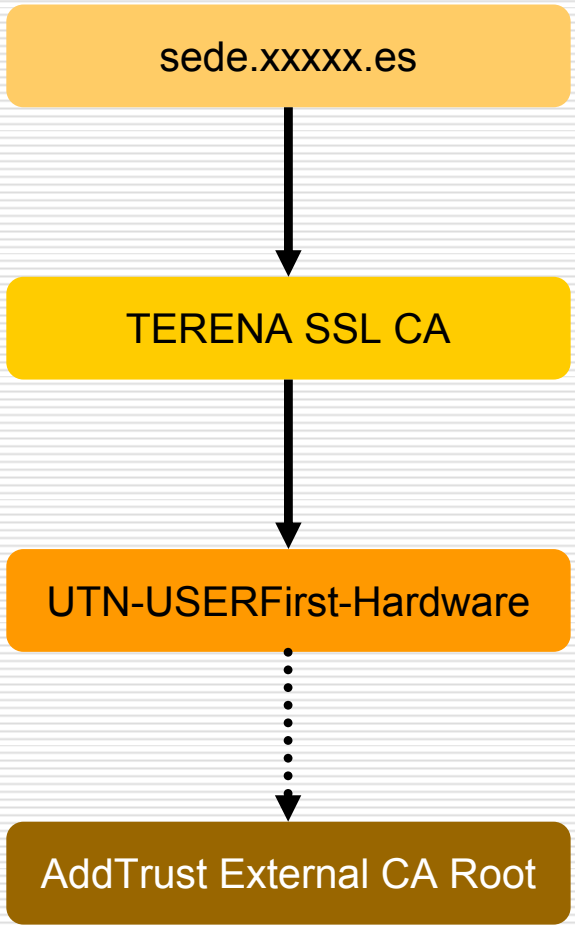


Entidades raíz: estadísticas



Jerarquía de certificados

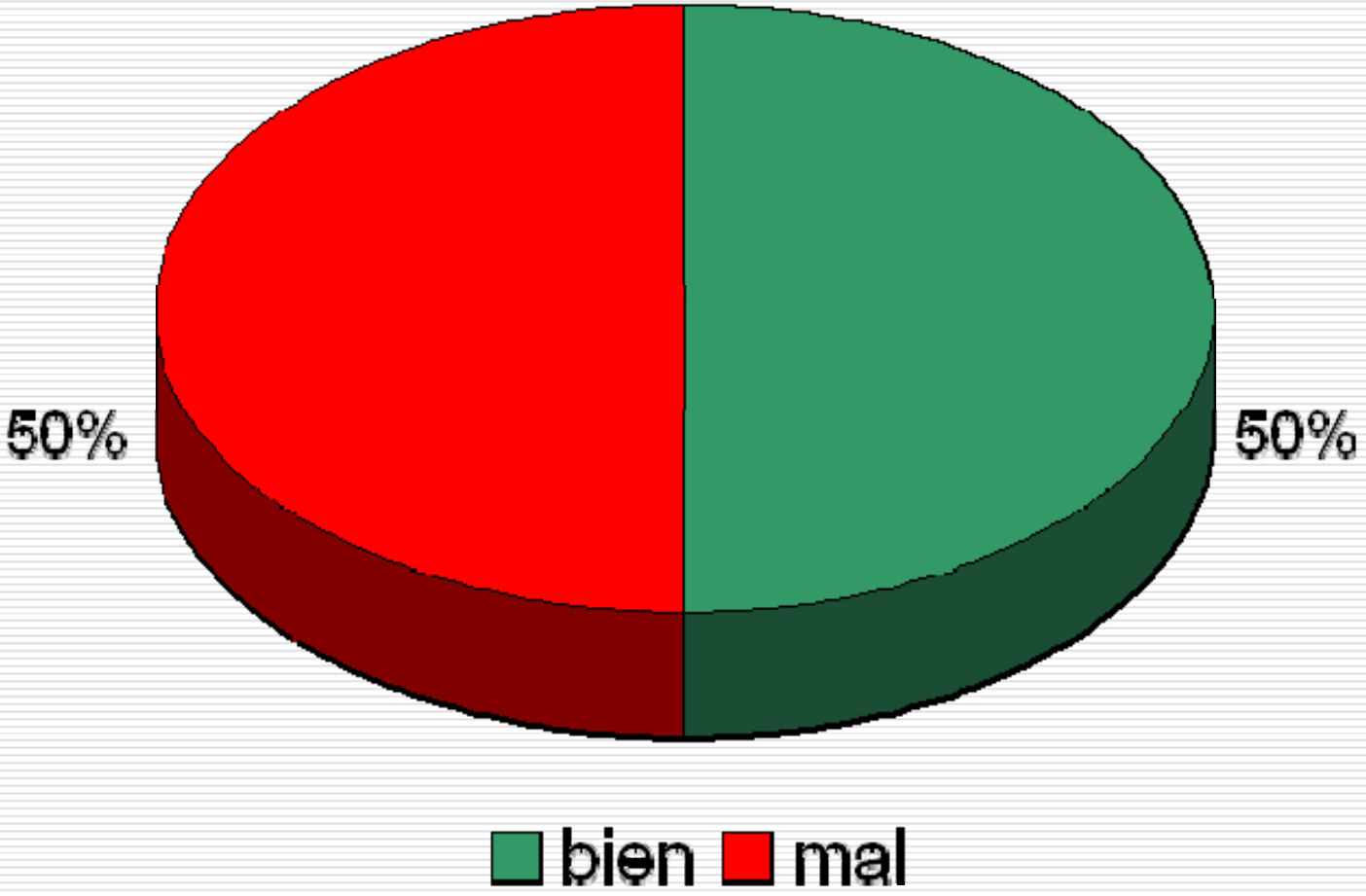
- El servidor ha de entregar su certificado digital y los certificados digitales de todas las entidades intermedias de la jerarquía
- Las entidades intermedias han de entregarse en orden: primero la que firma el certificado del servidor y subiendo hasta la entidad raíz
- La entidad raíz puede no entregarse, ya que el cliente ha de tenerla en su propio almacén
- Cada navegador reacciona de una manera ante un fallo de jerarquía



Jerarquía de certificados: ejemplo

- Jerarquía no entregada correctamente

Jerarquía de certificados: estadísticas

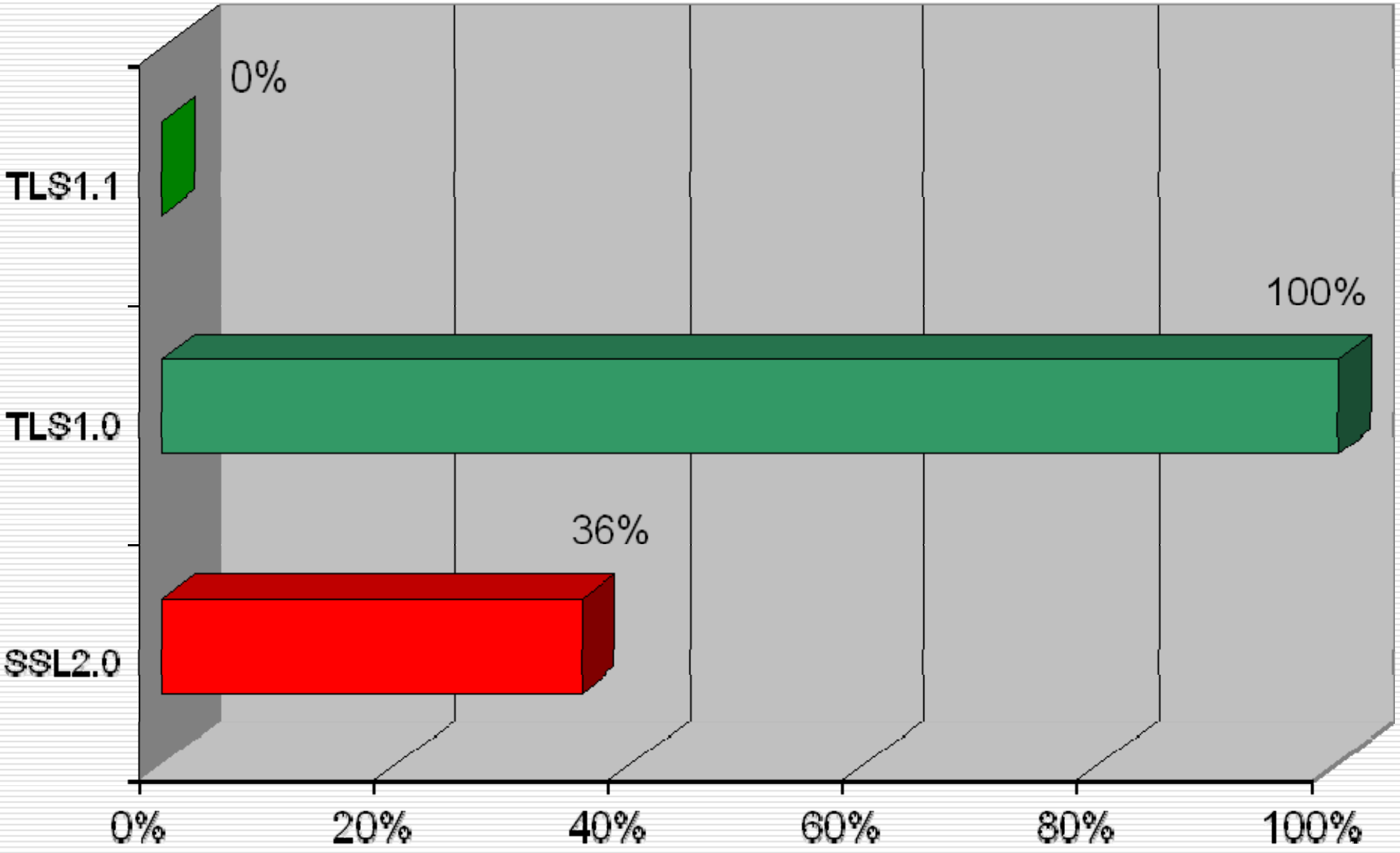


Protocolos

- **SSL** (*Secure Sockets Layer*) es el protocolo original desarrollado en 1995 por Netscape
- La versión 2 está 'rota' y no debería usarse nunca
- La versión 3 es prácticamente similar al protocolo **TLS** (*Transport Layer Security*) v 1.0, el sucesor
- TLS tiene dos versiones posteriores (1.1 y 1.2) con mejoras respecto a la anterior
- Se aconseja configurar:
  - **SSL 3.0 / TLS 1.0** (seguro* y con muy amplio soporte)
  - **TLS 1.1** (muy seguro y con bastante soporte)
  - **TLS 1.2** (muy seguro y poco soportado)
  - en ningún caso SSL 2.0

** ver cifrado*

Protocolos: estadísticas



Suites criptográficas

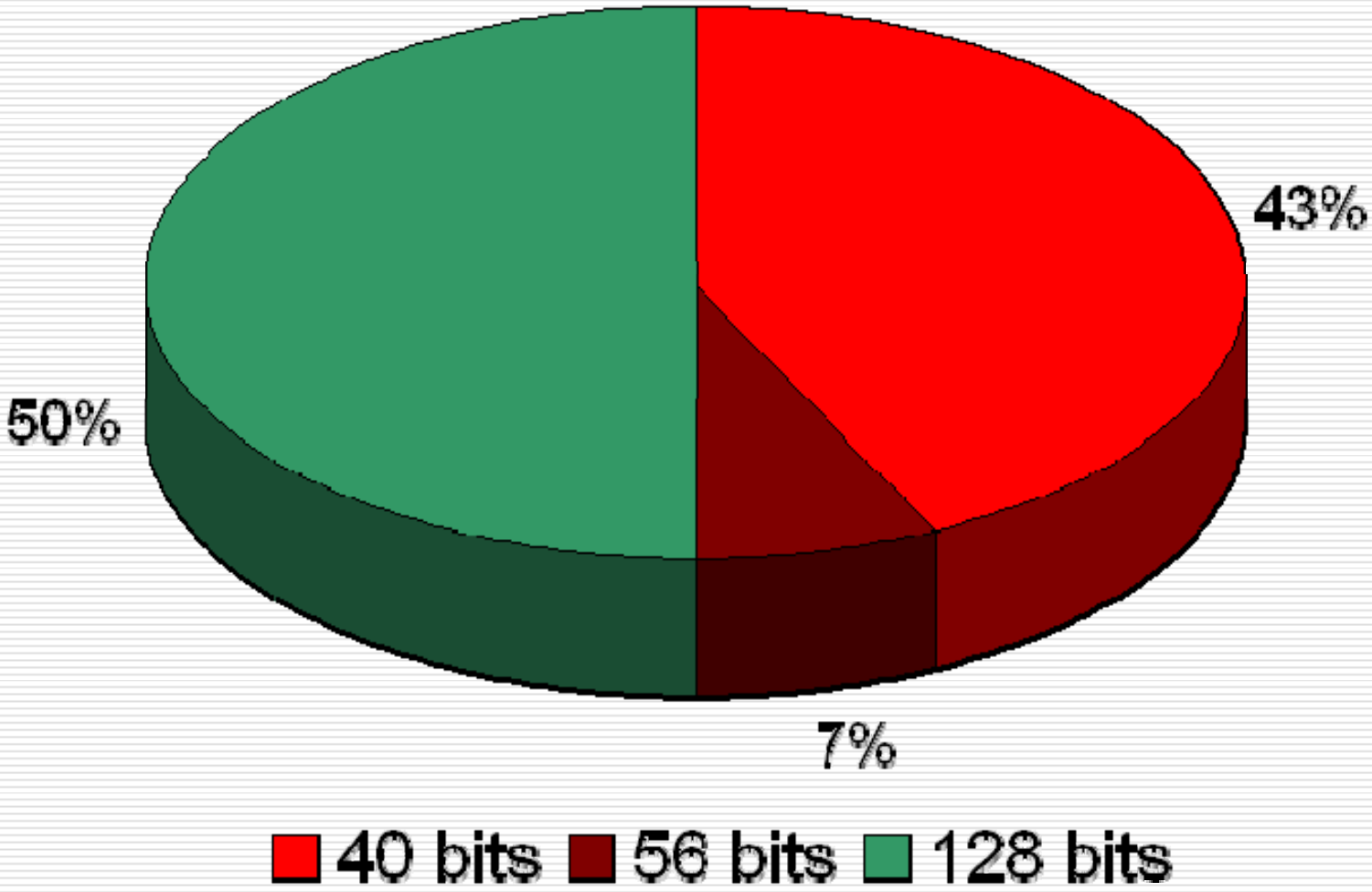
- Las sesiones SSL manejan distintos aspectos:
  - **intercambio de claves**: permite generar claves secretas
  - **autenticación**: se permite el uso de certificados digitales para autenticar sólo al servidor, a ambos o a ninguno
  - **cifrado**: desde nulo a DES, AES, RC4, …
  - **resumen**: código MAC para verificar la integridad
- Una suite de seguridad especifica los parámetros empleados para cada uno de estos aspectos
  - ej.: **TLS_RSA_WITH_AES_256_CBC_SHA256**

TLS	*protocolo*
RSA	*int. claves*
AES_256_CBC	*cifrado*
SHA256	*MAC*

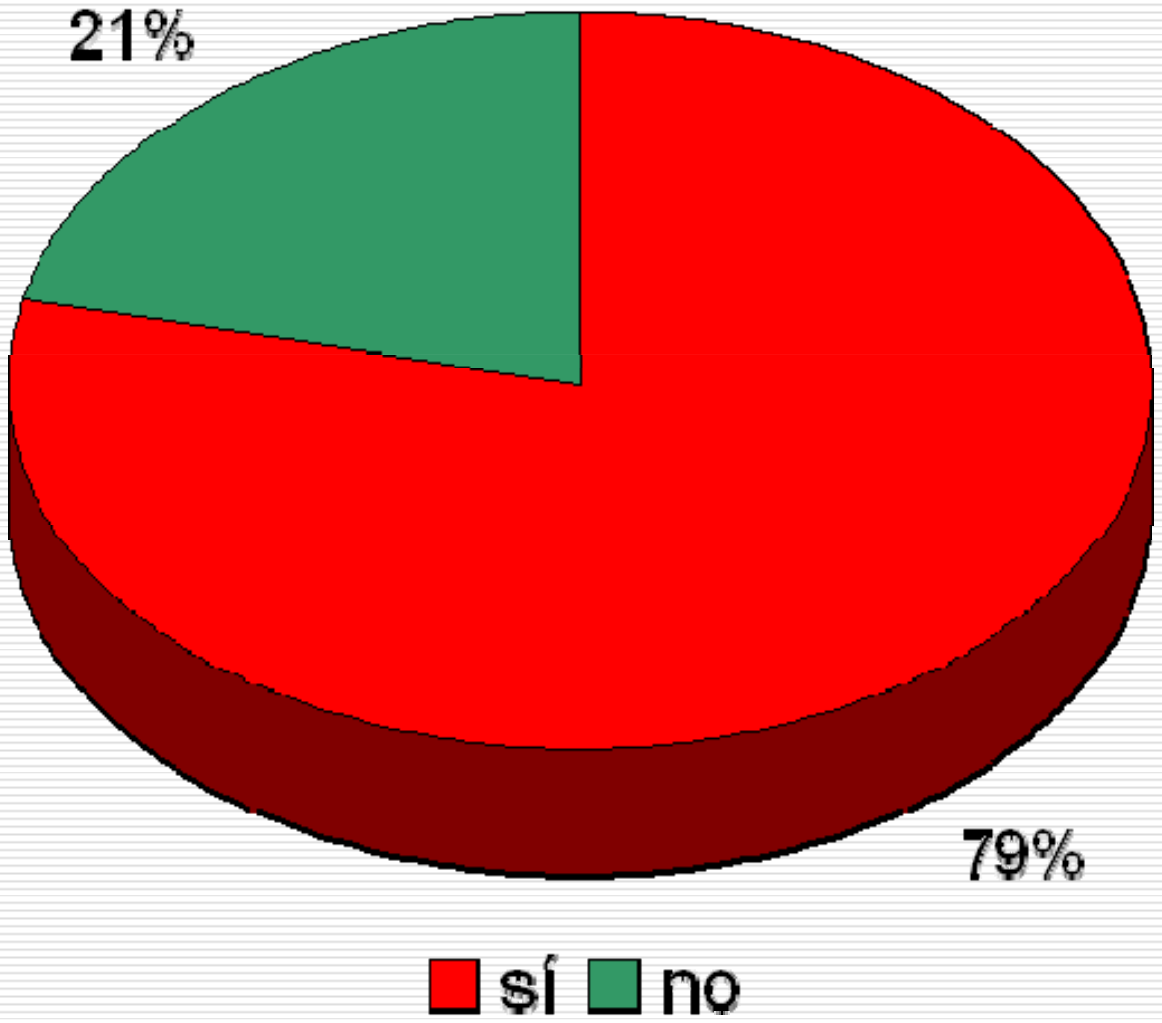
Cifrado

- Tiene que utilizarse cifrado y el tamaño mínimo de clave ha de ser de **128 bits**
- Cualquier cifrado con un tamaño de clave menor puede romperse con los medios al alcance de cualquiera de nosotros
- Debido al ataque **BEAST** (dado a conocer en 2011, a partir de una vulnerabilidad de 2004, corregida en 2006), en SSL 3.0 / TLS 1.0 debería utilizarse únicamente RC4
  - TLS 1.1 no es vulnerable a BEAST
  - el ataque requiere un cifrado por bloques y un cierto 'control' sobre la aplicación web

Cifrado (tamaño de clave): estadísticas



Cifrado: vulnerabilidad a BEAST

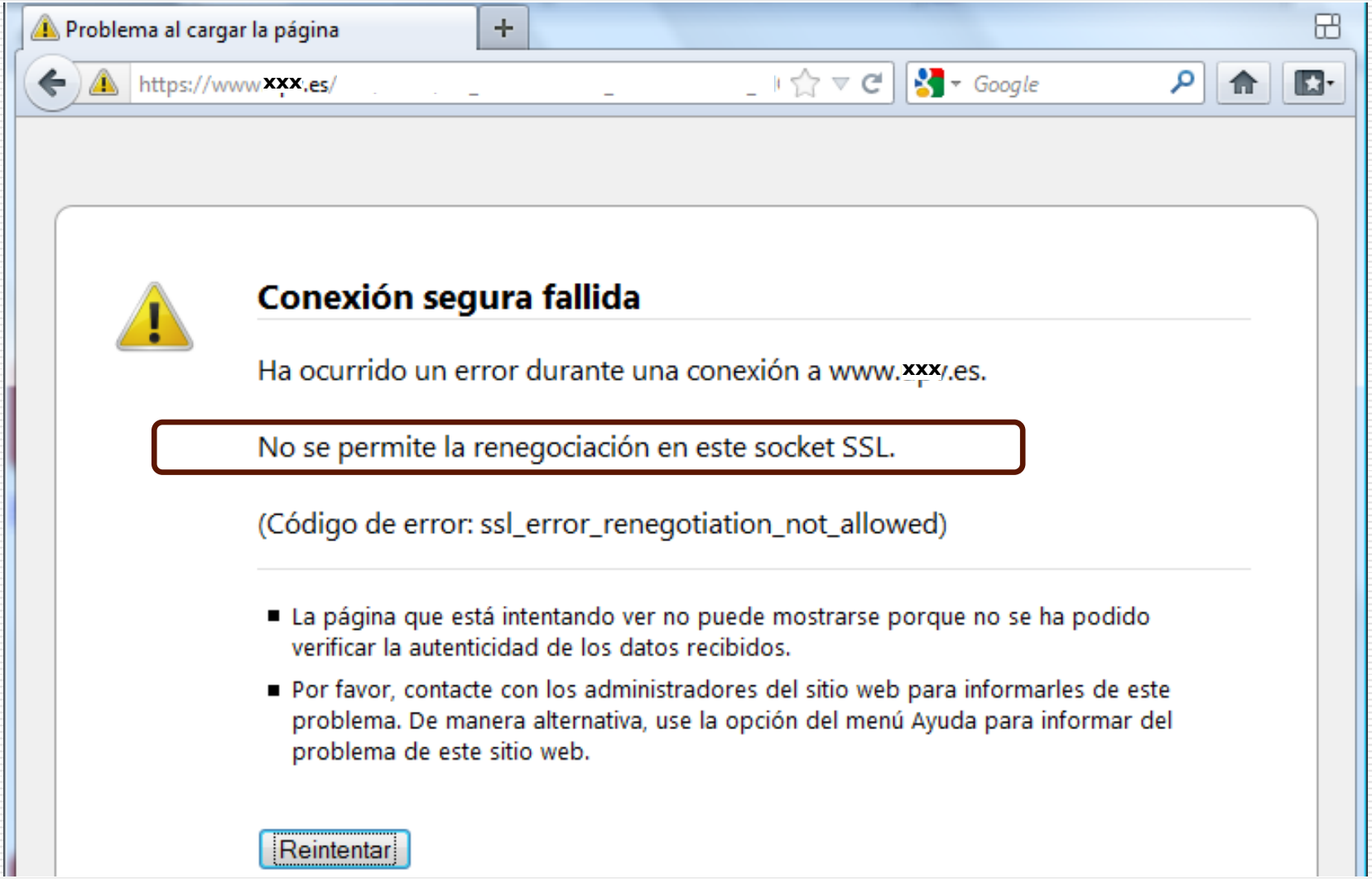


Renegociación

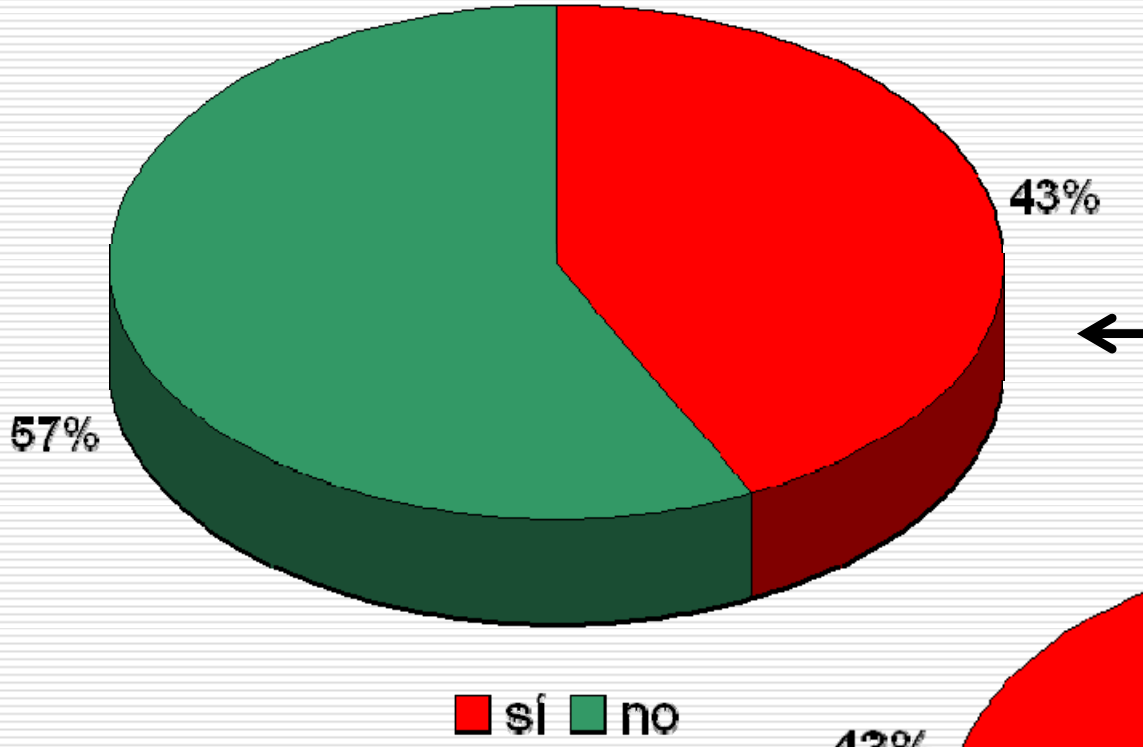
- El canal SSL puede pararse momentáneamente y permitir que los extremos renegocien la sesión
- Si el cliente es el que inicia la renegociación, podemos tener problemas
  - no es necesario y hay ataques descritos
- Sólo el servidor ha de poder iniciar la renegociación
- En el 2009 se descubrió un problema en la renegociación de TLS, corregido en la RFC 5746
  - algunos navegadores protegen a sus usuarios impidiendo las conexiones a servidores Web que no la implementan
- La renegociación tiene que permitirse, pero sólo de manera segura

Renegociación: ejemplo

- Institución que admite renogociación no segura



Renegociación: estadísticas



Renegociación iniciada por el cliente

Renegociación segura

HSTS

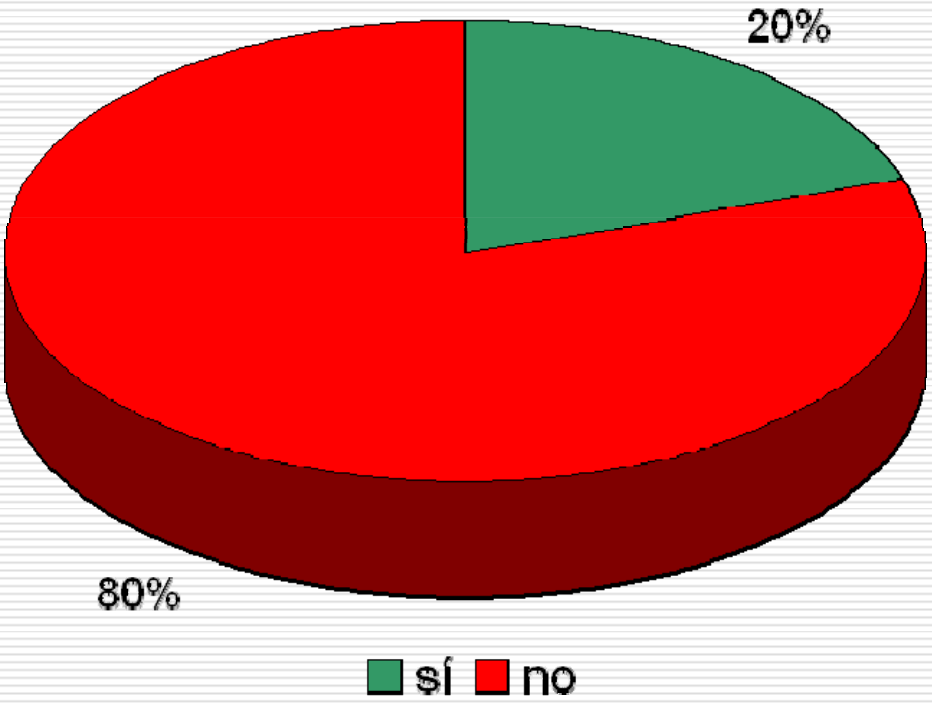
- **HSTS** *(HTTP Strict Transport Security)* es un mecanismo que permite 'exigir' el uso de HTTPS
  - se trata de una cabecera HTTP
- El propio navegador se encargaría de utilizar siempre HTTPS, a partir de la primera visita
- Está en versión borrador y no está soportado por todos los navegadores
  - pero aún así es conveniente implementarlo: no tiene coste y puede proteger a algunos usuarios
- Permite evadir ciertos tipos de ataques y evita errores de nuestro propio sitio web (por ejemplo: contenido mixto)

HSTS: estadísticas

Cookies

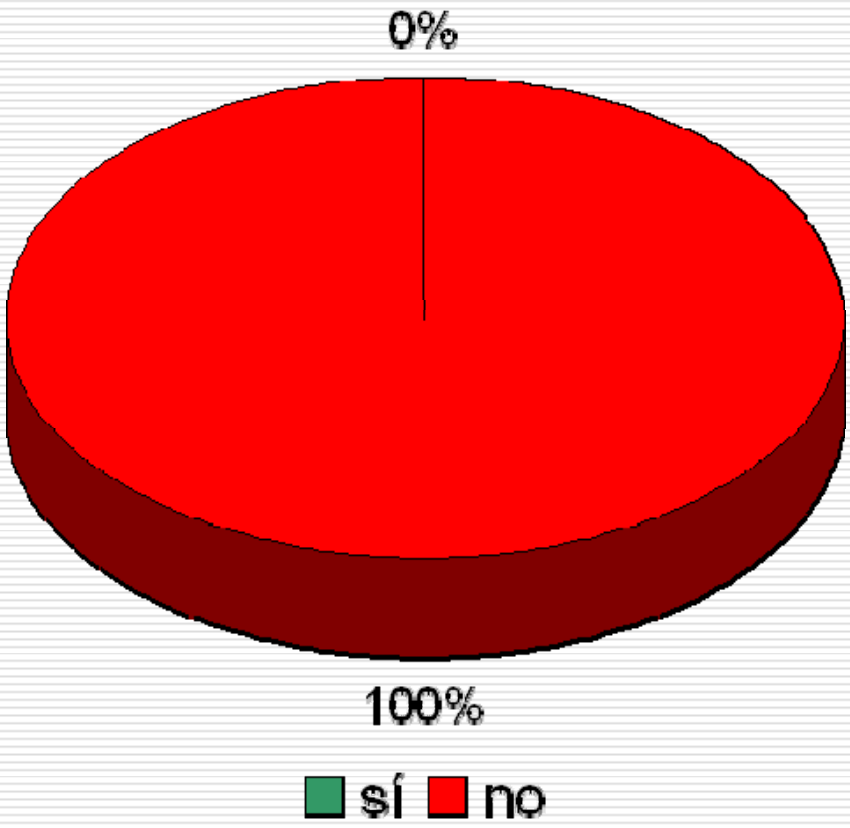
- Las cookies permiten gestionar las sesiones y han de protegerse tanto como las credenciales
  - un atacante puede suplantar a la víctima robándole (o adivinando) sus credenciales o la cookie de sesión
- Cuando se establece una cookie se puede marcar con dos niveles de protección:
  - **HTTPonly**: la cookie no será accesible mediante JavaScript, lo que evitará robos a través de XSS, etc.
  - **Secure**: la cookie sólo se enviará a través de un canal seguro, lo que evitará inspecciones, etc.
- En este estudio se han buscado las cookies de sesión en la carga inicial de la sede
  - puede que algunas sedes no la establezcan todavía

Cookies: estadísticas



cookies seguras

cookies HTTPonly



Contenido mixto

- Toda la sede electrónica debería servirse por HTTPS
  - todas las páginas y todos los elementos de las páginas
- Si alguna página está accesible por HTTP, el usuario será vulnerable a ataques de tipo *Man In The Middle*
- Si dentro de una página se cargan elementos mediante el protocolo HTTP, el usuario se encontrará con una 'advertencia' de seguridad
  - algunos navegadores muestran la advertencia explícitamente y le preguntan al (pobre) usuario
  - algunos navegadores utilizan la opción más segura (e informan al usuario y le permiten cambiar la opción)
  - algunos navegadores utilizan la opción menos segura (e informan, indirectamente, al usuario)

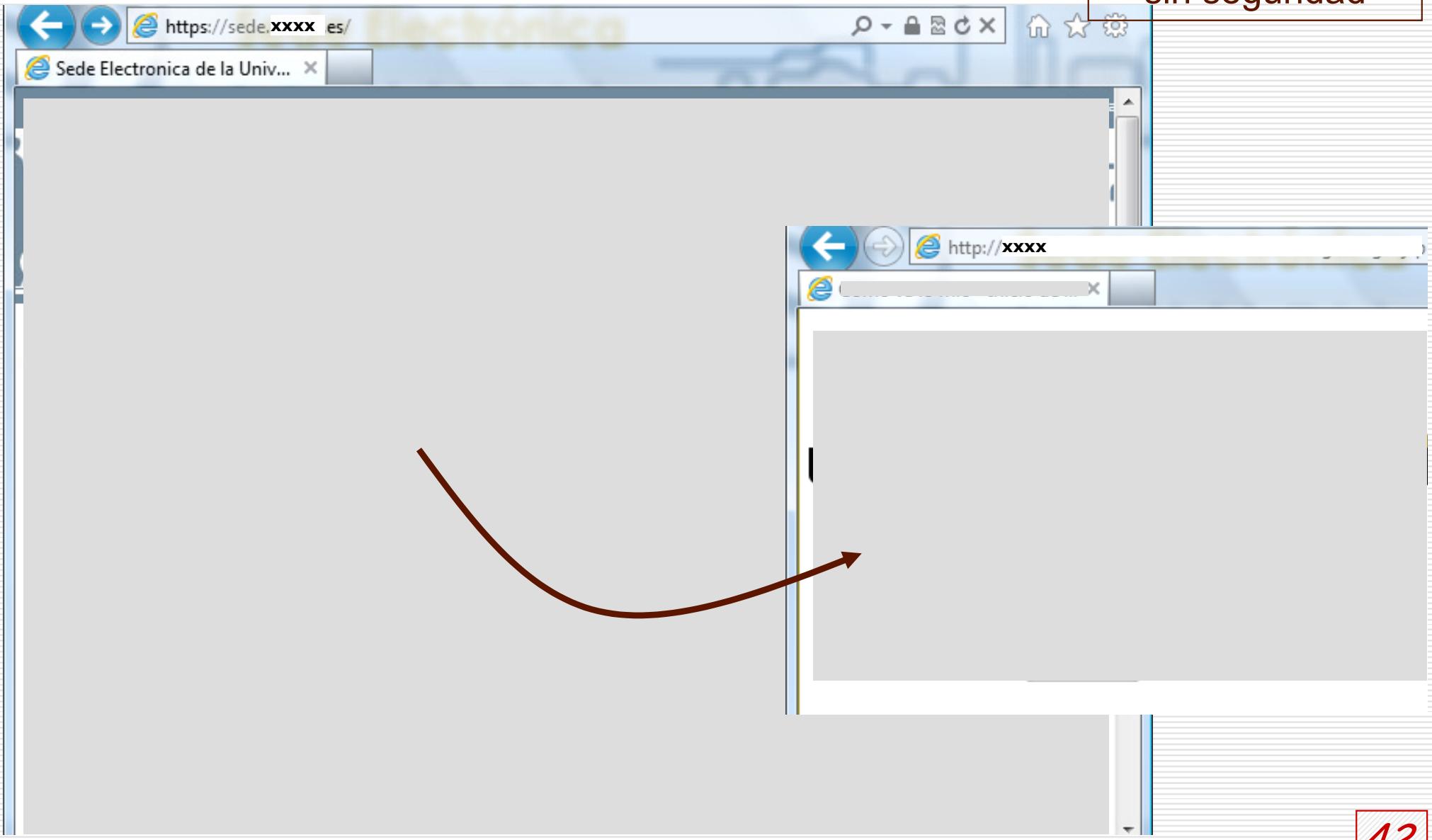
Contenido mixto: ejemplo

- Institución con contenido mixto

elementos seguros
y no seguros
en la misma página

Contenido mixto: ejemplo

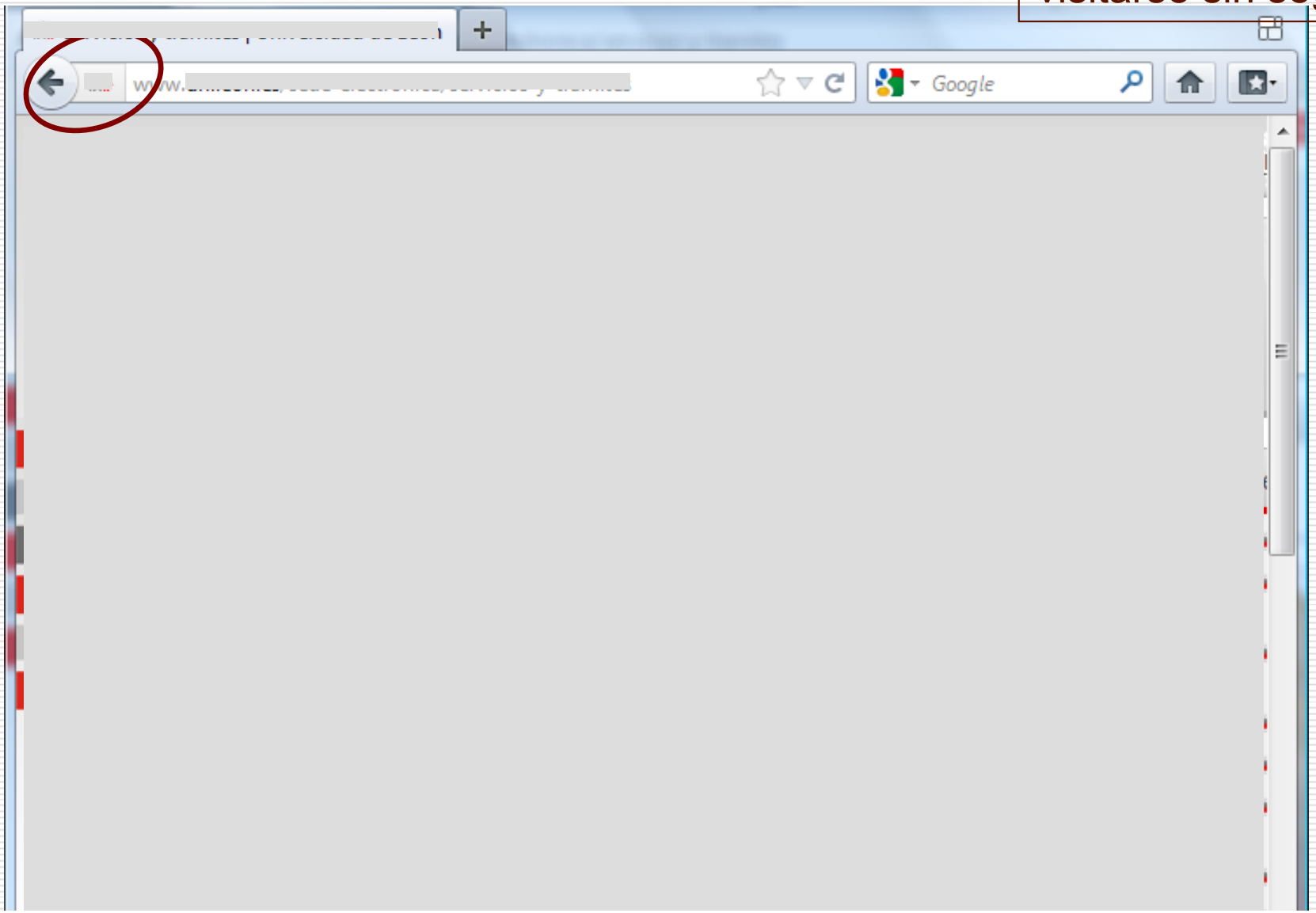
- Institución con contenido mixto



Contenido mixto: ejemplo

- Institución con contenido mixto

páginas que pueden visitarse sin seguridad



Resumen

- Configura tu servidor web teniendo en cuenta:
  - **certificado**: clave mínima de 2048 bits y conteniendo todos los nombres DNS que llevan al servicio
    - protegiendo la clave privada correctamente en todo momento
  - **jerarquía** de entidades intermedias: se entregan todas al cliente y en el orden 'natural'
  - **protocolos**: implementa TLS 1.1 y SSL 3.0/TLS 1.0
    - también es conveniente ofrecer TLS 1.2 (para el futuro)
  - **cifrado**: utiliza claves con un tamaño mínimo de 128 bits
    - si el protocolo es SSL 3.0/TLS 1.0 utiliza RC4, nunca por bloques
  - **renegociación**: siempre segura e iniciada por el servidor
  - sitio **web**: todas las páginas y elementos por HTTPS
    - genera la cabecera HSTS
  - **cookies**: márcalas como seguras y HTTPonly

Referencias

- SSL/TLS Deployment Best Practices
  - Ivan Ristic; Qualys SSL Labs
  - https://www.ssllabs.com/
- SSL/TLS Hardening and Compatibility report 2011
  - Thierry ZOLLER; G-SEC
  - http://www.g-sec.lu/tools.html

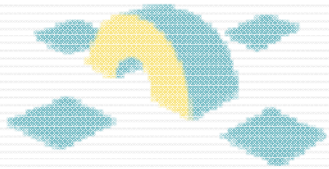
Configurando SSL/TLS

GRACIAS POR

Hacia la seguridad real...

TU ATENCIÓN

Miguel Macías Enguídanos
miguel.macias@upv.es



IRIS

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

XXXIII Grupos de Trabajo
Cáceres, 06/06/2012