ANF - Protection des données par le chiffrement

Le chiffrement sur les services SSL/TLS

Hervé Ballans - Matthieu Herrb

12 février 2016

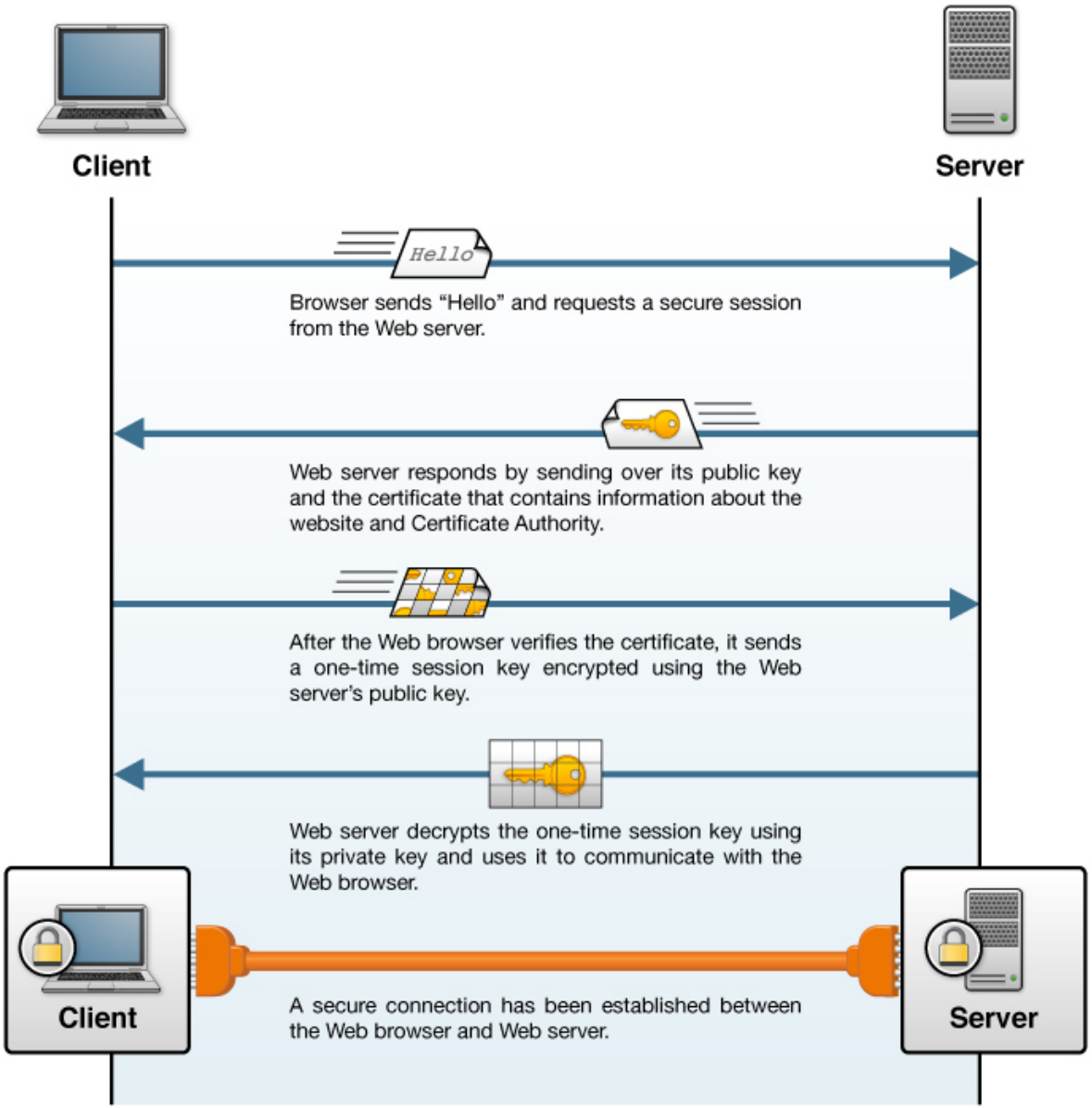
Tous les services réseau doivent être sécurisés, donc chiffrés

Sécurisation des échanges via TLS

- TLS est un protocole de sécurisation des échanges sur TCP/IP
- Objectifs de sécurité garantis :
  - l'**authentification** du serveur ;
  - la **confidentialité** des données échangées (session chiffrée)
  - l'**intégrité** des données échangées
- Anciennes versions : SSL ...
- Modification des protocoles au niveau de la couche transport
  - `http` (80) $\longrightarrow$ `https` (443)
  - `imap` (143) $\longrightarrow$ `imaps` (993)
  - ...
  - ou bien `STARTTLS` sur le même port

Autres protocoles : *IPSec, DNSSec, SSH, Kerberos,...* non traités ici.

Session TLS : exemple



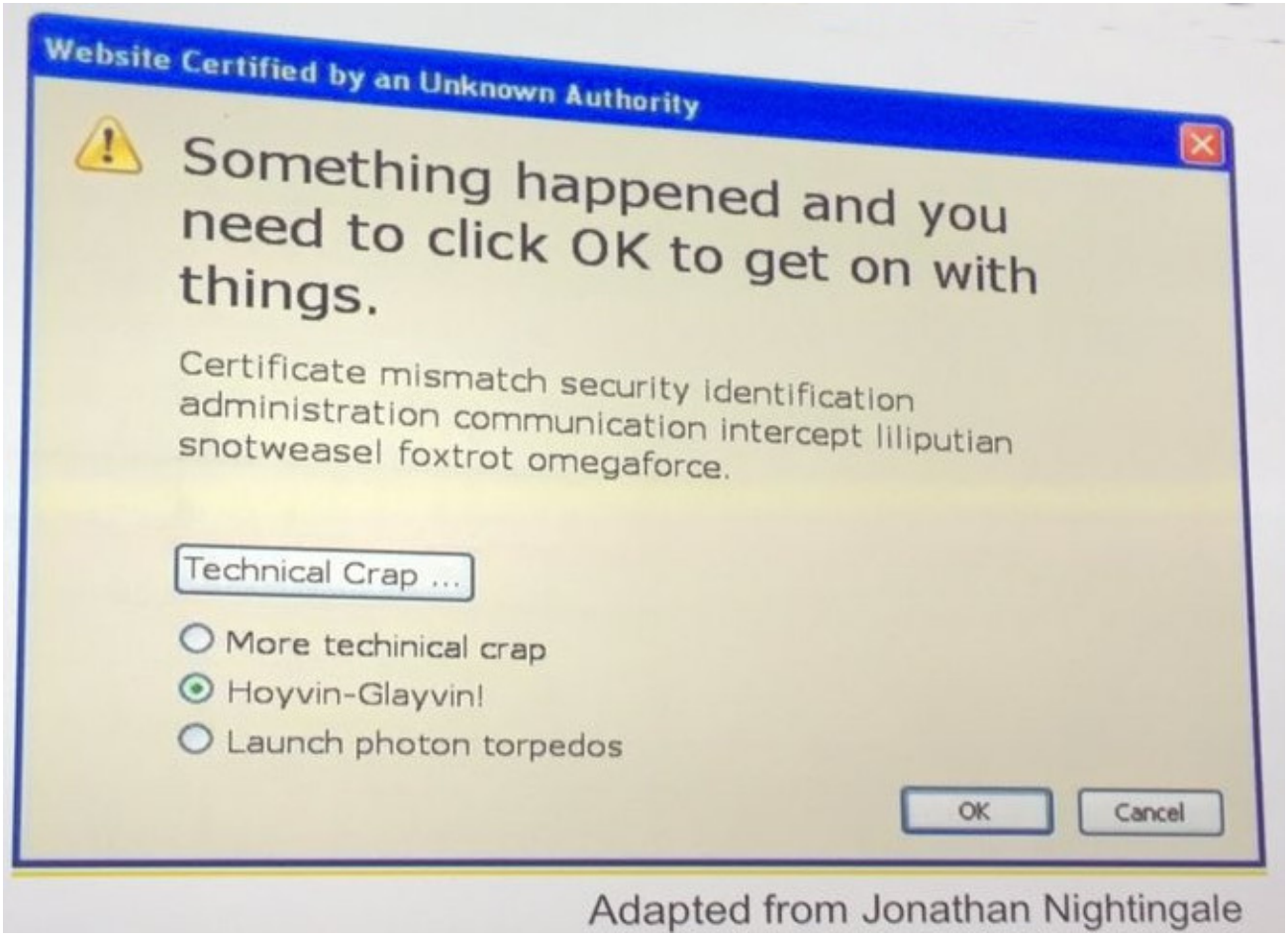
Risques...

Nombreux problèmes...

- Confiance dans les AC racine,
- Vérification des certificats (signature, date de validité, nom,...)
- Avancées de la cryptanalyse (*R.I.P.* RC4, MD5, SHA-1,...)
- Bugs dans les implémentations (Heartbleed, Freak...)
- Mauvaises pratiques (Poodle, Logjam...)
- Clés privées mal protégées (ou publiées sur Github)
- Logiciels obsolètes (Navigateurs, Java,...)
- ...

Le chiffrement n'est utile que s'il est correctement implémenté.

Risques


Website Certified by an Unknown Authority
Something happened and you need to click OK to get on with things.
Certificate mismatch security identification administration communication intercept liliputian snotweasel foxtrot omegaforce.
Technical Crap ...
More techinical crap
Hoyvin-Glayvin!
Launch photon torpedos
OK
Cancel
Adapted from Jonathan Nightingale

Recommandations

Attention à la validité des certificats utilisés :

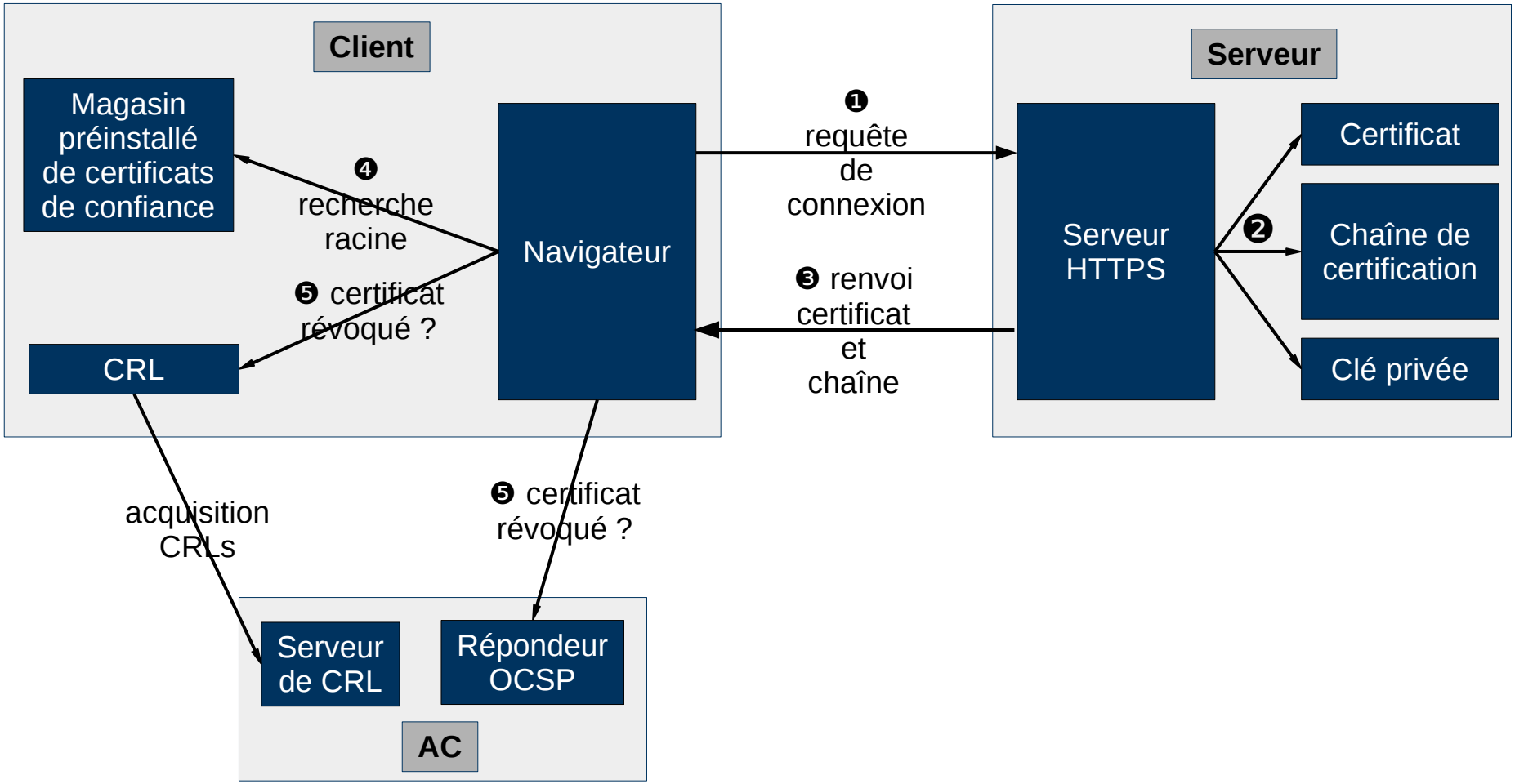
- **Bannir les certificats auto-signés**
- Superviser (nagios,...) l'expiration des certificats utilisés
- Fournir la chaîne de certification aux clients

Renforcer les protocoles de chiffrement :

- TLSv1..TLSv1.2 (supprimer SSLv2 et SSLv3)
- Clés 2048 bits, voir 3072 (RGS)
- Algorithmes modernes, pas export, avec perfect forward secrecy
- Paramètres Diffie-Hellman de longueur suffisante

Mettre à jour les configurations / suivre les avis de sécurité.

Vérification du certificat serveur



Confiance dans les AC ?

Outils / protocoles pour renforcer les vérifications :

- **Cert Patrol** (Firefox) - avertit l'utilisateur des changements de certificat sur les sites déjà visités.
- **HPKP** HTTP Protocol Key Pinning - interdit changement de certificat pour sites connus
- **DANE/TLSA** Publication du certificat du site via DNS(Sec)

Le CA-Browser Forum

- Groupement d'autorités de certification + éditeurs de logiciels (navigateurs)
- Définit des règles de conduite / bonnes pratiques.
- En particulier : types de certificats:

  DV : Domain Validated - vérification de la propriété du nom de domaine (enregistrement DNS / cookie sur serveur)

  OV : Organisation Validated - vérification de la possession du nom de domaine par une organisation (via annuaires externes)

  EV : Extended validation - vérification rigoureuse des éléments

Obtenir des certificats

TCS (Accord géant / Digicert)

- CNRS $\rightarrow$ Portail DSI
- Université(s) $\rightarrow$ Direct Renater

Types de certificats Digicert:

SSL Plus - un seul nom

Multi-Domain SSL - 1-4 noms

Wildcard Plus - wildcard + domaine

EV SSL Plus Extended Validation - un nom

EV Multi-domain EV 1-4 noms

Autres possibilités

Exemple : pour domaines achetés directement chez un registrar pour un projet, une conférence, non listés coté Renater/TCS

- achat du certificat auprès du registrar (ex. Gandi)
- Let's encrypt (gratuit mais limitations)
- ...

Algorithmes

Asymétrique :

- RSA factorisation nombres premiers au moins 2048bits
- ECDSA Courbes elliptiques clés min 256bits.
- DSS à éviter.

Symétrique :

- AES Standard 128 ou 256 bits
- 3DES 112 bits - à éviter
- RC4 à fuir

HMAC :

- MD5 à éviter
- SHA SHA-1 déprécié
- SHA256 SHA-2

Perfect Forward Secrecy

Définition - Confidentialité Persistante

La découverte par un adversaire de la clé privée d'un correspondant (secret à long terme) ne compromet pas la confidentialité des communications passées.

Échange initial de clé privée basée sur un protocole de Diffie-Hellman.
Deux variantes:

- EDH basé sur RSA
- ECDHE basé sur courbes elliptiques

Génération de paramètres DH maison pour EDH :

```
openssl dhparam 2048 -out /etc/pki/tls/private/dh2048.pem
```

Analyse d'un serveur : cryptcheck

[HTTPS] www.[redacted].fr (24/01/2016 18:42:03) **C**

Scores C	
Protocole	60 / 100
Échange de clef	100 / 100
Chiffrement	50 / 100
Total	68.0 / 100

Protocoles	TLSv1_2 TLSv1_1 TLSv1
Clefs	Certificat : RSA 4096 bits Diffie Hellman : ECC 521 bits
Dangers	DES3

Algorithme	Clef	DH	
TLSv1_2			
ECDHE-RSA-AES256-SHA	256 bits	ECC 521 bits	PFS
ECDHE-RSA-AES256-SHA384	256 bits	ECC 521 bits	PFS
ECDHE-RSA-AES128-SHA	128 bits	ECC 521 bits	PFS
ECDHE-RSA-AES128-SHA256	128 bits	ECC 521 bits	PFS
AES256-SHA	256 bits		
AES256-SHA256	256 bits		
AES128-SHA	128 bits		
AES128-SHA256	128 bits		
DES-CBC3-SHA	112 bits		DES3
TLSv1_1			
ECDHE-RSA-AES256-SHA	256 bits	ECC 521 bits	PFS
ECDHE-RSA-AES128-SHA	128 bits	ECC 521 bits	PFS
AES256-SHA	256 bits		
AES128-SHA	128 bits		
DES-CBC3-SHA	112 bits		DES3
TLSv1			
ECDHE-RSA-AES256-SHA	256 bits	ECC 521 bits	PFS
ECDHE-RSA-AES128-SHA	128 bits	ECC 521 bits	PFS
AES256-SHA	256 bits		
AES128-SHA	128 bits		
DES-CBC3-SHA	112 bits		DES3

En pratique : serveurs web

SNI permet les hôtes virtuels avec HTTPS sur une seule adresse IP.

- Un seul certificat avec plusieurs `subjectAltName`
- Un certificat par hôte virtuel.
- Un certificat de type *wildcard* si tous les hôtes virtuels sont dans le même domaine.

Utiliser un reverse proxy devant les serveurs qui ne supportent pas du HTTPS moderne.

En pratique : apache

```
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLKeyFile /etc/pki/tls/private/localhost.key
SSLCertificateChainFile /etc/pki/tls/certs/DigiCert.crt

SSLProtocol +TLSv1.2 +TLSv1.1 +TLSv1 -SSLv3 -SSLv2

SSLCipherSuite EECDH+AES:EDH+AES+aRSA

SSLHonorCipherOrder on
```

En pratique : nginx

/etc/nginx/nginx.conf

```
        ssl_certificate /etc/pki/tls/certs/localhost-with-chain.crt;
        ssl_certificate_key /etc/pki/tls/private/localhost.key;
        ssl_prefer_server_ciphers On;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers EECDH+AES:EDH+AES+aRSA;
        ssl_dhparam /etc/pki/tls/private/dh2048.pem;
```

Concaténer les certificats intermédiaires au certificat du site.

En pratique : tomcat

Java 7 ou 8 minimum.

https://www.sslshopper.com/article-how-to-disable-weak-ciphers-and-ssl-2-in-tomcat.html

```
<connector port="443" maxhttpheadersize="8192" address="127.0.0.1"
enablelookups="false" disableuploadtimeout="true" acceptCount="100"
scheme="https" secure="true" clientAuth="false" SSLEnabled="true"
sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
ciphers="TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
    TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,,
    TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA,
    TLS_RSA_WITH_AES_256_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA"
keystoreFile="mydomain.key" keystorePass="password"
truststoreFile="mytruststore.truststore" truststorePass="password"/>
```

En pratique : IIS

$$\text{IIS} >= 7.0$$

1. Open the Group Policy Object Editor (i.e. run gpedit.msc in the command prompt).
2. Expand Computer Configuration, Administrative Templates, Network, and then click SSL Configuration Settings.
3. Under SSL Configuration Settings, open the SSL Cipher Suite Order setting.
4. Set up a strong cipher suite order. See this list of Microsoft's supported ciphers and Mozilla's TLS configuration instructions.

En pratique : postfix

/etc/postfix/main.cf

```
smtpd_tls_security_level = may
smtpd_tls_cert_file = /etc/pki/tls/certs/localhost.pem
smtpd_tls_key_file = /etc/pki/tls/private/localhost.key
smtpd_tls_mandatory_protocols = !SSLv2, !SSLv3
smtpd_tls_mandatory_ciphers = high
smtpd_tls_mandatory_exclude_ciphers = aNULL, MD5
tls_high_cipherlist = EECDH+AES:EDH+AES+aRSA
smtpd_tls_eecdh_grade = strong
smtpd_tls_dh1024_param_file = /etc/pki/tls/private/dh2048.pem
```

En pratique : dovecot

/etc/dovecot/conf.d/10-ssl.conf

```
ssl_cert = </etc/letsencrypt/live/tls.laas.fr/cert.pem
ssl_key = </etc/letsencrypt/live/tls.laas.fr/privkey.pem
ssl_dh_parameters_length = 2048
ssl_protocols = !SSLv2 !SSLv3
ssl_cipher_list = EECDH+AES:EDH+AES+aRSA;
ssl_prefer_server_ciphers = yes
```

En pratique : navigateurs et clients TLS

- Maintenir à jour !
- Vérifier la prise en charge des protocoles cryptographiques forts et l'abandon des protocoles faibles.

Récapitulatif : les points clés

1. Algorithmes et longueur de clé du certificat serveur
   RSA 3072 bits SHA256
2. Protocoles SSL/TLS
   TLSv1 TLSv1.1 **TLSv1.2**
3. Algorithmes et longueur de clé pour échange clé privée (PFS)
   ECDHE 256 / EDH 2048
4. Algorithme et longueur de clé chiffrement symétrique
   AES 128 **256**
5. Algorithme intégrité (HMAC)
   SHA256 SHA384
6. Vérification du certficat au niveau des clients.

Vérification d'une configuration

Basé web :

- Cryptcheck https://tls.imirhil.fr/
- SSLLabs (Qualys) https://www.ssllabs.com/ssltest/index.html

En ligne de commande : CryptCheck
https://github.com/aeris/cryptcheck

Futur

- TLS 1.3 en cours de standardisation
  - supprime algorithmes obsolètes
  - ajout algo symétrique ChaCha20 + Poly1305
  - ajout courbe elliptique ed25519 ?
- Démocratisation HSM ?
- et après...

Bibliographie

- CryptCheck, vérifiez vos implémentations de TLS, *Aeris*, https://blog.imirhil.fr/2015/09/02/cryptcheck-verifiez-implementations-tls.html
- RFC 7525 - Recommendations for Secure Use of TLS and DTLS, *Stéphane Bortzmeyer*, http://www.bortzmeyer.org/7525.html
- RFC 7507 - TLS Fallback Signaling Cipher Suite Value (SCSV) for Preventing Protocol Downgrade Attacks, *Stéphane Bortzmeyer*, http://www.bortzmeyer.org/7507.html
- RFC 6696 The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA, *Stéphane Bortzmeyer*, http://www.bortzmeyer.org/6698.html
- RGS Annexe 4 : Authentification Serveur https://references.modernisation.gouv.fr/sites/default/files/RGS_fonction_de_securite_AuthentificationServeur_V2_3.pdf
- Certificats X.509 : quelle confiance leur accorder ? *Giles Carré*, JRES 2015, Montpellier.
- Premiers retours sur Let's Encrypt, *Pierre-Yves Bonnetain*, Resist décembre 2015, Toulouse. http://www.ossir.org/resist/supports/cr/2015/2015-12-15/2015-12-15-LetsEncrypt.pdf
- The Transport Layer Security (TLS) Protocol Version 1.3 draft-ietf-tls-tls13-latest https://tlswg.github.io/tls13-spec/