Comerç electrònic – Desenvolupament d'una solució per a l'anàlisi de la seguretat *web*

Nom Estudiant: Oriol Secall i Gasulla

Programa: Màster Universitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions (MISTIC)

Nom Consultor: María Francisca Hinarejos Campos

Centre: Universitat Oberta de Catalunya

Data Lliurament: 12/2018

Índex

Acrònims 5
Resum 7
1. Introducció 8
1.1 Context i justificació del treball 8
1.2 Objectius del treball 8
1.2.1 Objectius 8
1.3 Enfocament i mètode seguit 9
1.4 Planificació del treball final de màster 9
1.4.1 Taula de fites 10
1.4.2 Gantt 12
1.5 Recorregut treball final de màster 12
2. Anàlisi per a la implementació de l'extensió 14
2.1 Paràmetres de seguretat a analitzar 15
2.1.1 Certificat digital 15
2.1.2 *Cipher suite* 18
2.1.3 Paràmetres de seguretat descartats 18
2.2 Extensions per *Mozilla Firefox* 19
2.2.1 *WebExtensions* 19
2.2.2 Llibreries externes 21
2.3 Anàlisi d'extensions actuals en el mercat 21
2.3.1 Estructura de SSLeuth 24
2.4 Anatomia d'una *WebExtension* 26
2.5 Instal·lació de les *WebExtensions* 28
2.6 Mòdul JS WebPack 29
2.7 Anàlisi dels algoritmes i característiques de seguretat 30
2.8 Criteri puntuació 31
3. Desenvolupament d'una extensió per a l'anàlisi de la connexió a un servidor *web* 35
3.1 Algoritme de puntuació 35
3.1.1 Diagrama de flux de l'algoritme 37
3.1.2 Fórmules càlcul puntuacions 38
3.1.3 Taules ponderacions algoritmes 39
3.2 Implementació 44
3.2.1 Estructura de fitxers 44
3.2.2 Estructura de codi 45
3.2.2 Paràmetres de seguretat proporcionats per la API 48
3.3 Informació no accessible 53

3.4 Joc de proves i verificació 54
3.4.1 Proves i verificació 55
3.4.3 Anàlisi de la *World Wide Web* 66
3.5 Comparativa amb altres solucions semblants 67
3.6 Definició possibles millores futures 69
4. Conclusions generals 70
5. Bibliografia consultada 71
Annex A “Exemple d’informació proporcionada per l’extensió” 74
Annex B “Exemple contingut fitxer local generat per l’extensió” 79
Annex C “Manual d’instal·lació” 83
Annex D “Manual d’usuari” 85

Acrònims

AEAD Authenticated Encryption with Associated Data

AES Advanced Encryption Standard

AIA Authority Information Access

API Application Programming Interface

ASN.1 Abstract Syntax Notation One

C Country

CA Certificate Authority

CBC Cipher Block Chaining

CCM Counter with CBC-MAC

CCN Centre Criptològic Nacional

CERT Computer Emergency Response Team

CLI Command line interface

CN Common Name

CP Certificate Policies

CRL Certificate Revocation List

CSS Cascading Style Sheets

CSV Comma Separated Values

DER Distinguish Encoding Rules

DH Diffie-Hellman

DHE Diffie–Hellman Ephemeral

DN Distinguished Name

DNS Domain Name System

DSA Digital Signature Algorithm

DSS Digital Signature Standard

ECC Elliptic Curve Cryptography

ECDH Elliptic-curve Diffie–Hellman

ECDHE Elliptic-curve Diffie–Hellman Ephemeral

ECDSA Elliptic Curve Digital Signature Algorithm

EV Extended Validation

GCM Galois/Counter Mod

HMAC Message Authentication Code

HPKP HTTP Public Key Pinning

HSTS HTTP Strict Transport Security

HTML5 HyperText Markup Language 5

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

JS JavaScript

JSON JavaScript Object Notation

NIST National Institute of Standards and Technology

O Organization

OCSP Online Certificate Status Protocol

OID Object IDentifier

OU Organizational Unit

PEM Privacy Enhanced Mail

PKI Public Key Infrastructure

RFC Request For Comments

RSA Rivest, Shamir y Adleman

SHA Secure Hash Algorithm

SSL Secure Sockets Layer

TLS Transport Layer Security

TSP Time-Stamp Protocol

URI Uniform Resource Identifier

URL Uniform Resource Locator

UTF-8 Unicode Transformation Format 8-bit

Resum

La seguretat de les connexions *web* a Internet consisteix en aconseguir crear un canal de comunicació entre el client i el servidor, en el que es compleixin els tres conceptes bàsics per a la seguretat: autenticació, integritat i confidencialitat.

Aquests conceptes s'aconsegueixen amb protocols que implementen seguretat en la comunicació; no obstant, la implementació és variable, i el nivell de seguretat pot ser diferent per a cada lloc *web*, on normalment es regeix per trobar un balanç entre cost computacional i seguretat.

Els serveis *web* a Internet estan en ascens, com també els llocs *web* on la seguretat és crítica per tal que l'usuari dipositi la seva confiança. Els atacs informàtics i les vulnerabilitats no son una excepció, i els usuaris de navegadors *web* cada vegada més son conscients de la importància d'implementacions de solucions de seguretat per resistir a possibles atacs.

Tots aquests fets fan necessari aportar a l'usuari eines per què pugui valorar què tan segura és la seva connexió, i així obtenir confiança en el lloc *web* visitat. Així doncs, l'objectiu d'aquest treball és, un cop s'han estudiat possibles solucions i característiques de seguretat a nivell *web*, desenvolupar un mòdul que permeti analitzar quina és la qualitat de la seguretat implementada en un servidor web.

1. Introducció

1.1 Context i justificació del treball

La seguretat de qualsevol pàgina (*site*) *web* que es navegui és clau per proporcionar confiança en l'usuari per tal de interactuar amb totes les funcionalitats que es disposa i clau també per a l'organització per protegir els recursos i els serveis que pugui tenir.

Avui en dia hi ha un gran nombre de servidors *webs* on l'oferta de serveis és molt gran i en expansió (banca, compra de bens i serveis...), posant al servei dels usuaris tasques on s'ha de confiar degut a l'entrega de dades sensibles o molt sensibles, tals com targetes de crèdit, informació personal, etc.

Els navegadors més moderns ens ofereixen la possibilitat de conèixer si la connexió és segura però no que tant segura és; el concepte de seguretat només implica que estigui xifrat i que el certificat no estigui revocat o caducat (en la gran majoria de navegadors). Però es queda curt en proporcionar informació de més valor pel usuari, com el nivell de seguretat, que sigui segur no és sinònim de seguretat absoluta.

Hem de recordar que el estàndard SSL i TLS [1] no està exempt de vulnerabilitats, en els últims anys s'han detectat múltiples vulnerabilitats [2] (BEAST, CRIME, BREACH, HEARTBLEED, POODLE, FREAK, Logjam), doncs, conèixer quin conjunt de xifrat usa i quina valoració té, és una dada important per conèixer el grau de seguretat.

Per anar una mica més enllà, es planteja la necessitat d'una extensió pel navegador *web* que permeti a l'usuari ser coneixedor de quin grau de seguretat implementa el servidor *web* i, d'aquesta manera, poder decidir la navegació o no, o simplement quines funcionalitats usarà en funció de la seguretat implementada.

Existeixen solucions en el mercat que realitzen aquestes tasques, no obstant, no s'ha trobat cap compatible amb les versions més recents dels navegadors que ja no donen suport als *add-on*. És per aquest motiu que existeix una demanda per programar aquesta extensió com a *WebExtension* [3] i no com la actual *add-on* que la fa inservible pels navegadors més recents, com també més segurs.

1.2 Objectius del treball

L'objectiu d'aquest treball és desenvolupar a partir de solucions ja existents en el mercat una extensió que sigui compatible amb les versions més recents de clients *web* per tal de poder avaluar el nivell de seguretat que un *site* (pàgina *web*) implementa a través dels navegador més recent del conegut *Mozilla Firefox*, i així, gràcies a la API de *WebExtension* [4] poder portar-lo fàcilment a altres navegadors.

1.2.1 Objectius

- Desenvolupar una extensió compatible amb la versió més recent de *Firefox*.

- Mostrar informació de valor per l'usuari sobre la seguretat implementada en el *site* que es s'accedeix.
- Establir un criteri per obtenir i mostrar una valoració global i detallada del nivell de seguretat del *site* que s'accedeix.

Per altre banda, aquesta extensió ha de ser *user-friendly* i capaç de donar informació de valor per a l'usuari i així poder prendre la decisió a nivell de seguretat. Es necessita poder obtenir informació de com de segura és una connexió a un *site* determinat. Actualment hi ha solucions natives en la majoria de navegadors que permeten comprovar certs paràmetres tals com autenticitat (a través del certificat) i el xifrat usat. Però aquesta mera comprovació es queda curta darrera les possibilitats que hi ha a analitzar la informació sobre la seguretat.

Necessitem poder saber que tan forta és la encriptació de les comunicacions:

- De totes les suites que existeixen i son suportades, quines son les més valorades?
- Com de segur és el certificat?
- Quins paràmetres podem valorar per oferir informació a l'usuari sobre seguretat?
- Es pot donar un resultat global sobre la seguretat a l'usuari de la URL visitada?

Existeixen actualment solucions parcials a aquestes preguntes però no compatibles amb les versions més recents dels navegadors (ex.: *Firefox* 57+). Doncs, s'ha de donar resposta amb l'evolució dels *add-ons* que són els anomenats *WebExtensions*.

1.3 Enfocament i mètode seguit

Es parteix de la base que les extensions han evolucionat de les antigues *add-ons* a les noves *WebExtensions* on la API canvia i la metodologia a nivell de programació, com també la compatibilitat (no retro compatibles).

Es decanta per una W*ebExtension* per la falta d'existència d'aquest tipus d'extensió de solucions semblants o iguals a la que es vol desenvolupar. A més a més, cal destacar la portabilitat d'aquest tipus d'extensió a altres navegadors *web* (a través de la API).

Una extensió *web* o *add-on* permet obtenir informació i/o agregar funcionalitats a la navegació de forma còmoda per a l'usuari ja que no és intrusiva a nivell visual, apareix com a una extensió i és tractable fàcilment i sense impactar l'experiència de l'usuari, a més a més de ser dinàmic.

1.4 Planificació del treball final de màster

Aquest treball és planificat durant el primer semestre del curs 2018-2019 amb inici el mes de setembre i final al de gener. S'opta per mostrar una taula de fites i un diagrama de Gantt per a realitzar el seguiment.

1.4.1 Taula de fites

	Nom	Durada	Inici	Final	Descripció ampliada
1	**PAC 1: Definició i Planificació**	12 dies	19/09/2018	30/09/2018	Entrega PAC 1.
1.1	Investigació projecte	7 dies	19/09/2018	25/09/2018	Conèixer el tipus de producte a desenvolupar, a alt nivell com i per a què.
1.2	Definició i planificació	5 dies	26/09/2018	30/09/2018	Definir i planificar el projecte perquè tingui èxit.
2	**PAC 2: Anàlisis**	28 dies	01/10/2018	28/10/2018	Entrega PAC 2.
2.1	Anàlisi i definició dels requisits	1 dia	01/10/2018	01/10/2018	Analitzar i definir quins són els requisits demanats per a dur a terme la investigació i buscar una solució.
2.2	Investigació API *WebExtension*	1 dia	02/10/2018	02/10/2018	Estudi de les WebExtensions per accedir a la informació necessària. Característiques, propietats (API) i comparativa amb un *add-on*.
2.3	*State of art*	3 dies	03/10/2018	05/10/2018	Investigar y analitzar altres possibles solucions existents similars amb especial èmfasis a 'SSLeuth'.
2.4	Definició de la solució	23 dies	06/10/2018	28/10/2018	Definir a alt nivell els passos per obtenir la solució als requisits definits.
3	**PAC 3 : Desenvolupament del treball**	40 dies	29/10/18	07/12/18	Entrega PAC 3.

3.1	Preparació entorn de treball	1 dia	29/10/2018	29/10/2018	Instal·lar i configurar l'entorn per desenvolupar la solució. Importar i configurar les llibreries necessàries per a desenvolupar el producte.
3.2	Desenvolupament de la *WebExtension* (producte)	30 dies	30/10/2018	28/11/2018	Desenvolupament del producte a lliurar.
3.3	Joc de proves i verificació	4 dies	29/11/2018	02/12/2018	Definir, executar i les proves per obtenir resultats del producte i realitzar les possibles correccions al producte.
3.4	Possibles correccions del producte	3 dies	03/12/2018	05/12/2018	Correccions a realitzar en funció del resultat del test (3.3) y tornar a verificar (cíclic).
3.5	Comparativa amb altres soluciones semblants i definició possibles millores futures	2 dies	06/12/2018	07/12/2018	Possible retorn al punt 3.4 comparant amb altres solucions existents. Definir possibles millores futures a implementar.
4	**PAC 4: Memòria del treball**	24 dies	08/12/18	31/12/18	Entrega PAC 4.
4.1	Redacció i correcció (continuada)	89 dies	08/12/2018	28/12/2018	Redacció i correcció continuada en totes les etapes. Sintetitzar totes les entregues.
4.2	Revisió general memòria	3 dies	29/12/2018	31/12/2018	Revisió i possibles correccions final.
5	Elaboració presentació	7 dies	01/01/2019	07/01/2019	Elaboració de la presentació del treball final de màster.

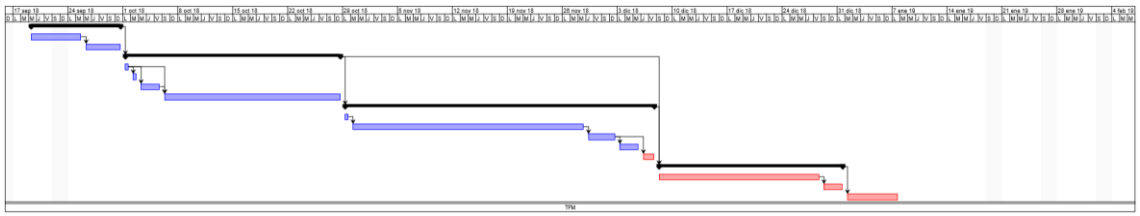
Taula 1: Taula de fites

1.4.2 Gantt

	Nombre	Inicio	Terminado	Predecesores
1	**PAC 1: Definició i Planificació**	**19/09/18 8:00**	**30/09/18 17:00**	
2	Investigació projecte	19/09/18 8:00	25/09/18 17:00	
3	Definició i planificació	26/09/18 8:00	30/09/18 17:00	2
4	**PAC 2: Anàlisis**	**1/10/18 8:00**	**28/10/18 17:00**	**1**
5	Anàlisi i definició dels requisits	1/10/18 8:00	1/10/18 17:00	
6	Investigació WebExtension	2/10/18 8:00	2/10/18 17:00	5
7	State of art	3/10/18 8:00	5/10/18 17:00	5
8	Definició de la solució	6/10/18 8:00	28/10/18 17:00	5;7
9	**PAC 3 : Desenvolupament del treball**	**29/10/18 8:00**	**7/12/18 17:00**	**4**
10	Preparació entorn de treball	29/10/18 8:00	29/10/18 17:00	
11	Desenvolupament de la WebExtension (pro...	30/10/18 8:00	28/11/18 17:00	10
12	Joc de proves i verificació	29/11/18 8:00	2/12/18 17:00	11
13	Possibles correccions del producte	3/12/18 8:00	5/12/18 17:00	12
14	Comparativa amb altres soluciones semblant...	6/12/18 8:00	7/12/18 17:00	12
15	**PAC 4: Memòria del treball**	**8/12/18 8:00**	**31/12/18 17:00**	**4;9**
16	Redacció i correcció (continuada)	8/12/18 8:00	28/12/18 17:00	
17	Revisió general memòria	29/12/18 8:00	31/12/18 17:00	16
18	Elaboració presentació	1/01/19 8:00	7/01/19 17:00	15
TFM				

Taula 2: Taula de Gantt

Diagrama de Gantt:



Il·lustració 1: diagrama de Gantt

1.5 Recorregut treball final de màster

S'ha optat per començar la redacció de la memòria al mateix moment que es comença la primera fita (investigació del projecte) per tal que sigui una redacció continua i al final només s'hagi de revisar i realitzar les correccions i millores pertinents.

Un cop es conegui quins son els requisits, quina és la problemàtica, quina solució es demana, llavors es començarà a definir quines eines es requereixen per començar el desenvolupament del producte final. La part més important és conèixer a quina informació es pot accedir i com, com també quines limitacions ens podem trobar.

Quan el producte tingui forma i es pugui testejar, es definiran les proves pertinents bàsiques i es corregirà el que sigui necessari pel funcionament esperat. Es realitzaran comparacions i contrastos amb altres possibles solucions per veure possibles millores a implementar o simplement recollir-les per deixar-les reflectides com a futures possibles.

Finalment es revisarà la memòria, s'elaborarà la presentació d'aquesta i es realitzarà en si la presentació per a la defensa del treball final de màster. El pla de treball s'ha realitzat a través de l'eina OpenProject (*Open source*) utilitzant el conegut diagrama de Gantt per poder seguir amb èxit el projecte.

2. Anàlisi per a la implementació de l'extensió

Quan es realitza l'establiment de comunicacions segures per un servei *web* els protocols que habitualment s'usen son els de SSL o TLS. Actualment existeix fins la versió TLS 1.3 (2018) [5] i de la SSL la 3.0 (1995), aquest últim protocol ja està obsolet segons el RFC 7568 [6], ja que es considera insegur (com també l'anterior versió 2.0 RFC 6176 [7]).

Versions del protocol per establir un canal segur:

TLS: TLS 1 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246), TLS 1.3 (RFC 8446).

SSL: SSL 2.0 (RFC 6176) i SSL 3.0 (RFC 6101) (ambdós obsolets).

Aquests protocols (SSL/TLS) proporcionen autenticació i confidencialitat de la informació entre extrems sobre xarxes insegures (Internet) mitjançant l'ús de criptografia. Normalment només el servidor és autenticat mentre que el client es manté sense fer-ho.

Fases bàsiques per establir la comunicació segura:

- Negociar [8] (*handshake*) entre les parts el conjunt d'algoritmes criptogràfics que s'usarà en la comunicació.
    - Algoritmes possibles a negociar:
        - Criptografia clau pública: RSA, DH, DSA o Fortezza.
        - Xifrat simètric: RC2, RC4, IDEA, DES, Triple DES i AES.
        - Funcions *hash*: MD5 o SHA.
- Intercanvi de claus públiques i autenticació basada en certificats digitals.
    - Algoritmes intercanvi de claus (definits per TLS 1.2 en el RFC 5246):
        - RSA
        - DH
        - DHE
        - ECDH
        - ECDHE
    - Algoritmes autenticació:
        - DSA
        - ECDSA
        - RSA
    - Algoritmes autenticació i intercanvi de claus (combinació):
        - RSA
        - DH i RSA
        - ECDHE i RSA
        - DH i DSA

        - ECDHE i ECDSA
- Xifrat del tràfic basat en xifrat simètric.
    - Alguns dels algoritmes usats pel xifrat:
        - AES GCM
        - AES CCM
        - AES CBC
        - Camellia GCM
        - Camellia CBC
        - ChaCha20+Poly1305
    - Algoritmes més comuns per calcular el *hash* (integritat dades):
        - MD5
        - SHA-1
        - SHA-2(256 o 384)

2.1 Paràmetres de seguretat a analitzar

Entre els paràmetres a analitzar hem de tenir clar quina informació es negocia i s'usa:

- Certificat(s) digital X.509 de tota la cadena de certificació (entitat final i les CA).
- Protocol (versió SSL/TLS)
- *Cipher Suite* [9] (indica quin conjunt d'algoritmes s'usaran)

2.1.1 Certificat digital

X.509 (RFC 5280) [10] especifica un format estàndard per a certificats de clau pública i un algoritme de validació de la ruta de certificació (cadena de certificats).

La seva sintaxis es defineix usant el llenguatge ASN.1 [11], i els formats de codificació més comuns són [12] DER o PEM.

2.1.1.1 Camps del certificat X.509

Els camps que són d'interès per a valorar la seguretat de la connexió s'indica amb el caràcter asterisc '*'.

- Versió
- Número de sèrie del certificat

- ID del algoritme utilitzat per la CA per firmar (típicament RSA o DSA)*
- Emissor (CA)
- Validesa*
    - No abans de
    - No després de
- Subjecte (subjecte titular), expressat en notació DN: CN, OU, O i C. El subjecte pot ser una persona, un servidor o un servei.*
- Informació de la clau pública del subjecte
    - Algoritme i mida de clau pública*
    - Clau pública del subjecte
- Identificador única de emissor (opcional)
- Identificador únic de subjecte (opcional)
- Extensions (opcional i només per a la versió 3)*
    - *Authority Key Identifier**
    - *Subject Key Identifier*
    - *Key Usage** (**crítica**)
    - *Certificate Policies **
    - *Basic Constraints** (**crítica**)
    - CRL *Distribution Points*
- Algoritme usat per firmar el certificat*
- Firma digital del certificat

Les extensions no existeixen en tots els de la cadena (opcionals i només per a la versió 3) i/o tenen qualitats diferents en cadascun d'ells. Existeixen els crítics i no crítics, els primers no poden tenir valors incorrectes i s'haurien de comprovar per a tota la cadena. Si aquests no són correctes podríem considerar la connexió no segura ja que no podem validar el certificat o com a mínim, han d'obtenir una puntuació molt baixa.

2.1.1.1.1 Extensions del certificat d'interès

A continuació es descriuen les extensions que poden ser d'interès per a valorar la seguretat de la connexió.

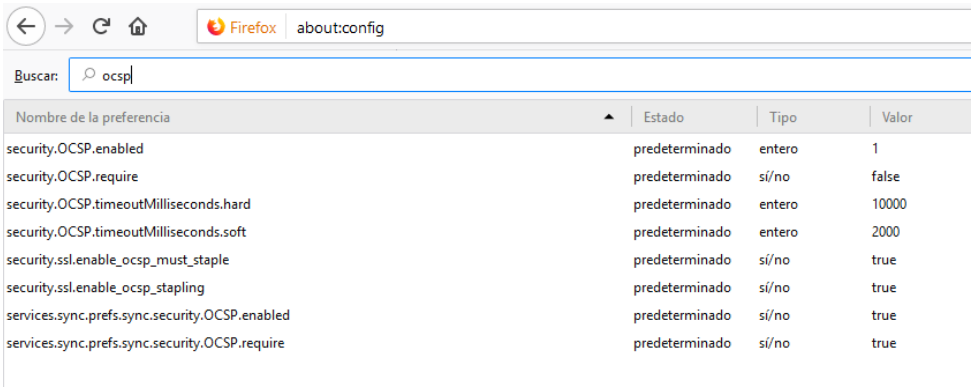
Crítiques:

- *Certificate Key Usage**: indica per a què es pot utilitzar el certificat (firmar un certificat, autenticació i/o xifratge informació durant intercanvi de claus).

- *Certificate Basic Constraints**: indica si el propietari del certificat es d'una CA, i en cas afirmatiu s'indica el número de CA intermèdies que pot expedir el certificat.

No crítiques:

- *CRL Distribution Points*: conté las URI on es pot trobar el fitxer '.crl' que conté els certificats revocats per la CA que ha firmat el certificat. En el certificat arrel no es troba.
- *Authority Information Access*: inclou les URI en les que s'ha de realitzar la petició OCSP per obtenir la informació de revocació del certificat. En el certificat arrel no es troba i per defecte el navegador *Mozilla Firefox* no obliga (no és requisit) a fer la comprovació OCSP per validar el certificat, encara que es pot configurar com a requeriment (configuració navegador).



Il·lustració 2: about:config del navegador web Mozilla Firefox, configuració per a OSCP

- *Certificate Subjecte Alt Name**: conté tots els noms de domini pel qual el certificat és vàlid.
- *Certificate Policies**: en el cas dels certificats de servidors aquests poden passar per un procés de validació de identitat per poder obtenir un certificat amb EV [13] que inclouen el OID de la CA que els ha expedit.

Aquestes extensions son molt interessants per a poder valorar la qualitat de la connexió, tals com els noms alternatius (si coincideix el domini), l'ús o usos de la clau, les restriccions d'ús de la clau i mètodes per consultar l'estat de revocació. Si algun d'aquests no son correctes s'haurien de considerar com a crítiques i puntuar molt negativament el resultat final.

2.1.1.2 Cadena completa de certificació

Perquè el certificat sigui fiable ha de disposar de la cadena completa fins el certificat arrel (CA *root*) que s'ha de trobar instal·lat en el navegador (es confia en els certificats instal·lats) i tots ells han ser vàlids en el temps (normalment: entitat final <-- CA intermedi <-- CA arrel).

Cadascun dels certificats tenen paràmetres a comprovar individualment, encara que poden ser diferents perquè aquests tenen propòsits diferents. Els següents camps s'analitzaran a tots els certificats:

- Algoritme de firma: algoritme usat per la CA emissora per firmar el certificat.

- Període de validesa: indica a partir de quin moment és vàlid i fins quan.
- Algoritme clau pública i mida de la clau: el propietari del certificat té la clau privada que és la parella d'aquesta clau pública (criptografia de clau asimètrica). Aquesta clau pot ser obtinguda a través de diferents algoritmes i de diferents mides (mida de la clau en bits).

2.1.2 *Cipher suite*

Un *string* que conté la informació del conjunt d'algoritmes [14] que s'utilitzaran (negociats): protocol SSL/TLS, l'algoritme *hash* per verificar la integritat de la informació (HMAC), el sistema de xifrat simètrica entre servidor i client per a la confidencialitat (*Bulk Cipher*), l'algoritme de intercanvi de claus (*Key Exchange*) i per la firma.

En el cas del protocol TLSv1.3 la informació de l'algoritme per a l'intercanvi de les claus no apareix contingut en el *string* del *cipher suite*, s'utilitzarà directament la API per obtenir el valor i en cas que retorni un valor inesperat s'usarà un valor estàtic a determinar o un càlcul en funció d'altres paràmetres.

Per altre banda, aquí també obtindrem informació si s'usa el *Perfect Forward Secrecy*; ja que, depèn de l'algoritme que s'usi per l'intercanvi de claus (*Key Exchange*) que són les versions efímeres de Diffie-Hellman (DHE y ECDHE).

Significa que cada sessió tindrà claus noves i aleatòries, doncs si s'aconseguís obtenir alguna clau d'alguna sessió no serviria per desxifrar informació d'altres sessions, ni prèvies ni posteriors. Aquest aspecte és important i de força valor per valorar la seguretat de la connexió.

S'ha d'analitzar quins algoritmes són més forts i més segurs per poder obtenir una estimació de la qualitat de la seguretat de la connexió; tant pel xifrat simètric com per la integritat de les dades (autenticació) i com pel del intercanvi de claus.

2.1.3 Paràmetres de seguretat descartats

A continuació es descriuen paràmetres de seguretat per a la connexió *web* que s'han descartat per diferents motius:

- HPKP: no tindrà pes en la puntuació ja que es considera obsolet [15] (eliminat en la versió Google Chrome 68) i la intenció és que sigui reemplaçat pel *Certificate transparency* en el temps.
- HSTS: s'ha descartat que tingui pes en la puntuació ja que es valoren individualment els elements que influeixen en una connexió HTTPS ja establerta. No obstant, s'informarà de la seva implementació en el servidor [16].
- *Certificate transparency*: s'informarà del estat però no tindrà pes en la puntuació ja que encara es troba en una fase experimental [17].

2.2 Extensions per *Mozilla Firefox*

Les extensions de navegador *web* son complements (*add-on*) que s'agreguen a aquest i permeten afegir funcionalitats vàries a través de llenguatge *web* (JS, HTML i CSS). Les versions anteriors a la 57 funcionen amb els coneguts *add-on,* però actualment s'usen noves extensions que es desenvolupen a partir de la API anomenada *WebExtension*.

Les solucions per analitzar la seguretat de la connexió a un servidor *web* que existeixen en el mercat actualment estan programades com a *add-on* i no son compatibles per les noves versions de *Mozilla Firefox* (57+), com a exemples trobem:

- Calomel SSL Validation [18]
- SSLeuth [19]
- CipherFox [20]

2.2.1 *WebExtensions*

El concepte de *WebExtension* fa que les extensions s'hagin de programar diferent a les conegudes *add-on*; l'objectiu és millorar la portabilitat de les extensions a altres navegadors i facilitar el desenvolupament. Encara que per altre banda, també ha portat limitacions tals com la d'accés a interfícies amb funcionalitats vàries que a dia d'avui no estan incloses a la nova API que es va publicar el novembre del 2017.

Aquesta nova API de *Firefox* és pública a la pàgina *web* per a desenvolupadors, en la que es pot trobar un llistat de funcionalitats que aquesta proporciona a una extensió [21]. Ens interessa sobretot les funcionalitats per accedir a la informació a nivell de seguretat del navegador, doncs, a continuació enumerem les que ens interessa:

- webRequest.getSecurityInfo() [22]

  Aquest mètode permet accedir a la informació de seguretat detallada sobre la connexió TLS associada a una petició en concreta a partir de la versió 62.

  Paràmetres d'interès:

  - certificates: *Array* de 'CertificateInfo' (objecte) amb cada certificat (X.509) de la cadena de certificats, des del servidor *web* fins el *root*. Es pot indicar que ens passi el certificat en format ANS.1 codificat en DER (*raw*), aspecte clau per poder accedir a la informació del certificat que no mostra el certificat descodificat (es mostren certs camps però no tots).
  - *ciphersuite* (opcional): ens retorna un s*tring* amb el *cipher suite* usat per a la connexió amb el format que especifica el protocol SSL/TLS (RFC 5246#appendix-A.5).
  - *errorMessage* (opcional): ens retorna un s*tring* amb un missatge d'error (descriptiu) recollit de la llista d'errors interns del navegador *web* (error en el *handshake*, certificat caducat o revocat, etc.). En principi no s'usarà, però es tindrà en compte.

- certificateTransparencyStatus (opcional): retorna un s*tring* indicant l'estat del 'Certificate Transparency' ("not_applicable", "policy_compliant", "policy_not_enough_scts", "policy_not_diverse_scts").
- hsts (opcional): ens retorna un booleà si el *host* usa el *Strict Transport Security*.
- *isDomainMismatch* (opcional): ens retorna un booleà que indica si el domini del servidor coincideix amb el del certificat. També el comprova en el *subject alternative names*, doncs ens pots estalviar accedir a la part d'extensions per comprovar-ho.
- *isNotValidAtThisTime* (opcional): ens retornarà un booleà indicant si la data actual no està entre el període de validesa (ha caducat o no és encara vàlid).
- *isUntrusted* (opcional): ens retornarà un booleà indicant si la cadena al certificat arrel no s'ha pogut construir.
- *state*: ens retorna un *string* indicant estats de la connexió, només ens interessa:
  - "insecure": la connexió no és TLS.
  - "secure": la connexió és TLS.
  - "broken": ha fallat el *handshake*.
- keaGroupName (opcional): si el valor de la propietat 'state' és igual a 'secure' ens retorna un *string* amb el nom de l'algoritme per a l'intercanvi de claus en la petició. Aquesta informació també la trobem en el *Cipher suite* fins el protocol TLS v1.2.
- *protocolVersion*: ens retornarà un *string* amb la versió del protocol SSL/TLS utilitzat.
- *signatureSchemeName*: si el valor 'state' és 'secure' ens retorna un *string* amb l'esquema per a la signatura (autenticació) usada en la petició.

- CertificateInfo:
  - *fingerprint*: ens retorna un objecte amb el sha-1 i sha256 del certificat codificat amb DER. S'usarà per a presentar-lo.
  - *isBuiltinRoot*: ens retorna un booleà que indica si el certificat es troba entre els certificats *root* instal·lats en el navegador.
  - *Issuer*: ens retorna un *string* com a DN amb el nom de l'organització que ha emès el certificat (qui el firma). S'usarà per a presentar-lo.
  - *serialNumber*: ens retorna un s*tring* amb el número de sèrie (rfc5280#section-4.1.2.2) del certificat. S'usarà per a presentar-lo.
  - *Subject*: ens retorna un *string* (DN) amb el nom de l'organització que va emetre el certificat. S'usarà per a presentar-lo.
  - *validity*: ens retorna un objecte amb el període de validesa (mil·lèsimes de segons des de dijous, 1 gener del 1970) del certificat, 'start' i 'end', ambdós números. S'usarà per a presentar-lo.

Ens trobem que la API no permet accedir a la informació de la clau pública (només el *hash*) i així poder obtenir valors d'interès tals com algoritme usat per crear-la i la mida de la clau. Tampoc podem accedir als valors de les extensions, ni crítiques ni no crítiques, excepte el valor de EV de l'extensió *Certificate Policies* que ens retorna el mètode "isExtendedValidation".

Després de fer consultes a la documentació de la API i a vàries comunitats de suport (Bugzilla i Stackoverflow) he arribat a la conclusió que la única manera d'obtenir aquesta informació és a través d'un dels valors retornats (certificates.rawDER) que es passa com a objecte (*array* de *strings* en UTF-8 del certificat codificat en DER del certificat en notació ASN.1).

Es necessiten llibreries de tercers per tal de descodificar-lo tals com 'asn1js' i 'pkijs' que són públiques; no obstant, les *WebExtensions* tenen la limitació de no permetre la inclusió de llibreries (mòduls) [23] dins d'un JS. Doncs s'ha hagut de cercar algun mètode extern per aconseguir l'objectiu de usar aquestes llibreries.

2.2.2 Llibreries externes

Per aconseguir la inclusió d'aquestes s'usarà l'eina *WebPack*, que ens permet a través d'una configuració concreta empaquetar tants mòduls (llibreries) com vulguem en un sol JS i així poder carregar-lo sense problema en la *WebExtension*.

- *asn1js*:

  ASN1.js és una llibreria *JavaScript* que implementa aquest estàndard. ASN.1 és la base de qualsevol estructura X.509 de dades relacionada i molts altres protocols l'usen a la *web*.

- *pkijs*:

  PKIjs és una llibreria JS que implementa formats que son usats en aplicacions PKI (firma, xifrat, peticions de certificats, OCSP i TSP peticions/respostes). Està construït en WebCrypto (Web Cryptography API) i no necessita cap *plug-in*.

  Pot treballar amb objectes de dades HTML5 (ArrayBuffer, Uint8Array, Promises, Web Cryptography API, etc.).

  Tot tipus de ASN.1 *strings*, tals com el 'UniversalString', 'UTF8String' i 'BMPString' (amb l'ajuda de ASN1js).

2.3 Anàlisi d'extensions actuals en el mercat

Després d'una investigació de les extensions actuals, ja sigui *WebExtentions* o com a *add-ons*, s'han analitzat quines dades s'estudien de la connexió segura com també a alt nivell l'estructura d'aquesta. No s'ha entrat a profunditzar en l'arquitectura del complement ja que estan programades (les que s'han trobat) com a *add-on* i no *WebExtension*, doncs, no ens aportarà pràcticament valor a nivell de codi ni estructura.

El que sí s'ha estudiat és l'arquitectura i codi a alt nivell d'alguna *WebExtension* a mode d'exemple [24] referenciada a la documentació de la API per tal de conèixer el funcionament i les possibilitats d'aquestes extensions.

Les diferències entre les solucions trobades en el mercat són molt semblants, en la majoria de casos és més un canvi en el disseny i/o interfície gràfica, però he trobat a faltar solucions que valorin la connexió amb criteris i així proporcionar aquesta informació d'interès a l'usuari.

A continuació anem a mencionar les més destacades:

- *CipherFox*: ens permet accedir a la informació de la seguretat de la connexió, el *Cipher suite* utilitzat i a tota la informació de la cadena de certificació. No indica cap paràmetre com a positiu o negatiu, sinó que només informa.

  Semblant a *SSLeuth* amb la diferència que no puntua (no valora) la connexió, ni la valora a nivell visual/descriptiu, sinó que només informa. L'usuari ha de ser coneixedor de la informació (avançat) per poder treure conclusions sobre la seguretat de la connexió. No emmagatzema ni exporta les dades obtingudes per una posterior consulta. No comprova la revocació del certificat.

  Calomel SSL Validation: Té capacitat per a valorar la seguretat de la connexió i accedir a la mateixa tal com ho fa *SSLeuth*, encara que el càlcul de la puntuació es realitza amb altres criteris. No accedeix a tota la cadena de certificació ni tampoc la valora implícitament.

  Tampoc permet la personalització de la ponderació dels camps a estudiar per a la seguretat. Tampoc exporta les dades ni les emmagatzema pel seu tractament. No comprova la revocació del certificat.

- *SSLeuth*: La solució que s'apropa més al objectiu que volem aconseguir, un *add-on* que permet accedir a la informació de la connexió i mostra tota la informació de valor per a valorar globalment si la connexió és segura o no ho és(de 0 a 10).

  No requereix que l'usuari sigui avançat ja que puntua cada apartat i ressalta els camps de forma visual (color verd, blau o vermell). Permet configurar la ponderació de cada camp estudiat per calcular la qualitat de la seguretat de la connexió.

  Trobo a faltar que calculi valors de tota la cadena de certificació. No fa menció del protocol OCSP (no es comprova la revocació) ni tampoc compara el valor obtingut amb un màxim que s'hagi pogut obtenir mai (cap de les extensions ho compara amb cap referència).

  Calcula la qualitat també de tots els dominis que s'estableix una connexió des de la pàgina visitada, fent el mateix càlcul. La part de configuració dels *Cipher suites* permet establir quines *suites* habilitar (obsolets i/o deshabilitats per defecte en certs navegadors, sobretot els moderns).

  El *add-on* també té la funcionalitat de comunicar si la connexió no és HTTPS (no usa el protocol segur) i informar de la possible versió alternativa (segura); ofereix la URL canviant HTTP per HTTPS però sense comprovar que realment existeixi.

  Els paràmetres que proporciona *SSLeuth* son:

  - *Cipher suite*:
    - Versió protocol (SSL/TLS)
    - Algoritme de intercanvi de claus (també valora el *Forward Secrecy*)
    - Algoritme de xifrat simètric
    - Algoritme de *hash*

- Certificat servidor *web*:
  - Clau pública del servidor (algoritme i mida clau)
  - Algoritme per a la firma del servidor
  - Validesa del certificat del servidor (també si és estès)
- Els paràmetres que **no** proporciona *SSLeuth* son:
  - Extensions del certificat
  - Informació de revocació del certificat
  - Valoració sobre els certificats que conformen la cadena

Ponderació:

A continuació es pot observar com calcula la valoració global de la connexió, aquesta informació s'ha obtingut des del panell de preferències de l'extensió i del fitxer 'ciphersuites.js' inclòs en l'extensió:

- Global:
  - *Cipher suite*: 40%
  - *Forward secrecy*: 20%
  - *Extended validation*: 10%
  - Estat de la connexió: 10%
  - Estat del certificat: 10%
  - Signatura: 10%

- *Cipher Suite*:
  - Algoritme de intercanvi de claus: 30%
  - *Bulk Cipher*: 30%
  - HMAC: 40%

El fet que sigui HTTPS et dona 1 punt sobre 10, com també que el certificat sigui vàlid (període de validesa), i per altre banda si és estesa la validació (EV) obté 1 punt extra. El *forward secrecy* el valora en el global encara que és una propietat derivada del algoritme per al intercanvi de claus (els efímers) que cau en el *cipher suite*.

Podem observar que el protocol usat (SSL/TLS) no es pondera en el *cipher suite*, observant el codi traiem la conclusió que no ho inclou enlloc.

A continuació es mostra una taula resum amb el que analitza i el que no el *add-on* SSLeuth:

Analitza	No Analitza
Versió TLS/SSL	Extensions del certificats
Algoritme intercanvi claus	Informació de revocació (es pot usar protocol OCSP per aconseguir-ho)
Algoritme de xifrat simètric	Realitzar comprovacions en tots els certificats de la cadena completa
Algoritme de *hash*	Tenir una puntuació de referència (ex.: la més segura o segures de Internet).
Algoritme creació clau pública del servidor i mida de la clau	*Certificate Transparency* (experimental però a considerar informar)
Algoritme de la firma del certificat del servidor	HTTP Strict Transport Security (considerar informar)
Validesa del certificat del servidor *web*	

2.3.1 Estructura de SSLeuth

S'analitzarà a alt nivell com està estructurada la informació a nivell de codi i les funcions; s'ha de tenir en compte que l'extensió que programarem es basa en una altre API (*WebExtensions* API) i l'accés a la informació de seguretat no és obtinguda de la mateixa manera, ni tan sols l'estructura de l'extensió.

El fitxer que s'observarà és del de 'cipher-suites.js' que s'usarà com a referència per a la ponderació i nomenclatura dels algoritmes i protocols de la connexió segura.

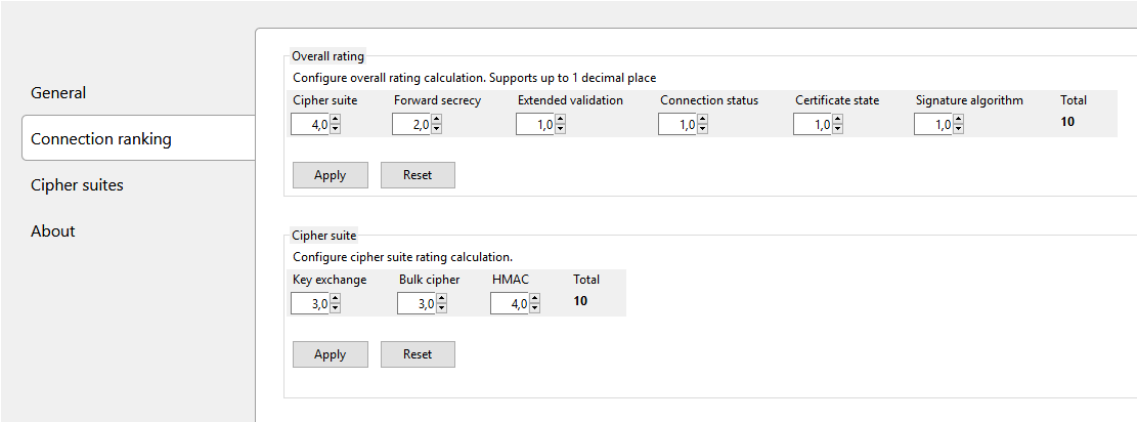
- 'cipher-suites.js':

  Dos constants format JSON amb la següent informació:

  - *Cipher Suites*:
    - *keyExchange*: nom, *rank*, *pfs*, ui i notes. (ui -> descripció)
    - *Authentication*: nom, *rank*, mida de la clau mínima, ui, cert i notes.
    - *bulkCipher*: nom, *rank*, ui i notes.
    - HMAC: nom, *rank*, ui, sigui i notes.
    - Puntuacions: nom –> valor numèric
    - Pesos puntuació : nom -> valor numèric
  - *connectionRating*: conceptes i pesos.

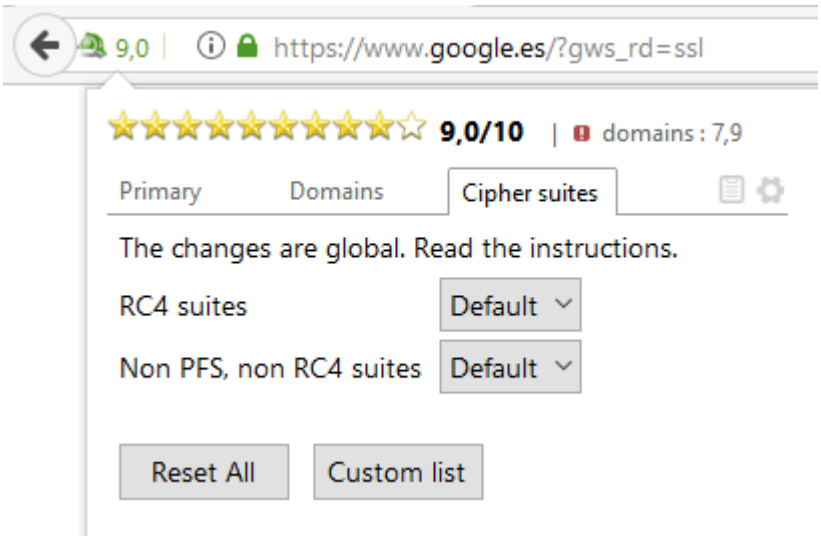
2.3.1.1 Funcionalitats

A continuació podem veure què ens deixa configurar l'extensió a nivell de càlcul i, així també conèixer què valora més i menys per a obtenir la puntuació global (pesos):



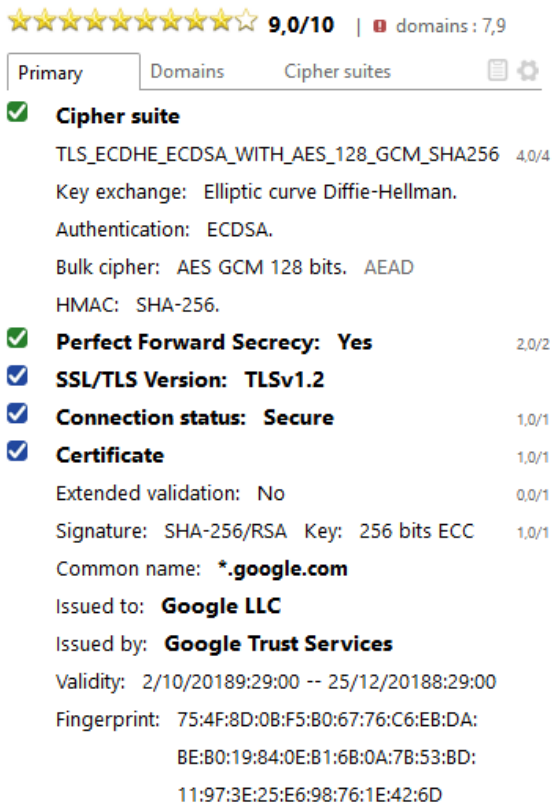
Il·lustració 3: panell de configuració SSLeuth

Aquest complement permet a l'usuari habilitar alguns c*ipher suites* que estan normalment deshabilitats, aquesta funcionalitat no la tindrem en compte per a la nova extensió ja que no aporta cap valor a conèixer si la connexió establerta és segura o no, i quin valor numèric té. Tampoc la part de 'Domains' ja que no analitza directament la connexió al servidor *web* que hem fet la petició, sinó altres dominis que internament es fan peticions.



Il·lustració 4: funcionalitat habilitació suites insegures

A continuació podem observar quina informació ofereix a l'usuari a través de l'extensió:



Il·lustració 5: informació connexió SSLeuth

Tenim informació sobre el certificat i sobre la connexió, com també la puntuació per cada part sobre el total i la puntuació global. Aquest complement no exporta la informació ni l'emmagatzema en local per a un posterior tractament o consulta.

2.4 Anatomia d'una *WebExtension*

La estructura d'una extensió [25] es compon d'una col·lecció de fitxers, empaquetats per a la seva distribució i instal·lació, a continuació es mostrarà aquesta:

- manifest.json: tota extensió ha de contenir aquest fitxer (format JSON) amb aquest mateix nom. Conté les metadades bàsiques de l'extensió: nom, versió, permisos i enllaços a altres fitxers de l'extensió.
    - Camps obligatoris:
        - applications (només suportat en Gecko, motor de presentació)
        - manifest_version (versió del manifest, actualment la 2)
        - name (nom de l'extensió)
        - version (versió de l'extensió)
    - Camps opcionals:

- background
- browser_action
- page_action
- content_scripts
- default_locale (només si la carpeta 'locales' pels idiomes existeix)
- description
- permissions
- web_accessible_resources

- *Background page*: implementa la lògica de llarga execució. Podrà estar format per HTML i JS.
- *Browser action*: implementen els botons de la barra d'eines del navegador. Els components que el formin (HTML, JS i CSS) hauran d'estar a una carpeta anomenada 'popup'. S'haurà de indicar també quines icones el conformaran.
- *Page action*: implementen botons a la barra de direccions. Igual que el cas anterior, el contingut haurà d'estar inclòs en una carpeta anomenada 'popup' i indicar les icones a usar.
- Content scripts: *scripts* que interactuen amb les pàgines *web* (per exemple per modificar elements del HTML en la pàgina visitada). Podrà estar format per JS i CSS.
- Default_locale: contindrà els diferents fitxers de propietats pels possibles diferents idiomes (opcional).
- description: descripció de la extensió.
- permissions: declaració del permisos que l'extensió necessita.
- *Options page*: defineixen una interfície per a l'usuari per poder veure i canviar les configuració de l'extensió (HTML, JS i CSS).
- *Web accessible resources*: recursos (HTML, CSS i JS) que es poden incloure a l'extensió i es volen fer accessibles als scripts en segon pla i els scripts de les pàgines.

Estructura directoris extensió:

/rankSSL

/icons (icones de l'extensió)

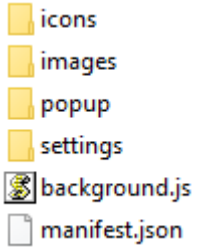
/images (imatges que s'incorporaran al 'popup' de l'extensió)

/popup (html, css i JavaScript pel 'popup' de l'extensió)

/settings (fitxers per al panell de opcions/configuració extensió)

background.js

manifest.js

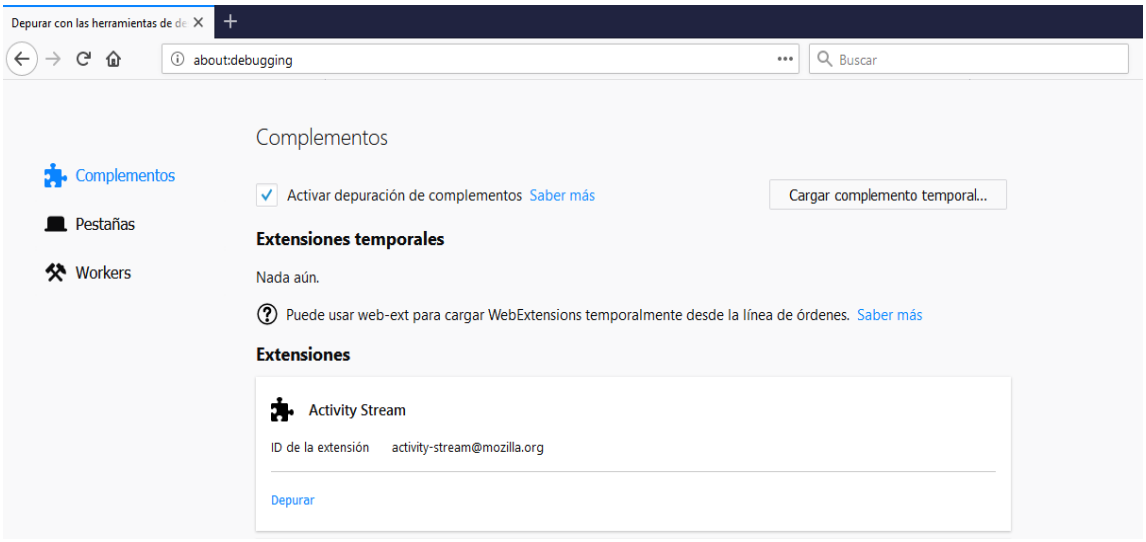


Il·lustració 6: directori arrel extensió

2.5 Instal·lació de les *WebExtensions*

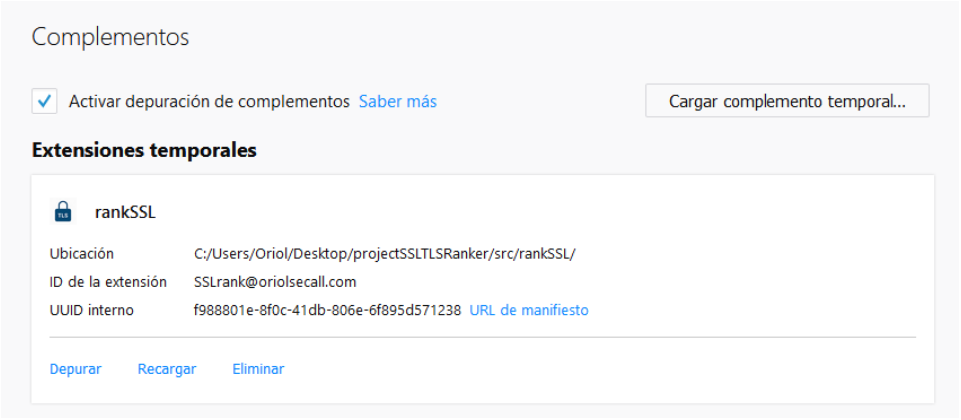
Per poder desenvolupar l'extensió s'haurà de poder carregar-la al navegador i provar-la; s'ha de tenir en compte que aquesta no estarà publicada al repositori oficial (ha de passar la verificació pertinent) d'extensions de *Firefox*, doncs, el navegador proporciona la possibilitat de instal·lar-la temporalment, i així poder fer *debugging* (depurar-la) i provar-la (si s'intenta instal·lar sense verificar es mostra un error).

Una de les maneres és escrivint 'about:debugging' en la barra de direccions del navegador per tal així de carregar el complement (extensió) temporalment (s'ha de seleccionar el fitxer del manifest). A més a més, si activem la casella 'Activar depuración de complementos' ens permetrà a través de la consola de desenvolupament depurar l'extensió i programar-la amb garanties.



Il·lustració 7: about:debugging Mozilla Firefox

En la següent imatge podem observar les opcions de 'Depurar' per iniciar la consola de depuració, 'Recargar' per reinstal·lar l'extensió i 'Eliminar' per descarregar-la del navegador.



Il·lustració 8: extensió temporal per depurar al navegador Mozilla Firefox

Una alternativa és usar l'eina 'web-ext' que s'ha de instal·lar amb 'npm' [26] (requisit tenir instal·lat prèviament 'nodejs' [27]). Un cop instal·lat, si ens dirigim al directori de l'extensió i cridem la comanda 'web-ext run' es carregarà al navegador temporalment de forma automàtica.

2.6 Mòdul JS WebPack

Aquesta eina [28], tal i com s'ha comentat anteriorment, ens permetrà introduir llibreries que d'altre manera seria impossible degut a les restriccions de les *WebExtensions*. La idea d'aquesta eina és empaquetar tots els mòduls amb dependències JS i crear un 'bundle' que els agruparà tots; és a dir, un sol fitxer JS amb totes les dependències i el codi del principal.

Per tal de instal·lar-la i poder usar les llibreries (mòduls) es necessita l'eina 'npm'; un administrador de paquets de JS que ens permetrà instal·lar el mateix 'web-pack' i els mòduls que es necessitin directament per línia de comanda.

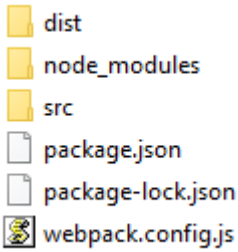
D'aquesta forma podrem incloure el fitxer generat 'bundle.js' al fitxer HTML que carregarà el codi (*popup* de l'extensió) que processa la informació sobre la seguretat i ens 'saltem' la 'restricció' que ens trobem per desenvolupar l'extensió.

Per poder instal·lar-lo necessitem l'eina 'npm' i introduir el directori font de l'extensió com a subdirectori 'src' de l'estructura 'web-pack'. Les llibreries que es necessitin també s'instal·laran usant l'eina 'npm' a nivell local degut a la facilitat que proporciona (ex.: "npm i pkijs" i ja tenim els mòduls instal·lats i disponibles a la carpeta 'node_modules' de l'arrel).

Estructura del directori per usar web-pack:

/rateWebServerSSL_TLS

- /dist (aquí surt el fitxer 'bundle.js')
- /node_modules (mòduls instal·lats externs per 'npm')
- /src (aquí col·locarem la *WebExtension*)
  - /rankSSL (*WebExtension* 'rankSSL')
- package.json (configuració web-pack)
- package-lock.json
- webpack.config.js (configuració web-pack)



Il·lustració 9: directori arrel web-pack

Instal·lació i producció fitxer 'bundle.js':

1) npm init -y

2) npm install webpack webpack-cli

3) npm install --save lodash

4) npx webpack --config webpack.config.js

5) Modificar el fitxer Package.json:

```
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
+   "build": "webpack"
  },
```

6) npm run build

7) El fitxer 'bundle.js' col·locar-lo en el html desitjat a carregar-lo com a script (*JavaScript*).

Fitxer 'webpack.config.js':

- entry: fitxer *JavaScript* on es declaren les dependències a usar.
- output: el nom del fitxer que es generarà i agruparà totes les dependències i el directori on es col·locarà.

```
const path = require('path');

module.exports = {
  entry: './src/rankSSL/popup/main.js',
  output: {
    filename: 'bundle.js',
    path: path.resolve(__dirname, 'dist')
  }
};
```

Il·lustració 10: contingut fitxer webpack.config.js

2.7 Anàlisi dels algoritmes i característiques de seguretat

Necessitem comparar els algoritmes utilitzats per usar el criteri emprat i poder fer una valoració quantitativa, doncs, s'ha cercat quines fonts poden ser d'interès, a part, és clar, de la pròpia informació continguda en el *add-on* SSLeuth ja existent ja analitzat.

A part dels RFC dels algoritmes que es poden usar en l'establiment de la connexió segura podem usar addicionalment altres fonts per determinar la qualitat i/o força d'aquests.

- (Qualys) [29]

  Ens permet obtenir tota la informació de seguretat sobre la connexió a un servidor *web* que ens interessa i una valoració global. També ofereix llistes amb les pitjors i millors *web sites* a nivell de seguretat. Molt útil per comprar les nostres valoracions i establir en més detall el criteri si és necessari (hi ha documentació de bones pràctiques i informació sobre algoritmes i protocols).

- (BadSSL) [30]

  Ens permetrà posar a prova les valoracions en diferents escenaris ja que disposa de dominis amb certificats amb característiques diferents: revocats, caducats, auto-firmats, amb arrel no de confiança, amb diferents algoritmes, etc.

- (OpenSSL) [31]

  OpenSSL té documentació sobre els diferents protocols SSL/TLS i els *Cipher suites* que s'usen; informació de molt valor per a poder valorar quins algoritmes s'usen en cada fase.

- (CCN-Cert) [32]

  El Centre criptològic nacional té documentació de bones pràctiques per a configurar servidors *web* on s'indica informació sobre els algoritmes que són de valor pel criteri de la puntuació.

- (RFC) [33]

  S'han consultat els RFC necessaris però a destacar els RFC 8247 i 7465 que informen d'algoritmes per al intercanvi de claus i els no a utilitzar.

- (Mozilla *community*) [34]

  Aquí també podem trobar documentació sobre característiques de seguretat per a connexions HTTP o HTTPS d'interès, com per exemple el HPKP.

- (Wikipedia) [35]

  Trobem informació sobre el protocol SSL/TLS i possibles algoritmes que hi participen, com també sobre la seguretat en connexions a nivell general i també específicament.

- (BlueKrypt) [36]

  Es mostra informació sobre recomanacions de mida de claus per algoritmes de generació de claus en funció de l'any.

- (IBM) [37]

  Ens servirà per conèixer algoritmes més comuns amb els seus respectius OID.

- (JavaDoc) [38]

  Conté informació sobre algoritmes existents per encriptar informació.

- (OWASP) [39]

  Trobem informació sobre implementacions dèbils pel protocol SSL/TLS.

2.8 Criteri puntuació

Primer de tot s'ha consultat el RFC 5246 per poder conèixer bé el protocol SSL/TLS i poder tenir un criteri base per valorar els diferents algoritmes que s'usen en el protocol.

Qualys, Inc. (NASDAQ: QLYS) és un proveïdor pioner i líder en seguretat basada en el núvol i solucions de compliment amb més de 9300 clients en més de 100 països (la majoria dels 100 Forbes Global i Fortune 100), doncs, s'utilitzarà per estudiar possibles valoracions de la qualitat de la connexió a nivell de la seguretat.

Segons Qualys, aquests següents factors son motius per puntuar zero la connexió global:

- Domini no coincideix
- Certificat no és encara vàlid
- Certificat ha caducat
- Certificat auto-firmat

- No es confia en el certificat (CA desconeguda o error en la validació)
- Certificat revocat
- Firma del certificat insegura (MD2 o MD5)
- Clau insegura

Agafarem com a vàlid aquest criteri excepte que el certificat tingui una firma amb l'algoritme MD5, que es valorarà individualment per l'apartat de l'algoritme de firma. La validesa del certificat farà caure substancialment la puntuació global però no a 0 directament, ja que encara es poden valorar altres conceptes individualment que aporten seguretat a la connexió.

Veiem ara quins pesos i puntuacions estableix per a valorar la connexió global segons la guia que podem trobar a Qualys [40]:

- Global:

Category	Score
Protocol support	30%
Key exchange	30%
Cipher strength	40%

- Protocol:

Protocol	Score
SSL 2.0	0%
SSL 3.0	80%
TLS 1.0	90%
TLS 1.1	95%
TLS 1.2	100%

El SSL 3.0 es considera obsolet i es puntuarà en qualsevol cas amb 0% dels punts possibles.

- *Key Exchange*:

Key exchange aspect	Score
Weak key (Debian OpenSSL flaw)	0%
Anonymous key exchange (no authentication)	0%
Key or DH parameter strength < 512 bits	20%
Exportable key exchange (limited to 512 bits)	40%
Key or DH parameter strength < 1024 bits (e.g., 512)	40%
Key or DH parameter strength < 2048 bits (e.g., 1024)	80%
Key or DH parameter strength < 4096 bits (e.g., 2048)	90%
Key or DH parameter strength >= 4096 bits (e.g., 4096)	100%

- *Cipher suite*:

Cipher strength	Score
0 bits (no encryption)	0%
< 128 bits (e.g., 40, 56)	20%
< 256 bits (e.g., 128, 168)	80%
>= 256 bits (e.g., 256)	100%

A partir del 21 de Gener del 2014 es van realitzar uns canvis importants a tenir en compte:

- La mida de les claus per sota de 2048 bits es considerant 'weak' i les per sota 1024 insegures.
- Per aconseguir màxima puntuació es necessita el protocol TLS 1.2. o superior.
- Firmes amb MD5 es consideren 'weak'.
- SHA1 es considera insegur.
- Es necessita el xifratge amb propietat AEAD i/o *Forward Secrecy* per tal d'aconseguir la màxima puntuació.

SSLeuth puntuació algoritmes:

- *Key Echange*:
  - ECDHE :10
  - ECDH: 9
  - DHE: 9
  - DH: 9
  - RSA: 6
  - RSA_EXPORT: 2
- *Authentication*:
  - RSA: 10 (2048)
  - ECDSA: 10 (256)
  - DSS: 10 (2048)
- *BulkCipher*:
  - CHACHA20_POLY1305: 10 (AEAD)
  - AES_256_GCM: 10 (AEAD)
  - AES_128_GCM: 10 (AEAD)
  - AES_256_CBC: 8
  - AES_128_CBC: 8
  - CAMELLIA_256_CBC: 8
  - CAMELLIA_128_CBC: 8
  - SEED_CBC: 8
  - 3DES_EDE_CBC: 7

- HMAC:
    - SHA512: 10
    - SHA384: 10
    - SHA256: 10
    - SHA224: 8
    - SHA: 4
    - MD5: 2

3. Desenvolupament d'una extensió per a l'anàlisi de la connexió a un servidor *web*

En aquest apartat es realitzarà el desenvolupament de l'extensió per a *Mozilla Firefox*; començarem utilitzant la API de W*ebExtensions* per accedir a la informació sobre la seguretat de la connexió *web* i seguidament, els paràmetres que no hem pogut obtenir directament utilitzarem altres llibreries (mòduls) per tal d'obtenir-la i aconseguir els objectius establerts.

Les dades que s'obtinguin s'usaran per alimentar l'algoritme estudiat en l'apartat 2 i així extreure un resultat que ens doni informació sobre la qualitat de la seguretat en la connexió. Un cop es tingui una extensió funcional i que proporcioni la informació desitjada, es començaran a fer probes per tal de depurar i/o millorar l'extensió.

A tenir en consideració que el navegador *web Mozilla Firefox* actualment ve amb una configuració per defecte on certs algoritmes considerats febles o insegurs no son acceptats (es bloqueja l'accés a la pàgina *web*), doncs, hi ha certs escenaris (algoritmes emprats pel servidor) que sense realitzar canvis manualment en el navegador no es podran arribar a donar.

Per exemple, no considera confiable la cadena de certificació si el certificat del CA intermedi està firmat amb SHA-1. Per realitzar canvis es poden fer des de 'about:config' [41] apartat 'security' i/o usant, com a recomanació, l'extensió 'toggle Cipher' que ens permet habilitar certs algoritmes deshabilitats per defecte per seguretat.

3.1 Algoritme de puntuació

Un cop coneixem totes les possibilitats que tenim per accedir a la informació desitjada (a nivell de seguretat de la connexió) concretem l'algoritme que calcularà la puntuació total sobre la qualitat de la seguretat.

Aquest algoritme primerament determina si es pot valorar el càlcul (protocol segur de HTTP), si ho és llavors entrem a la part on es realitzaran les comprovacions més crítiques de la seguretat, que ens cas que no es superin ens portarà a un resultat de puntuació zero, i en cas de superar-se a un valor quantitatiu.

Comprovacions crítiques:

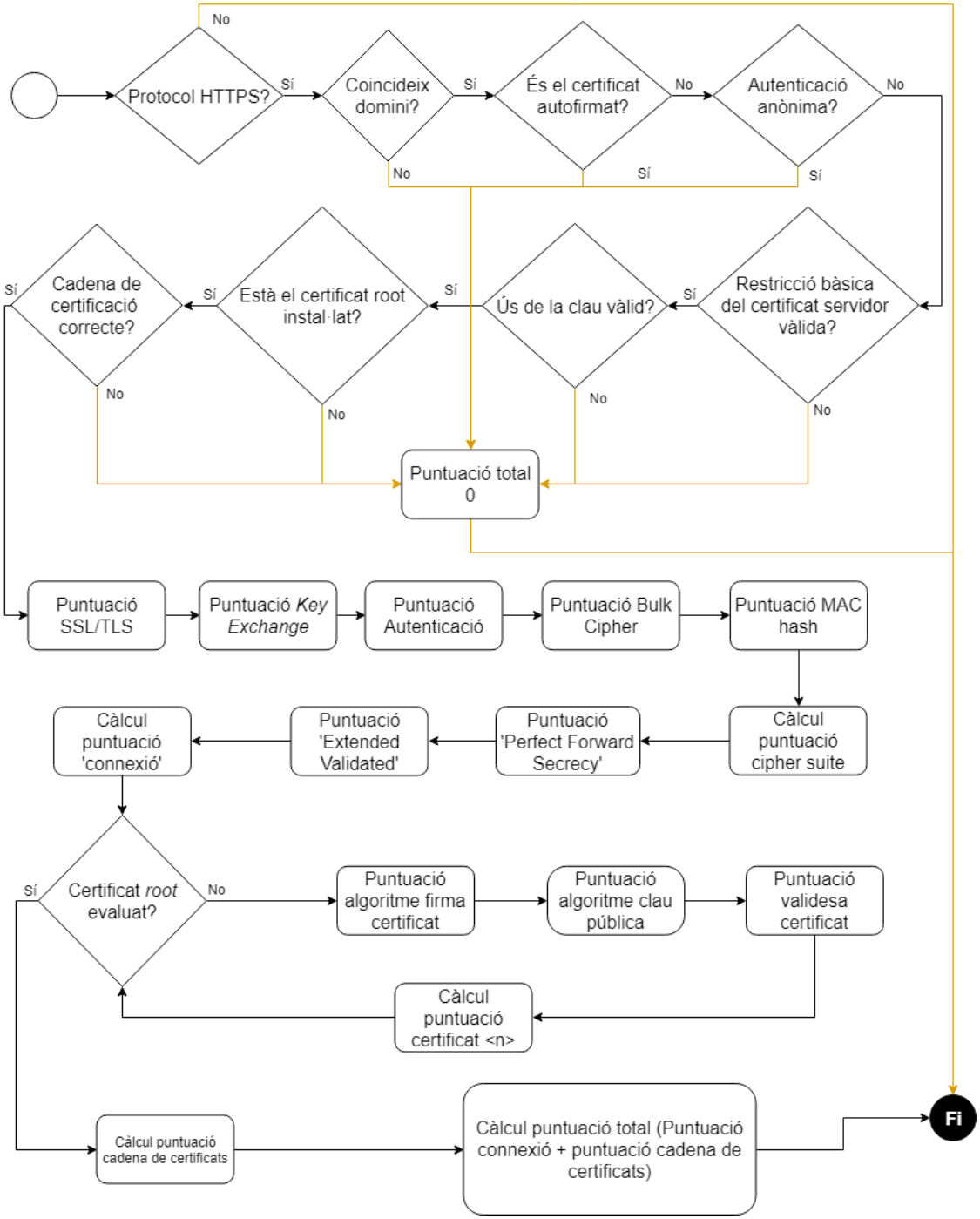
- Coincideix el domini: si no coincideix el domini, el certificat no aplica al lloc *web* que consultem i determinem que la puntuació és zero ja que la connexió és totalment insegura.
- Certificat auto-firmat: un certificat auto-firmat no ha estat firmat per una autoritat certificadora, doncs, no hi podem confiar.
- Cadena de certificació correcta: si no s'ha construït la cadena de certificació completa significa que no podem confiar en el certificat.
- Autenticació anònima servidor: servidor web no s'autentica.
- Certificat 'root' instal·lat en el navegador: si no es troba el certificat *root* instal·lat no podem confiar en el certificat de l'entitat final que ha estat firmat per la autoritat certificadora i que aquesta a la vegada ha estat firmada pel certificat arrel. Se sobreentén

que si el certificat instal·lat (per part de l'usuari o venia per defecte) al navegador és perquè l'usuari hi confia totalment.

- Ús de la clau del certificat vàlid: si es detecta un ús de clau no vàlid per les restriccions d'ús del certificat (CA o entitat final) es valorarà amb un zero.
- Restricció de l'ús del certificat servidor vàlid: si el camp de les extensions 'basic constraints' indica un ús que no correspon en el que hauria de ser, es valorarà amb una puntuació a zero.

Posteriorment després d'aquestes comprovacions es realitzaran els càlculs (ponderació) a través de les consultes pertinents per tal d'obtenir un resultat quantitatiu i mostrar-lo a través de l'extensió. Aquests càlculs es basen en vàries fórmules que tenen pesos (de 0 a 1) i variables (puntuació de 0 a 10) que es mostren en el punt 3.3.2. Cal recordar que aquests pesos son configurables a través de les opcions de l'extensió.

3.1.1 Diagrama de flux de l'algoritme



3.1.2 Fórmules càlcul puntuacions

3.1.2.1 Puntuació total

Puntuació total = (Puntuació connexió · Pes connexió) + (Puntuació cadena certificació · Pes Cadena certificació);

- Puntuació connexió: es valora només els paràmetres que afecten directament a la connexió del client amb el servidor *web*.
- Puntuació cadena certificació: es valoren els mateixos paràmetres en tota la cadena de certificació i el total és el sumatori de tots ells amb els pesos que li correspon a cadascun.

3.1.2.2 Puntuació connexió

Puntuació connexió = (Puntuació *cipher suite* · Pes *cipher suite*) + (Puntuació EV · Pes EV) + (Puntuació PFS · Pes PFS);

- Puntuació *cipher suite*: ens proporciona molta informació per a valorar la seguretat del connexió.
- Puntuació EV: és un camp que només es troba en el certificat de l'entitat final, doncs s'ha considerat agregar-lo en el càlcul de la puntuació de la connexió i deixar els mateixos paràmetres de càlcul de la cadena de certificació per igual. EV aporta més fiabilitat en la connexió a nivell d'autenticació (domini ha estat validat de forma estesa). La puntuació és binaria, o nul·la o màxima (0 o 10).
- Puntuació PFS: S'ha valorat l'opció de valorar l'algoritme efímer més positivament i no incloure'l com a variable en el càlcul de la puntuació de la connexió, no obstant, es considera determinant, ja que agrega molta seguretat.
  El xifratge és només vàlid per a la sessió establerta i la considero una variable de càlcul per a la connexió, encara que sigui una propietat derivada de l'algoritme emprat en el *key exchange*. La puntuació és binaria, o nul·la o màxima (0 o 10).

3.1.2.3 Puntuació cipher suite

Puntuació *cipher suite* = (Puntuació protocol TLS · Pes protocol TLS) + (Puntuació *Key Exchange* · Pes *Key Exchange*) + (Pes Autenticació · Puntuació Autenticació) + (Puntuació *Bulk Cipher* · Pes *Bulk Cipher*) + (Puntuació *Hash* · Pes *Hash*);

- Puntuació protocol TLS: es prendrà el valor en funció del protocol emprat (SSL o TLS).
- Puntuació *Key Exchange*: en funció de l'algoritme emprat.
- Puntuació Autenticació: en funció de l'algoritme emprat.
- Puntuació *Bulk Cipher*: en funció de l'algoritme emprat, en cas que utilitzi el mode AEAD s'informarà i es valorarà positivament.
- Puntuació *Hash*: es valora l'algoritme emprat.

3.1.2.4 Puntuació cadena certificació

Puntuació cadena certificació = $\sum_{i=0}^{n}$ Pes Certificat$_i$ · ((Puntuació validesa · Pes Validesa) + (Puntuació Algoritme Clau Pública · Pes Algoritme Clau Pública) · (Puntuació Algoritme Firma · Pes Algoritme Firma));

Aquesta puntuació és obtinguda a partir de la suma de tots els certificats de la cadena i aquesta es valora sempre amb els mateixos paràmetres (validesa del certificat en el temps, algoritme i mida de la clau pública, com també l'algoritme i mida emprat per a la firma del certificat), no obstant que cada certificat té el seu pes sobre el càlcul final determinat.

- Puntuació validesa: la puntuació és binaria, o és vàlid en el temps o no ho és (0 o 10).
- Puntuació algoritme Clau Pública: en funció de l'algoritme emprat i la mida de la clau.
- Puntuació algoritme Firma: en funció de l'algoritme emprat i la mida de la clau.

Tal i com s'ha explicat en el capítol 2 les valoracions dels algoritmes s'han extret de vàries fonts, tals com dels RFC: 8247 i 7465, del NIST, de Qualys, de BlueKrypt, de CCN-CERT i s'han valorat *add-ons* ja publicades on ja existien valors en les variables (algoritmes) a calcular la puntuació.

S'ha analitzat específicament l'algoritme ECDSA (RFC 6979 [42]) i la implementació de la EC en el protocol TLS en els RFC: 7748 [43] , 4492 [44], 5656 [45], ja que la API ens proporciona informació pel *key Exchange* i *signature* (autenticació).

3.1.3 Taules ponderacions algoritmes

3.1.3.1 Taula puntuació cipher suite

Versió protocol SSL/TLS		***Key Exchange***		***Signature* (Autenticació)**		***Bulk Cipher***		**MAC *hash***	
SSL 0.2	0	ECDHE (x2559, 256 bits)	9	ECDSA-P192-SHA160 (192 bits)	4	CHACHA20_POLY1305 (256 bits i AEAD)	10	SHA 512	10
SSL 0.3	0	ECDHE (x448, 448 bits)	10	ECDSA-P224-SHA224 (224 bits)	5	AES_256_GCM (256 bits i AEAD)	10	SHA 384	10
TLS 1.0	0	ECDHE (P192, 192 bits)	4	ECDSA-P256-SHA256 (256 bits)	9	AES_128_GCM (128 bits i AEAD)	9	SHA 256	9
TLS 1.1	6	ECDHE (P224, 224 bits)	5	ECDSA-P384-SHA384 (384 bits)	10	AES_128_CCM_8 (128 bits i AEAD)	9	SHA 224	4
TLS 1.2	10	ECDHE (P256, 256 bits)	9	ECDSA-P521-SHA512	10	AES_128_CCM (128 bits i AEAD)	9	SHA1	3

				(521 bits)					
TLS 1.3	10	ECDHE (P384, 384 bits)	10	RSA-PSS-RIPEMD-128 (128 bits)	2	AES_256_CBC (256 bits)	8	MD5	1
		ECDHE (P521, 521 bits)	10	RSA-PSS-RIPEMD-160 (160 bits)	3	AES_128_CBC (128 bits)	7	MD2	0
		ECDH (x2559, 256 bits)	9	RSA-PSS-SHA1	2	CAME-LLIA_256_CBC (256 bits)	8		
		ECDH (x448, 448 bits)	10	RSA-PSS-SHA224	5	CAME-LLIA_128_CBC (128 bits)	7		
		ECDH (P192, 192 bits)	4	RSA-PSS-SHA256	9	SEED_CBC	1		
		ECDH (P224, 224 bits)	5	RSA-PSS-SHA384	10	3DES_EDE_CBC (112 bits)	1		
		ECDH (P256, 256 bits)	9	RSA-PSS-SHA512	10	RC4_128 (128 bits)	1		
		ECDH (P384, 284 bits)	10	RSA-PKCS1-RI-PEMD-128	2	DES_CBC	1		
		ECDH (P521, 521 bits)	10	RSA-PKCS1-RI-PEMD-160	3	DES40_CBC	1		
		DHE	8	RSA-PKCS1-SHA1	2	RC2_CBC_40	1		
		DH	7	RSA-PKCS1-SHA224	5	RC4_40	1		
		RSA	6	RSA-PKCS1-SHA256	9	RC4	1		
				RSA-PKCS1-SHA384	10	DES	1		
				RSA-PKCS1-SHA512	10				
				DSS	4				
				ANON	0				

Taula 3: Puntuació algoritmes cipher suite

La API no proporciona informació de la mida de la clau per a RSA, per DH/DHE ni per DSS (DSA), doncs només s'aplica pels algoritmes que apliquen la corba el·líptica (ECDH, ECDHE, EC-DSA). S'ha valorat el grup (mida de la clau) usat per a aplicar la corba el·líptica (ECC) i per RSA s'ha considerat el *hash* emprat per determinar la puntuació.

La valoració en funció de la mida de la clau, la versió de l'algoritme per a la funció *hash*, xi-fratge (*bulk cipher*) i clau pública s'ha extret de les referencies que BlueKrypt publica de dife-rents organitzacions (NIST, ANSII, BSI...) per a l'any 2018. Si algun algoritme pateix alguna vul-nerabilitat la puntuació cau dràsticament.

3.1.3.2 Taula longitud de claus RSA / ECC

RSA		ECC	
1024 bits	4	≥ 160 bits	4
2048 bits	9	≥ 224 bits	9
≥ 3072 bits	10	≥ 256 bits	10
≥ 7680 bits	10	≥ 384 bits	10
≥ 15360 bits	10	≥ 512 bits	10

Taula 4: puntuacions mida de la clau per a RSA i ECC

3.1.3.3 Taula puntuacions algoritmes de firma certificat

ECDSA		RSA		DSA	
ECDSA_SHA_224	8	RSA_SHA_224	8	DSA_SHA1	0
ECDSA_SHA_256	9	RSA_SHA_256	9	DSA_SHA224	8
ECDSA_SHA_384	10	RSA_SHA_384	10	DSA_SHA256	10
ECDSA_SHA_512	10	RSA_SHA_512	10	DSA_SHA3_1	2
ECDSA_SHA_1	0	RSA_SHA1	0	DSA_SHA3_224	10
ECDSA_MD5	0	RSA_MD5	0	DSA_SHA3_256	10
ECDSA_SHA3_224	8	RSA_SHA3_224	8		
ECDSA_SHA3_256	9	RSA_SHA3_256	9		
ECDSA_SHA3_384	10	RSA_SHA3_384	10		
ECDSA_SHA3_512	10	RSA_SHA3_512	10		

Taula 5: puntuació algoritmes firma certificat

3.1.3.4 Exemple de puntuació màxima

Com un lloc *web* pot rebre el 100% de la puntuació?

Després de superar les valoracions crítiques (cas contrari cau a puntuació 0) el protocol utilitzat ha de ser el TLS en la versió 1.2 cap endavant, amb un algoritme per a l'intercanvi de la clau efímer (PFS) amb una mida de la clau de més de 256 bits i una signatura (autenticació) emprant ECDSA (clau de més de 256 bits) o RSA (amb funció *hash* de més de 256 bits).

El xifratge simètric de les dades ha ser de més de 128 bits i mode GCM o CCM per tal de poder aplicar la forma d'encriptació AEAD, el *hash* de les dades ha de ser superior als 256 bits (MAC *hash*).

Els algoritmes per a la clau pública dels certificats (tots els de la cadena) han d'utilitzar una mida de la clau superior a 2048 bits per a RSA o 224 bits per a ECC, i la funció *hash* ha de ser mínim la versió SHA-2 (més de 256 bits). Per últim el certificat de l'entitat final ha d'estar validat de forma estesa (EV).

3.1.3.5 Taula de pesos per defecte

La puntuació de cada concepte compren valors de 0 a màxim de 10.

Variable	Pes (0% a 100%)
Cadena de certificats	
Certificat entitat final	50%
Certificat CA (*intermediate*)	25%
Certificat CA *root*	25%
Certificat	
Validesa certificat	20%
Algoritme clau pública certificat	30%
Algoritme firma certificat	50%
Cipher suite	
Versió protocol SSL/TLS	10%
Algoritme intercanvi de claus (*Key Exchange*)	30%
Algoritme firma (Autenticació)	10%
Algoritme xifrat en bloc (*Bulk Cipher*)	20%
Algoritme *hash* (MAC *hash*)	30%
Connexió	
Cipher suite	70%
Extended Validation (EV)	15%
Perfect forward secrecy (PFS)	15%
Puntuació total	
Connexió	60%
Cadena de certificació	40%

Taula 6: pesos dels conceptes a analitzar

Exemple càlcul puntuació total:

$Certificat_i$ = (20%) · Validesa certificat + (50%) · Algoritme clau pública certificat + (30%) ·Algoritme firma certificat;

Cadena de certificats = (50%) · Certificat entitat final + (25%) · Certificat CA (*intermediate*) + (25%) · Certificat CA *root*;

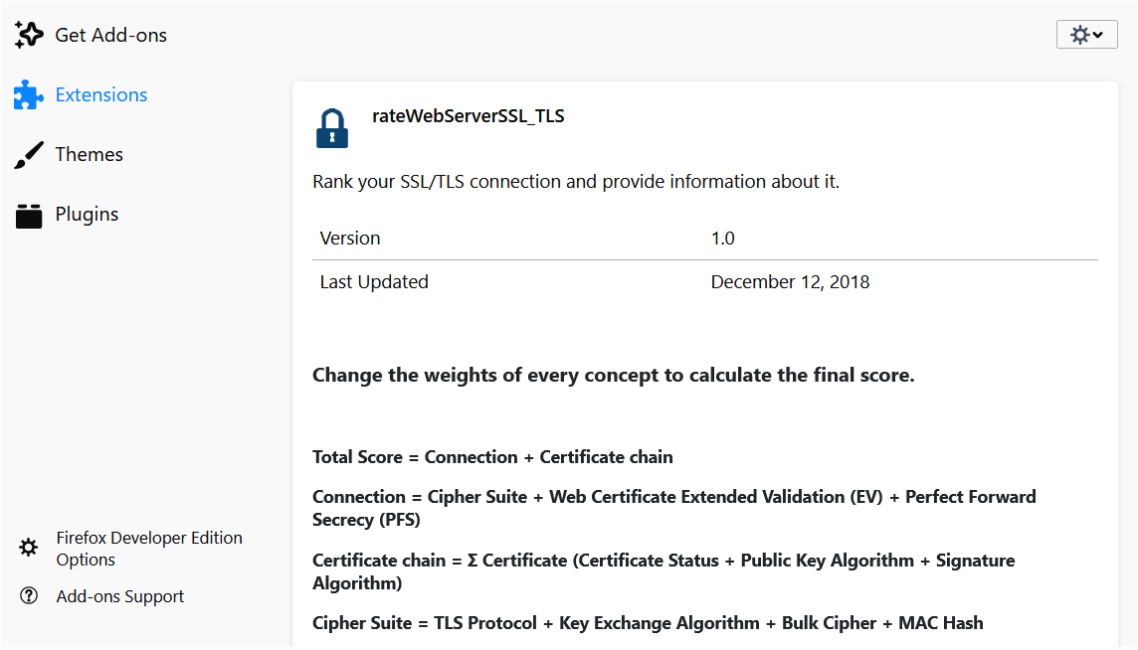
Cipher suite = (10%) · Versió protocol SSL/TLS + (30%) · Algoritme intercanvi de claus (*Key Exchange*) + (10%) · Algoritme firma (Autenticació) + (20%) · Algoritme xifrat en bloc (*Bulk Cipher*) + (30%) · Algoritme *hash* (MAC *hash*);

Connexió = (70%) · C*ipher suite* + (15%) · EV + (15%) · PFS;

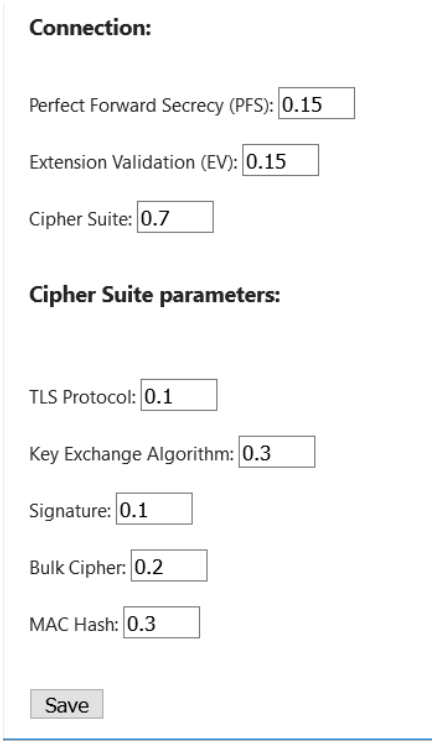
Puntuació total = (60%) · Connexió + (40%) · Cadena de certificació;

3.1.3.5 Opcions personalitzades dels pesos

En les opcions de l'extensió es permet que l'usuari personalitzi els pesos pels càlculs. El que no es permet és canviar l'algoritme de càlcul ni els valors de les puntuacions dels diferents algoritmes a analitzar.



Il·lustració 11: part panell de opcions (configuració) extensió



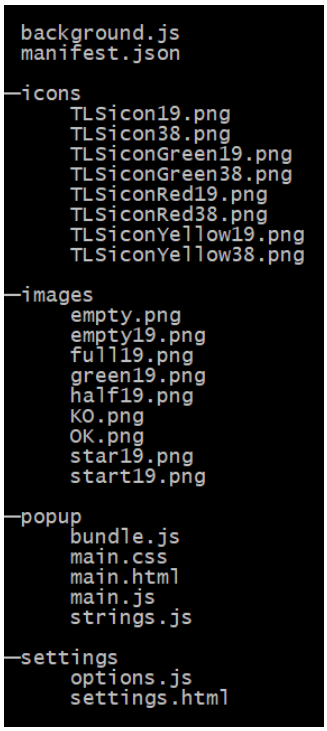
Il·lustració 12: part panell de opcions (configuració) amb els pesos elements de seguretat a analitzar

3.2 Implementació

A continuació en aquest apartat es mostrarà l'estructura a nivell de codi i de fitxers de l'extensió, una breu descripció de les funcions creades, com s'implementa l'algoritme per obtenir la puntuació total i que s'ha usat de la API nativa.

3.2.1 Estructura de fitxers

Estructura en forma d'arbre del directori arrel de l'extensió:



Il·lustració 13: estructura en forma d'arbre directori arrel extensió

A continuació descrivim de l'arrel de l'extensió els següents fitxers i directoris:

- 'manifest.json': fitxer manifest, declarem el nom de l'extensió i permisos necessaris, entre altres.
- 'background.js': fitxer que contindrà el *listener* per capturar les capçaleres i passar-les al script principal ('main.js') on es farà el tractament de la informació sobre la seguretat.
- '/settings': conté el *script* ('options.js') i el fitxer HTML que mostra les opcions a configurar de l'extensió. El script conté una funció per emmagatzemar els valors personalitzats i restaurar-los, doncs, servirà perquè l'usuari pugui canviar els pesos de les variables de la puntuació total. La configuració s'emmagatzema a nivell local i cada vegada que s'obre l'extensió es llegeixen les preferències.

- '/images': conté totes les imatges que s'usaran en l'extensió; les estrelles que van en funció de la puntuació total i les icones de validesa o de incorrecte. Totes les imatges usades aniran al *popup* de l'extensió ('main.html').
- '/icons': conté les icones de l'extensió (icona de l'extensió en el navegador). Trobem la icona 'neutral' anterior a prémer el botó de l'extensió (candau blau) que no indica cap valoració de la seguretat i la del color que correspongui en la valoració de la seguretat obtinguda (groc, vermell o verd). Per a cada icona trobem 2 mides, 19 i 38 píxels tal i com es recomana per part de *Firefox*.
- '/popup': s'inclouen tots els fitxers per mostrar el *popup* quan es prem el botó de l'extensió.

    - bundle.js: conté el main.js i totes les llibreries contingudes en aquest. Tal i com indica el nom és un 'bundle', una agrupació de tot el codi a nivell de *script* i és el que es carrega com a script en el fitxer HTML.

      S'usa aquesta estructura com a solució per a poder importar mòduls externs que d'altre manera seria impossible degut a restriccions de les *WebExtensions*. Aquest fitxer és generat a través de l'eina 'web-pack' que s'ha explicat en el capítol 2.

    - main.html: fitxer HTML on incloure la informació a mostrar a l'usuari. Hi ha una base fixe i la resta que es genera dinàmicament es fa a partir de llenguatge 'javascript' en el 'main.js' que posteriorment recau a 'bundle.js' quan s'empaqueta.

    - main.css. fitxer d'estil per a 'main.html'.

    - strings.js: conté *strings* que s'usaran en el 'main.js' tals com els OID [46].

    - main.js: el *script* principal que conté totes les funcions per generar la informació que es mostra en el *popup (*implementa l'algoritme de càlcul*)*. Aquest és passat amb tots els seus mòduls a importar al fitxer 'bundle.js', tal i com s'ha explicat anteriorment a través de l'eina 'web-pack'.

      Quan hi hagi canvis en aquest fitxer s'ha de generar de nou el script 'bundle.js' a través de l'eina per tal que 'main.html' agafi el nou fitxer amb els canvis realitzats.

3.2.2 Estructura de codi

Es descriurà les funcions creades i com l'algoritme obté el resultat numèric de la qualitat a nivell de seguretat de la connexió.

main.js:

- Variables:
    - Les puntuacions s'emmagatzemen en variables globals inicialitzades a 0. D'aquesta manera totes les funcions podran escriure/obtenir el valor quan sigui necessari.
    - Els textos (*strings*) que es comparteixen entre funcions també són globals.

- Variable amb format JSON ('json_string') s'inicialitza amb valors 'undefined' i és de un *scope* global, també per tal d'habilitar totes les funcions que escriguin a mesura que avança la seqüència del codi.

- Funcions:
  - object function dataExtractionCert (certificat):
    - Ens retornarà informació sobre el certificat que no és accessible a través de la API de les *webExtensions*. Aquesta informació es trobarà continguda en un objecte amb els següents *key-value* com es mostra en la il·lustració 12:

      1. El certificat en format JSON (x509).
      2. Noms de les extensions que són crítiques.
      3. Els usos de la clau del certificat.
      4. Les restriccions bàsiques del certificat.
      5. Els usos estesos de la clau del certificat.
      6. La informació sobre AIA del certificat.
      7. Objecte amb informació sobre la clau pública.
      8. Els noms alternatius de domini.
      9. L'algoritme de la firma del certificat.
      10. L'algoritme de la clau pública i la mida de la clau (concatenat).

```
return {

    x509: x509,
    criticalExtensions: criticalExtensions,
    keyUsages: keyUsages,
    basicConstraints: basicConstraints,
    eKeyUsages: eKeyUsages,
    aia: aia,
    spki: spki,
    san : san,
    algoritmeFirmaUsat: algoritmeFirmaUsat,
    algoritmeClauPublica : algoritmeClauPublica

};
```

Il·lustració 14: part del codi funció 'dataExtractionCert', objecte que retorna

La funció també escriu en les variables globals de puntuació per l'algoritme de la firma i de la clau pública.

  - void function logTabs(tabs):
    - funció per obtenir la funció activa. Ens servirà per determinar en quina pestanya (*tab*) es troba l'usuari.
  - void function onError(err):
    - funció per registrar possibles errors.
  - object getX509Ext (extensions, value):
    - funció per obtenir les extensions del certificat i passar-les a text (del OID al *string* corresponent). Ens retorna una variable tipus *object* amb el valor de l'extensió i el valor parcejat.
  - object function bold_size_text_TextNodeLarge(paraula):
    - funció usada per crear estil en texts amb una mida de la font gran i en negreta, retorna un element tipus 'span'.
  - object function bold_size_text_TextNodeMid(paraula):

        - funció usada per crear estil en texts amb una mida de la font mitjana i estil en negreta, retorna un element tipus 'span'.
    - void function print_puntuacions():
        - Imprimeix les puntuacions a mostrar en el fitxer main.html.
    - void function print_usos_clau_estesos(dadesCertificat):
        - Imprimeix els usos estesos de la clau en el fitxer main.html. Se li passa com argument l'objecte que retorna la funció dataExtractionCert().
    - void function print_validity_valid(message):
        - Imprimeix la validesa del certificat (dates) i fa el càlcul de quant de temps queda. Aquesta funció es crida si és vàlid en el temps el certificat. Se li passa com argument l'objecte rebut sobre la seguretat.
    - void function print_validity_INvalid(message):
        - Imprimeix la validesa del certificat (dates) i fa el càlcul de quant de temps queda. Aquesta funció es crida si el certificat és invàlid en el temps. Se li passa l'objecte com a argument rebut sobre la seguretat.
    - void function print_DNS(dadesCertificat):
        - Imprimeix els noms alternatius de domini que són vàlids pel certificat. Se li passa com argument l'objecte que retorna la funció dataExtractionCert().
    - void function print_basic_constraints(dadesCertificat):
        - Imprimeix les restriccions bàsiques del certificat en el fitxer main.html. Se li passa com argument l'objecte que retorna la funció dataExtractionCert().
    - void function print_aia(dadesCertificat):
        - Imprimeix la informació sobre accés a l'autoritat del certificat. Se li passa com argument l'objecte que retorna la funció dataExtractionCert().
    - void function print_key_usages(dadesCertificat):
        - Imprimeix la informació sobre els usos de la clau del certificat. Se li passa com argument l'objecte que retorna la funció dataExtractionCert().
    - void function print_issued(message,dadesCertificat):
        - Imprimeix la informació sobre l'emissor (DN) del certificat. Se li passa com argument l'objecte que retorna la funció dataExtractionCert() i el objecte rebut sobre la seguretat com a missatge.
    - void function print_issuer(message,dadesCertificat):
        - Imprimeix la informació sobre el subjecte (DN) del certificat i informa si el domini coincideix. Se li passa com argument l'objecte que retorna la funció dataExtractionCert() i el objecte rebut sobre la seguretat com a missatge.
    - void function print_serial_number(message):
        - Imprimeix el número de sèrie del certificat. Se li passa com argument el objecte rebut sobre la seguretat com a missatge.
    - void function print_algoritme_firma(dadesCertificat):
        - Imprimeix l'algoritme usat per firmar el certificat. Se li passa com argument l'objecte que retorna la funció dataExtractionCert().
    - void function print_PK(dadesCertificat):
        - Imprimeix l'algoritme emprat per la clau pública del certificat. Se li passa com argument l'objecte que retorna la funció dataExtractionCert().
    - void function print_fingerprints(message):
        - Imprimeix el *fingerprint* (sha-1 i sha256) del certificat. Se li passa com argument l'objecte que retorna la funció dataExtractionCert().
    - void function print_critical_extensions(dadesCertificat):

        - Imprimeix el nom de les extensions crítiques del certificat. Se li passa com argument l'objecte que retorna la funció dataExtractionCert().
    - void function print_estrelles():
        - Imprimeix i calcula les estrelles a mostrar en la puntuació total. Els valors no són discrets, doncs és capaç de mostrar mitges estrelles i senceres.
    - void function print_header_cert(name,color):
        - Imprimeix el titular del tipus de certificat que es mostrarà a continuació en el color passat com a paràmetre, el nom pot ser arbitrari.
    - void function get_weights_score():
        - Obté els pesos pels càlculs salvats com a preferències o els de per defecte en cas contrari, i els passa a les variables globals dels pesos de les ponderacions.
    - void function change_icon_color(color):
        - Canvia el color de la icona de l'extensió en funció de la puntuació obtinguda. Se li passa el color desitjat.
    - void function value_protocol_SSL(message):
        - Calcula la puntuació del protocol TLS/SSL que s'usa en la connexió i la guarda en la variable global de la puntuació corresponent. Se li passa com argument l'objecte rebut sobre la seguretat com a missatge.
    - void function value_cipher_suite(message):
        - Calcula la puntuació del *cipher suite* que s'usa en la connexió i la guarda en la variable global de la puntuació corresponent. Se li passa com argument l'objecte rebut sobre la seguretat com a missatge.
    - void function print_total_score(puntuacioTOTAL):
        - Imprimeix la puntuació total en el main.html amb el color que correspongui al total de la puntuació. Se li passa el valor de la puntuació total (*integer*).
    - Void async function store_local_values(dades):
        - Emmagatzema a nivell local (*storage*) informació rellevant de la seguretat a nivell de connexió en format JSON i, se li passa una variable tipus objecte amb les dades a emmagatzemar.
    - Void function print_pfs():
        - Imprimeix la propietat PFS al main.html.

3.2.2 Paràmetres de seguretat proporcionats per la API

Tal i com s'ha analitzat en el capítol 2, la API nativa ens proporciona mètodes per tal de capturar les dades que determinen la seguretat de la connexió però no tots els que necessitem per aconseguir l'objectiu establert. A continuació mostrarem quina informació obtenim directament amb les classes i mètodes que ens proporciona la API per a les *WebExtensions* i quines no.

Utilitzarem la 'JavaScript APIs' que es pot usar (entre altres) en el *script* que s'executarà en *background* (segon pla) per accedir a la informació de la seguretat:

- La classe 'webRequest' té el mètode 'getSecutiryInfo()' que amb el *listener* 'onHeader-Recieved' (que escolta quan rep capçaleres en la petició HTTP) podrem accedir (llegir) a les propietats TLS d'una petició (ens interessa protocol HTTPS).

Aquest *listener* se li indica quins protocols escolta (HTTPS i HTTP), se li passa la funció que rebrà el valor retornat (informació de la seguretat HTTP) i el paràmetre 'blocking' que la documentació indica.

```
browser.webRequest.onHeadersReceived.addListener(logSubject,
  {urls: ["https://*/*","http://*/"],types: ["main_frame"]},
  ["blocking"]
);
```

Il·lustració 15: listener del script 'background.js' per obtenir les capçaleres de una petició web

Un cop el *listener* obtingui informació de les capçaleres li passarà la informació a la funció 'logSubject'. Aquesta funció utilitza el mètode 'getSecurityInfo()' de la classe 'webRequest' per tal d'obtenir la informació de les capçaleres que ens interessa (TLS).

A la funció 'logSubject' rep els paràmetres 'details.requestID' (conté l'identificador de la petició HTTP) i el JSON (opcional) amb camps 'certificateChain' i 'rawDER' amb valor *true* per tal d'obtenir, en primer lloc, tots els certificats de la cadena, i en segon lloc, els certificats en format 'raw DER'.

Aquests dos paràmetres son d'especial interès, primer per obtenir tota la cadena de certificats i segon, per tal de poder accedir a la informació que no és proporcionada en l'objecte tipus 'certificat' retornada per la API nativa (s'usaran llibreries externes per descodificar la informació DER).

```
async function logSubject(details) {
    try {
            gettingInfo = await browser.webRequest.getSecurityInfo(details.requestId,{"certificateChain": true,"rawDER": true });
    }
    catch(onError) {
    }
}
```

Il·lustració 16: funció per filtrar les capçaleres sobre la seguretat de una petició web

Encara que aquesta informació s'emmagatzemarà a la variable 'gettingInfo' cada vegada que el 'listener' rebi capçaleres sobre el protocol TLS (sigui protocol segur o no), les enviarem al script 'main.js' quan el 'trigger' s'accioni. Aquest *trigger* és simplement un missatge que conté un *string* amb el contingut 'init'. L'usuari clica la icona de l'extensió i aquesta envia (main.js) el missatge 'init' al script 'background.js' que acciona la transferència de la variable 'gettingInfo' amb tota la informació sobre la seguretat que la API proporciona (webRequest.getSecurityInfo()).

```
browser.runtime.sendMessage("init");
```

Il·lustració 17: part de codi del script main.js on s'envia missatge amb contingut 'init'

```
function handleMessage(message, sender) {
  if (message === "init") {
    browser.runtime.sendMessage(gettingInfo);
  } else {
    console.log("No s'ha pogut enviar missatge amb la informació de seguretat");
  }
}
```

Il·lustració 18: funció per enviar informació sobre la seguretat al script principal

Flux per a l'obtenció i enviament de la informació sobre seguretat:

1. L'usuari fa petició al servidor *web*.
2. *listener* (onHeadersReceived) captura les capçaleres de la petició i resposta i les emmagatzema a la variable 'gettingInfo' a través de la funció 'logSubject()'.
3. L'usuari clica l'extensió 'rankSSL'.
4. 'main.js' envia missatge al script en segon pla 'background.js' amb el contingut 'init'.
5. *listener* (onMessage) del script en segon pla obté el missatge i crida la funció 'handleMessage()'.
6. La funció 'handleMessage()' envia la variable 'gettingInfo' al script 'main.js'.
7. listener (onMessage) del script main.js captura el missatge que conté la variable 'gettingInfo'.
8. Comença el tractament, càlcul i impressió en el *popup* de la informació sobre la seguretat.

Per obtenir la informació sobre la seguretat del lloc *web* es necessita fer la petició al servidor, és a dir, si es canvia de 'tab' sense recarregar la pàgina es mostrarà la informació en el *popup* de l'última pàgina peticionada al servidor *web*.

3.2.2.1 Propietats extretes del objecte 'webRequest.SecurityInfo'

- certificates
  - Tota la cadena de certificats (certificat X.509) on traurem certa informació d'interès que s'indica en el punt 3.1.2. Retorna un objecte tipus 'certificateInfo'.
- certificateTransparencyStatus
  - Ens indicarà l'estat de CTS. Retorna un *string*.
- cipherSuite
  - Ens servirà per obtenir el *string* que indica el *cipher suite* i així, conèixer paràmetres com el 'Key Exchange', l'algoritme usat per signar, algoritme per xifrar la comunicació (*bulk cipher*) i l'algoritme per calcular el 'MAC hash'. Retorna un *string* amb el *cihpersuite* negociat entre client i servidor.
- hsts
  - Ens indica si usa HSTS. Retorna *true* or *false*.
- isDomainMismatch

    - S'usarà per conèixer si el domini que es consulta coincideix amb el del certificat. Retorna *true* or *false*.
- isExtendedValidation
    - Ens evita haver d'entrar a consultar l'extensió del certificat 'Certificate policies' i consultar si té la propietat 'Extended Validated' que és més costós que usant el mètode directament de la API. Retorna *true* or *false*.
- isNotValidAtThisTime
    - Només ens servirà per el certificat del servidor (entitat final) però ens evita cert esforç a nivell de codi si volem consultar la validesa en el temps del certificat. Retorna *true* or *false*.
- isUntrusted
    - Ens indicarà si s'ha pogut construir tota la cadena de certificació i si és de confiança (certificat arrel està instal·lat en el navegador). Retorna *true* or *false*.
- keaGroupName
    - S'usarà pel protocol TLS versió 1.3 ja que no podem extreure el valor de l'algoritme d'intercanvi de claus en el mateix *cipher suite*. Retorna un *string*.
- protocolVersion
    - Ens servirà per conèixer quina versió del protocol SSL/TLS s'ha negociat entre client i servidor. Retorna un *string*.
- signatureSchemeName
    - Ens retorna l'algoritme per signar per part del servidor. Retorna un *string*.
- state
    - S'usarà per conèixer si s'ha establert una connexió HTTPS o HTTP o ha fallat. Retorna un *string* amb els possibles valors 'secure', 'insecure', 'broken' o 'weak'. Ens interessa les dues primeres.

3.2.2.2 Propietats extretes en el objecte tipus certificat (certificateInfo)

A continuació es mostra quina informació es pot obtenir del certificat que ens retorna la API i així podem observar quines carències té per justificar la descodificació de l'objecte certificat tipus 'rawDER'. Necessitarem llibreries externes per descodificar-lo ja que la API nativa no ens ofereix aquesta possibilitat.

webRequest.getSecurityInfo().certificates:

Ens retorna un *array* d'objectes tipus 'CertificateInfo' si s'ha indicat que es vol la cadena completa de certificats (webRequest.getSecurityInfo() amb paràmetre opcional 'certificateChain' a 'true'). En cas contrari ens retorna el certificat de l'entitat final (servidor *web*).

Propietats 'certificateInfo':

- fingerprint.sha1 i fingerprint.sha256
    - Mostrarem el *fingerprint* en el *popup* de l'extensió, merament informatiu. Retornen un *string*.
- isBuiltInRoot
    - S'usarà per comprovar que el certificat arrel està instal·lat en el navegador i indicar que és un aspecte correcte o incorrecte. Retorna *true* or *false*.
- issuer

    - S'usarà per indicar l'emissor del certificat. Retorna un *string* amb el DN.
- rawDER (per obtenir-lo s'ha de passar el paràmetre opcional 'rawDER' com a 'true' a webRequest.getSecurityInfo())
    - Retorna un *array* de números amb el certificat(s) codificat en DER. Obtindrem de tota la cadena de certificats el *array* i el descodificarem per extreure la informació que no podem accedir directament des de les propietats esmentades.
- serialNumber
    - Mostrarem el número de sèrie merament informatiu. Retorna un *string*.
- subject
    - Mostrarem el 'issued to' (emès a) que és l'organització (O del DN) propietària del certificat.
- validity.start i validity.end
    - Ens retorna la data d'inici del certificat i la data de finalització, la usarem per calcular si és vàlid el certificat de la CA i el del *root*, com també per mostrar les dades a nivell informatiu. Retorna un *string*.

Aquestes propietats ens seran d'interès, però com podem observar, ens falta informació de valor per poder executar l'algoritme de puntuació establert en el capítol 2. Les carències a través de la informació que ens proporciona el mètode 'webRequest.SecurityInfo()' són: algoritme i mida de la clau pública i firma del certificat, informació sobre l'estat de revocació del certificat i per últim, accés a totes les extensions possibles del certificat (excepte la propietat EV de la CP en el certificat del servidor ja que la podem extreure directament a través de la API).

Aquesta informació ha d'estar continguda en la propietat 'rawDER' però prèviament hem d'aconseguir descodificar-lo per poder tractar-lo i obtenir aquesta informació.

3.2.2.2.1 Propietats extretes de 'certificateInfo.rawDER'

Per tal d'obtenir la informació sobre l'algoritme i mida de la clau pública i firma s'ha hagut de descodificar el certificat en aquest format (*array* de números). Per aconseguir-ho s'ha usat la llibreria 'pkijs' [47] i 'ansj1' [48] juntament amb funcions del mòdul 'pvutils' [49] a la propietat 'rawDER' de 'certificateInfo'.

La funció que s'encarrega d'extreure les dades i retorna dades d'interès és 'object dataExtractionCert(certificateInfo certificat)' que retorna un objecte amb les dades:

- x509: certificat X.509 en format JSON (*string*).
- criticalExtensions: *array* de *strings* amb els OID de les extensions crítiques.
- keyUsages: objecte.
    - 'critical': indica si l'extensió està marcada com a crítica (*boolean*).
    - 'purposes': indica els propòsits de la clau del certificat (*string*).
- basicConstraints: objecte.
    - 'cA': indica si el certificat està restringit a ser usat per una CA (*boolean*).
    - 'critical': indica si l'extensió està marcada com a crítica (*boolean*).
- eKeyUsages: objecte.

    - ‘critical’: indica si l’extensió està marcada com a crítica (*boolean*).
    - ‘purposes’: indica els propòsits estesos de la clau del certificat (*string*).
- aia: objecte.
    - ‘critical’: indica si l’extensió està marcada com a crítica (*boolean*).
    - ‘descriptions’: objecte.
        - ‘method’: indica el mètode per accedir a la informació de l’autoritat (*string*).
        - ‘location’: indica la URI per accedir a la informació (*string*).
- spki: objecte amb la informació sobre la clau pública.
    - ‘kty’: algoritme emprat per a calcular la clau pública (string).
    - ‘keysize’: mida de la clau (integer).
- san: objecte.
    - ‘altNames’: *array* de noms alternatius (DNS) pel que el certificat és vàlid (*array* de *string*).
    - ‘critical’: indica si l’extensió està indicada com a crítica (*boolean*).
- algoritmeFirmaUsat: algoritme i mida de la firma del certificat (*string*).
- algoritmeClauPublica: es concatena l’algoritme i la mida de la clau pública certificat (*string*).

3.3 Informació no accessible

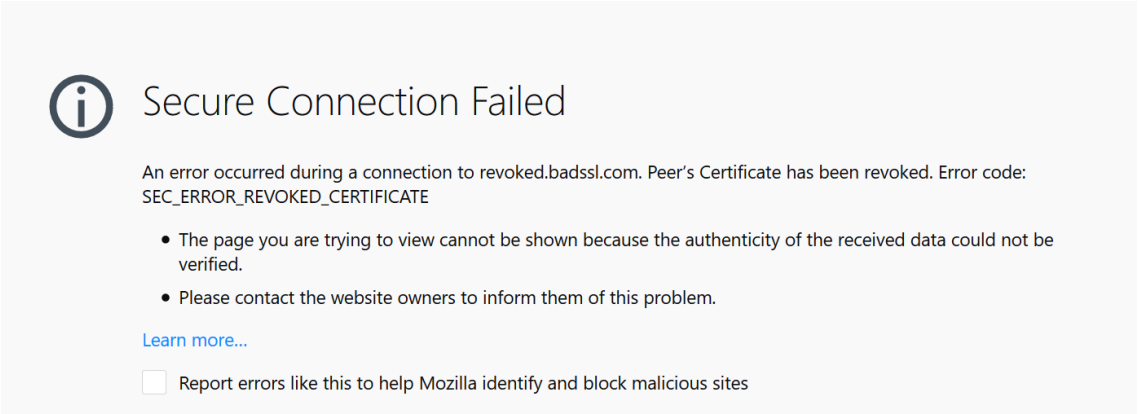
Tal i com hem pogut observar en el punt anterior, no existeix cap propietat obtinguda de la API de les W*ebExtensions* per tal d’esbrinar l’estat de revocació del certificat i així implementar-lo en el càlcul de la puntuació total de la seguretat al lloc *web*.

Una de les maneres, tal i com s’ha analitzat en el capítol 2, és utilitzant el protocol OCSP que permet fer la consulta en un servidor i aquest ens retorna l’estat de revocació. Tampoc existeix cap mètode natiu per fer la consulta, i les llibreries que s’han usat per accedir a la informació del certificat codificat DER proporcionen mètodes per construir (amb certa complexitat) la petició i la resposta però no fan la consulta al servidor OCSP.

Tampoc es pot accedir a la informació OCSP *staple* (‘TLS Certificate Status Request extension’) que s’obté en el *handshake*, doncs si en un futur la API permet fer la consulta d’aquesta informació seria molt interessant per comprovar l’estat de revocació del certificat, i més quan el OCSP és qüestionat [50].

Després de investigar mòduls de tercers que permetin la consulta, s’ha decidit deixar aquesta característica com evolutiu de l’extensió ja que no s’ha tingut èxit en la implementació ni la concreció de la realització de la consulta ni captura de la resposta.

No obstant la premissa anterior, el navegador està configurat per fer la consulta OCSP al servidor corresponent segons el certificat, i si aquest està revocat, informa a l’usuari explícitament. En cas que ja estigui marcat com a revocat el navegador no permet la navegació al lloc *web* (error: ‘SEC_ERROR_REVOKED_CERTIFICATE’).



Il·lustració 19: missatge del navegador Firefox al accedir a un lloc web amb certificat revocat

Un certificat revocat no és acceptat per defecte pel navegador *web*, doncs no es podrà accedir a la pàgina sol·licitada. Encara que seria d'interès que l'estat de revocació estigués present en l'algoritme de càlcul de la puntuació final, pel fet que es considera aquesta comprovació crítica, no incideix al càlcul total; és a dir, si fos revocat o la consulta al OCSP també ho indica, llavors la puntuació total seria 0 i no faria variar un altre possible valor.

També existeix la deficiència d'obtenir la mida de la clau pels algoritmes que no apliquen la corba el·líptica (ECC), ni per l'intercanvi de claus (*Key Exchange*) ni per l'autenticació (*Signature*); s'ha valorat sempre que s'ha pogut però pels algoritmes RSA, DH, DHE, DSA, DSS no s'ha analitzat la mida de clau. Seria interessant actualitzar l'extensió si la API la milloren i agreguen la funcionalitat en un futur.

La resta de informació que ens interessa per complir els objectius s'ha pogut aconseguir (algoritme clau pública, firma del certificat i per últim les extensions del certificat), doncs, concretem la única informació que no s'ha aconseguit i que era de valor és l'estat de revocació del certificat.

3.4 Joc de proves i verificació

Per provar l'extensió s'ha utilitzat la versió 'Mozilla firefox developer' Edition de 64 bits (65.0b4) i l'extensió empaquetada com a tal (*add-on*) sense signar/verificar per part de Mozilla. Les proves consistiran en navegar a pàgines amb protocol HTTPS i HTTP, verificar que la informació té coherència, contrastar-la amb la informació que proporciona el navegador sobre el certificat i la connexió.

Es consultaran pàgines amb algoritmes diferents per posar a prova l'extensió i la que més ens servirà és la 'badssl.com' que conté una bateria d'enllaços a pàgines amb diferents escenaris (algoritmes a analitzar), protocols SSL/TLS i estats de certificats (validesa, autosignat...).

Checklist a verificar per a les pàgines *web* amb protocol HTTPS per tal de conèixer que l'extensió funciona correctament:

1. *Rating* estrelles correcte (es mostren les estrelles plenes i mitges plenes quan s'escaigui i tingui coherència amb la puntuació total).

2. Puntuació correcte (s'ha traçat les puntuacions i càlculs a través de la consola del navegador *web* amb la funció *log* de a classe 'console' per comprovar si es compleix l'algoritme).

3. Domini consultat correcte (es mostra el domini correctament i coincideix amb el del certificat de l'entitat final).

4. Informació presentada coherent i consistent (tota la informació que es mostra en el *popup* té el format esperat i coherència).

6. Canvi de color icona extensió correcte (La icona de l'extensió canvia de color en funció de la puntuació obtinguda en el càlcul de la qualitat de la seguretat de la connexió).

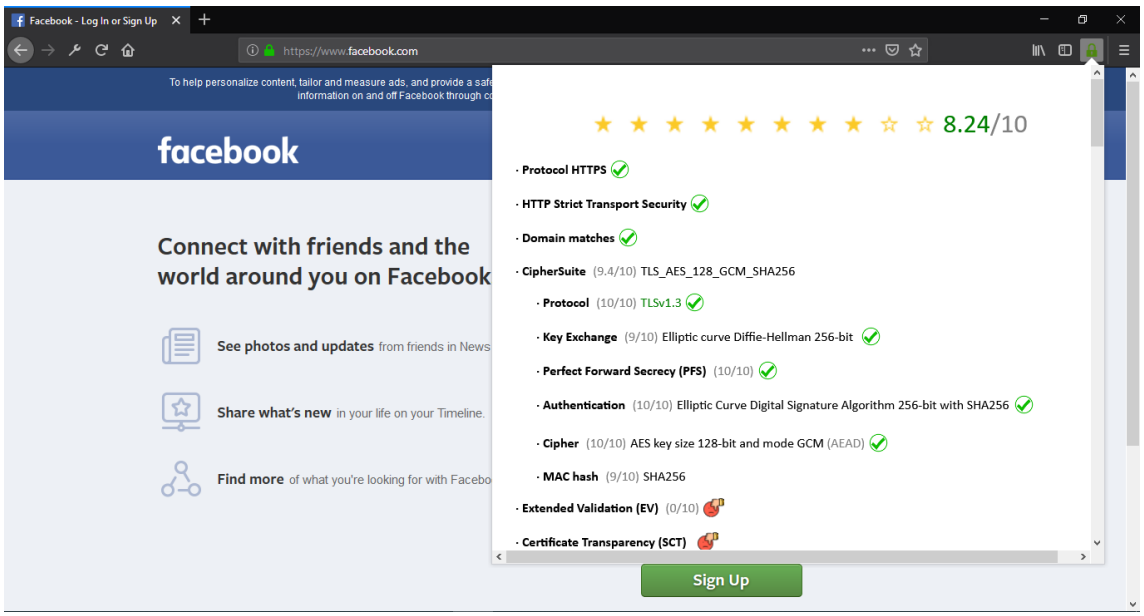
3.4.1 Proves i verificació

A continuació es mostra el resultat proporcionat per l'extensió al accedir a una pàgina *web* amb protocol HTTPS, es considerà correcte si tots els punts del *checklist* comentat anteriorment son correctes. Els 5 primers dominis consultats s'han seleccionat de la pagina *web* 'The Moz Top 500' on es poden trobar els top 500 dominis més visitats de Internet.

S'ha decidit afegir la consulta al domini de Qualys per valorar la qualitat de la seguretat d'un lloc *web* que proporciona aquest mateix servei. La pàgina de la universitat UOC s'ha utilitzat per comprovar si el format (JSON) i la informació continguda és correcte segons contrast amb el que mostra l'extensió en el navegador i els requeriments del *checklist*. La resta de dominis son de 'badssl.com' on podem obtenir diferents escenaris a nivell d'algoritmes i estats del certificat.

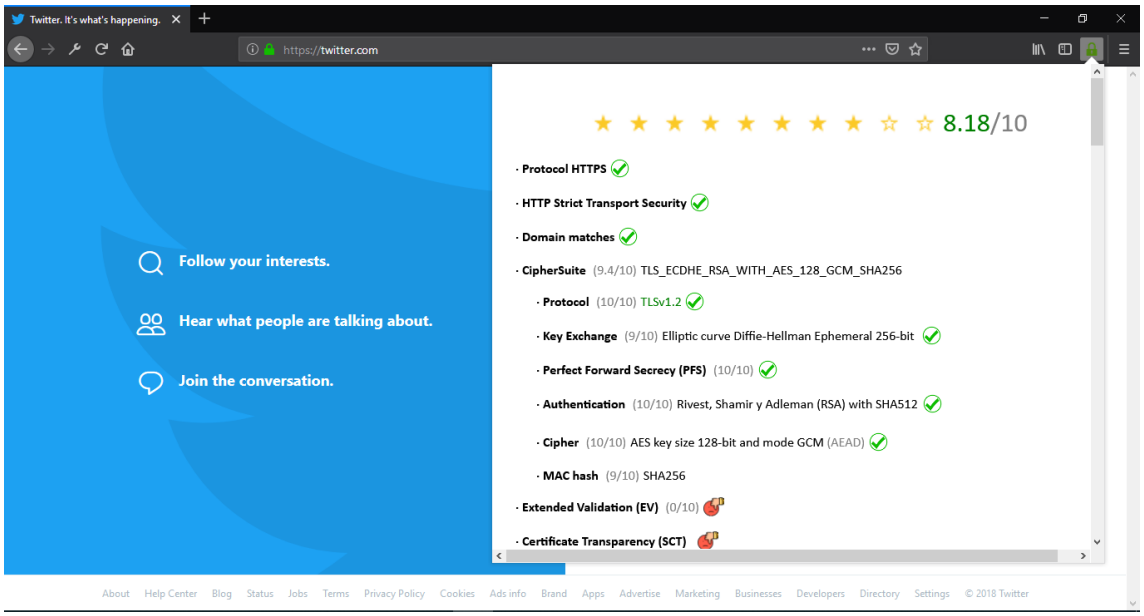
Proves:

1. https://facebook.com



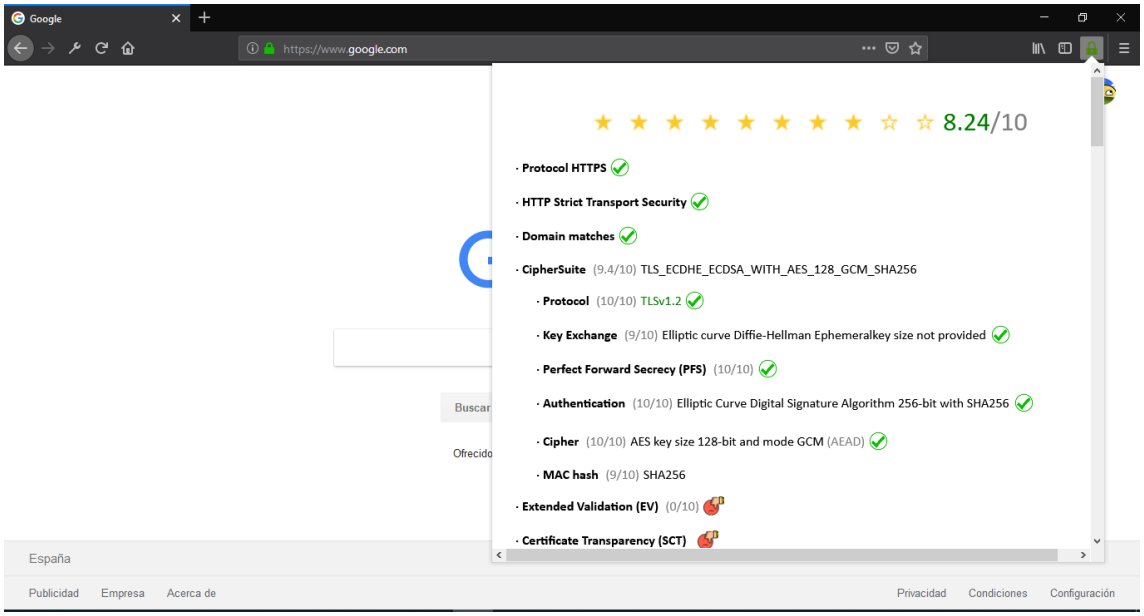
Il·lustració 20: test extensió rankSSL al domini 'facebook.com'

2. https://twitter.com



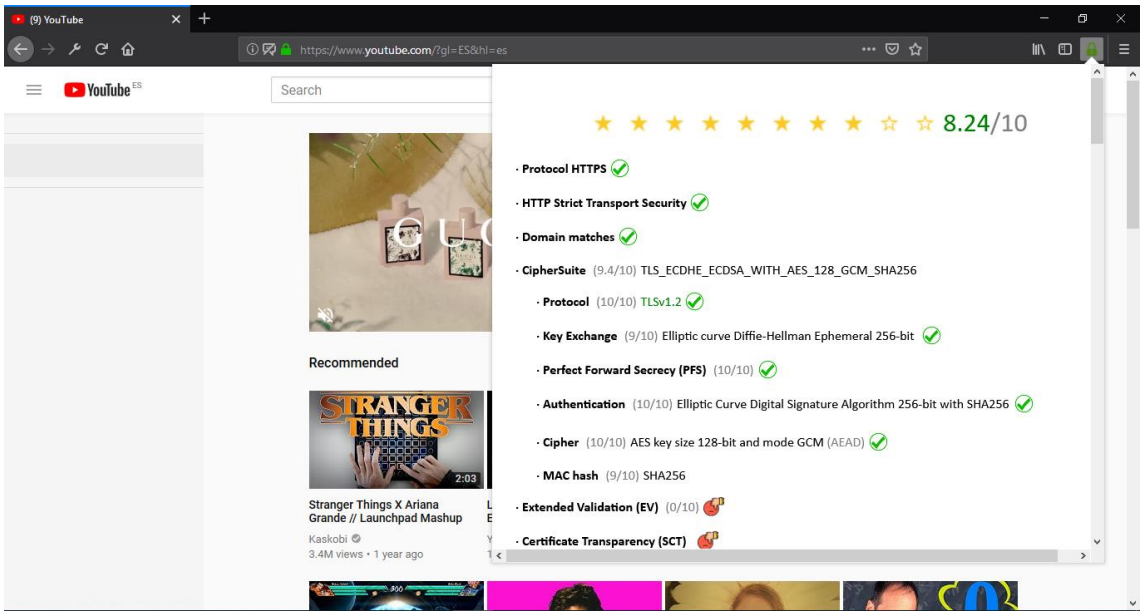
Il·lustració 21: test extensió rankSSL al domini 'twitter.com'

3. https://google.com



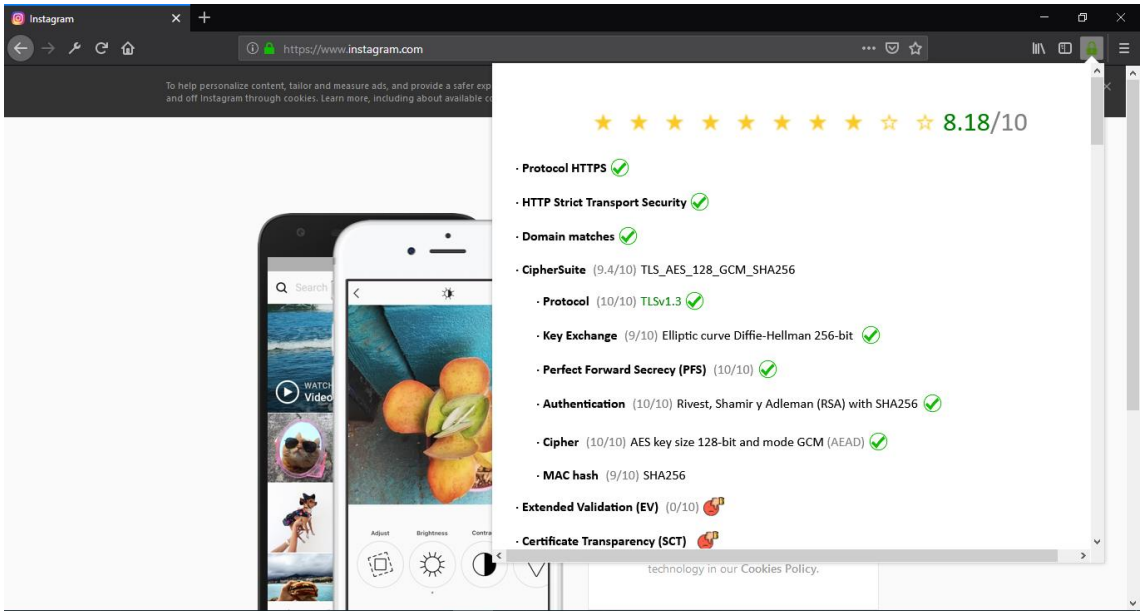
Il·lustració 22: test extensió rankSSL al domini 'google.com'

4. https://youtube.com



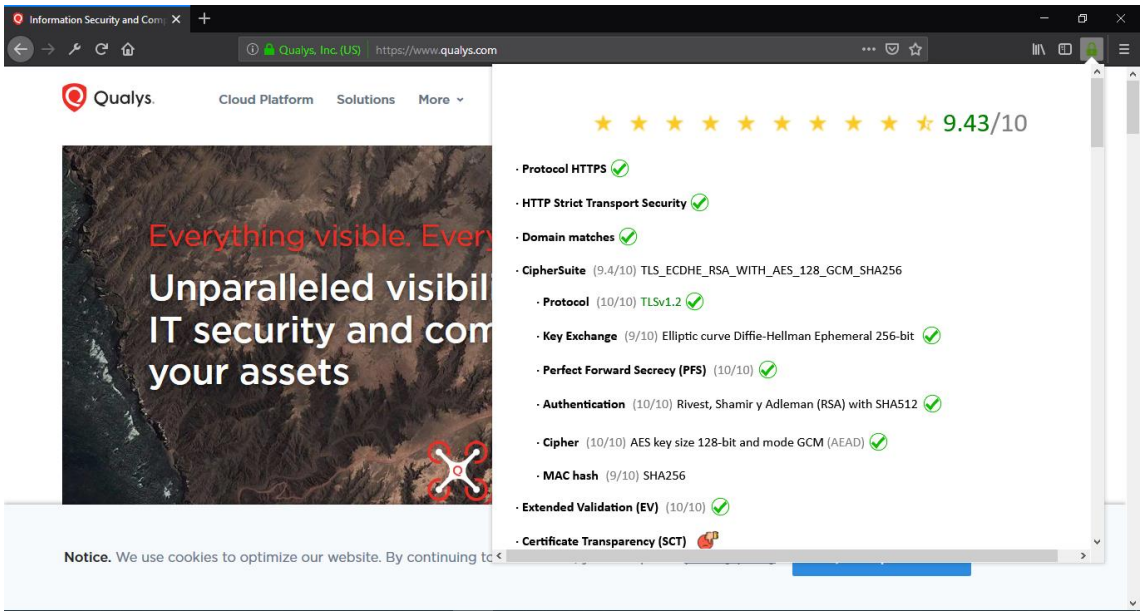
Il·lustració 23: test extensió rankSSL al domini 'youtube.com'

5. https://instagram.com



Il·lustració 24: test extensió rankSSL al domini 'instagram.com'

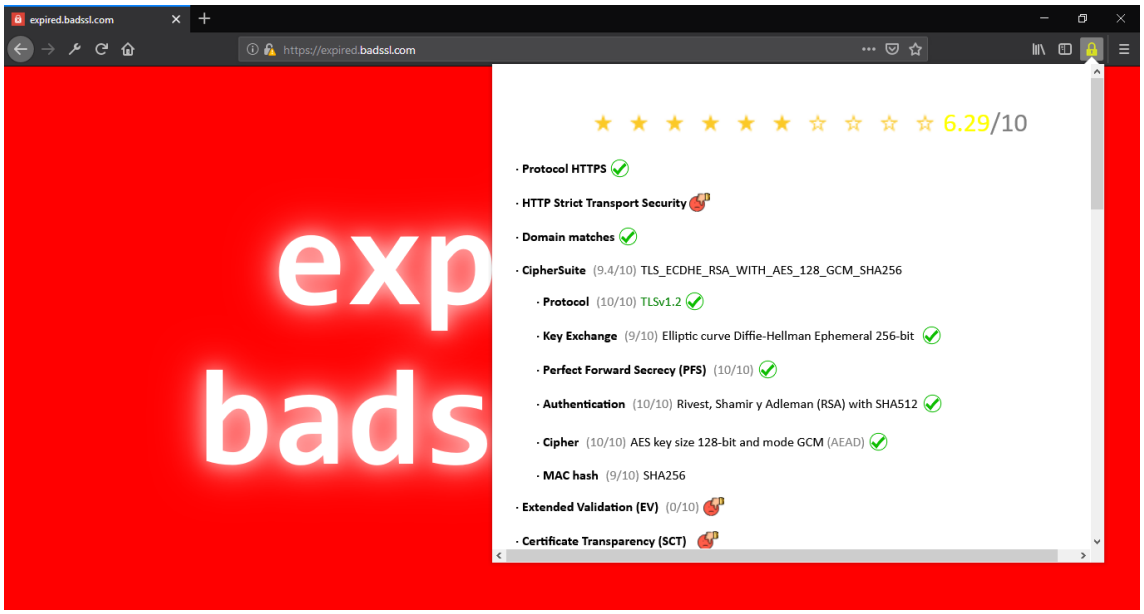
6. https://qualys.com



Il·lustració 25: test extensió rankSSL al domini 'qualys.com'

7. https://expired.badssl.com

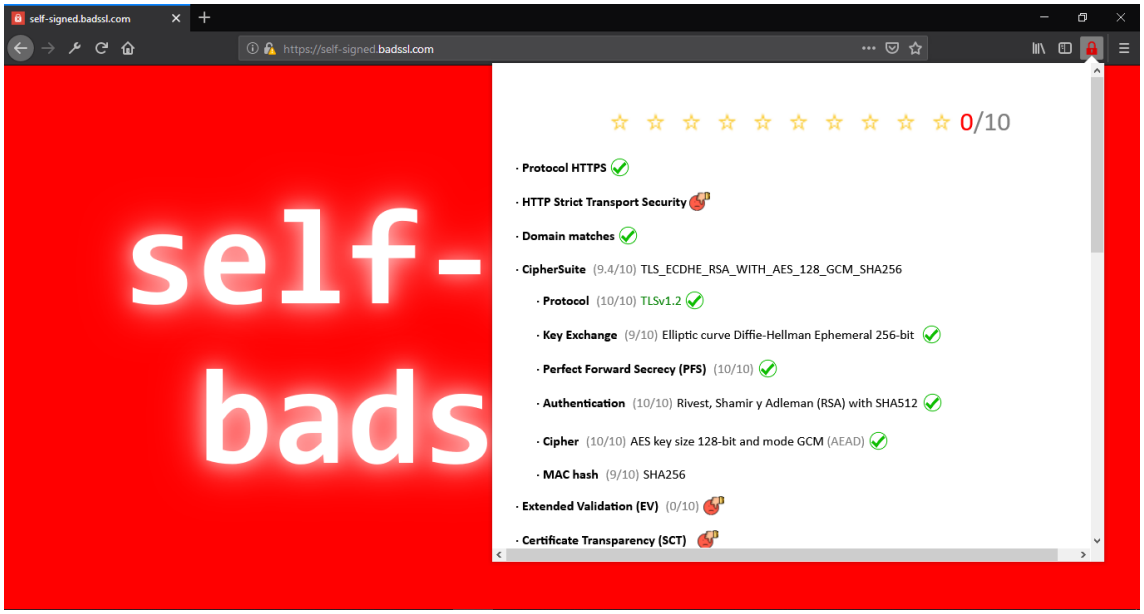
Domini amb el certificat caducat, es pot observar que la puntuació baixa força encara que la resta de paràmetres siguin molt bons (tals com el *cipher suite* que té 9.4 sobre 10).



Il·lustració 26: test extensió rankSSL al domini 'expired.badssl.com'

8. https://self-signed.badssl.com

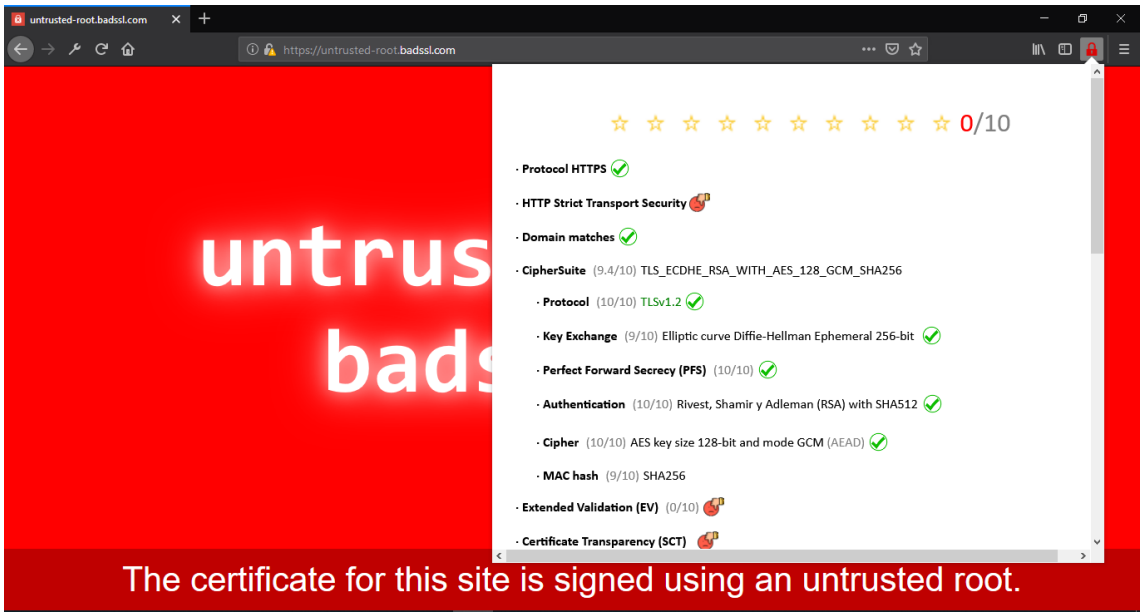
Cas amb el certificat auto-firmat, es pot observar com la puntuació és zero.



Il·lustració 27: test extensió rankSSL al domini 'self-signed.badssl.com'

9. https://untrusted-root.badssl.com

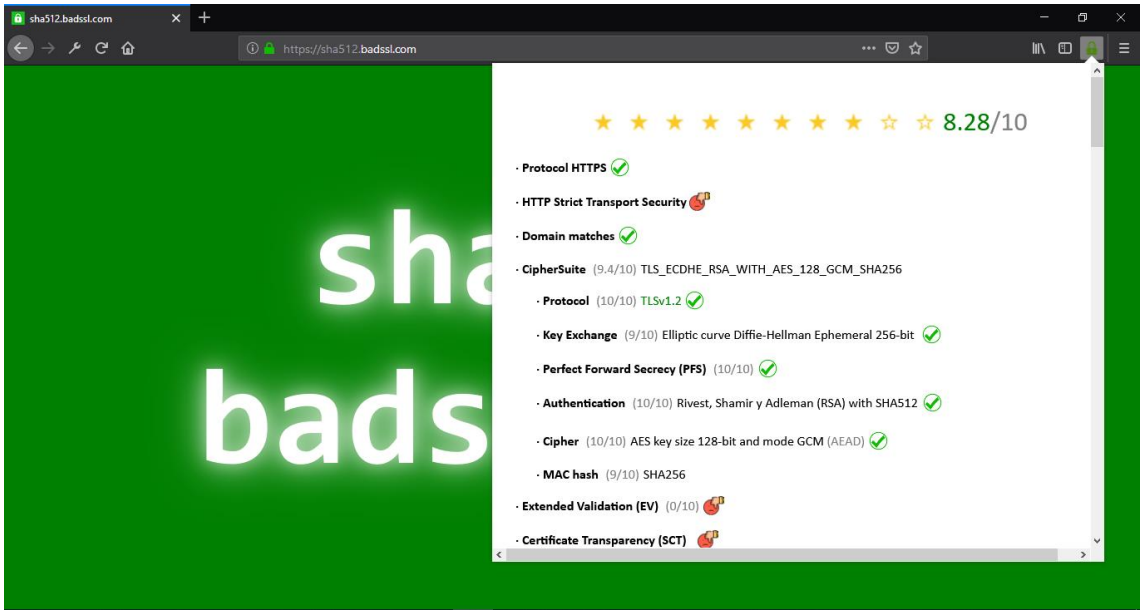
Cas on el certificat *root* no es troba instal·lat en el navegador, és a dir, no és de confiança. Es pot observar com la puntuació cau a zero.



Il·lustració 28: test extensió rankSSL al domini 'untrusted-root.badssl.com'

10. https://sha512.badssl.com

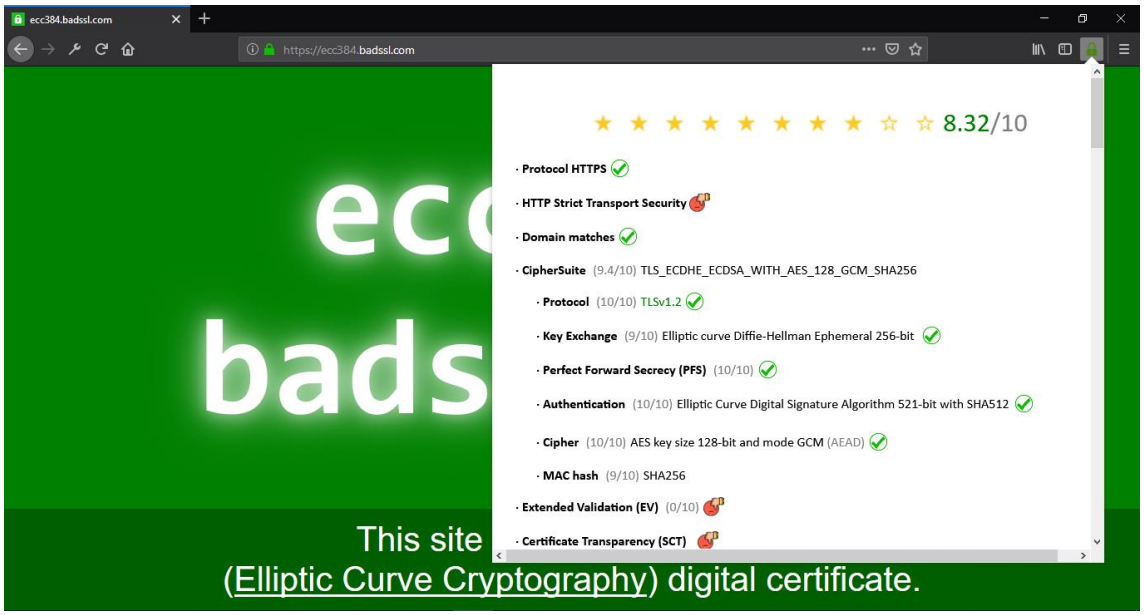
Cas on el certificat del servidor *web* (entitat final) ha estat firmat amb l'algoritme RSA amb SHA512.



Il·lustració 29: test extensió rankSSL al domini 'sha512.badssl.com'

11. https://ecc384.badssl.com

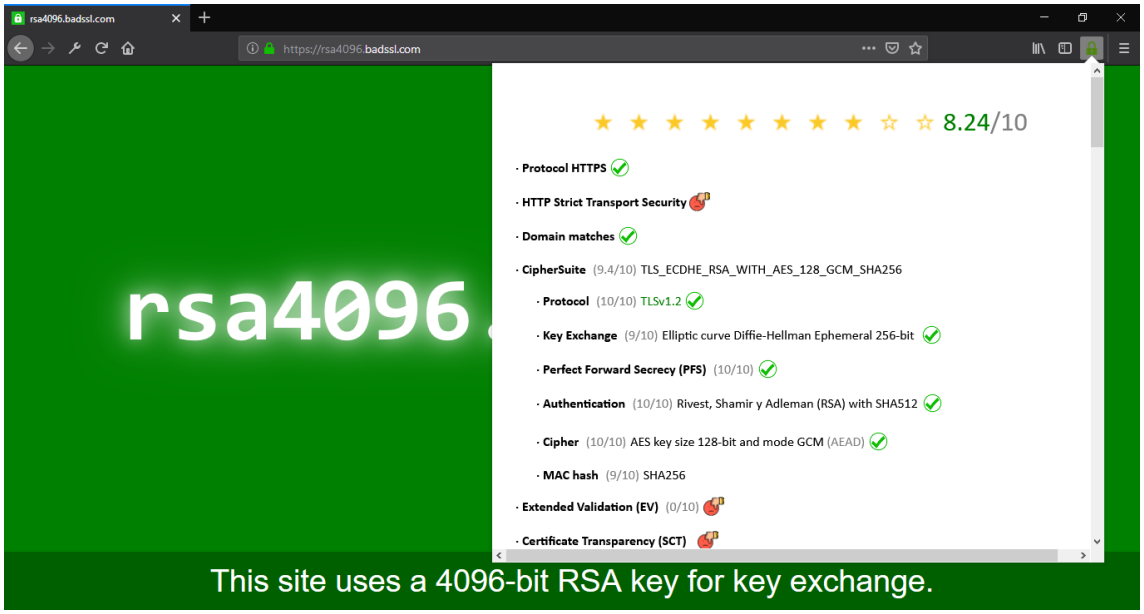
Cas on la clau pública del certificat del servidor *web* usa l'algoritme ECC amb una mida de la clau de 384 bits.



Il·lustració 30: test extensió rankSSL al domini 'ecc384.badssl.com'

12. https://rsa4096.badssl.com

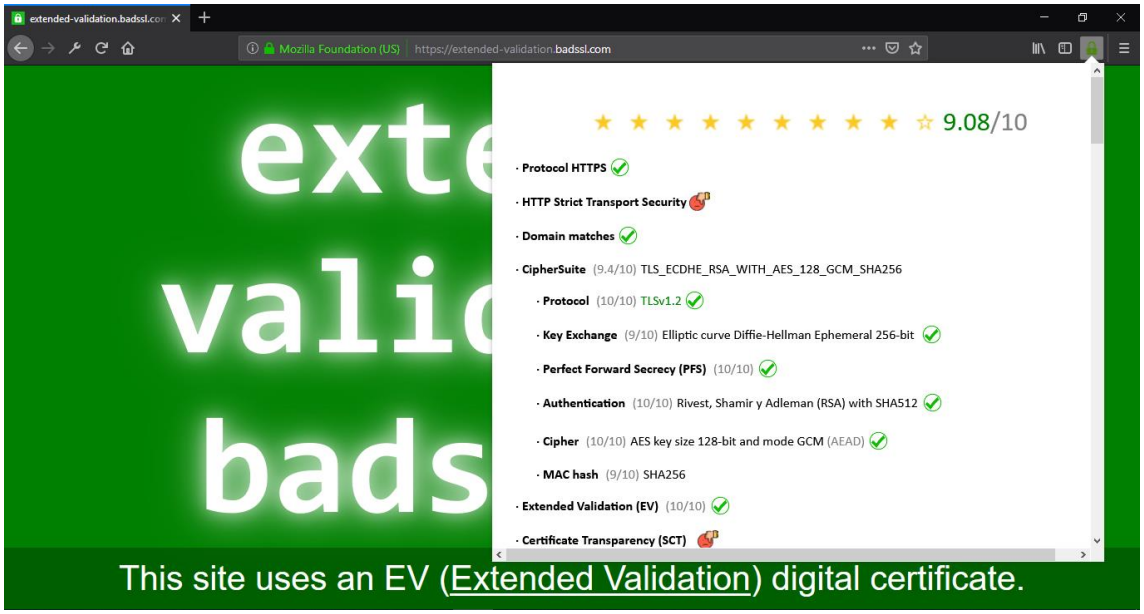
Cas on el intercanvi de claus (*Key Exchange*) entre client i servidor ha estat realitzat amb l'algoritme RSA amb una mida de la clau de 4096 bits.



Il·lustració 31: test extensió rankSSL al domini 'rsa4096.badssl.com'

13. https://extended-validation.badssl.com

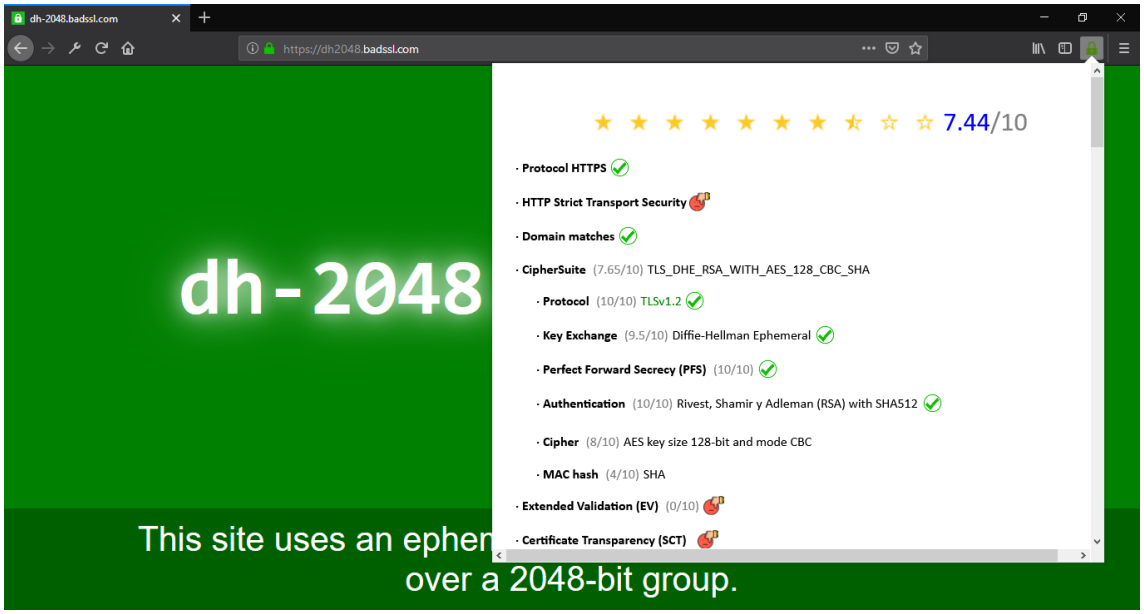
Cas on la propietat 'extended validation' la conté el certificat del servidor *web*. Es pot observar una bona puntuació total, gràcies també a la bona puntuació de la resta de paràmetres.



Il·lustració 32: test extensió rankSSL al domini 'extended-validation.badssl.com'

14. https://dh2048.badssl.com

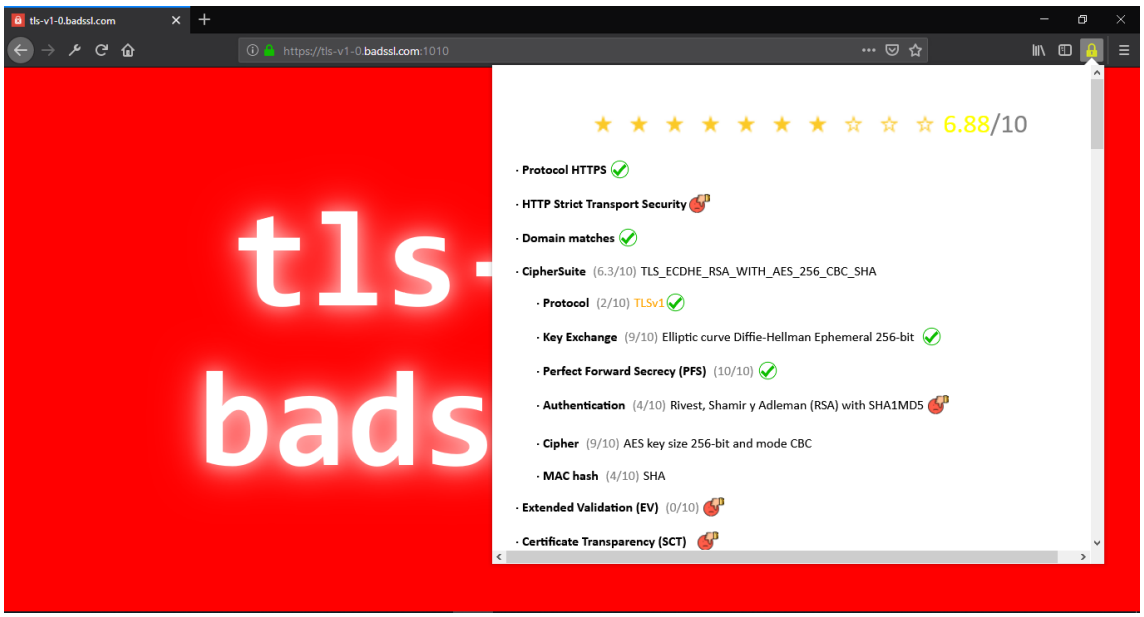
Cas on el intercanvi de claus s'ha realitzat utilitzant l'algoritme DH amb una mida de la clau de 2048 bits.



Il·lustració 33: test extensió rankSSL al domini 'dh-2048.badssl.com'

15. https://tls-v1-0.badssl.com:1010

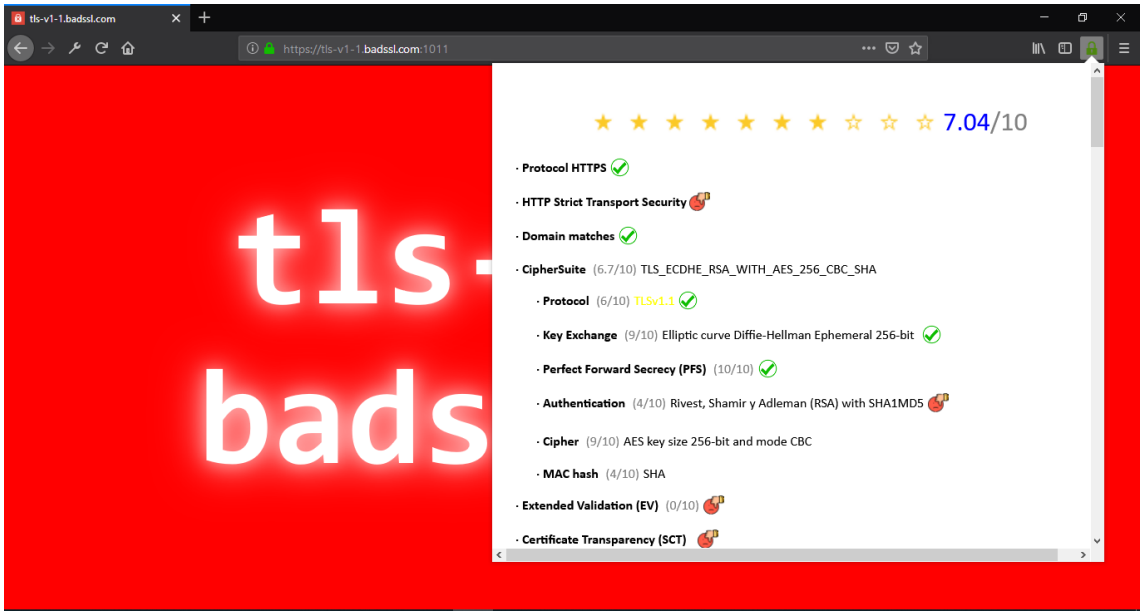
Cas on s'usa el protocol TLS versió 1.0. Es pot observar una davallada de la puntuació total.



Il·lustració 34: test extensió rankSSL al domini 'tls-v1-0.badssl.com'

16. https://tls-v1-1.badssl.com:1011

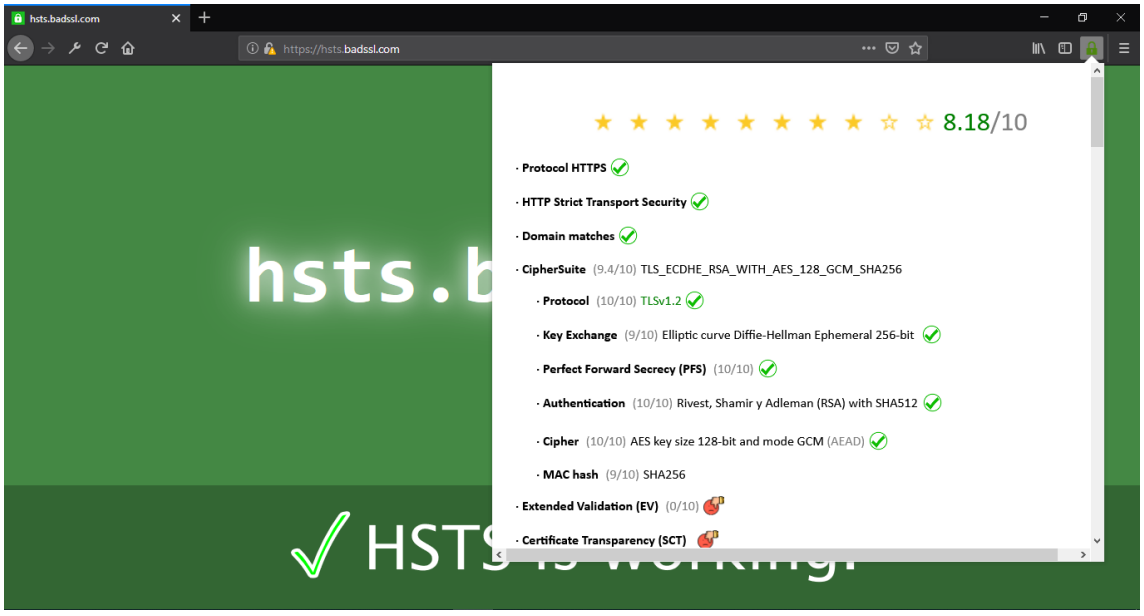
Cas on s'usa el protocol TLS versió 1.1.



Il·lustració 35: test extensió rankSSL al domini 'tls-v1-1.badssl.com'

17. https://hsts.badssl.com

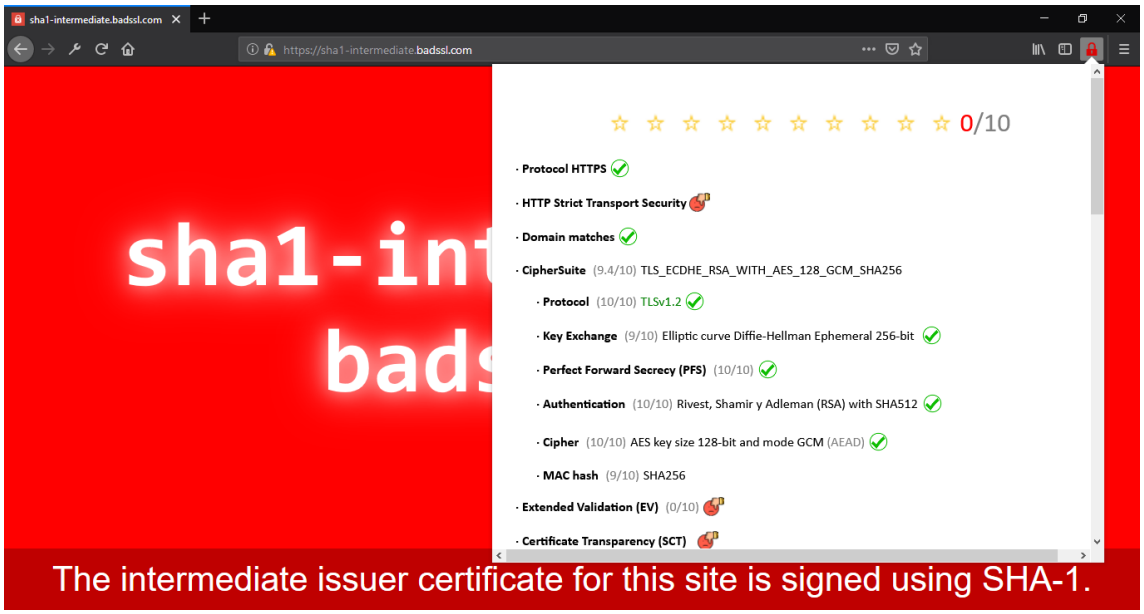
Cas on s'usa la propietat 'hsts' ('HTTP Strict Transport Security').



Il·lustració 36: test extensió rankSSL al domini 'hsts.badssl.com'

18. https://sha1-intermediate.badssl.com

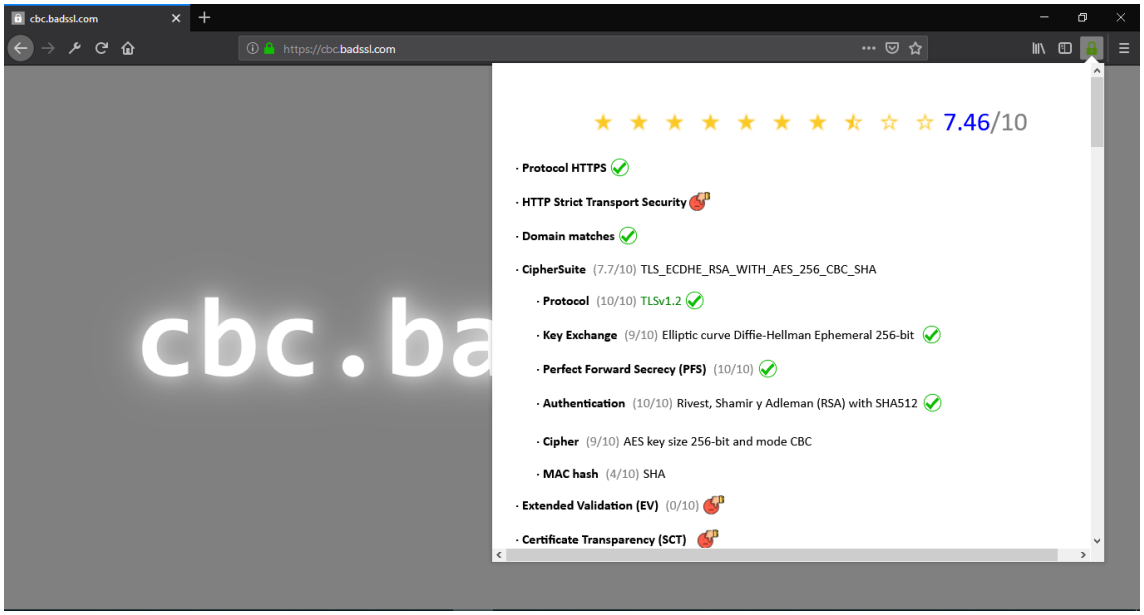
Cas on el certificat CA està firmat amb l'algoritme SHA-1. Es pot observar que la puntuació cau a 0 degut a que el navegador (versions més recents) no confia en el certificat pel algoritme de la firma emprat, doncs, no es pot construir la cadena de certificació de forma satisfactòria.



Il·lustració 37: test extensió rankSSL al domini 'sha1-intermediate.badssl.com'

19. https://cbc.badssl.com

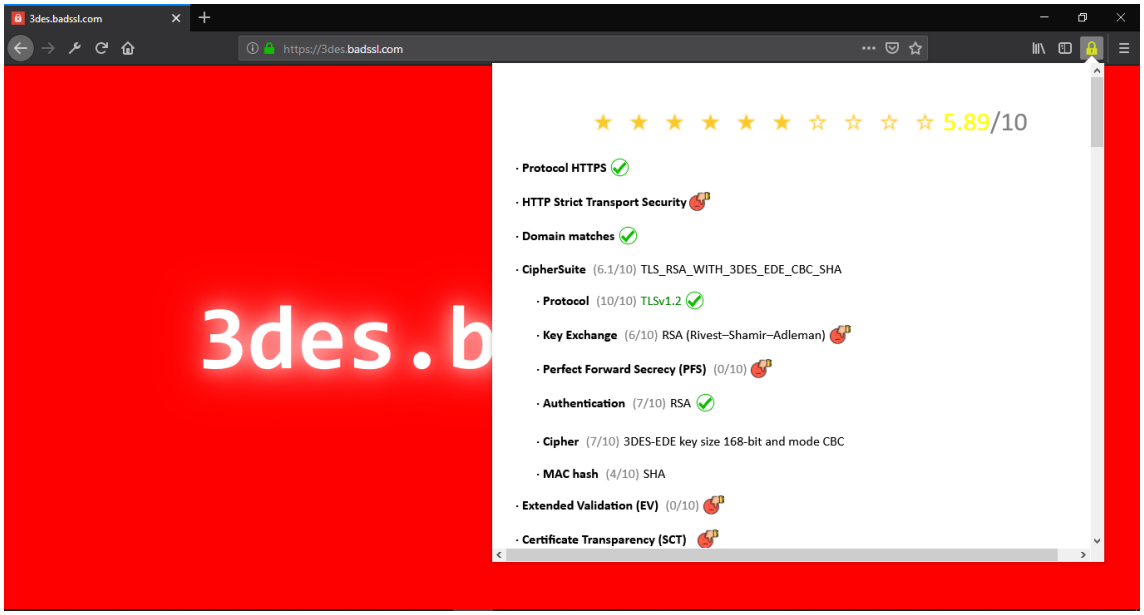
Cas on s'usa un algoritme pel xifratge (*Bulk cipher*) mode CBC.



Il·lustració 38: test extensió rankSSL al domini 'cbc.badssl.com'

20. https://3des.badssl.com

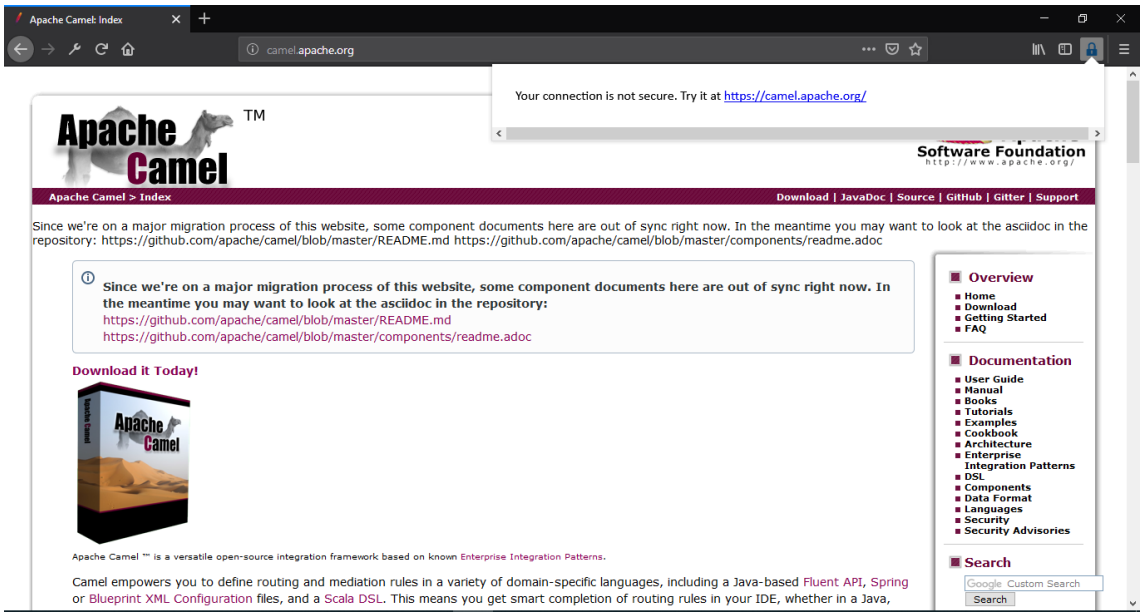
Cas on s'usa l'algoritme 3DES i mode CBC pel xifratge (*Bulk cipher*).



Il·lustració 39: test extensió rankSSL al domini '3des.badssl.com'

21. http://camel.apache.org

Cas on la connexió no utilitza el protocol segur de HTTP, es pot observar que ho indica i mostra una URL alternativa possible amb el protocol HTTPS.



Il·lustració 40: captura pantalla on es mostra missatge en cas que el lloc web no sigui protocol HTTPS

22. https://www.uoc.edu

En l'annex B es pot observar un exemple de la informació emmagatzemada nivell local (format JSON) pel domini 'uoc.edu' quan es consulta i s'executa l'extensió.

3.4.2.1 Resultat de les proves

L'extensió per cada domini consultat ha passat el *checklist* de forma satisfactòria. S'ha calculat la puntuació correctament o s'ha anul·lat a zero quan es requeria (ex.: domini no coincideix amb el certificat), la informació presentada té el format esperat i és coherent. També s'ha contrastat de forma satisfactòria, amb la informació sobre els certificats i la connexió a nivell de seguretat que el navegador proporciona (*Mozilla Firefox*).

Pel que fa a la versió no segura del protocol HTTP el comportament també és correcte; mostra la possible versió de la URL amb el protocol HTTPS (s'informa però no es verifica).

3.4.3 Anàlisi de la *World Wide Web*

Per poder tenir una valoració aproximada de quina és la seguretat de la *web* a Internet s'han valorat els 500 dominis més visitats (segons 'The Moz Top 500') de forma automatitzada i així, poder treure'n una conclusió general (actualment existeixen més de 300 milions de dominis [51]);

Per poder fer la tasca de forma automàtica s'ha creat un script BASH que rep els dominis extrets de la pàgina web, mou el cursor a la posició de l'extensió en el navegador *web*, es prem aquest, s'emmagatzema el 'popup' generat (fitxer HTML), i un cop tenim tots els dominis a nivell local, amb l'eina 'sed' per a Linux filtrem les dades que ens interessa (domini i puntuació global).

Posteriorment he passat les dades a un fitxer CSV i he pogut filtrar dades i treure estadístiques (mitjana aritmètica) a través de programari per a fulls de càlcul.

Ens trobem que dels 500 dominis més visitats n'hi ha que implementen el protocol HTTPS i HTTP, doncs s'ha procedit a extreure dos dades:

- Pels 500 dominis amb només protocol HTTPS (439 dominis) la puntuació global mitjana és de 8,44.
- Pels 500 dominis (es consideren protocol HTTP i HTTPS) la puntuació global mitjana és de 7.44.

3.5 Comparativa amb altres solucions semblants

L'extensió és molt semblant a altres existents tals com s'ha comentat en el capítol 2 del treball. La diferència rau, i com a objectiu s'ha aconseguit, en no només mostrar la informació de la seguretat de la connexió i a nivell de certificat, sinó que es té en consideració tota la cadena de certificació i a més a més, existeix un algoritme de puntuació que es mostra a l'usuari.

A nivell de puntuació és molt semblant però amb una lleugera reducció degut a la consideració de tota la cadena de certificats i la mida de claus pels diferents algoritmes per a l'any 2018.

Si ho comparem amb l'extensió de referència ('SSLeuth') podrem diferenciar que:

1. Mostra informació sobre tota la cadena.
2. Considera més algoritmes (tals com SHA3...)
3. Compatible amb els navegadors més moderns (*Mozilla Firefox* 57+) i fàcilment portable a altres (W*ebExtensions*).
4. Mostra més informació sobre la connexió i del(s) certificat(s).
5. Informa de la propietat *Certificate Transparency Status*.
6. Informa de la propietat HSTS.
7. Emmagatzema informació sobre la seguretat a nivell local en format JSON per a posterior consultes i/o tractaments.

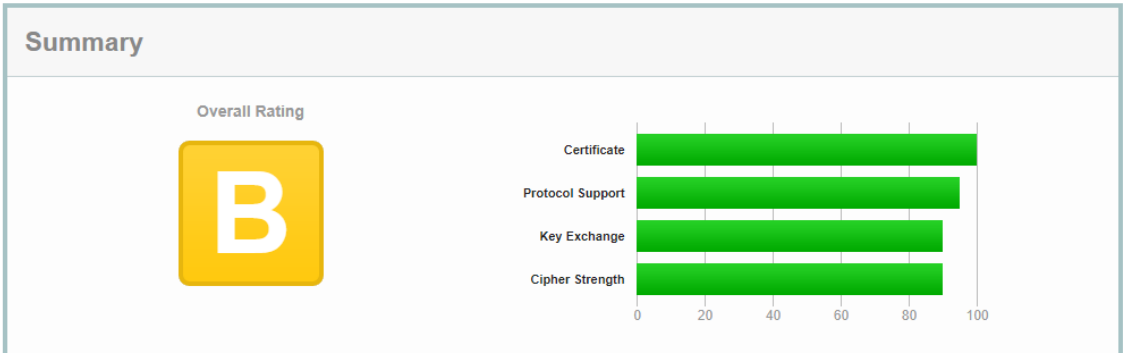
Comparem la puntuació amb el servei que ofereix Qualys per obtenir un resultat sobre la qualitat de la seguretat d'un domini. Per comparar, s'han seleccionat les URL amb la més puntuació més alta (qualys.com), una amb puntuació entre 7 i 9 punts (facebook.com) i una de zero (expired.badssl.com).

D'aquesta manera podrem comprovar com es comporta Qualys respecte l'extensió pel cas d'una que es considera molt segura, un que és simplement segura i una amb una particularitat que fa caure a zero la puntuació en el cas de l'extensió.

Comparativa:

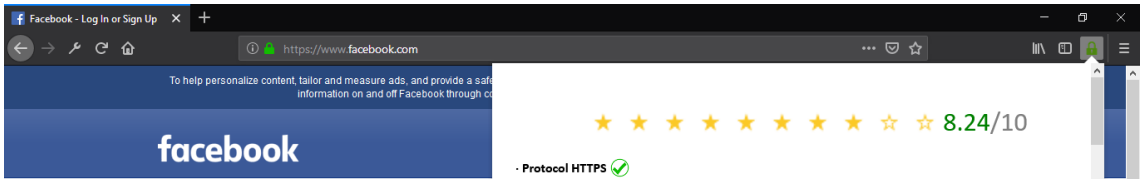
1. URL: https://www.facebook.com

- Qualys:



Il·lustració 41: valoració del servei 'SSL Server Test' de Qualys de la seguretat servidor web 'facebook.com'

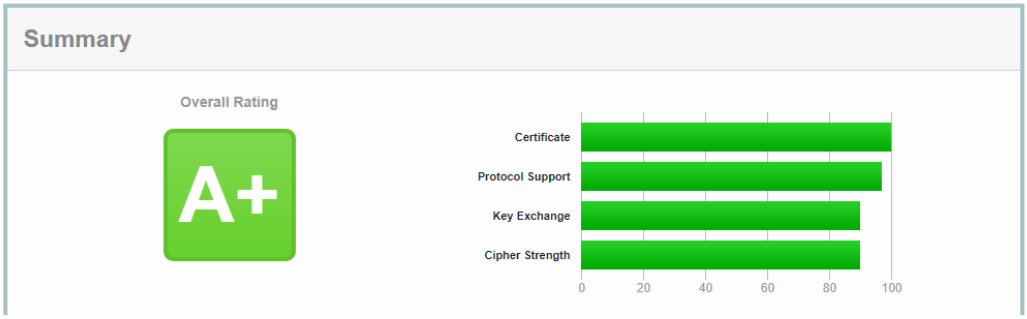
- RankSSL:



Il·lustració 42: valoració extensió rankSSL de la seguretat servidor web 'facebook.com'

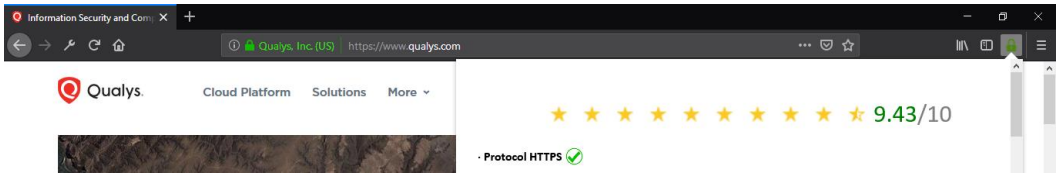
2. URL: https://www.qualys.com

- Qualys:



Il·lustració 43: Valoració del servei 'SSL Server Test' de Qualys de la seguretat servidor web 'qualys.com'

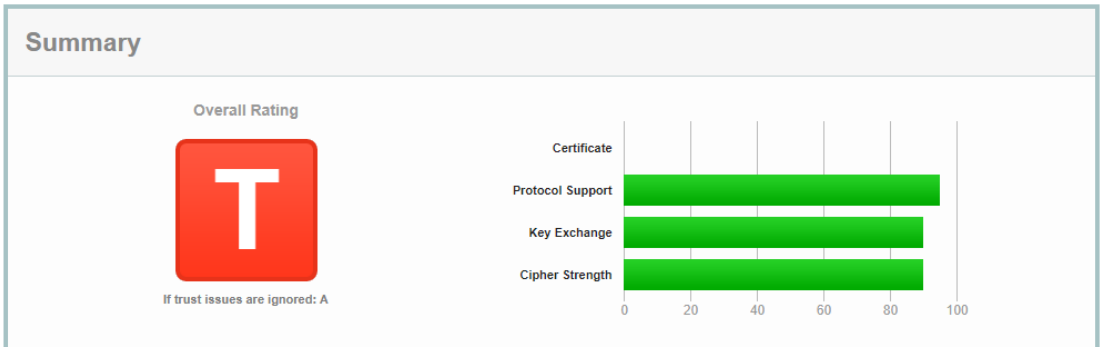
- RankSSL:



Il·lustració 44: valoració extensió rankSSL de la seguretat servidor web 'qualys.com'

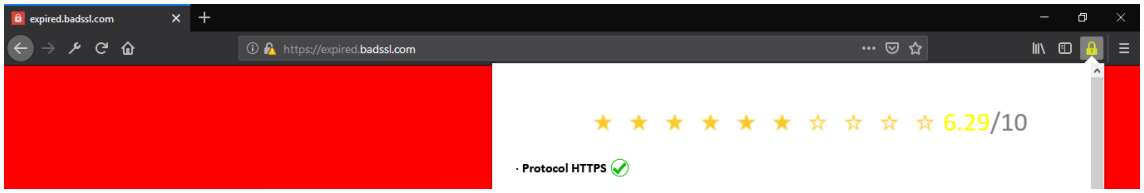
3. URL: https://expired.badssl.com

- Qualys:



Il·lustració 45: valoració del servei 'SSL Server Test' de Qualys de la seguretat servidor web 'expired.badssl.com'

- rankSSL:



Il·lustració 46: valoració extensió rankSSL de la seguretat servidor web 'expired.badssl.com'

'rankSSL' puntua baix la seguretat de la connexió a diferència de Qualys que la puntuació del certificat l'avalua a 0, i com a conseqüència, la puntuació total és 'T' (significa que no és confiable la connexió degut a que el certificat està caducat). Si no es té en consideració la validesa del certificat, llavors la puntuació és 'A' que és pràcticament igual a la de 'rankSSL' (sense considerar la validesa estaria al voltant de 8 aproximadament).

Conclusions comparativa:

La puntuació és similar a la que mostra el servei de Qualys, referent internacional, amb la consideració que l'algoritme de puntuació és diferent i no és numèric; no obstant, és força semblant i la informació que es mostra (la de valor per l'usuari que navega) també ho és.

Podríem considerar que la validesa del certificat és crítica i puntuar 0 la puntuació global, però si el certificat no ha estat revocat, que podria ser el pitjor escenari possible, s'ha decidit que tingui un pes en la puntuació i que no sigui crític.

Per últim, s'ha de tenir en compte que Qualys és un servei *web* a diferència de l'extensió 'rankSSL' que està integrada en el navegador, fet que proporciona com a mínim accessibilitat i usabilitat.

3.6 Definició possibles millores futures

Possibles millores de l'extensió:

- Inclusió comprovació estat revocació a través del protocol OCSP.
- Analitzar i considerar la mida de la clau emprada per a l'autenticació i el intercanvi de claus en la puntuació en els casos que la API actualment no proporciona la informació.
- Automatització per poder valorar en *batch* varis dominis.
- Inclusió del concepte 'Certificate Transparency' en l'algoritme de càlcul de la puntuació quan es consideri adequat.
- Possibilitat de modificar la puntuació dels algoritmes per part de l'usuari.
- Mostrar puntuació global sense prémer botó extensió.

4. Conclusions generals

S'ha aconseguit complir amb els objectius establerts en el punt 1.2, doncs es pot parlar d'èxit en el desenvolupament del treball. No obstant, l'extensió, tal i com s'ha comentat en l'apartat de millores, té deficiències parcials per poder valorar totalment la qualitat de la seguretat en una connexió a un servidor *web*.

Aquestes deficiències son en gran part degut a la immaduresa del sistema de desenvolupament d'extensions (la API de *WebExtensions*), que per una banda es guanya portabilitat (multi navegador) però s'han perdut moltes funcionalitats i falta de documentació/informació comparat amb el sistema anterior. Aquest fet ha dificultat molt el desenvolupament de l'extensió allargant-lo considerablement.

S'ha hagut de buscar solucions de tercers per poder agregar funcionalitats que de per si la API no proporciona i, en la meva opinió, les hauria d'incorporar ja que en el sistema anterior sí ho feia. Solució que ha provocat moments difícils i crítics per poder seguir el desenvolupament degut a la recerca i/o creació d'aquests, i no poder destinar més temps en el contingut a analitzar.

L'extensió té un grau de capacitat potent per seguir desenvolupant-la i poder incorporar nous algoritmes, nova puntuació i, fins i tot, canviar l'algoritme de càlcul de la puntuació. En cas que la API per a les *WebExtensions* millora (s'incorporen noves funcionalitats tals com OCSP o es corregeixen errors) seria interessant augmentar la versió de l'extensió.

Pel que fa a nivell de seguretat, després d'utilitzar l'extensió per valorar llocs *web* observo que en general la seguretat per llocs *web* molt coneguts és força alta; es busca una relació de seguretat i rendiment compensada (es compleixen els requisits mínims actuals en seguretat de vàries organitzacions internacionals).

La quantitat d'algoritmes i les combinacions entre aquests és complexa i és canviant; si es troben vulnerabilitats, si la mida de la clau emprada es consideri insuficient, quan hi hagi noves versions de protocols TLS, nous algoritmes, s'hauria d'actualitzar l'extensió, doncs la considero oberta a canvis que plasmin la realitat del moment a nivell de seguretat per a la *web*.

Pel que fa la seguretat de la *web* a Internet, dels 500 dominis més visitats a Internet amb una connexió segura, es pot considerar que és alta (semblant a la que implementa 'https://google.es').

Però de tots aquests dominis, degut a que n'hi ha que no implementen una connexió HTTPS (puntuació global 0), es pot considerar la seguretat mig-alta, però una mica lluny del que s'hauria d'exigir per l'any actual (màxim trobat 9,45 de la *web* de Qualys). També cal tenir en compte que els serveis que ofereixen aquests dominis consultats és força heterogeni i seria més precís realitzar estudis per sector i/o servei.

5. Bibliografia consultada

[1] “The Transport Layer Security (TLS) Protocol Version 1.2”, https://tools.ietf.org/html/rfc5246, [Online; accedit 19-Setembre-2018]. 1.1

[2], “TLS/SSL Explained – Examples of a TLS Vulnerability and Attack, Final Part”, https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part, [Online; accedit 19-Setembre-2018]. 1.1

[3] “Browser Extensions” https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions, [Online; accedit 19-Setembre-2018]. 1.1

[4]“Webextensions API”, https://developer.mozilla.org/en-US/Add-ons/, [Online; accedit 26-Setembre-2018]. 1.2

[5] “The Transport Layer Security (TLS) Protocol Version 1.3”, https://tools.ietf.org/html/draft-ietf-tls-tls13-28, [Online; accedit 1-Octubre-2018]. 2

[6] “Deprecating Secure Sockets Layer Version 3.0”, https://tools.ietf.org/html/rfc7568, [Online; accedit 1-Octubre-2018]. 2

[7] “Prohibiting Secure Sockets Layer (SSL) Version 2.0”, https://tools.ietf.org/html/rfc6176, [Online; accedit 1-Octubre-2018]. 2

[8], “An overview of the SSL or TLS handshake”, https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_7.1.0/com.ibm.mq.doc/sy10660_.htm, [Online; accedit 1-Octubre-2018]. 2

[9] “Comparación de implementaciones TLS”, https://es.wikipedia.org/wiki/Comparación_de_implementaciones_TLS, [Online; accedit 1-Octubre-2018]. 2.1

[10], “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, https://tools.ietf.org/html/rfc5280, [Online; accedit 1-Octubre-2018]. 2.1.1

[11], “Introduction to ASN.1”, https://www.itu.int/en/ITU-T/asn1/Pages/introduction.aspx, [Online; accedit 1-Octubre-2018]. 2.1.1

[12], “X.690”, https://en.wikipedia.org/wiki/X.690, [Online; accedit 01-Octubre-2018]. 2.1.1

[13] “Extended Validation Certificate”, https://en.wikipedia.org/wiki/Extended_Validation_Certificate, [Online; accedit 01-Octubre-2018]. 2.1.1.1.1

[14] “¿Qué es un conjunto de cifrado (cipher suite) y cómo funciona en SSL/TLS?”, https://www.deacosta.com/que-es-un-conjunto-de-cifrado-cipher-suite-y-como-funciona-en-ssltls/, [Online; accedit 2-Octubre-2018]. 2.2

[15] “Public Key Pinning Extension for HTTP”, https://tools.ietf.org/html/rfc7469, [Online; accedit 30-Octubre-2018]. 2.7

[16] “HTTP Strict Transport Security (HSTS)”, https://tools.ietf.org/html/rfc6797, [Online; accedit 2-Octubre-2018]. 2.2.1

[17] “Certificate Transparency”, https://tools.ietf.org/html/rfc6962, [Online; accedit 2-Octubre-2018]. 2.2.1

[18] “Calomel SSL Validation”, https://calomel.org/firefox_ssl_validation.html, [Online; accedit 2-Octubre-2018]. 2.2

[19] “SSLeuth”,https://github.com/sibiantony/ssleuth, [Online; accedit 2-Octubre-2018]. 2.2

[20] “CipherFox”, https://github.com/gavinhungry/cipherfox, [Online; accedit 2-Octubre-2018]. 2.2

[21], “JavaScript APIs”, https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API, [Online; accedit 2-Octubre-2018]. 2.2.1

[22] “webRequest.getSecurityInfo()”, https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/webRequest/getSecurityInfo, [Online; accedit 2-Octubre-2018]. 2.2.1

[23] “Web technology for developers”, https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import, [Online; accedit 2-Octubre-2018]. 2.2.1

[24] “Example Firefox add-ons created using the WebExtensions API”, https://github.com/mdn/webextensions-examples, [Online; accedit 2-Octubre-2018]. 2.3

[25] “A WebExtension Guide”, https://dev.to/christiankaindl/a-webextension-guide-36ag, [Online; accedit 2-Octubre-2018]. 2.4

[26] “npmjs”, https://www.npmjs.com/, [Online; accedit 6-Octubre-2018]. 2.5

[27] “Node.js”, https://nodejs.org/en/, [Online; accedit 6-Octubre-2018]. 2.5

[28] “webpack”, https://webpack.js.org/, [Online; accedit 6-Octubre-2018]. 2.6

[29] “SSL Server Test”, https://www.ssllabs.com/ssltest/, [Online; accedit 6-Octubre-2018]. 2.7

[30] “A memorable site for missconfiguration on HTTPS”, https://badssl.com/, [Online; accedit 6-Octubre-2018]. 2.7

[31] “ciphers - SSL cipher display and cipher list tool.”, http://openssl.cs.utah.edu/docs/apps/ciphers.html, [Online; accedit 6-Octubre-2018]. 2.7

[32] “CCN-CERT BP/07 Recomendaciones implementación HTTPS”, https://www.ccn-cert.cni.es/informes/informes-de-buenas-practicas-bp/2109-ccn-cert-bp-07-recomendaciones-implementacion-https-1.html, [Online; accedit 6-Octubre-2018]. 2.7

[33] “Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)”, https://tools.ietf.org/html/rfc8247, [Online; accedit 6-Octubre-2018]. 2.7

[34] “MDN web docs HTTP”, https://developer.mozilla.org/en-US/docs/Web/HTTP, [Online; accedit 6-Octubre-2018]. 2.7

[35] “Comparación de implementaciones TLS”, https://es.wikipedia.org/wiki/Comparaci%C3%B3n_de_implementaciones_TLS, [Online; accedit 6-Octubre-2018]. 2.7

[36] “BlueKrypt, Cryptographic Key Length Recommendation” https://www.key-length.com/en/3/, [Online; accedit 6-Octubre-2018]. 2.7

[37] “Object identifiers”, https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.gska100/sssl2oids.htm, [Online; accedit 6-Octubre-2018]. 2.7

[38] “AlgorithmID”, http://javadoc.iaik.tugraz.at/iaik_jce/current/iaik/asn1/structures/AlgorithmID.html, [Online; accedit 6-Octubre-2018]. 2.7

[39] “Testing for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection (OTG-CRYPST-001)”, https://www.owasp.org/index.php/Testing_for_Weak_SSL/TLS_Ciphers,_Insufficient_Transport_Layer_Protection_(OTG-CRYPST-001), [Online; accedit 16Octubre-2018]. 2.7

[40] “SSL Server Rating Guide”, https://www.ssllabs.com/projects/rating-guide/index.html, [Online; accedit 6-Octubre-2018]. 2.8

[41] “About:config entries”, http://kb.mozillazine.org/About:config_entries, [Online; accedit 29-Octubre-2018]. 3

[42] ” Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)”, https://tools.ietf.org/html/rfc6979, [Online; accedit 29-Octubre-2018]. 3.1.2.4

[43], “Elliptic Curves for Security”, https://tools.ietf.org/html/rfc7748, [Online; accedit 29-Octubre-2018]. 3.1.2.4

[44] “Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)”, https://tools.ietf.org/html/rfc4492, [Online; accedit 29-Octubre-2018]. 3.1.2.4

[45] “Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer”, https://tools.ietf.org/html/rfc5656, [Online; accedit 29-Octubre-2018]. 3.1.2.4

[46], “OID Repository”, http://oid-info.com/, [Online; accedit 29-Octubre-2018]. 3.1.2.4

[47] “PKIjs”, https://www.npmjs.com/package/pkijs, [Online; accedit 29-Octubre-2018]. 3.1.2.4

[48] “ASN1js”, https://www.npmjs.com/package/asn1js, [Online; accedit 29-Octubre-2018]. 3.1.2.4

[49] “pvutils”, https://github.com/PeculiarVentures/pvutils, [Online; accedit 29-Octubre-2018]. 3.1.2.4

[50] “Revocation is broken”, https://scotthelme.co.uk/revocation-is-broken/, [Online; accedit 30-Octubre-2018]. 3.2

[51] “¿Cuántos dominios hay en el mundo?”, http://www.solopiensoentic.com/cuantos-dominios-hay, [Online; accedit 22-Diciembre-2018]. 4

Annex A "Exemple d'informació proporcionada per l'extensió"

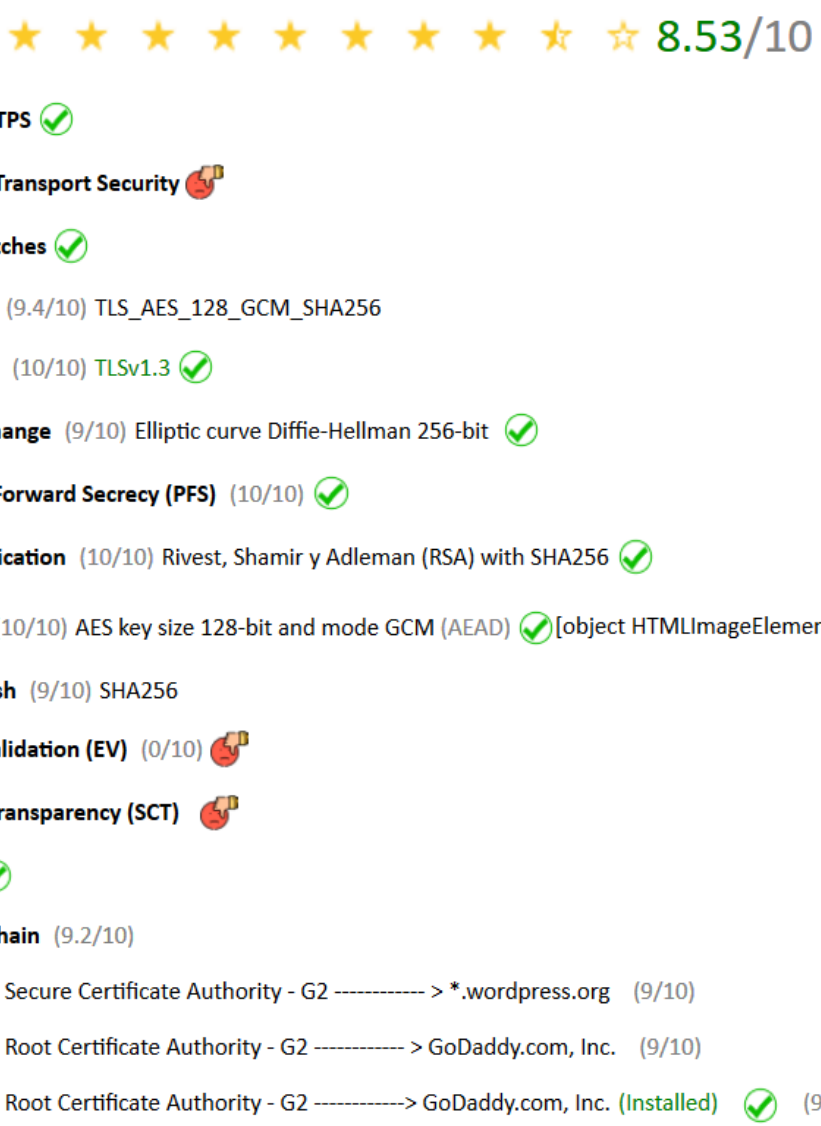
A continuació es descriu a alt nivell com s'estructura i quina informació proporciona l'extensió. Quan es realitzi una petició a un servidor *web* que utilitzi el protocol HTTPS i posteriorment es realitzi la pulsació a la icona de l'extensió, observarem la informació següent (consulta realitzada a 'https://wordpress.org').

Es mostraran tantes estrelles com el valor de la puntuació total arrodonida, es considera també les mitges estrelles (valors entre 0.35 i 0.65). Al costat dret de les estrelles trobem la puntuació total del màxim possible que pendrà el color en funció del valor (vermell, groc, blau i verd).

Seguidament tenim per a cada concepte analitzat si es compleix amb una icona de 'check' color verd ( ), o en cas contrari una icona amb una cara vermella i una mà amb el pulgar cap a baix ( ). Aquestes icones indiquen que el concepte té una configuració desitjable per a la seguretat o no la té. També, en cas de possibles valors numèrics de puntuació, tals com d'un algoritme emprat, es mostra la icona de correcta si es supera la puntuació de 8 sobre 10.

Si el concepte analitzat té pes sobre la puntuació total (en el algoritme de càlcul) s'ha indicat quina puntuació s'obté sobre el màxim possible sempre. En la il·lustració 18 podem observar un analisis dels conceptes crítics a nivell de la connexió i del certificat de l'entitat final. Per acabar aquest primera secció es mostra les organitzacions (la O del DN o el CN si no es troba en el certificat) que conformen la cadena de certificació (emissor i emès).

A partir d'aquí, es mostrará informació detallada de cada certificat que conforma la cadena de certificació (entitat final, CA intermedi i CA arrel); els conceptes importants son l'algoritme emprat per a la firma i la clau pública, que es valoren quantitativament, que el domini coincideixi (només certificat entitat final) amb el consultat, el ús de la clau del certificat i les restriccions bàsiques (propietats crítiques), la resta son merament informatius.



Il·lustració 47: primera secció informació de l'extensió mostrada sobre la seguretat

----- End-entity Certificate -----

Certificate is valid

Start: 06-11-2017 18:42:01. Started hace un año

End: 15-12-2020 21:11:21. Expires en 2 años

Issuer:

CN=Go Daddy Secure Certificate Authority - G2,OU=http://certs.godaddy.com/repository/,O="GoDaddy.com, Inc.",L=Scottsdale,ST=Arizona,C=US

Issued:

*.wordpress.org **(matched)** **URL visited:** wordpress.org

CN=*.wordpress.org,OU=Domain Control Validated

Alternative names: *.wordpress.org wordpress.org

Serial number: 00:A7:62:D0:DC:E1:E5:E5:72

Signature algorithm: RSA with SHA256 (9/10)

Public key algorithm: RSA 2048-bit (9/10)

SHA1:

5B:92:E6:49:F7:8E:FA:4E:AB:37:28:FB:2A:C0:D2:5F:EE:8C:03:C2

SHA256:

CC:3C:B3:35:89:A7:C5:94:65:25:E0:58:7F:AE:24:5E:C5:13:8B:78:D9:6E:14:AE:13:10:D2:8B:B3:59:95:DF

Critical extensions:

Basic Constraints
Key Usages

Key usages (Critical extension):

Digital Signature

Key Encipherment

Certificate constraints (critical extension): End-entity

Extended key usages:

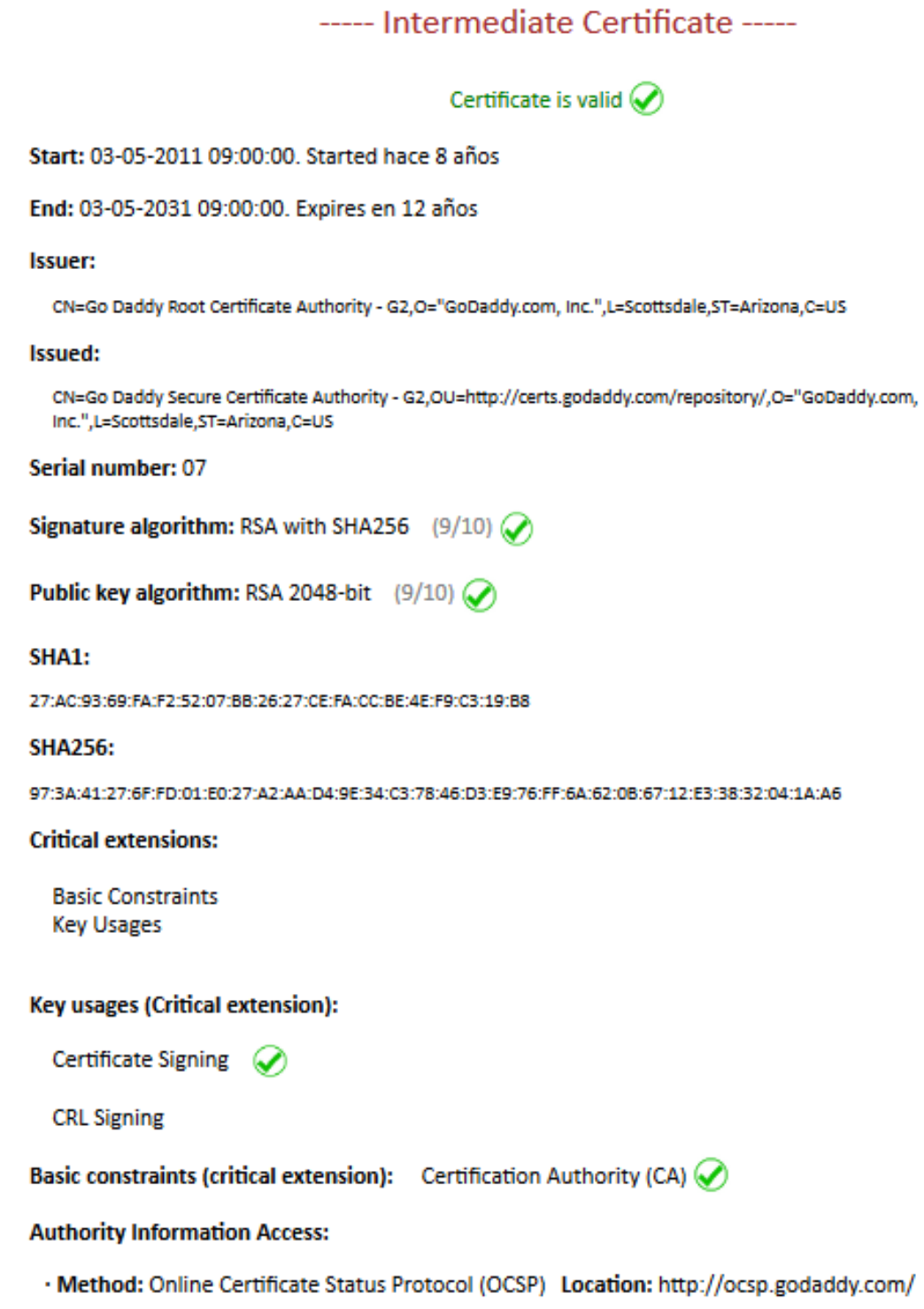
Server Authentication

Client Authentication

Authority Information Access:

- **Method:** Online Certificate Status Protocol (OCSP) **Location:** http://ocsp.godaddy.com/
- **Method:** CA Issuers **Location:** http://certificates.godaddy.com/repository/gdig2.crt

Il·lustració 48: segona part informació (certificat entitat final) de l'extensió mostrada sobre la seguretat



Il·lustració 49: tercera part informació (certificat intermedi) de l'extensió mostrada sobre la seguretat

Il·lustració 50: quarta part informació (certificat arrel) de l'extensió mostrada sobre la seguretat

Annex B “Exemple contingut fitxer local generat per l’extensió”

Exemple contingut fitxer emmagatzemat a nivell local al consultar un domini y valorar la seguretat per l’extensió rankSSL (domini ‘uoc.edu’):


```
{
 "URL": "www.uoc.edu",
 "dades": {
  "https": true,
  "score": 8.195999999999998,
  "url": "www.uoc.edu",
  "critics": {
   "domainMatches": true,
   "trustable": true
  },
  "scores": {
   "connection": {
    "weight": 0.6,
    "score": 8.009999999999998,
    "cipherSuite": {
     "score": 9.299999999999999,
     "weight": 0.7,
     "value": "TLS_AES_128_GCM_SHA256",
     "protocol": {
      "name": "TLSv1.3",
      "score": 10,
      "weight": 0.1
     },
     "keyExchange": {
      "score": 10,
      "weight": 0.3
     },
     "signature": {
      "name": "ECDSA-P256-SHA256",
      "score": 10,
      "weight": 0.1
     },
     "bulkCipher": {
      "name": "TLS_AES_128_GCM",
      "score": 8,
```

```
      "weight": 0.2,
      "AEAD": true
     },
     "MAChash": {
      "name": "SHA256",
      "score": 9,
      "weight": 0.3
     }
    },
    "extendedValidation": {
     "score": 0,
     "weight": 0.15
    },
    "perfectForwardSecrecy": {
     "score": 10,
     "weight": 0.15
    },
    "certificateTransparency": {
     "name": “not_applicable”
    }
   },
  "chainCertificates": {
   "weight": 0.4,
   "score": 8.475,
   "certificates": {
    "certificateWeb": {
     "score": 9.5,
     "weight": 0.5,
     "publicKey": {
      "name": "Elliptic Curve 256",
      "score": 10,
      "weight": 0.3
     },
     "signature": {
      "name": "RSA with SHA-256",
      "score": 9,
      "weight": 0.5
     },
     "valid": {
      "value": true,
      "score": 10,
```

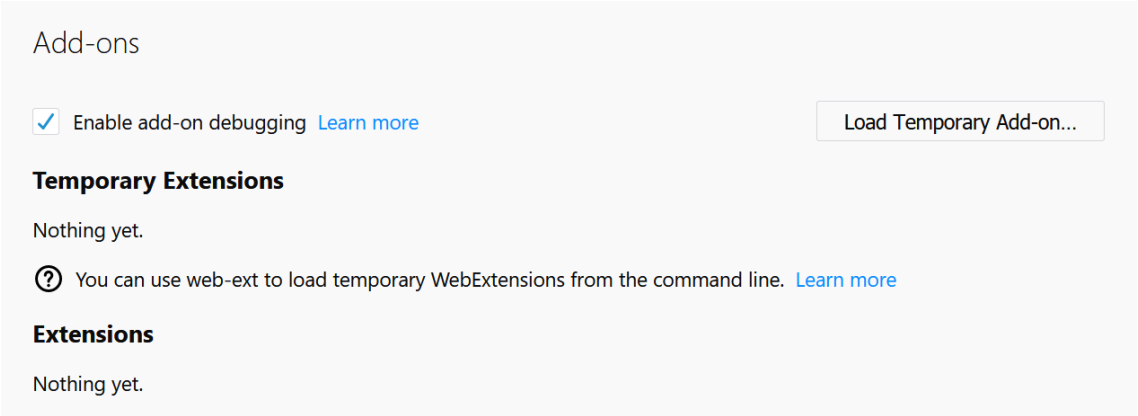
```
    "weight": 0.2
   },
   "basicConstraints": {
    "cA": false,
    "critical": true
   }
  },
  "certificateCA": {
   "score": 5.699999999999999,
   "weight": 0.25,
   "publicKey": {
    "name": "RSA 2048",
    "score": 9,
    "weight": 0.3
   },
   "signature": {
    "name": "RSA with SHA-1",
    "score": 2,
    "weight": 0.5
   },
   "valid": {
    "value": true,
    "score": 10,
    "weight": 0.2
   },
   "basicConstraints": {
    "cA": true,
    "critical": true
   }
  },
  "certificateCAroot": {
   "score": 9.2,
   "weight": 0.25,
   "publicKey": {
    "name": "RSA 2048",
    "score": 9,
    "weight": 0.3
   },
   "signature": {
    "name": "RSA with SHA-1",
    "score": 2,
```

```
          "weight": 0.5
         },
         "valid": {
          "value": true,
          "score": 10,
          "weight": 0.2
         },
         "basicConstraints": {
          "cA": true,
          "critical": true
         }
        }
       }
      }
     }
    }
}
```

Annex C "Manual d'instal·lació"

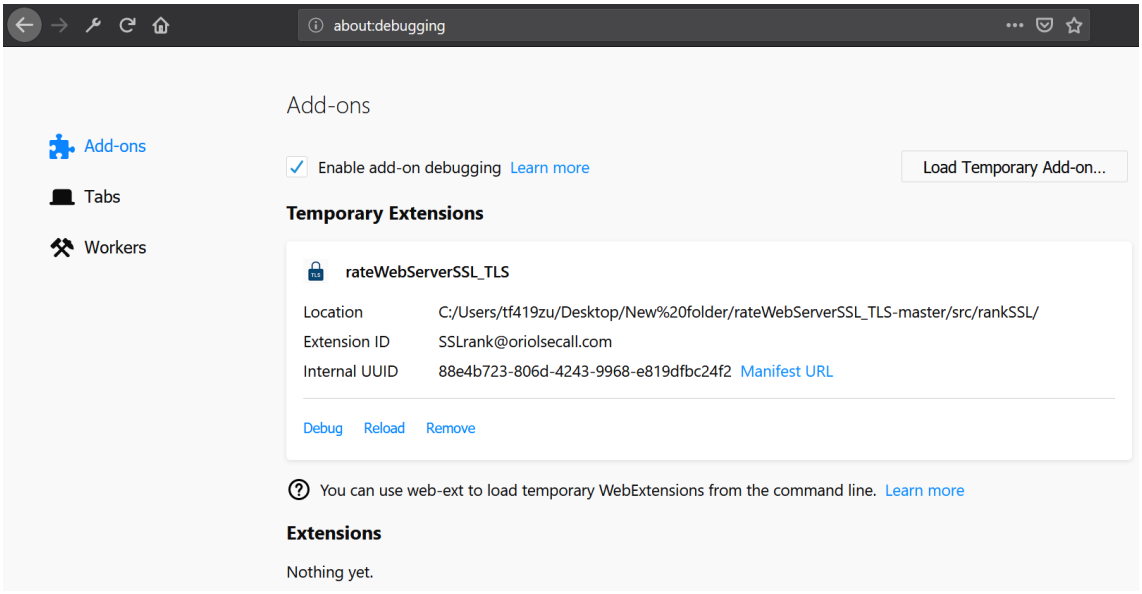
- Instruccions amb el codi font:

1. Obtenir el codi font.
2. En la ruta "/src/rankSSL/" trobem el fitxer manifest anomenat 'manifest.json' que és el usarem per carregar l'extensió al navegador *Mozilla Firefox*.
3. Obrir el navegador *Mozilla Firefox* i introduir a la barra de direccions 'about:debugging'.



Il·lustració 51: about:debugging Mozilla Firefox

4. Activar 'Enable add-on debugging' si es vol depurar i/o veure la consola del navegador amb els *logs* generats per l'extensió.
5. Prémer el botó a la part superior dreta 'Load Temporary Add-on' i seleccionar el fitxer del punt 2 'manifest.json' per pujar-lo al navegador.



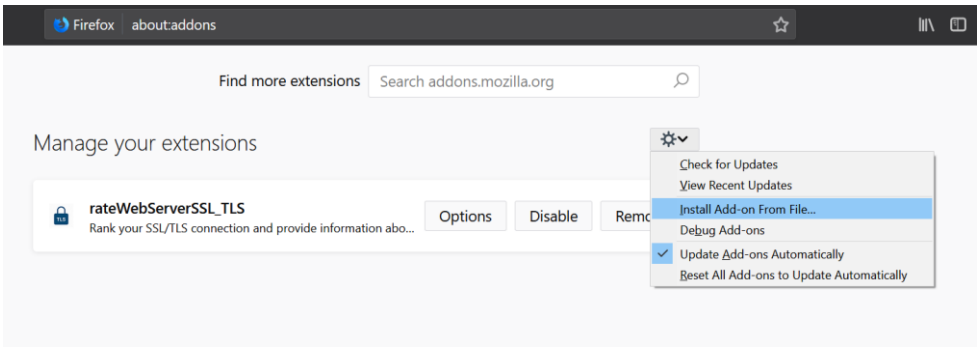
Il·lustració 52: pujar fitxer manifest de l'extensió.

6. Navegar a qualsevol pàgina *web* que usi protocol HTTPS per tal d'explotar totes les funcionalitats que ofereix l'extensió.

- Instruccions amb l'extensió empaquetada sense firmar/verificar:

1. Descarregar-se la versió de *Mozilla Firefox* ('Developer Edition').
2. Introduir a la barra de direccions "about:config" i acceptar els riscs a través del botó mostrat a la pantalla.
3. Canviar 'xpinstall.signatures.required' a valor *false*.
4. Instal·lar l'eina 'npm'.
5. Instal·lar l'eina 'web-ext' des de la línia de comandes (CLI).
   Comanda: npm install --global web-ext
6. Des de la línia de comandes (CLI) escriure la comanda:
   web-ext build -s <ruta_directori_font_extensió>
   ex.: web-ext build -s C:\Users\tf419zu\Desktop\New folder\rateWebServerSSL_TLS-master\src\rankSSL

   A tenir en compte que el directori font (*source*) és el que està contingut en el directori 'src/' i no l'arrel del codi font descarregat. Hem de indicar que el directori font és el 'rankSSL' que es l'arrel de l'extensió.
7. La comanda anterior ens crearà un directori nou anomenat 'web-ext-artifacts' i aquest contindrà l'extensió amb el nom 'rankssl-1.0' en format ZIP.
8. En el navegador *Mozilla Firefox* ('Developer Edition') escriure a la barra de direccions 'about:addons' o, alternativament des del menú dirigir-se a l'opció 'add-ons' i instal·lar l'extensió des del fitxer.



Il·lustració 53: instal·lar add-on des de fitxer local.

9. Seleccionar 'Install Add-on From File...' i seleccionar el fitxer extensió ZIP generat en el punt 5 i descrit en el 6 ('rankssl-1.0.zip').
10. Navegar a qualsevol pàgina *web* i prémer el botó de l'extensió amb la icona que es pot veure en la següent il·lustració:

Il·lustració 54: botó icona extensió

Annex D "Manual d'usuari"

1. Navegar a una pàgina *web* (en cas que no s'hagi realitzat el *handshake* completament l'extensió queda a l'espera de rebre la informació necessària per mostrar la informació).
2. Prémer el botó de l'extensió 'rankSSL'.
3. Veure la informació sobre la seguretat de la connexió HTTPS.
    a. Si la connexió no és xifrada es mostrarà la URL alternativa segura com a enllaç (no es dona garanties que existeixi la implementació HTTPS, només es mostra com a possible alternativa canviant el protocol HTTP per HTTPS).
    b. Si la connexió usa el protocol HTTPS es mostrarà una valoració global i informació sobre la seguretat (connexió i cadena de certificació).
4. Canvi de pestanya en el navegador.
    a. Per obtenir la informació sobre la seguretat de la connexió s'ha de fer la petició al servidor i que aquesta es carregui en el navegador.
    b. Si hi ha un canvi de pestanya, al prémer l'extensió es mostrarà la última informació carregada, és a dir, per conèixer de nou la informació del lloc *web* de la pestanya activa s'haurà de recarregar la pàgina *web* i tornar a prémer la icona de l'extensió.
5. Informació local
    a. Per a cada consulta a un servidor *web* amb protocol HTTPS s'emmagatzemarà la informació en format JSON a nivell local.
    b. La ruta és en el '%APPDATA%\Mozilla\Firefox\Profiles\' (Windows)
    c. Per a altres sistemes operatius consultar al navegador *web* on emmagatzema fitxers locals.