SIARS v2

Le chiffrement sur les services SSL/TLS

Hervé Ballans - Matthieu Herrb

9 octobre 2018

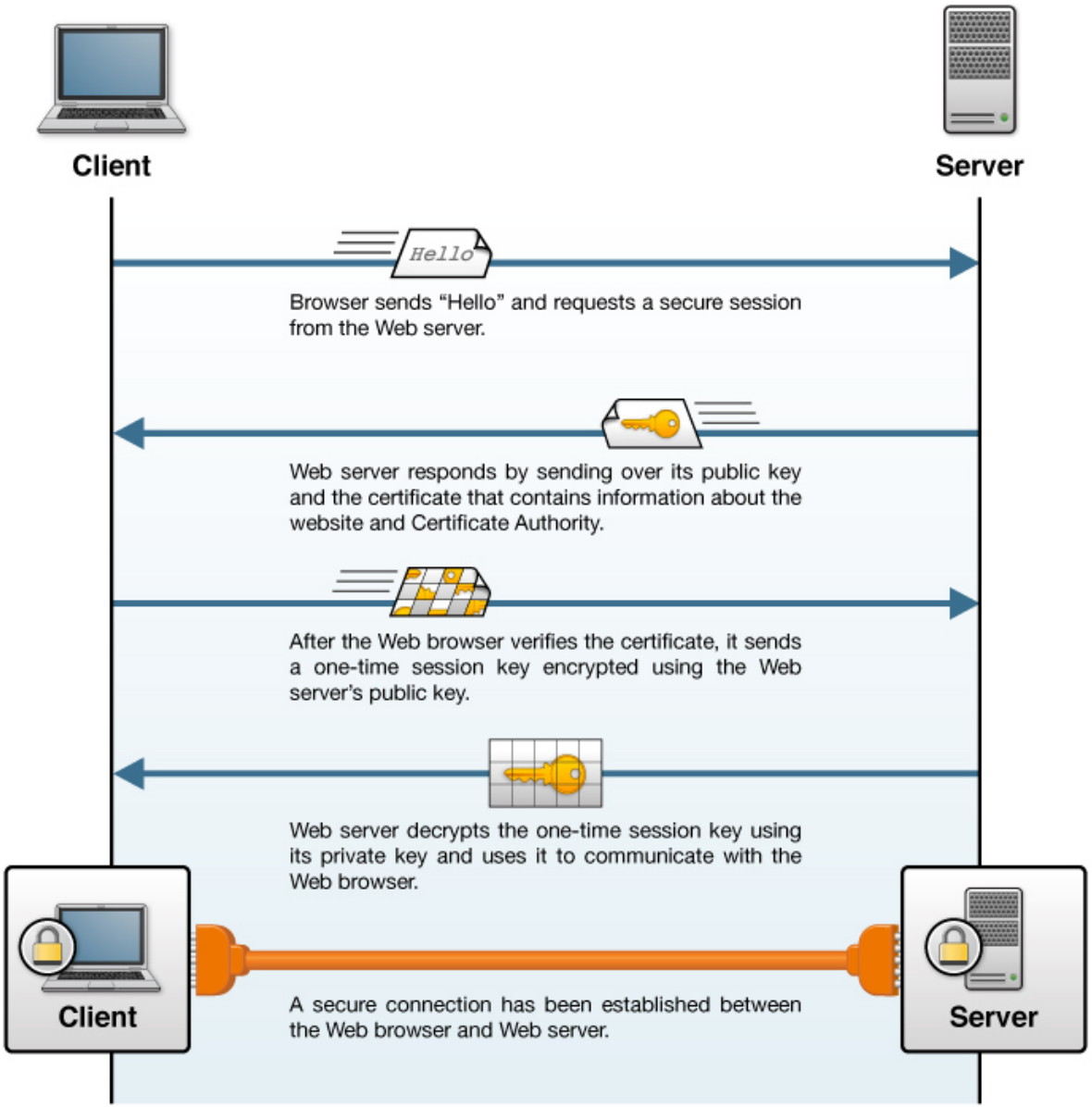
Tous les services réseau doivent être sécurisés, donc chiffrés

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 :

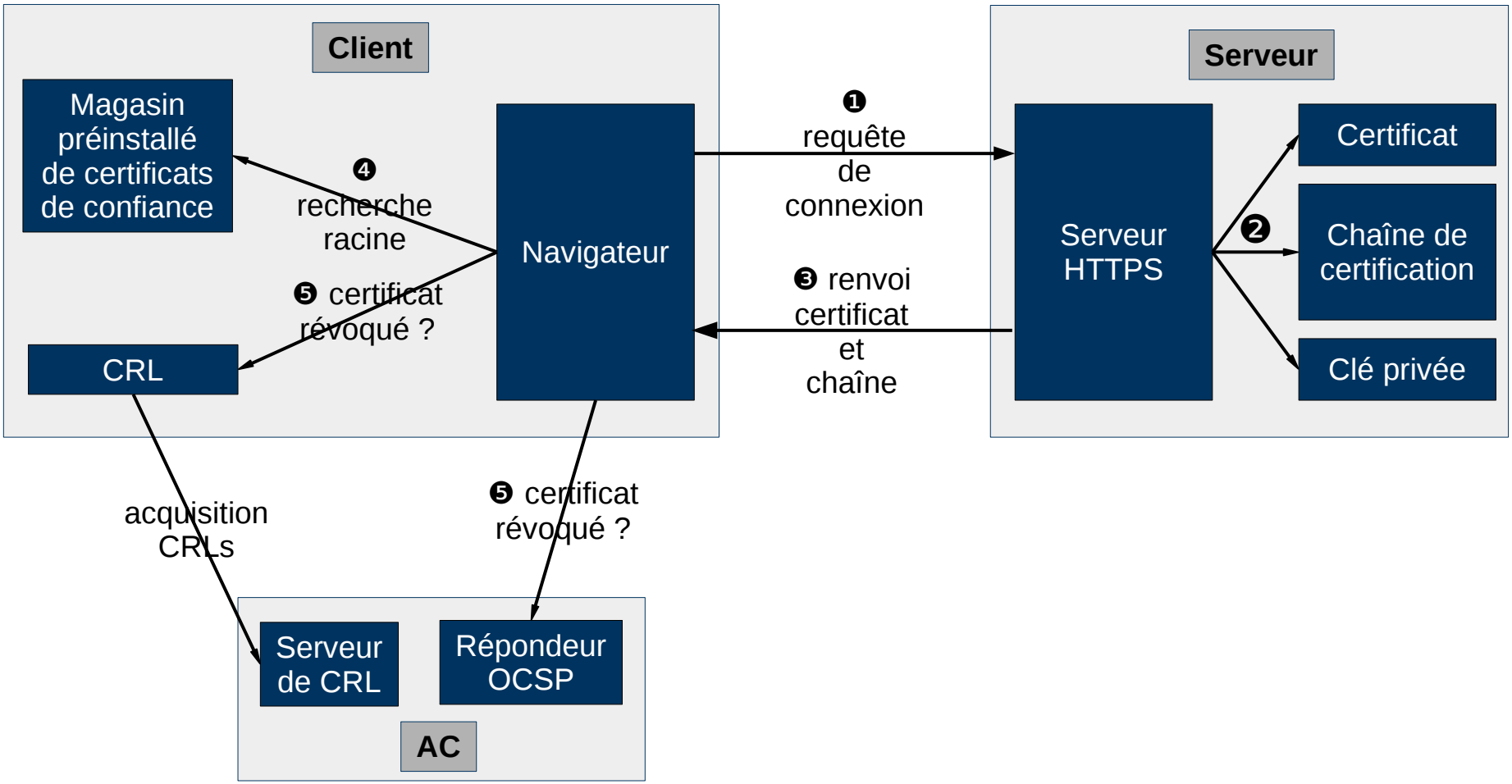
- **Bannir les certificats auto-signés**
- Superviser (nagios,...) l'expiration des certificats utilisés
- Fournir la chaîne de certification aux clients

Renforcer les protocoles de chiffrement :

- TLSv1.2..TLSv1.3 (supprimer SSLv2, SSLv3 et TLS 1.0/1.1)
- 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 :

- **HPKP** HTTP Protocol Key Pinning - interdit changement de certificat pour sites connus
- **HSTS** HTTP Strict Transport Security - interdit la connexion non chiffrée à un site connu.
- **CAA** DNS Certification Authority Authorization - enregistrement DNS annoncant l'AC utilisée par un domaine.
- **OCSP Stapling** - envoi des données de validation OSCP (signées par l'AC) au client directement lors de la négociation TLS.
- **DANE/TLSA** Publication du certificat du site via DNS(Sec)

DNSSEC est quasi indispensable pour la bonne sécurité d'un site DoH !

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

Terena Certificate Service (TCS - Accord Géant / Digicert)

- CNRS → Portail DSI
- Université(s) → Portail 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 (signature et échanges de clés):

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

Symétrique (chiffrement):

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

HMAC (intégrité):

MD5	à éviter
SHA	SHA-1 à éviter
SHA2	SHA-256, SHA-384

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

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. (Attention danger !)

Utiliser un reverse proxy devant les serveurs qui ne supportent pas les protocoles de HTTPS modernes.

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 all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
SSLCipherSuite EECDH+AESGCM:EEDH+AES
SSLHonorCipherOrder on

SSLOpenSSLConfCmd Curves secp384r1

Header always set Strict-Transport-Security \
                  "max-age=31536000; includeSubDomains"
```

Apache $\geq$ 2.4.8 accepte la chaîne de cerfication complète dans `SSLCertificateFile`

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 TLSv1.2;
        ssl_ciphers EECDH+AESGCM:EEDH+AES;
        ssl_dhparam /etc/pki/tls/private/dh2048.pem;

        add_header Strict-Transport-Security \
          "max-age=31536000; includeSubDomains; preload";
```

Concaténer les certificats intermédiaires au certificat du site.

En pratique : tomcat

Java 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+AESGCM:EEDH+AES
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_dh_parameters_length = 2048
ssl_protocols = !SSLv2 !SSLv3
ssl_cipher_list = ALL:!ADH:!LOW:!SSLv2:!3DES:!RC4:!MD5:!EXP:\
                  !aNULL:+HIGH
ssl_prefer_server_ciphers = yes
ssl_cert = </etc/ssl/certs/dovecot.crt
ssl_key = </etc/ssl/private/dovecot.key
```

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

Analyse d'un service : cryptcheck

[HTTPS] tls.laas.fr (Tue, 18 Sep 2018 12:42:11 +0000)

tls.laas.fr - 2001:660:6602:2::18 : 443

Scores A+	
Protocol	100 / 100
Key exchange	100 / 100
Cipher	90 / 100
Overall	96.0 / 100

Protocols	TLSv1_2
Keys	Certificats: RSA 4096 bits Diffie Hellman : ECC 384 bits
Good practices	PFS HSTS HSTS_LONG

	Key exchange		Authentication		Encryption				MAC		
Name	Type	Key size	Type	Key size	Type	Key size	Block size	Mode	Type	Size	PFS
TLSv1_2											
ECDHE-RSA-AES256-GCM-SHA384	ECDH	384	RSA	4096	AES	256	128	GCM	SHA384	384	PFS
ECDHE-RSA-AES128-GCM-SHA256	ECDH	384	RSA	4096	AES	128	128	GCM	SHA256	256	PFS

Proxys sortants

Proxy TLS sortant : filtrage d'URLs, anti-virus,...

- rompt la notion de chiffrement de bout en bout
- *compliqué* à mettre en œuvre avec TLS 1.3
- c'est sur le proxy que repose la sécurité

Problèmes :

- vérification des certificats du serveur
- négociation des algorithmes de chiffrement
- mesures de protection contre les attaques

implémentées dans une appliance qui n'est pas mise à jour aussi souvent que les navigateurs, d'où nombreuses vulnérabilités.

Futur

- Standard TLS 1.3 adopté en août 2018.
  - supprime algorithmes obsolètes (dont DSA)
  - ajout algo symétrique ChaCha20 + Poly1305
  - ajout courbe elliptique ed25519
  - supporté par OpenSSL 1.1.1, nginx 1.13.0, Firefox, Chrome
    pas encore avec Apache
- Démocratisation des HSM ?
- et après (cryptographie quantique...) ?

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 v2.0 Annexe 3 : Certificats électroniques de services applicatifs https://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_A3.pdf
  Annexe 4: Profils de certificats et Algorithmes Cryptographiques https://references.modernisation.gouv.fr/sites/default/files/RGS_v-2-0_A4.pdf
- Certificats X.509 : quelle confiance leur accorder ? *Giles Carré*, JRES 2015, Montpellier.
- Reparlons de Let's Encrypt, Linux FR, février 2016 https://linuxfr.org/news/reparlons-de-let-s-encrypt
- RFC 8446 : The Transport Layer Security (TLS) Protocol Version 1.3 http://www.bortzmeyer.org/8446.html
- The Sorry State of TLS Security in Enterprise Interception Appliances, *Louis Waked, Mohammad Mannan et Amr Youssef*, septembre 2018, https://arxiv.org/pdf/1809.08729.pdf