

horizon-backports - Feature - Fonctionnalité #5033

OpenSSL > 1.0 pour TLS 1.2

25/07/2019 22:54 - Eric Seigne

Statut:	Assigned - En cours	Début:	25/07/2019
Priorité:	Normale	Echéance:	
Assigné à:	Eric Seigne	% réalisé:	80%
Catégorie:		Temps estimé:	0.00 heure
Version cible:	Version 11.08		
Description			
À priori il faudrait passer sur OpenSSL version > 1.0 pour avoir le support de TLS 1.2 qui permettrait de ne pas avoir d'alerte sécurité "site web non sécurisé" quand on a un firefox récent sur les postes clients (et le filtrage des sites https du serveur).			

Historique

#1 - 26/07/2019 09:34 - Eric Seigne

Après avoir essayé les branches openssl 1.1 (version openssl-OpenSSL_1_1_1c.tgz) je repasse sur openssl1.0 (openssl-OpenSSL_1_0_2s)

Compilation & Installation dans une sous arborescence pour ne pas avoir de conflit avec la lib ssl 0.9.8 de la distrib

```
./configure --prefix=/usr/local --openssldir=/usr/local/openssl  
make  
make install
```

On passe ensuite à Squid, après quelques heures squid4 ne sera pas possible (C++11 / version de gcc trop ancienne pour pouvoir le compiler) on se rabat donc sur la dernière version de la branche 3 : la 3.5.28 du 15 juillet dernier :-)

Téléchargement depuis <http://www.squid-cache.org/Versions/v3/3.5/>

```
./configure --enable-ssl-crttd --with-openssl=/usr/local/ --without-gnutls --with-dl --with-cppunit-basedir=/usr  
r --enable-inline --enable-async-io=8 --enable-storeio="ufs,aufs,diskd" --enable-removal-policies="lru,heap" --  
enable-delay-pools --enable-cache-digests --enable-underscores --enable-icap-client --enable-follow-x-forward  
ed-for --enable-arp-acl --enable-esi --enable-zph-qos --disable-translation --with-filedescriptors=65536 --wi  
th-large-files --with-default-user=proxy --with-included-ltdl --with-dl  
  
make
```

#2 - 26/07/2019 11:08 - Eric Seigne

Bon, forcément ça ne passe pas tout seul ...

```
servecole squid[3698]: assertion failed: support.cc:1762: "0"
```

heureusement qu'on a le code source de squid pour aller voir ce qui fait cette assertion 0 ...

```
// SSL_get_certificate is buggy in openssl versions 1.0.1d and 1.0.1e

// Try to retrieve certificate directly from SSL_CTX object

#if SQUID_USE_SSLGETCERTIFICATE_HACK
    X509 ***pCert = (X509 ***)sslContext->cert;
    X509 * cert = pCert && *pCert ? **pCert : NULL;
#elif SQUID_SSLGETCERTIFICATE_BUGGY

    X509 * cert = NULL;

    assert(0);

#else
    // Temporary ssl for getting X509 certificate from SSL_CTX.

    Ssl::SSL_Pointer ssl(SSL_new(sslContext));
    X509 * cert = SSL_get_certificate(ssl.get());
#endif
```

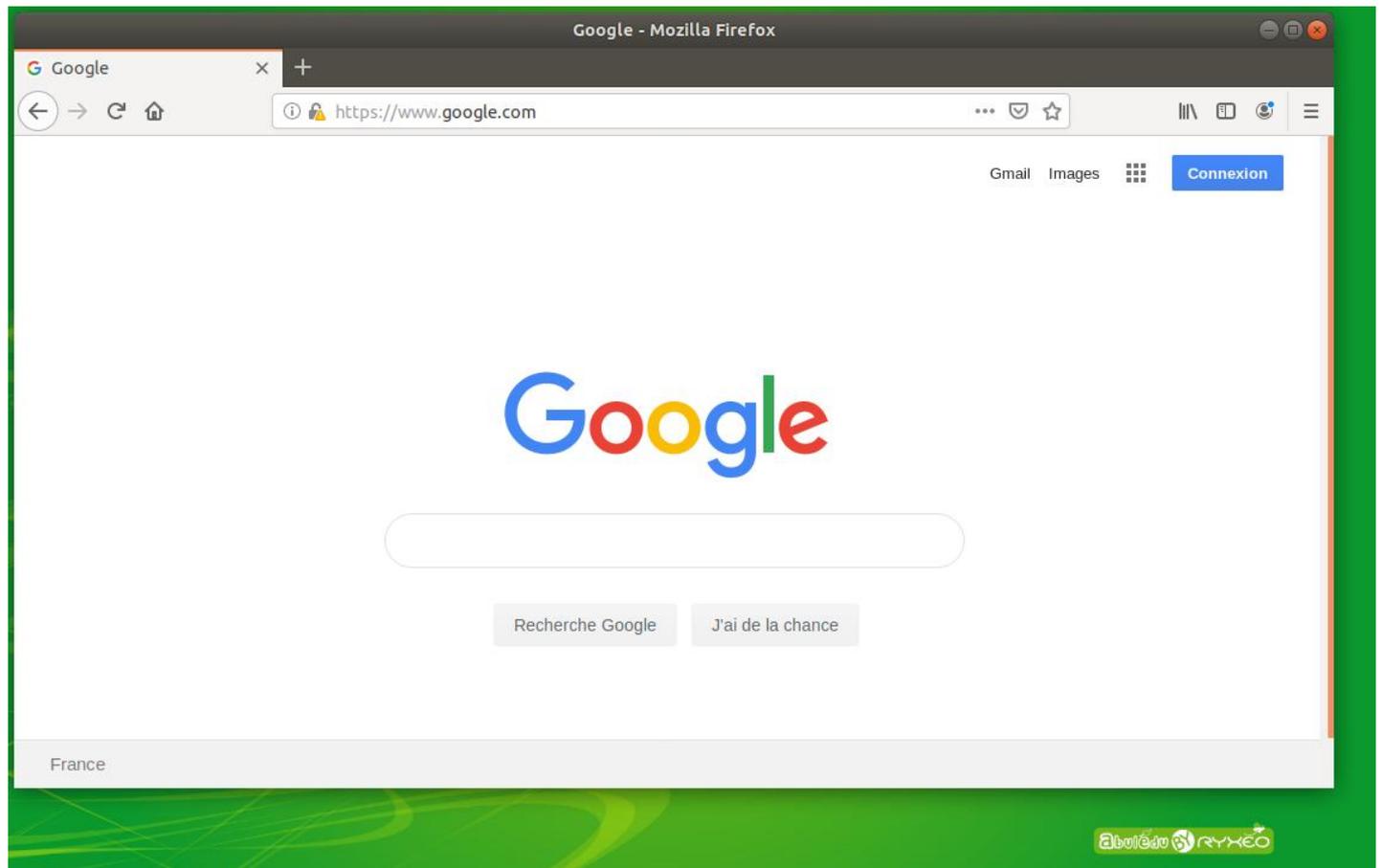
on est donc à priori dans le cas particulier "SQUID_SSLGETCERTIFICATE_BUGGY" lié à openssl versions 1.0.1d and 1.0.1e ... sauf qu'on a compilé la 1.0.2s ... bref modification du code, compilation et on relance le tout ...

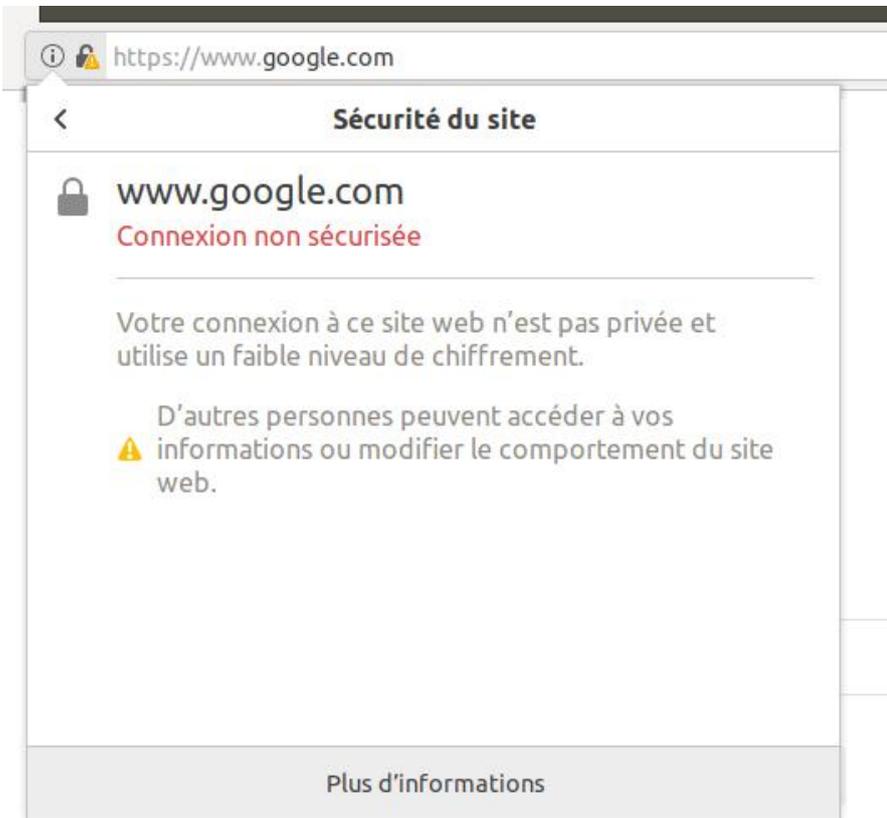
#3 - 26/07/2019 11:18 - Eric Seigne

- Fichier abuledu-ssl_tls1.2-ko_01.jpg ajouté
- Fichier abuledu-ssl_tls1.2-ko_02.jpg ajouté
- Fichier abuledu-ssl_tls1.2-ko_03.jpg ajouté
- Fichier abuledu-ssl_tls1.2-ok_01.jpg ajouté
- Fichier abuledu-ssl_tls1.2-ok_02.jpg ajouté
- Fichier abuledu-ssl_tls1.2-ok_03.jpg ajouté

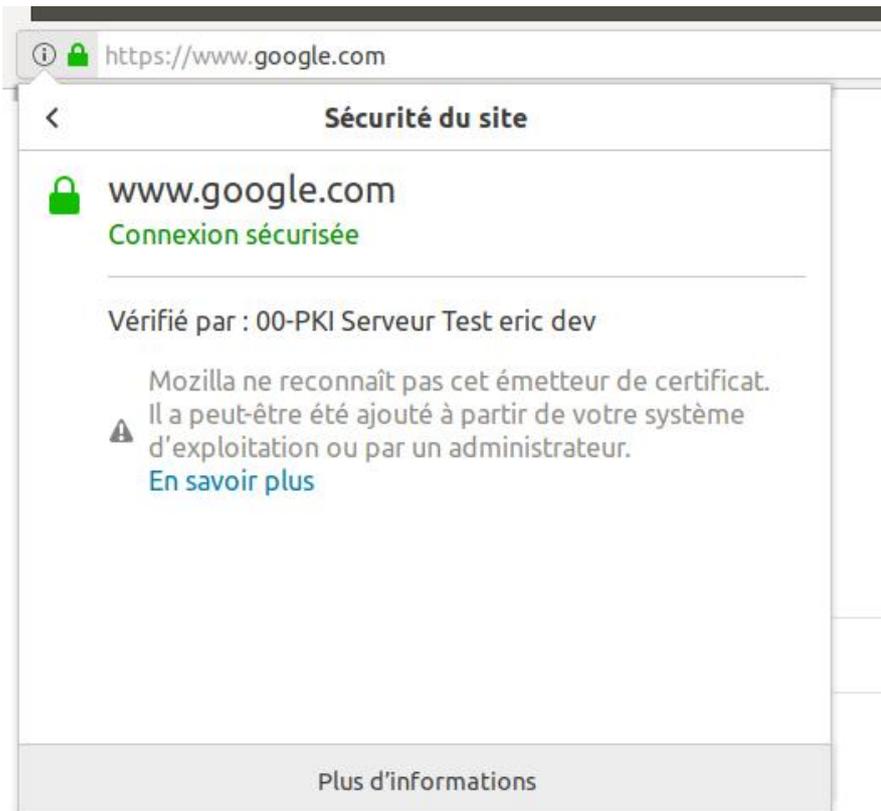
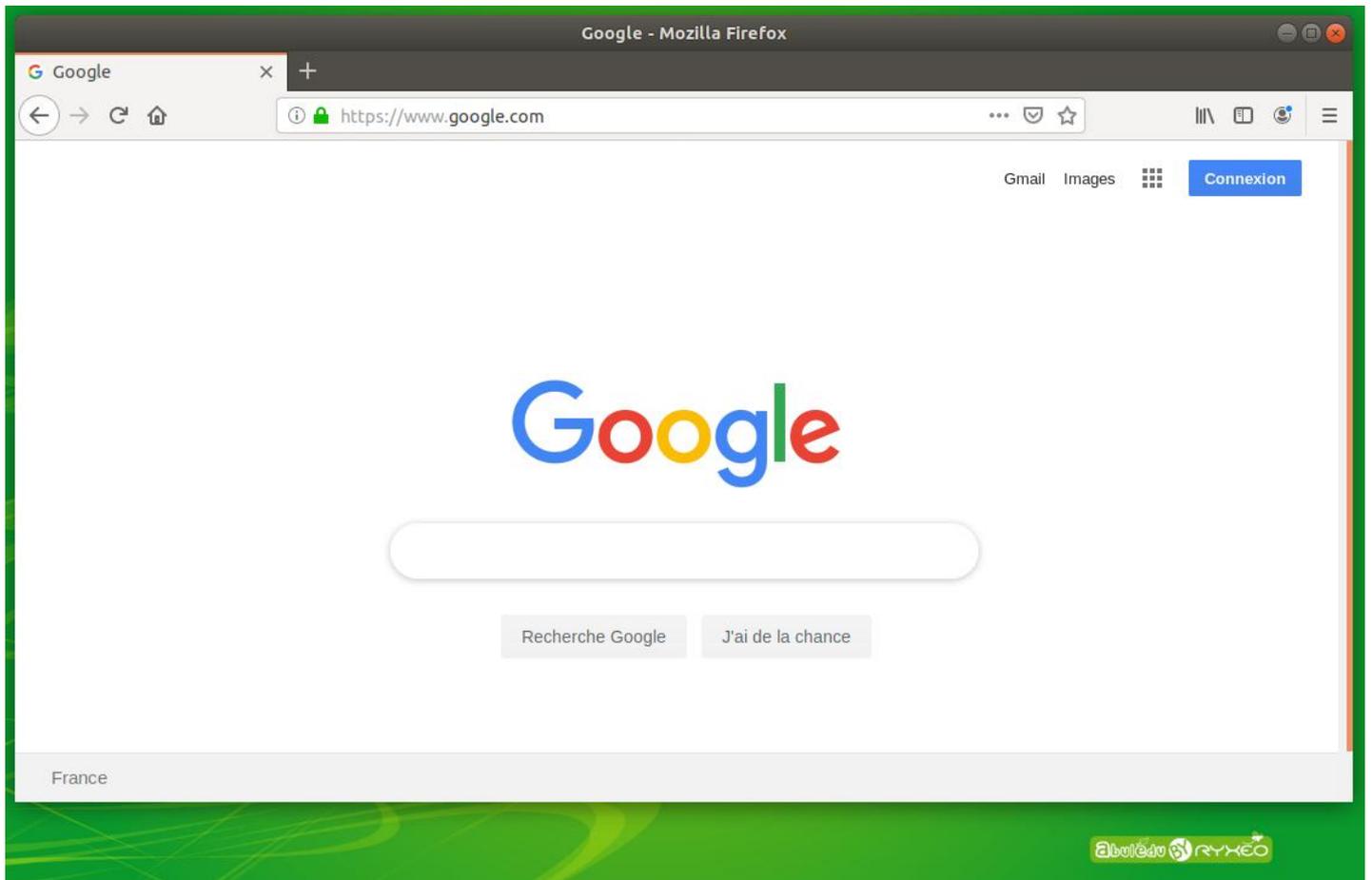
Et ça marche :)

Situation initiale:





Situation actuelle:



▼ Connexion :

- Version du protocole : TLSv1.2
- Suite de chiffrement : TLS_RSA_WITH_AES_128_CBC_SHA
- Groupe d'échange de clés : aucun
- Algorithme de signature : aucune

▼ Hôte www.google.com :

- HTTP Strict Transport Security : **Activé**
- Public Key Pinning : **Activé**

▼ Certificat :

- ▼ Émis à
 - Nom commun (CN) : www.google.com
 - Organisation (O) : Google LLC
 - Unité d'organisation (OU) : <Non disponible>
- ▼ Émis par
 - Nom commun (CN) : Serveur SSL local
 - Organisation (O) : 00-PKI Serveur Test eric dev
 - Unité d'organisation (OU) : 00-PKI Locale

Le protocole TLSv1.2 est donc bien maintenant utilisable :-)

Il reste à mettre tout ça en paquets ...

Et ça c'est une autre histoire ... mais on sait que c'est possible !

#4 - 01/08/2019 13:31 - Eric Seigne

- Fichier 20190801-test_tls.png ajouté

- % réalisé changé de 0 à 80

Copie du mail envoyé sur la liste support

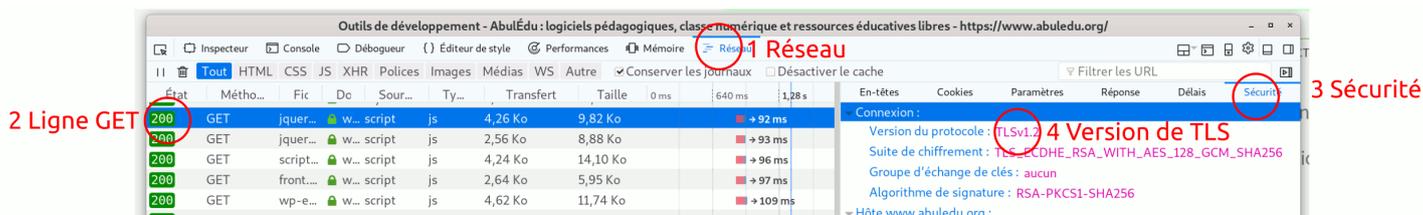
Le voilà ... le paquet qui apporte la gestion des sites en TLS v1.2

ATTENTION IL FAUT VALIDER CE PAQUET !

il me faut donc un ou deux testeur, le risque est de ne plus avoir du tout de filtrage actif donc merci de ne pas faire ça sur un serveur en prod avec des utilisateurs ... par contre une école ou un site qui n'a pas d'utilisateur autre que vous pendant quelques jours est tout à fait adapté pour le test ...

1. ouvrez une session utilisateur, ouvrez un navigateur web (firefox); appuyez sur F12 et allez sur le site <https://abuledu.org/>

2. une fois le site ouvert, regardez dans la fenêtre développeur de firefox, cliquez sur réseau puis sur la 1ere ligne GET puis sur "sécurité" pour voir en quelle version de TLS est faite la transaction, normalement TLS 1.0



3. ensuite pour faire la mise à jour, se connecter en abuladmin ou root directement sur le serveur

2. lancer la commande abuledu-upgrade

3. à la fin de la mise à jour rebootez le serveur, vous pouvez vérifier à l'aide de la commande suivante que la bonne version de squid s'est installée avant de faire le reboot

```
dpkg -l squid*
```

ça devrait donner la version 3.5.28-1lucid1~abuledu2

4. une fois le reboot terminé, relancez un navigateur web, retournez sur le même site web après avoir ouvert la fenetre des outils de developpement de firefox (F12) et vérifiez que vous êtes maintenant en TLS 1.2 ...

Merci d'avance pour vos tests, je n'ai pas pu aller jusqu'au bout de mes tests de mon côté par manque de matériels, je suis en train de réinstaller des clients lourds 15.08 par exemple, je n'ai pu tester qu'avec le 19.08 en cours de dev et je n'ai pas la moindre idée de comment vont se comporter les anciens firefox ...

si vous validez que tout marche (du 1er coup ça serait vraiment une belle surprise) on proposera à tout le monde de faire la mise à jour avant la rentrée :-)

Fichiers

abuledu-ssl_tls1.2-ko_02.jpg	24,6 ko	26/07/2019	Eric Seigne
abuledu-ssl_tls1.2-ko_01.jpg	47,9 ko	26/07/2019	Eric Seigne
abuledu-ssl_tls1.2-ko_03.jpg	28,8 ko	26/07/2019	Eric Seigne
abuledu-ssl_tls1.2-ok_01.jpg	45,6 ko	26/07/2019	Eric Seigne
abuledu-ssl_tls1.2-ok_02.jpg	27,2 ko	26/07/2019	Eric Seigne
abuledu-ssl_tls1.2-ok_03.jpg	23,2 ko	26/07/2019	Eric Seigne

