

# SEGURIDAD DE LA INFORMACIÓN

## TEMA 5 – PARTE A

### **SEGURIDAD EN REDES TCP/IP**

# Índice del tema

- Seguridad en la Capa de Transporte
- Seguridad en la Capa de Internet
- Firewalls en Redes
- Seguridad en la Capa de Acceso a Red: el caso de las Redes Inalámbricas

# SEGURIDAD EN LA CAPA DE TRANSPORTE





- Existen diversos tipos de amenazas que pueden aparecer al usar la Web, y los posibles puntos de ataque son:
  - el cliente Web,
  - el servidor Web,
  - la propia información entre cliente y servidor

	Threats	Consequences	Counter measures
<b>Integrity</b>	<ul style="list-style-type: none"><li>•Modification of user data</li><li>•Trojan horse browser</li><li>•Modification of memory</li><li>•Modification of message traffic in transit</li></ul>	<ul style="list-style-type: none"><li>•Loss of information</li><li>•Compromise of machine</li><li>•Vulnerability to all other threats</li></ul>	Cryptographic checksums
<b>Confidentiality</b>	<ul style="list-style-type: none"><li>•Eavesdropping on the Net</li><li>•Theft of info from server</li><li>•Theft of data from client</li><li>•Info about network configuration</li><li>•Info about which client talks to server</li></ul>	<ul style="list-style-type: none"><li>•Loss of information</li><li>•Loss of privacy</li></ul>	Encryption, web proxies
<b>Denial of Service</b>	<ul style="list-style-type: none"><li>•Killing of user threads</li><li>•Flooding machine with bogus requests</li><li>•Filling up disk or memory</li><li>•Isolating machine by DNS attacks</li></ul>	<ul style="list-style-type: none"><li>•Disruptive</li><li>•Annoying</li><li>•Prevent user from getting work done</li></ul>	Difficult to prevent
<b>Authentication</b>	<ul style="list-style-type: none"><li>•Impersonation of legitimate users</li><li>•Data forgery</li></ul>	<ul style="list-style-type: none"><li>•Misrepresentation of user</li><li>•Belief that false information is valid</li></ul>	Cryptographic techniques



01-10-20

**Firefox attacks: Homeland Security urges all users to update browsers immediately in rare warning**

**The issue is this:** Firefox versions for desktop older than the just-patched version contain a critical vulnerability that could allow an attacker to take control of a user's entire operating system—whether they use Windows or Mac. More alarming, the vulnerability is already being exploited in the wild, thus Homeland Security stepping in with the urgent plea for users to upgrade.

[Image: courtesy of Mozilla Foundation]

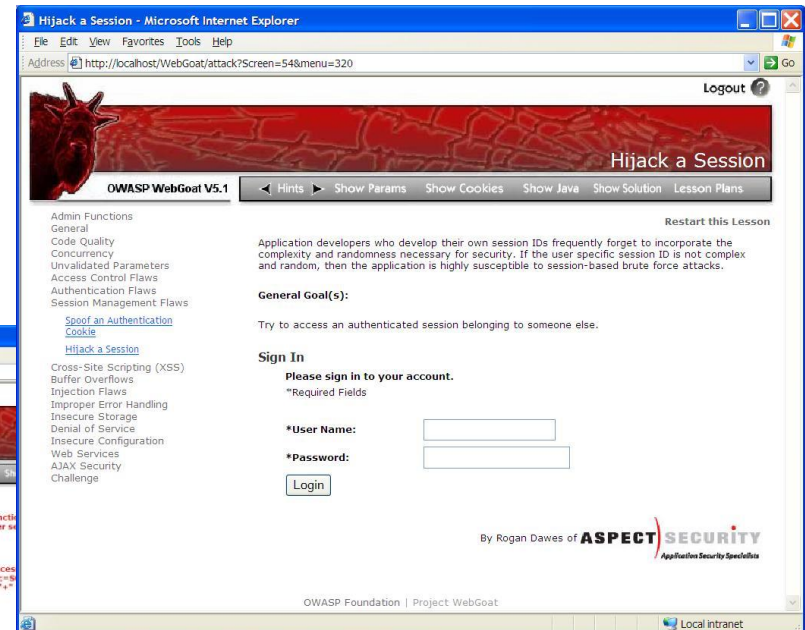
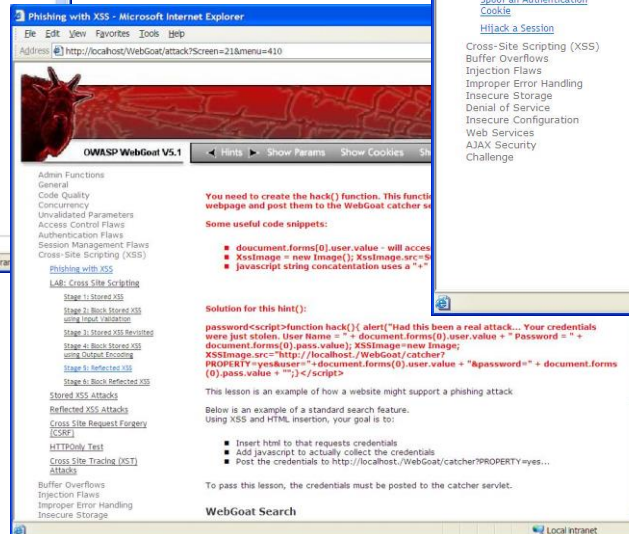
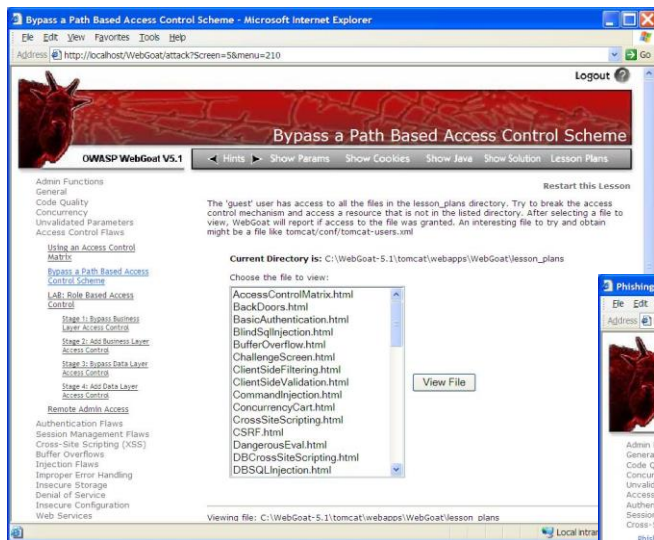
BY MICHAEL GROTHAUS 1 MINUTE READ

If you use Mozilla Firefox's web browser, you'll want to drop what you are doing right now and update it. That urging doesn't just come from Mozilla—it comes from the United States Department of Homeland Security's Cybersecurity and Infrastructure Security Agency (CISA).

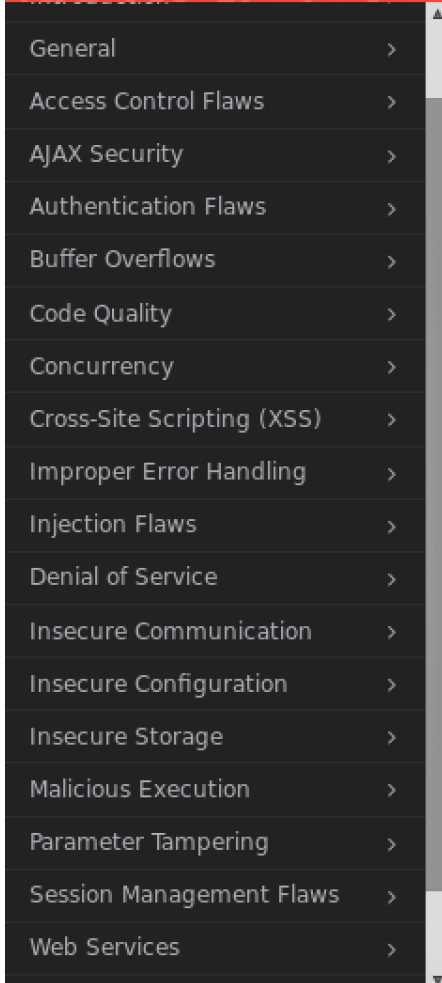
**The issue is this:** Firefox versions for desktop older than the just-patched version contain a critical vulnerability that could allow an attacker to take control of a user's entire operating system—whether they use Windows or Mac.

- **WebGoat:** [https://www.owasp.org/index.php/Proyecto\\_WebGoat\\_OWASP](https://www.owasp.org/index.php/Proyecto_WebGoat_OWASP)

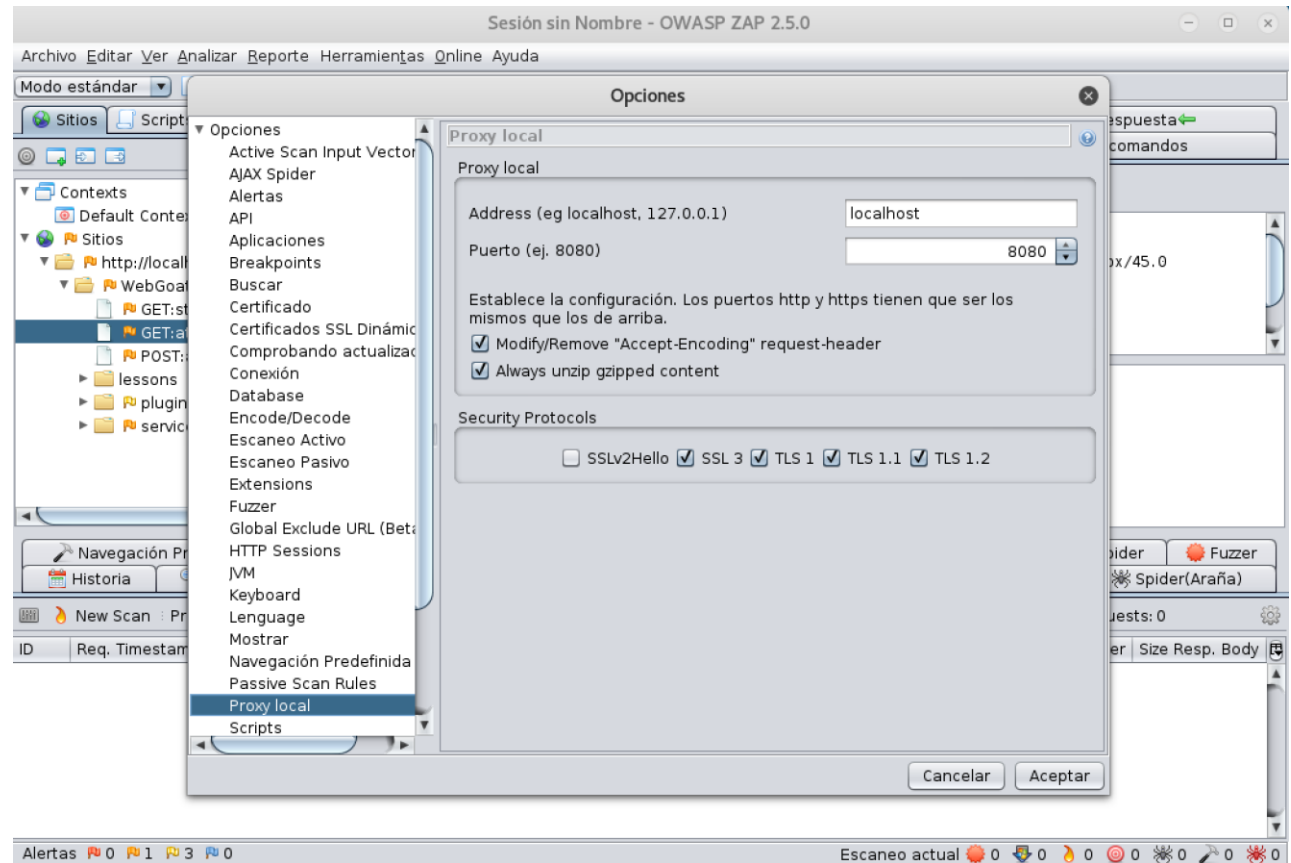
- Es una aplicación Web que contiene y muestra muchos tipos de agujeros de seguridad, diseñada por OWASP para mostrar vulnerabilidades de seguridad y enseñar seguridad en aplicaciones Web



# WebGoat



- 28 lecciones y cuatro laboratorios
- Requiere un proxy funcionando en el puerto 8080:



# SQL en WebGoat



WebGoat - Mozilla Firefox

WebGoat x Problem loading page x Preferences x +

localhost/WebGoat/start.mvc

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

Buffer Overflows >  
Code Quality >  
Concurrency >  
Cross-Site Scripting (XSS) >  
Improper Error Handling >  
Injection Flaws >  
Command Injection  
Numeric SQL Injection  
Log Spoofing  
XPATH Injection  
LAB: SQL Injection  
Stage 1: String SQL Injection  
Stage 2: Parameterized Query #1  
Stage 3: Numeric SQL Injection  
Stage 4: Parameterized Query #2  
String SQL Injection  
Modify Data with SQL Injection  
Add Data with SQL Injection  
Blind Numeric SQL Injection  
Blind String SQL Injection  
Denial of Service >

**THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT**

Implement a fix to block SQL injection into the fields in question on the Login page. Repeat stage 1. Verify that the attack is no longer effective.

**\* Login failed**

**Goat Hills Financial**  
Human Resources

Please Login

Neville Bartholomew (admin) ▾

Password ●●●●

Login

domain  
httpOnly false  
maxAge -1  
name JSESSIONID  
path  
secure false  
value D944781E27E237F8DEE2C7522478  
version 0

Params

Param	Value
Screen	6
menu	1100

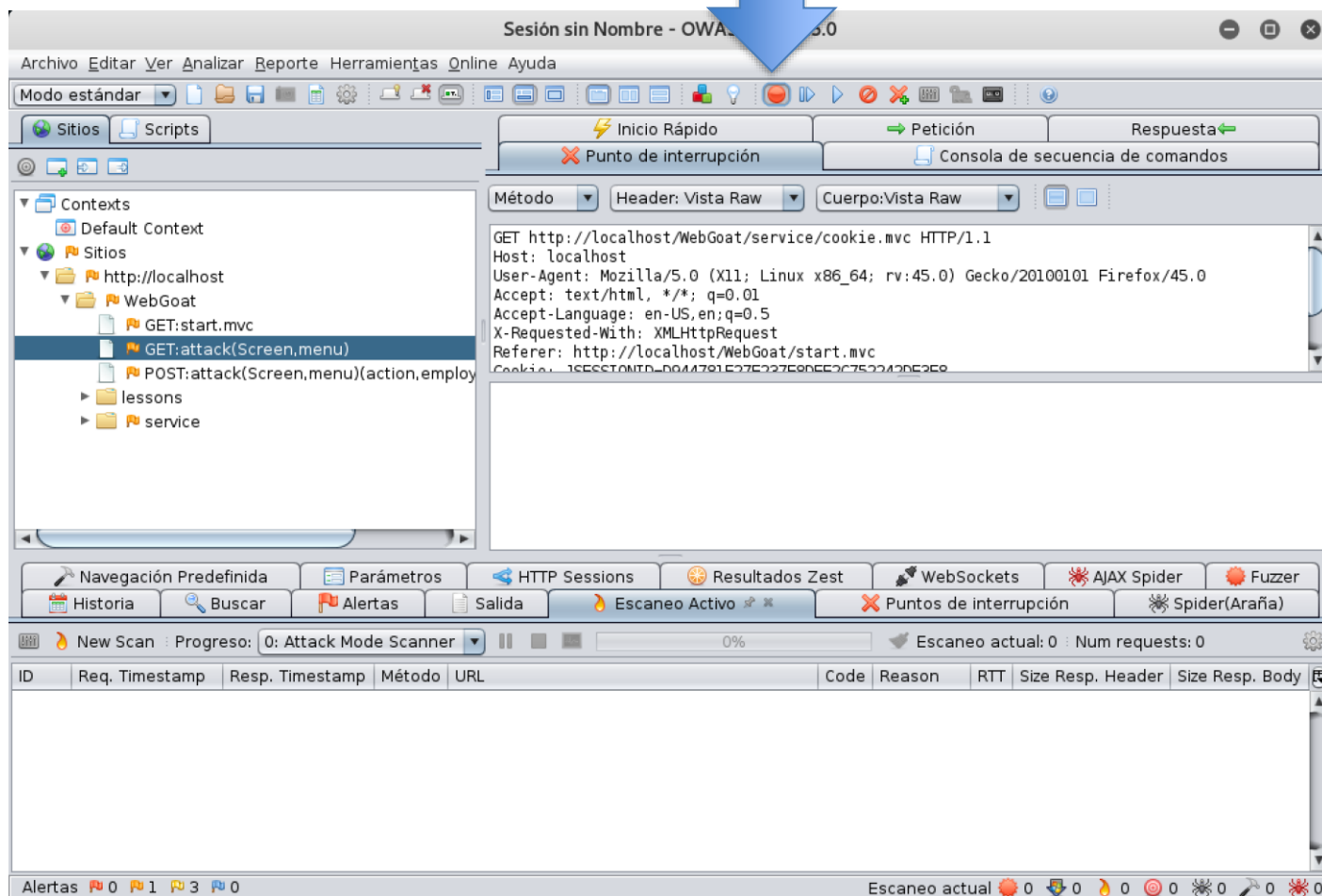
- Vamos a intentar entrar como admin



# SQL en WebGoat



- Activar el proxy para capturar las solicitudes

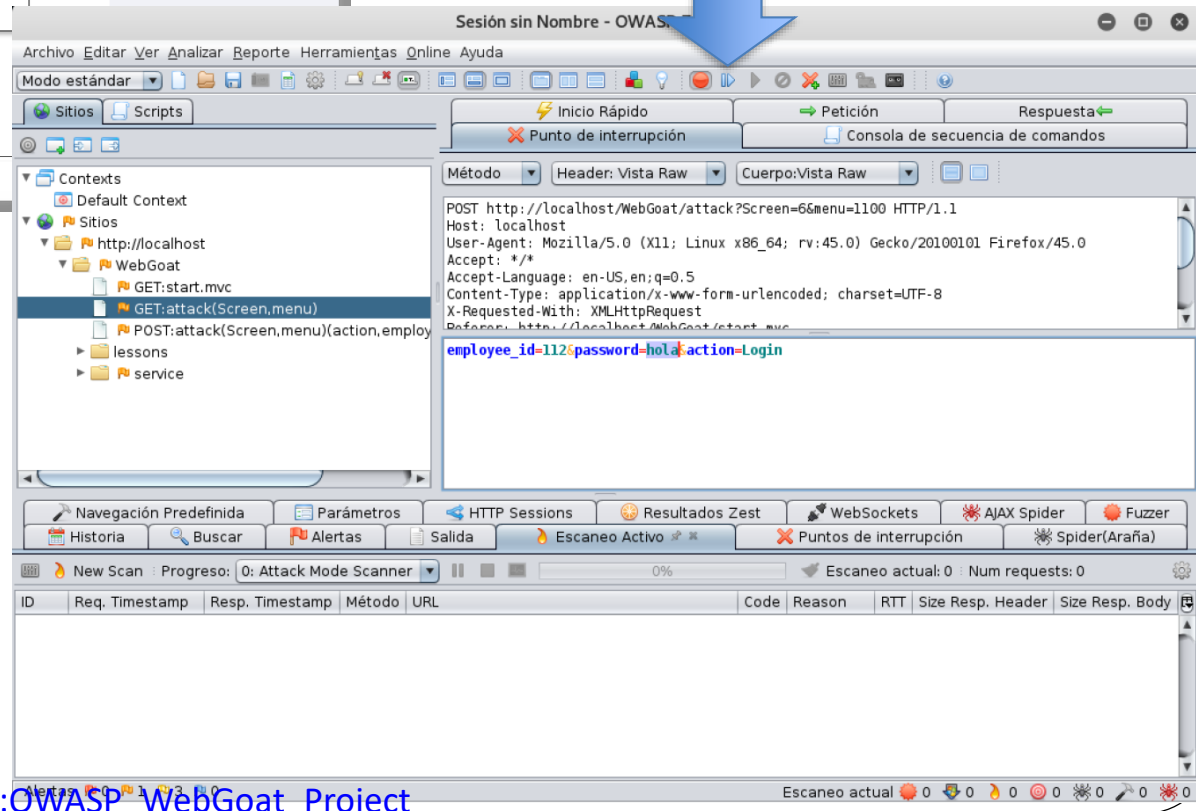
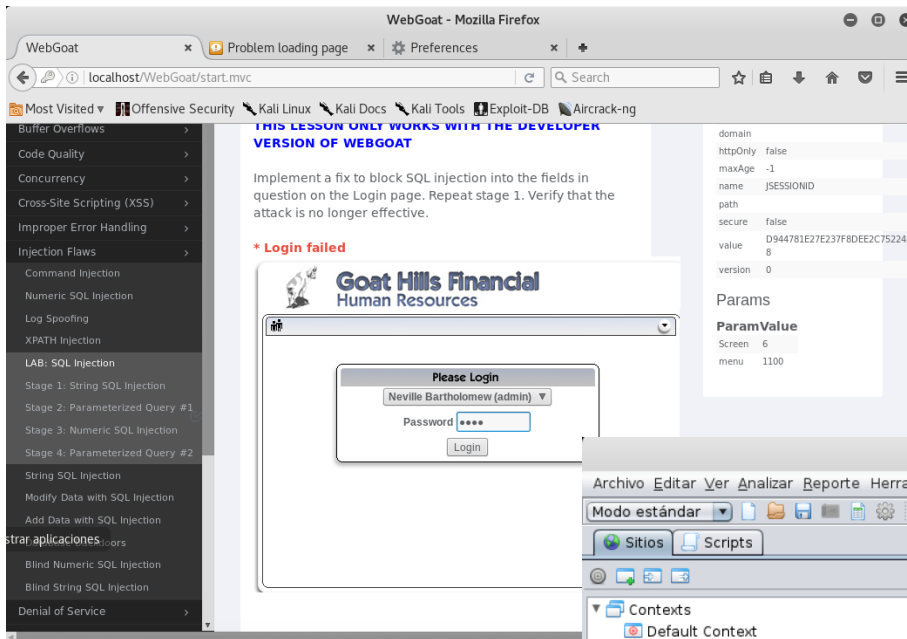


# SQL en WebGoat

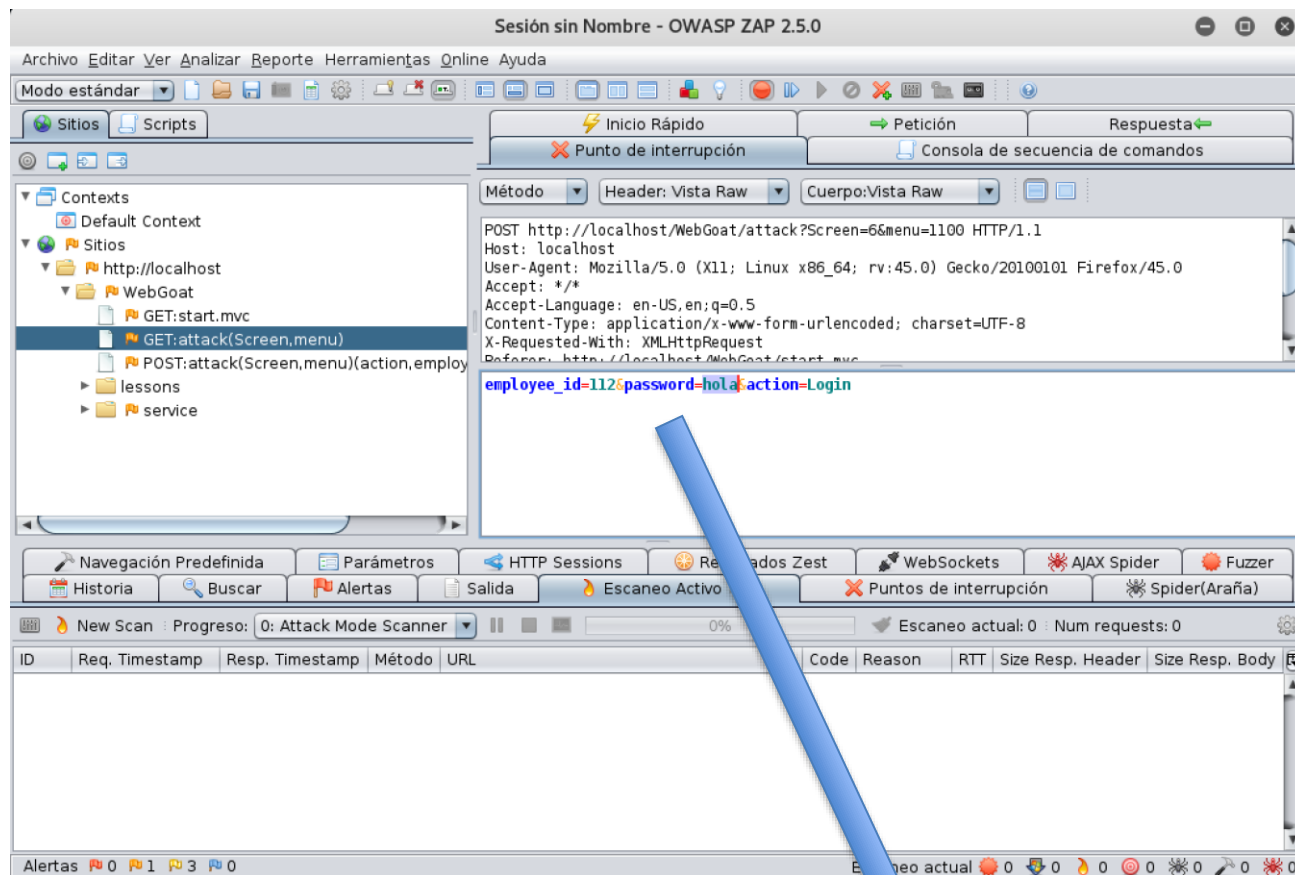


- Introducir una clave cualesquiera

Busca el siguiente breakpoint (la solicitud)



# SQL en WebGoat



- Realizar el ataque

Referer: http://localhost/WebGoat/start.mvc  
**employee\_id=112&password=' OR '1'='1&action=Login**

# SQL en WebGoat



## THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT

Implement a fix to block SQL injection into the fields in question on the Login page. Repeat stage 1. Verify that the attack is no longer effective.

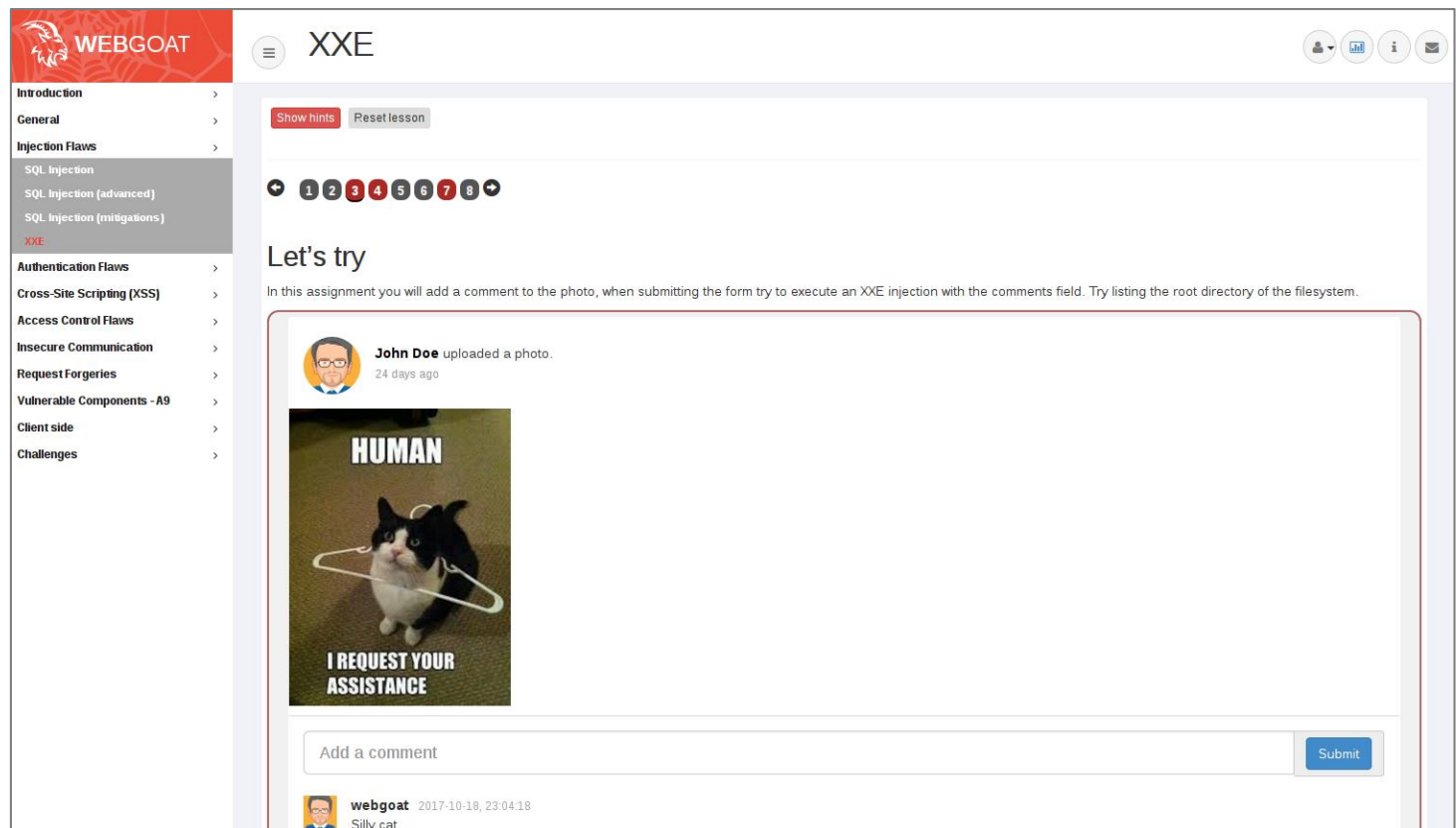
The screenshot shows the 'Goat Hills Financial Human Resources' web application. At the top, there is a logo of a goat and the text 'Goat Hills Financial Human Resources'. Below this, a navigation bar says 'Welcome Back Neville - Staff Listing Page'. The main content area has the text 'Select from the list below' followed by a dropdown menu. The dropdown menu is open, showing a list of staff members with their roles: Larry Stooge (employee), Moe Stooge (manager), Curly Stooge (employee), Eric Walker (employee), Tom Cat (employee), Jerry Mouse (hr), David Giambi (manager), Bruce McGuirre (employee), Sean Livingston (employee), Joanne McDougal (hr), and John Wayne (admin). To the right of the dropdown menu, there are four buttons: SearchStaff, ViewProfile, CreateProfile, and DeleteProfile. Below these buttons is a Logout button.



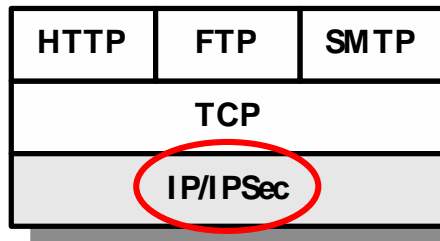
- Se accede en modo administrador

- **XXE (XML External Entity)**

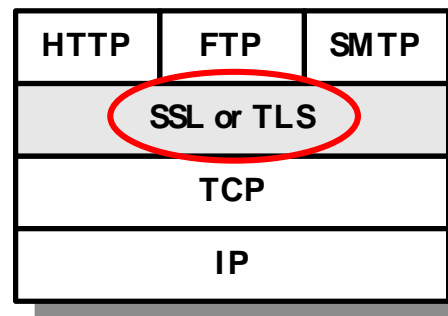
- Ataque de injection contra un sistema que gestiona entradas XML
- Este ataque puede liderar a lecturas o modificaciones no autorizadas, o DoS



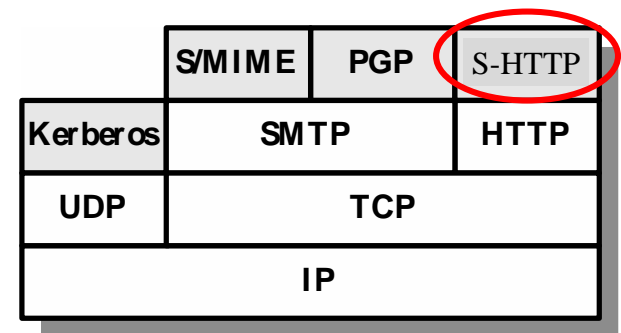
- De los tres puntos de ataques mencionados anteriormente (**cliente Web, servidor Web y tráfico**), nosotros centramos en la gestión del tráfico de red
- Así, para dar una solución al problema del **tráfico Web**, existen diferentes soluciones, dependiendo de la capa TCP/IP a considerar



(a) Network Level



(b) Transport Level



(c) Application Level

- El IETF formó a mediados de los 90 un Grupo de Trabajo denominado *Web Transaction Security (WTS)*
  - su objetivo: desarrollar los requisitos y las especificaciones para la provisión de servicios de seguridad en transacciones Web

## Web Transaction Security (wts) (concluded WG)

[Documents](#) | [Charter](#) | [History](#) | [List Archive »](#) | [Tools WG Page »](#)

### Description of Working Group

The goal of the Web Transaction Security Working Group is to develop requirements and a specification for the provision of security services to Web transaction, e.g., transactions using HyperText Transport Protocol (HTTP). This work will proceed in parallel to and independently of the development of non-security features in the HTTP Working Group. The working group will prepare two documents for submission as Internet Drafts; an HTTP Security Requirements Specification, and an HTTP Security Protocol Specification. The latter will be submitted as a Standards Track RFC.

### Goals and Milestones

Jul 1995	HTTP Security Requirements finalized at the Stockholm IETF. Submit HTTP Security Specification proposal(s) as Internet-Drafts.
Dec 1995	HTTP Security Specification finalized at the Dallas IETF, submit to IESG for consideration as a Proposed Standard.
Done	HTTP Security Requirements submitted as Internet-Draft.

Note: The data for concluded WGs is occasionally incorrect.

#### Group

Name: Web Transaction Security  
 Acronym: wts  
 Area: Security Area (sec)  
 State: Concluded  
 Charter: [charter-ietf-wts-01](#) (Approved)

#### Personnel

Chair: [Charlie Kaufman <charlie\\_kaufman@notesdev.ibm.com>](#)  
 Area Director: ?

#### Mailing List

Address: [www-security@nsmx.rutgers.edu](mailto:www-security@nsmx.rutgers.edu)  
 To Subscribe: [www-security-request@nsmx.rutgers.edu](mailto:www-security-request@nsmx.rutgers.edu)  
 Archive: <http://www.ns.rutgers.edu/www-security>

- Este grupo se centró en el desarrollo de una solución en la capa de aplicación, y diseñó el protocolo **SHTTP - Secure HyperText Transfer Protocol**, especificado en los documentos RFC que se observan abajo

## Web Transaction Security (wts)

**(concluded WG)**

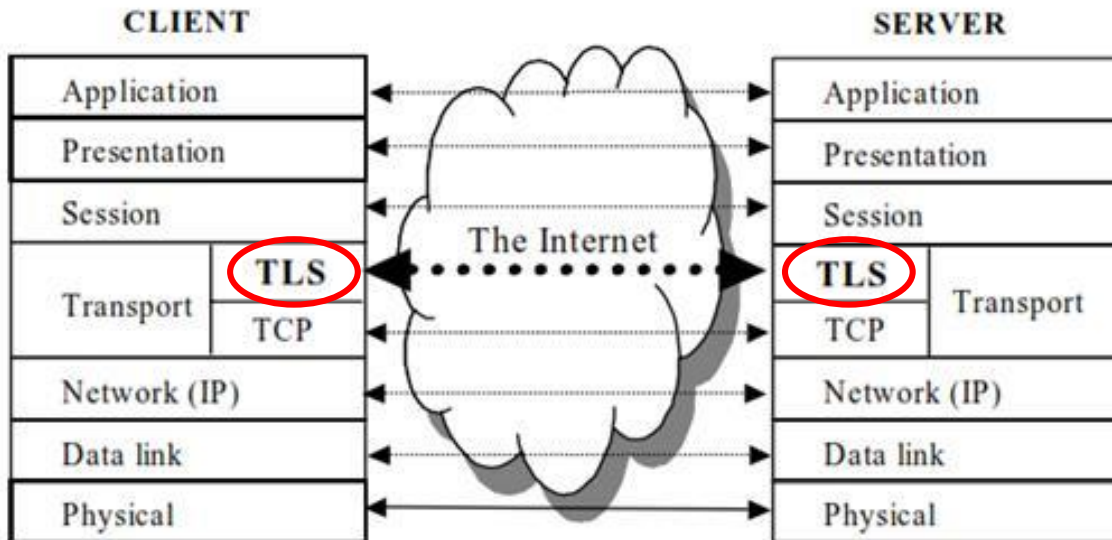
[Documents](#) | [Charter](#) | [History](#) | [List Archive »](#) | [Tools WG Page »](#)

Document	Title	Date	Status	IPR	Area Director
<b>RFCs</b>					
<a href="#">RFC 2084</a> ( <a href="#">draft-ietf-wts-requirements</a> )	Considerations for Web Transaction Security	1997-01	RFC 2084 (Informational)		
<a href="#">RFC 2659</a> ( <a href="#">draft-ietf-wts-shtml</a> )	Security Extensions For HTML	1999-08	RFC 2659 (Experimental)		
<a href="#">RFC 2660</a> ( <a href="#">draft-ietf-wts-shttp</a> )	The Secure HyperText Transfer Protocol	1999-08	RFC 2660 (Experimental)		
<b>Related Documents</b>					
Related Documents	Title	Date	Status	IPR	Area Director

- S-HTTP es una solución alternativa a HTTPS pero obsoleta
  - S-HTTP sólo cifra los datos enviados al servidor sin requerir un proceso de negociación y funcionando en el mismo puerto que HTTP
  - HTTPS, por el contrario, funciona sobre SSL protegiendo el dato antes y después de todo el proceso de transmisión de la misma – nota: SSL está actualmente obsoleta y se aplica TLS, y SSL se describe a continuación



- Por otro lado, en las mismas fechas, los desarrolladores de Netscape abordaron el problema, pero desde la capa de transporte
  - como una solución intermedia (ni en la capa alta ni en las bajas)
- El resultado fue el protocolo **SSL - Secure Sockets Layer**, una subcapa entre la de aplicación y la de transporte
  - más concretamente, SSL se sitúa por encima de TCP dado que este es orientado a la conexión y proporciona fiabilidad



- el objetivo del protocolo SSL es, por tanto, **crear conexiones seguras** y transmitir datos a través de esas conexiones
- la última versión producida fue la v3.0

# SSL - Secure Sockets Layer

- El protocolo SSL es un **protocolo cliente/servidor** que proporciona los siguientes servicios de seguridad entre los puntos que se comunican:
  - Autenticación de entidades y de origen de datos
  - Confidencialidad de la conexión
  - Integridad de la conexión
- Más concretamente, SSL emplea:
  - criptografía de **clave secreta** para la autenticación de los datos (mensajes) y para el cifrado de los mismos
  - criptografía de **clave pública** para la autenticación de las entidades y para el establecimiento de clave
    - básicamente, hay tres algoritmos de intercambio de clave en la especificación de SSL: **RSA, Diffie-Hellmann y Fortezza**
    - a pesar de la utilización de criptografía de clave pública, **no proporciona el servicio de no-repudio** (ni no-repudio de origen ni no-repudio de entrega)



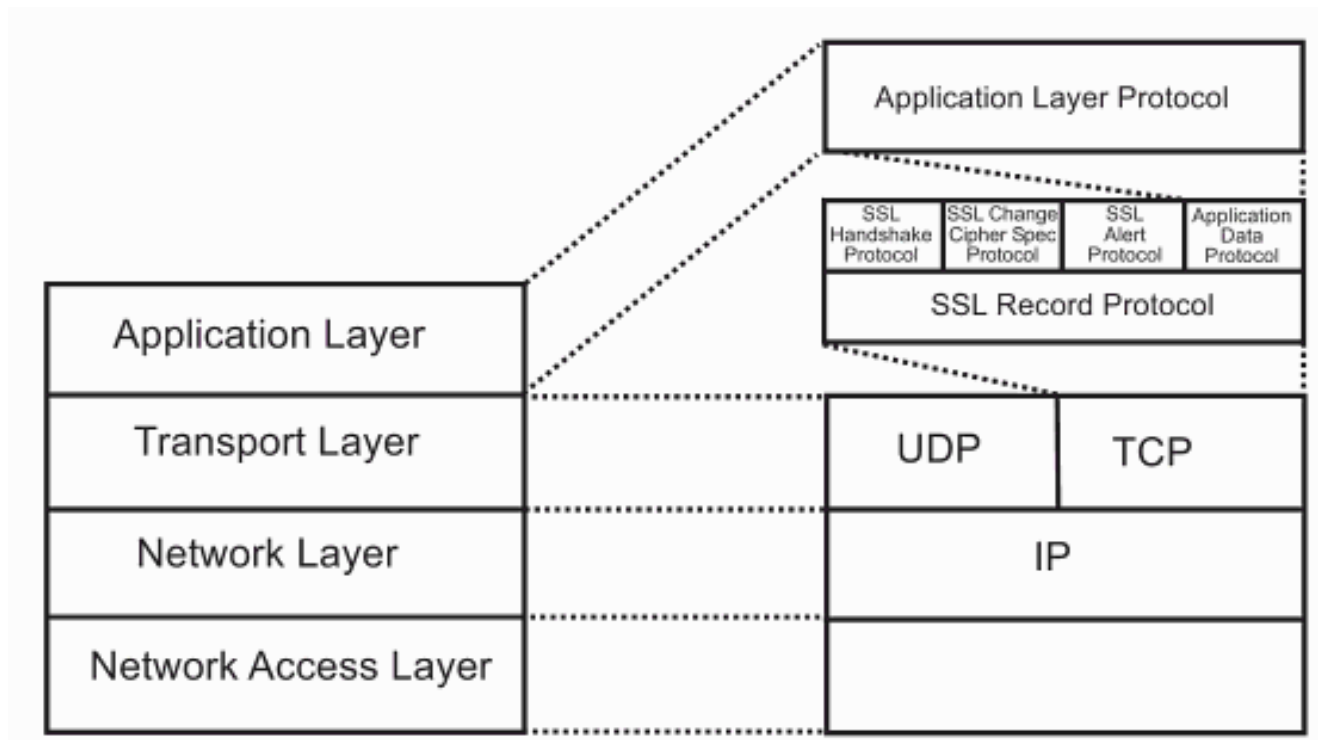
- Una ventaja del protocolo SSL es que es independiente del protocolo de la capa de aplicación
  - es decir, cualquier protocolo de aplicación basado en TCP se puede beneficiar de SSL (éste le dota de los servicios de seguridad mencionados)

Port Numbers Reserved for Application Protocols Layered over SSL/TLS

Protocol	Description	Port #
nsiops	IIOP Name Service over SSL/TLS	261
https	HTTP over SSL/TLS	443
nntps	NNTP over SSL/TLS	563
ldaps	LDAP over SSL/TLS	636
ftps-data	FTP Data over SSL/TLS	989
ftps	FTP Control over SSL/TLS	990
telnets	Telnet over SSL/TLS	992
imaps	IMAP4 over SSL/TLS	993
ircs	IRC over SSL/TLS	994
pop3s	POP3 over SSL/TLS	995
tftps	TFTP over SSL/TLS	3713
sip-tls	SIP over SSL/TLS	5061
...	...	...

- El protocolo SSL emplea, entre otros, estos dos conceptos:
  - **Sesión SSL**: asociación entre el cliente y el servidor en la que **se negocian los parámetros de seguridad** para todas las conexiones de esa sesión
  - **Conexión SSL**: realización de la **transmisión de datos** entre el cliente y el servidor, protegida criptográficamente según lo negociado en la sesión
- El ámbito de la funcionalidad de SSL es doble, como se desprende de lo anterior:
  1. Establecer una conexión segura (confidencial y autenticada) entre los puntos que se comunican
  2. Utilizar esa conexión para transmitir de forma segura los datos del nivel de aplicación entre el emisor y el receptor. Esta transmisión requiere a su vez de:
    - Dividir los datos en fragmentos más manejables
    - Procesarlos de forma individual
      - cada fragmento tratado se denomina **SSL record**

- Para llevar a cabo esa doble funcionalidad, SSL consta de dos subcapas y varios subprotocolos, como se observa en la siguiente figura:



- La subcapa alta contiene:
  - **SSL Handshake Protocol**: permite que los puntos de comunicación se autenticuen mutuamente, y que además, negocien un **cipher suite** y (opcionalmente) un método de compresión
  - **SSL Change Cipher Spec Protocol**: permite a los puntos de comunicación activar el cipher suite
  - **SSL Alert Protocol**: permite a los puntos de comunicación indicar posibles problemas potenciales e intercambiar los correspondientes mensajes de alerta
  - **SSL Application Data Protocol**: es el propio protocolo de la capa de aplicación (ej: HTTP) y alimenta al SSL Record Protocol
- La subcapa baja contiene:
  - **SSL Record Protocol**: fragmenta los datos de la capa de aplicación y los procesa de forma individual

- SSL Handshake Protocol

- Se utiliza antes de transmitir ningún dato de la capa de aplicación
- Es la parte más compleja de SSL porque permite al servidor y al cliente:
  - autenticarse mutuamente
  - negociar un algoritmo de cifrado y una función MAC
  - así como las claves a usar para proteger los datos del SSL record
- Consta de una serie de mensajes intercambiados entre el cliente y el servidor, con el formato:

1 byte	3 bytes	$\geq 0$ bytes
Type	Length	Content

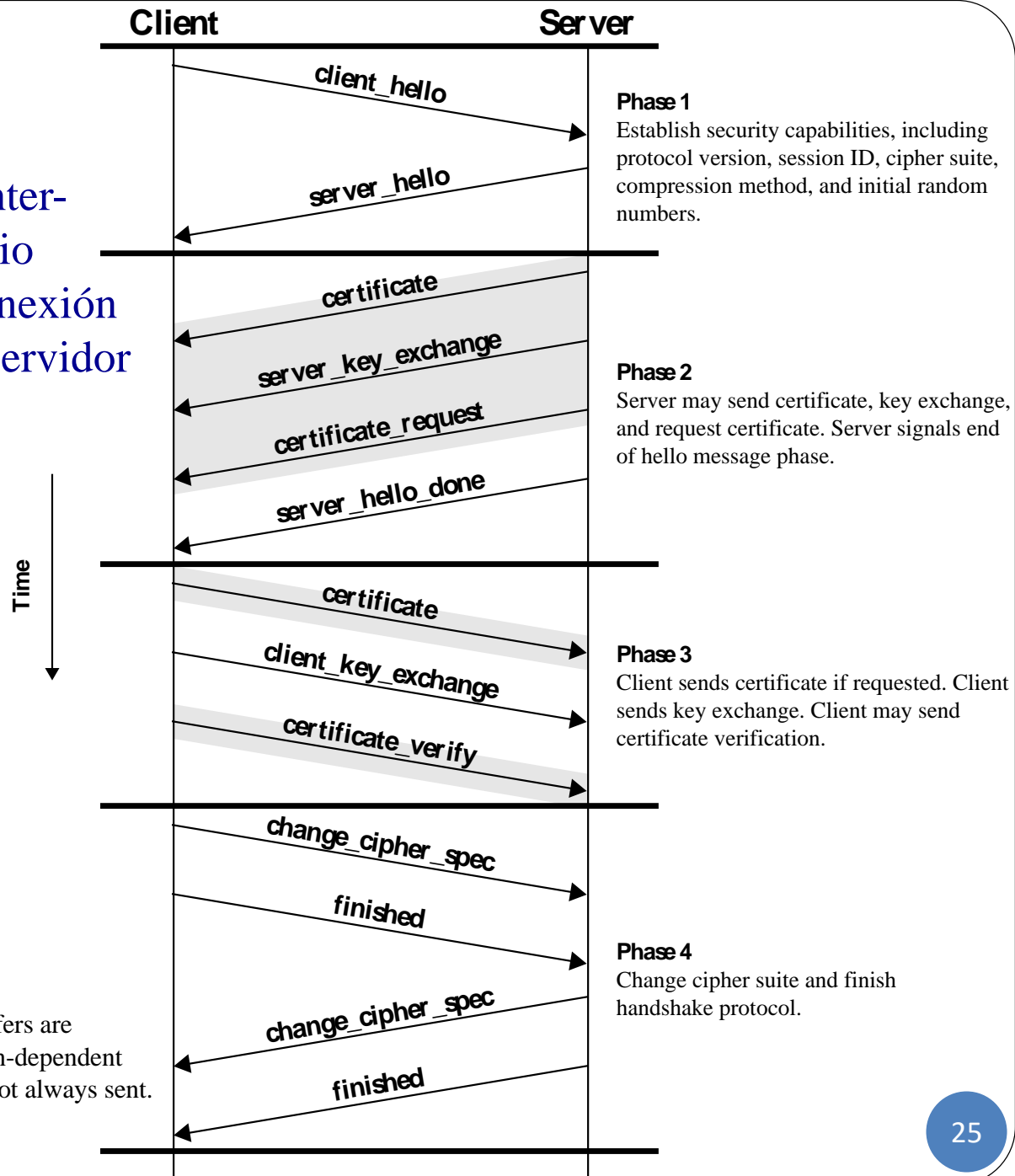
- Por lo tanto, cada mensaje tiene 3 campos:
  - *Type* (1 byte): indica uno de 10 posibles mensajes (ver siguiente tabla)
  - *Length* (3 bytes): longitud del mensaje en bytes
  - *Content* ( $\geq 0$  bytes): parámetros asociados con el mensaje (ver también siguiente tabla)

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value



- La figura muestra el intercambio inicial necesario para establecer una conexión lógica entre cliente y servidor

- Se observan 4 fases



- DHE (Diffie Helman efímero): Tanto el cliente como el servidor generan sus valores secretos ( $x$ ,  $y$ ,  $cli\_sec$ ,  $srv\_sec$ ) en cada negociación
  - Proporciona *forward secrecy* (FS) en la creación del secreto compartido
    - FS protege las comunicaciones y las sesiones establecidas en el pasado
      - Es decir, claves de sesión antiguas no serán derivadas
  - Negociación de Claves: DHE/RSA
    - **Server Key Exchange:** Servidor  $\rightarrow$  Cliente:  $p$ ,  $g$ ,  $pubKey = g^y \bmod p$ 
      - Estos parámetros están firmados por el Servidor
    - **Client Key Exchange:** Cliente  $\rightarrow$  Servidor:  $pubKey = g^x \bmod p$
    - $premaster\_secret = (g^x)^y \bmod p = (g^y)^x \bmod p = g^{xy} \bmod p$
  - Negociación de Claves: ECDHE (Curvas elípticas) - **TLS**
    - **Server Key Exchange:** Servidor  $\rightarrow$  Cliente:  $a$ ,  $b$ ,  $p$ ,  $G$ ,  $pubKey = G * srv\_sec$ 
      - Estos parámetros están firmados por el Servidor
    - **Client Key Exchange:** Cliente  $\rightarrow$  Servidor:  $pubKey = G * cli\_sec$
    - $premaster\_secret = server\_pubKey * cli\_sec = client\_pubKey * srv\_sec$   
 $= G * cli\_sec * srv\_sec$

- SSL Change Cipher Spec Protocol

- Es un protocolo muy simple que consta de un solo mensaje de un solo byte con valor 1 que permite activar el cipher suite

- SSL Alert Potocol

- Se usa para comunicar al otro punto de comunicación las alertas relacionadas con SSL, y cada mensaje de este protocolo consta de 2 bytes
  - Estos mensajes también se comprimen y se cifran de acuerdo con lo establecido en la sesión

- El primer byte toma el valor 1 (warning) o 2 (fatal) para informar de la severidad del mensaje. Si el nivel es fatal SSL termina la conexión de forma inmediata

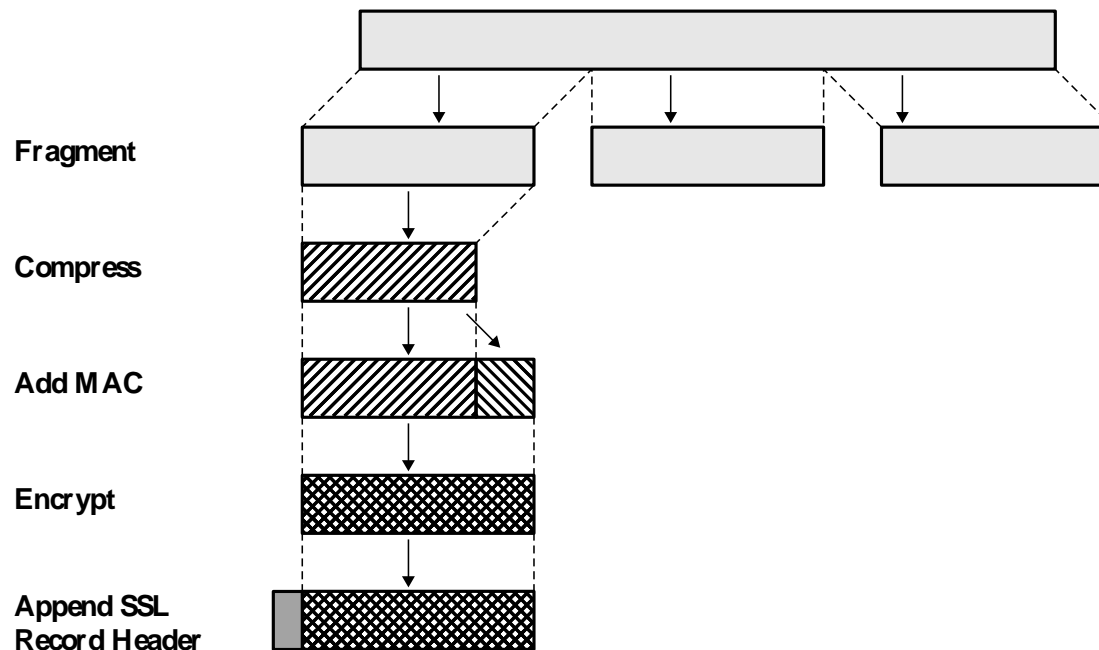
- Otras conexiones de la misma sesión pueden continuar pero no se producen nuevas conexiones dentro de la misma sesión

- El segundo byte contiene un código que indica la alerta específica

- Ejemplos: unexpected\_message, bad\_record\_mac, decompression\_failure, illegal\_parameter, ...

- SSL Record Protocol

- Toma los datos de la subcapa alta, los fragmenta en bloques manejables, los comprime de forma opcional, añade el MAC, cifra, y añade una cabecera
- El resultado final se transmite en un segmento TCP
- En recepción, los datos recibidos son descifrados, verificados, descomprimidos y reensamblados antes de entregarlos a la capa de aplicación



- Por lo tanto, este subprotocolo proporciona:
  - *Confidencialidad:*
    - el SSL Handshake Protocol define una **clave secreta compartida** que es utilizada **para el cifrado de los datos**
  - *Integridad de datos:*
    - el SSL Handshake Protocol también define **una clave secreta** compartida que es **utilizada para formar un MAC**
- En lo que a fragmentación se refiere, cada mensaje de la capa de aplicación se fragmenta en bloques de longitud  $2^{14}$  bytes (16384) o menor
- En lo que respecta al algoritmo de compresión, SSL no especifica ninguno
- En cuanto al código de autenticación de mensajes, se utiliza uno similar a HMAC

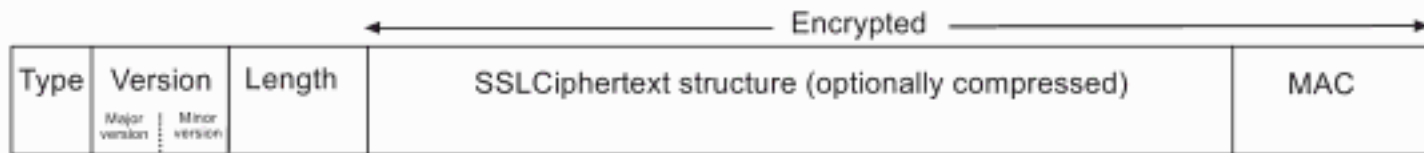


- El mensaje comprimido y el valor MAC se cifran utilizando **criptografía simétrica**. Los algoritmos que se pueden utilizar se muestran en la tercera columna de la tabla:

SSL Cipher Suites

CipherSuite	Key Exchange	Cipher	Hash
SSL_NULL_WITH_NULL_NULL	NULL	NULL	NULL
SSL_RSA_WITH_NULL_MD5	RSA	NULL	MD5
SSL_RSA_WITH_NULL_SHA	RSA	NULL	SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5	RSA_EXPORT	RC4_40	MD5
SSL_RSA_WITH_RC4_128_MD5	RSA	RC4_128	MD5
SSL_RSA_WITH_RC4_128_SHA	RSA	RC4_128	SHA
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	RSA_EXPORT	RC2_CBC_40	MD5
SSL_RSA_WITH_IDEA_CBC_SHA	RSA	IDEA_CBC	SHA
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	RSA_EXPORT	DES40_CBC	SHA
SSL_RSA_WITH_DES_CBC_SHA	RSA	DES_CBC	SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA	RSA	3DES_EDE_CBC	SHA
SSL_DH_DSS_EXPORT_WITH_DES40_CBC_SHA	DH_DSS_EXPORT	DES40_CBC	SHA
SSL_DH_DSS_WITH_DES_CBC_SHA	DH_DSS	DES_CBC	SHA
SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA	DH_DSS	3DES_EDE_CBC	SHA
SSL_DH_RSA_EXPORT_WITH_DES40_CBC_SHA	DH_RSA_EXPORT	DES40_CBC	SHA
SSL_DH_RSA_WITH_DES_CBC_SHA	DH_RSA	DES_CBC	SHA
SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA	DH_RSA	3DES_EDE_CBC	SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	DHE_DSS_EXPORT	DES40_CBC	SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA	DHE_DSS	DES_CBC	SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DHE_DSS	3DES_EDE_CBC	SHA
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	DHE_RSA_EXPORT	DES40_CBC	SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA	DHE_RSA	DES_CBC	SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DHE_RSA	3DES_EDE_CBC	SHA
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	DH_anon_EXPORT	RC4_40	MD5
SSL_DH_anon_WITH_RC4_128_MD5	DH_anon	RC4_128	MD5
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	DH_anon	DES40_CBC	SHA
SSL_DH_anon_WITH_DES_CBC_SHA	DH_anon	DES_CBC	SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	DH_anon	3DES_EDE_CBC	SHA
SSL_FORTEZZA_KEA_WITH_NULL_SHA	FORTEZZA_KEA	NULL	SHA
SSL_FORTEZZA_KEA_WITH_FORTEZZA_CBC_SHA	FORTEZZA_KEA	FORTEZZA_CBC	SHA
SSL_FORTEZZA_KEA_WITH_RC4_128_SHA	FORTEZZA_KEA	RC4_128	SHA

- El paso final del SSL Record Protocol es preparar una **cabecera** que consta de los siguientes campos:
  - *Content Type* (8 bits): protocolo de la subcapa alta de SSL de la que procede el fragmento:
    - 20 → SSL Change Cipher Spec Protocol
    - 21 → SSL Alert Protocol
    - 22 → SSL Handshake Protocol
    - 23 → SSL Application Data Protocol
  - *Major Version* (8 bits): versión de SSL en uso (para SSLv3, el valor es 3)
  - *Minor Version* (8 bits): versión menor en uso (para SSLv3, el valor es 0)
  - *Compressed Length* (16 bits): longitud en bytes del fragmento de texto en claro (o del fragmento comprimido si se ha utilizado compresión)

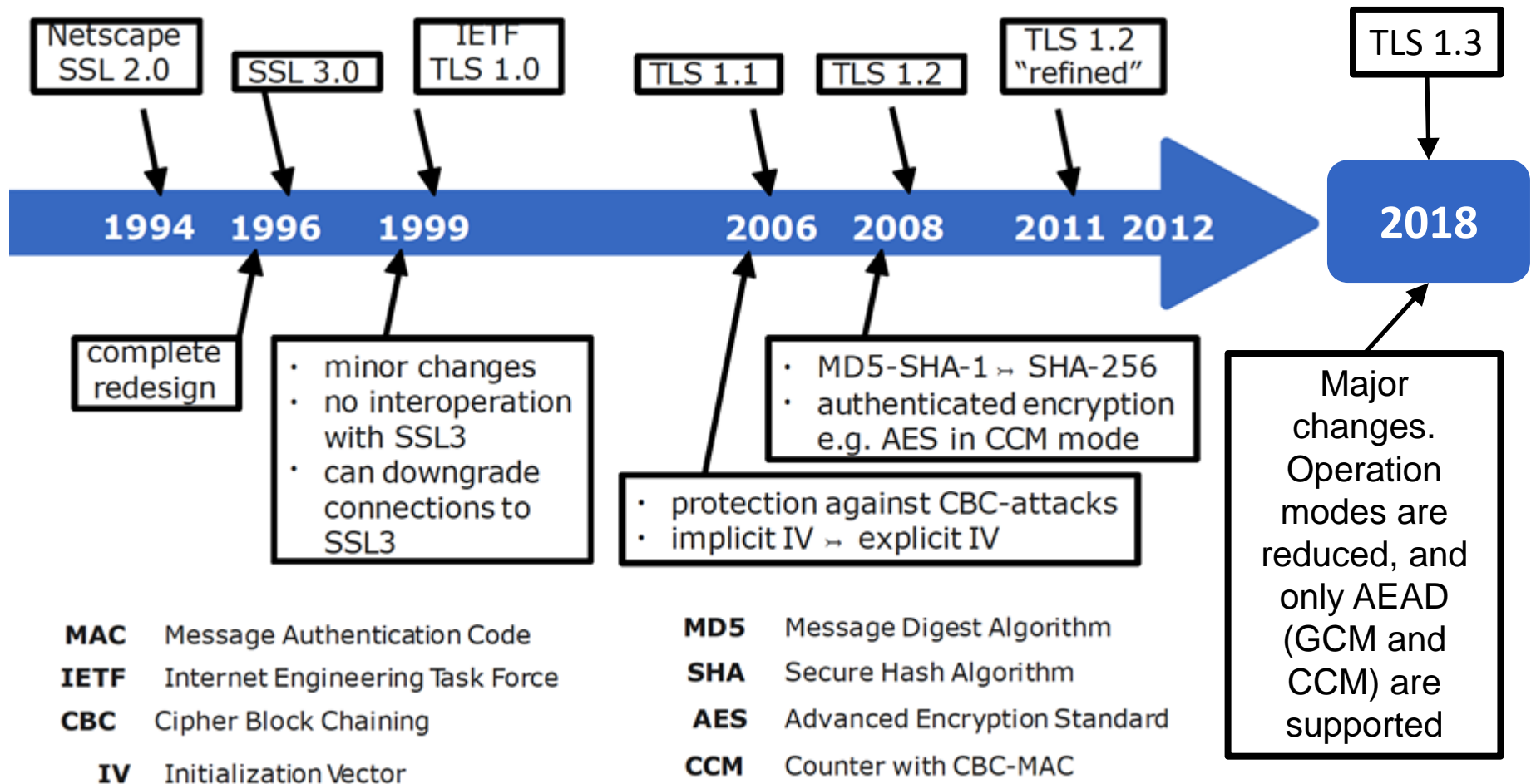


# Transport Layer Security

- TLS es una iniciativa de IETF para estandarizar SSL
- Se definió por primera vez (TLS 1.0) en 1999, en el RFC 2246. Como se menciona en ese RFC:

*“the differences between this protocol and SSL 3.0 are not dramatic, but they are significant to preclude interoperability between TLS 1.0 and SSL 3.0.”*
- Versiones posteriores han sido TLS 1.1, publicada en 2006 en el RFC 4346, TLS 1.2, publicada en 2008 en el RFC 5246, y TLS 1.3, publicada en 2018 en el RFC 8446
  - TLS 1.2 introduce cambios muy significativos, como la inclusión de **AES** en el cipher suite, la consideración de **SHA-256**, así como la consideración de **criptografía de clave pública basada en curvas elípticas**
    - Lo hemos visto anteriormente en el uso de **ECDHE** para la negociación
  - TLS 1.3 **reduce el tiempo de “handshake”**, y reduce el número de modos de operación soportados, limitándolo a **GCM** y **CCM**





Transport Layer Security (tls) – Charter

datatracker.ietf.org/wg/tls/charter/

SSL and TLS: Theory and Practice – Rolf Oppliger, Ph.D. – Google Libros

Transport Layer Security (tls) – Charter

**datatracker.ietf.org** Sign In

**Accounts**  
New Account

**Working Groups**  
Applications  
Internet  
Ops & Mgmt  
RAI  
Routing  
Security  
Transport  
Active WGs  
Chartering WGs  
BoFs  
Concluded WGs  
Non-WG Lists

**Drafts & RFCs**  
Document search:  
Submit a draft  
Sign in to track drafts

**Meetings**  
Agenda  
Materials  
Past Proceedings  
Upcoming

**Other Documents**  
IPR Disclosures  
Liaison Statements  
IESG Agenda

**Related Sites**  
Main IETF site  
IETF tools  
IAB  
RFC Editor  
IASA/IAOC/Trust  
IANA  
IRTF

**Transport Layer Security (tls)**

[Documents](#) | [Charter](#) | [History](#) | [List Archive »](#) | [Tools WG Page »](#)

### Description of Working Group

The TLS Working Group was established in 1996 to standardize a 'transport layer' security protocol. The working group began with SSL version 3.0. The TLS Working Group has completed a series of specifications that describe the Transport Layer Security protocol versions 1.0, 1.1, and 1.2, extensions to the protocol, and new ciphersuites to be used with TLS.

The primary goals of the WG are to maintain:

- The TLS protocol, RFC 5246;
- The DTLS protocol, draft-ietf-tls-rfc4347-bis.

Significant changes to the protocol, such as a new version 1.3, are not within scope of the working group unless they are explicitly added to the charter.

The secondary goals of the WG are to publish:

- Guidelines for Specifying the Use of TLS/DTLS;
- Recommendations for use of TLS (e.g., server ID);
- Extensions to TLS and DTLS; and,
- Cipher suites.

### Goals and Milestones

Done	Agreement on charter and issues in current draft.
Done	Final draft for Secure Transport Layer Protocol ('STLP')
Done	Working group 'Last Call'
Done	Submit to IESG for consideration as a Proposed Standard.
Done	First revised draft of TLS specification
Done	TSL 1.1 Specification
Done	First draft of TLS 1.2 specification, including CTR mode cipher suites
Done	First draft of specification for cipher suites with combined encryption/authentication modes
Dec 2011	Heartbeat Extension Sent to IESG

Version 4.36, 2012-11-07  
[Report a bug](#)

**Group**  
Name: Transport Layer Security  
Acronym: tls  
Area: Security Area (sec)  
State: Active  
Charter: [charter-ietf-tls-04](#) (Approved)

**Personnel**  
Chairs: [Eric Rescorla <ekr@networkresonance.com>](#)  
[Joseph Salowey <jsalowey@cisco.com>](#)  
[Eric Rescorla <ekr@rtfm.com>](#)  
Area Director: [Sean Turner <tumers@ieca.com>](#)  
Tech Advisor: [Allison Mankin <mankin@psg.com>](#)

**Mailing List**  
Address: [tls@ietf.org](#)  
To Subscribe: [https://www.ietf.org/mailman/listinfo/tls](#)  
Archive: [http://www.ietf.org/mail-archive/web/tls/](#)

**Jabber Chat**  
Room Address: [xmpp:tls@jabber.ietf.org](#)  
Logs: [http://jabber.ietf.org/logs/tls/](#)

RFCs					
<a href="#">RFC 2246</a> ( <a href="#">draft-ietf-tls-protocol</a> )	The TLS Protocol Version 1.0	1999-01	RFC 2246 (Proposed Standard) Obsoleted by <a href="#">RFC 4346</a> Updated by <a href="#">RFC 3546</a> , <a href="#">RFC 5746</a> , <a href="#">RFC 6176</a>		
<a href="#">RFC 2712</a> ( <a href="#">draft-ietf-tls-kerb-cipher-suites</a> )	Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)	1999-10	RFC 2712 (Proposed Standard)		
<a href="#">RFC 2817</a> ( <a href="#">draft-ietf-tls-http-upgrade</a> )	Upgrading to TLS Within HTTP/1.1	2000-05	RFC 2817 (Proposed Standard) <a href="#">Errata</a>		
<a href="#">RFC 2818</a> ( <a href="#">draft-ietf-tls-https</a> )	HTTP Over TLS	2000-05	RFC 2818 (Informational) Updated by <a href="#">RFC 5785</a> <a href="#">Errata</a>		
<a href="#">RFC 3268</a> ( <a href="#">draft-ietf-tls-ciphersuite</a> )	Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)	2002-07	RFC 3268 (Proposed Standard) Obsoleted by <a href="#">RFC 5246</a>		
<a href="#">RFC 3546</a> ( <a href="#">draft-ietf-tls-extensions</a> )	Transport Layer Security (TLS) Extensions	2003-06	RFC 3546 (Proposed Standard) Obsoleted by <a href="#">RFC 4366</a>		Steven Bellovin
<a href="#">RFC 3749</a> ( <a href="#">draft-ietf-tls-compression</a> )	Transport Layer Security Protocol Compression Methods	2004-05	RFC 3749 (Proposed Standard)		Steven Bellovin
<a href="#">RFC 4132</a> ( <a href="#">draft-ietf-tls-camellia</a> )	Addition of Camellia Cipher Suites to Transport Layer Security (TLS)	2005-07	RFC 4132 (Proposed Standard) Obsoleted by <a href="#">RFC 5932</a>		Russ Housley
<a href="#">RFC 4279</a> ( <a href="#">draft-ietf-tls-psk</a> )	Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)	2005-12	RFC 4279 (Proposed Standard)		Russ Housley
<a href="#">RFC 4346</a> ( <a href="#">draft-ietf-tls-rfc2246-bis</a> )	The Transport Layer Security (TLS) Protocol Version 1.1	2006-04	RFC 4346 (Proposed Standard) Obsoleted by <a href="#">RFC 5246</a> Updated by <a href="#">RFC 4366</a> , <a href="#">RFC 4680</a> , <a href="#">RFC 4681</a> , <a href="#">RFC 5746</a> , <a href="#">RFC 6176</a> <a href="#">Errata</a>		Russ Housley
<a href="#">RFC 4366</a> ( <a href="#">draft-ietf-tls-rfc3546bis</a> )	Transport Layer Security (TLS) Extensions	2006-04	RFC 4366 (Proposed Standard) Obsoleted by <a href="#">RFC 5246</a> , <a href="#">RFC 6066</a> Updated by <a href="#">RFC 5746</a> <a href="#">Errata</a>		Russ Housley
<a href="#">RFC 4492</a> ( <a href="#">draft-ietf-tls-ec</a> )	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)	2006-05	RFC 4492 (Informational) Updated by <a href="#">RFC 5246</a> <a href="#">Errata</a>	<a href="#">2</a>	Russ Housley
<a href="#">RFC 4785</a> ( <a href="#">draft-ietf-tls-psk-null</a> )	Pre-Shared Key (PSK) Ciphersuites with NULL Encryption for Transport Layer Security (TLS)	2007-01	RFC 4785 (Proposed Standard)		Russ Housley
<a href="#">RFC 5054</a> ( <a href="#">draft-ietf-tls-srp</a> )	Using the Secure Remote Password (SRP) Protocol for TLS Authentication	2007-11	RFC 5054 (Informational)		Tim Polk
<a href="#">RFC 5081</a> ( <a href="#">draft-ietf-tls-openpgp-keys</a> )	Using OpenPGP Keys for Transport Layer Security (TLS) Authentication	2007-11	RFC 5081 (Experimental) Obsoleted by <a href="#">RFC 6091</a>		Russ Housley
<a href="#">RFC 5246</a> ( <a href="#">draft-ietf-tls-rfc4346-bis</a> )	The Transport Layer Security (TLS) Protocol Version 1.2	2008-08	RFC 5246 (Proposed Standard) Updated by <a href="#">RFC 5746</a> , <a href="#">RFC 5878</a> , <a href="#">RFC 6176</a> <a href="#">Errata</a>	<a href="#">1</a>	Tim Polk
<a href="#">RFC 5288</a> ( <a href="#">draft-ietf-tls-rsa-aes-gcm</a> )	AES Galois Counter Mode (GCM) Cipher Suites for TLS	2008-08	RFC 5288 (Proposed Standard)		Pasi Eronen
<a href="#">RFC 5289</a> ( <a href="#">draft-ietf-tls-ec-new-mac</a> )	TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)	2008-08	RFC 5289 (Informational)		Pasi Eronen
<a href="#">RFC 5469</a> ( <a href="#">draft-ietf-tls-des-idea</a> )	DES and IDEA Cipher Suites for Transport Layer Security (TLS)	2009-02	RFC 5469 (Informational)	<a href="#">1</a>	Tim Polk
<a href="#">RFC 5487</a> ( <a href="#">draft-ietf-tls-psk-new-mac-aes-gcm</a> )	Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode	2009-03	RFC 5487 (Proposed Standard)		Pasi Eronen

- Es conveniente comentar que existe un protocolo llamado **DTLS** (**Datagram Transport Layer Security**) definido en el RFC 6347
  - Se utiliza para los protocolos basados en datagramas
    - Es decir, para los que se ejecutan por encima de UDP
  - Se creó en 2006, aunque la última versión es de Enero de 2012
  - Esta tomando un papel relevante en entornos restringidos (IoT)
   
<https://datatracker.ietf.org/wg/dice/documents/>

IETF Datatracker Groups Documents Meetings Other User Document search

## DTLS In Constrained Environments (dice) Concluded WG

About Documents Meetings History Photos Email expansions List archive Tools »


Document	Date	Status	IPR	AD / Shepherd
<b>RFC (1 hit)</b>				
RFC 7925 (was draft-ietf-dice-profile) Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things	2016-07 61 pages	Proposed Standard RFC		Stephen Farrell Zach Shelby

Atom feed: All changes Significant Subscribe to changes Export as CSV

ISOC IETF Trust RFC Editor IRTF IESG IETF IAB IASA & IAOC IETF Tools IANA

About | IETF Datatracker | Version 6.89.2 | 2018-12-19 | Report a bug: Tracker Email  
Python 2.7.13 | Django 1.11.17

# Test de TLS

[Solutions](#) [Network](#) [Features](#) [Pricing](#) [Company](#)

[Help Center](#) [Login](#) [Sign up](#)

## TLS Checker

Does your server or CDN support the latest TLS 1.3 to make your HTTPS connections fast and secure? This testing tool will quickly verify which SSL and TLS

### Check the SSL/TLS setup of your server

**Server check:** Enter your domain name.  
**CDN check:** Enter your CNAME (e.g. images.domain.tld) or CDN domain.

### Transport Layer Security only

Your SSL/TLS certificate can run on all SSL and TLS have already been deprecated.

[Home](#) [Projects](#) [Qualys Free Trial](#) [Contact](#)

You are here: [Home](#) > [Projects](#) > SSL Server Test

## SSL Server Test

This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet. **Please note that the information you submit here is used only to provide you the service. We don't use the domain names or the test results, and we never will.**

Hostname:

☒ Do not show the results on the boards

#### Recently Seen

<a href="#">sgi.pros.org.br</a>	Err
<a href="#">www.gwyll.eu</a>	
<a href="#">mail.cloudcomputingtechnolog...</a>	
<a href="#">na04.alma.exlibrisgroup.com</a>	
<a href="#">pros.org.br</a>	Err
<a href="#">joinhour.com</a>	
<a href="#">croixducanigou.com</a>	A
<a href="#">startwritehere.com</a>	A
<a href="#">plitz.sh</a>	A+
<a href="#">alumni.cern</a>	A+

#### Recent Best

<a href="#">plitz.sh</a>	A+
<a href="#">ceb.csob.cz</a>	A+
<a href="#">alumni.cern</a>	A+
<a href="#">startwritehere.com</a>	A
<a href="#">jxappgtw.jhahosted.com</a>	A
<a href="#">vid.wzsm.sx</a>	A
<a href="#">autokrat.co.uk</a>	A
<a href="#">simpleboard.io</a>	A
<a href="#">api.godex.io</a>	B
<a href="#">mail.cloudcomputingtechnolog...</a>	B

#### Recent Worst

<a href="#">kaorin2019.com</a>	T
<a href="#">makkapp.gdnc.gov.sa</a>	F
<a href="#">webmail.markocevic.com</a>	F
<a href="#">downloads.upd.kaspersky.com</a>	T
<a href="#">report.fundclear.com.tw</a>	F
<a href="#">wow.dport.com.ua</a>	F
<a href="#">webmail.cofili.com</a>	T
<a href="#">www.eepension.co.uk</a>	F
<a href="#">venus.ingrammicro.com</a>	F
<a href="#">inet-lnx.lnxnetwork.com</a>	T

SSL Report v2.0.5

Tema 5: Seguridad en Redes TCP/IP

37