SSL - TLS

Histoire, fonctionnement
Sécurité et failles

1. SSL/TLS qu'est-ce que c'est ?

- Chiffrement
- Authentification faible du serveur
- Authentification forte du serveur (EV, facultative)
- Authentification du client (facultative)

1995

2013

2. SSL, Histoire

- 1994-1996 : Netscape développe SSL
- (1988)-1995 : la norme X.509 de certificats
- 1995 : création de Thawte & Verisign
- 1999-2008-2011 : l'IETF et TLS, les RFC
- 2005 : CA/Browser Forum & Extended Validation

3. TLS, Aujourd'hui

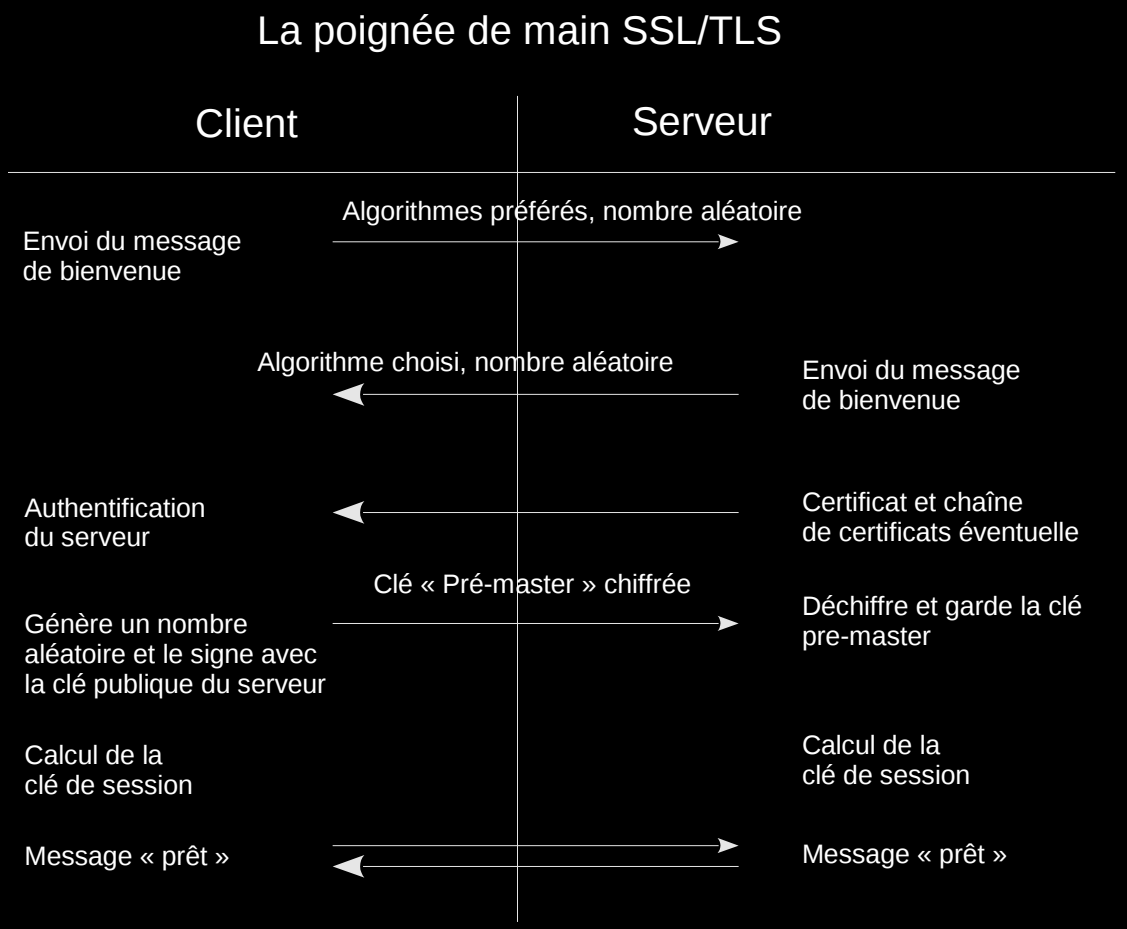
- On utilise TLS habituellement dans HTTPS
- Les autres protocoles ont été mis à jour pour savoir utiliser SSL/TLS aussi
- Exemple : pop(s) imap(s), smtp(s), puis pop+starttls, imap+starttls, smtp+starttls
- Autres protocoles : xmpp, vnc, ftp, irc …
- D'autres protocoles utilisent identification et chiffrement, mais sans SSL/TLS
- Exemple : DNS > DNSSEC, RSH > SSH

4. Exemple de session TLS

- http://brassens.heberge.info/form.php
- `sudo tcpdump -n 'host 91.194.60.2' -A`
- On voit les informations passer en clair

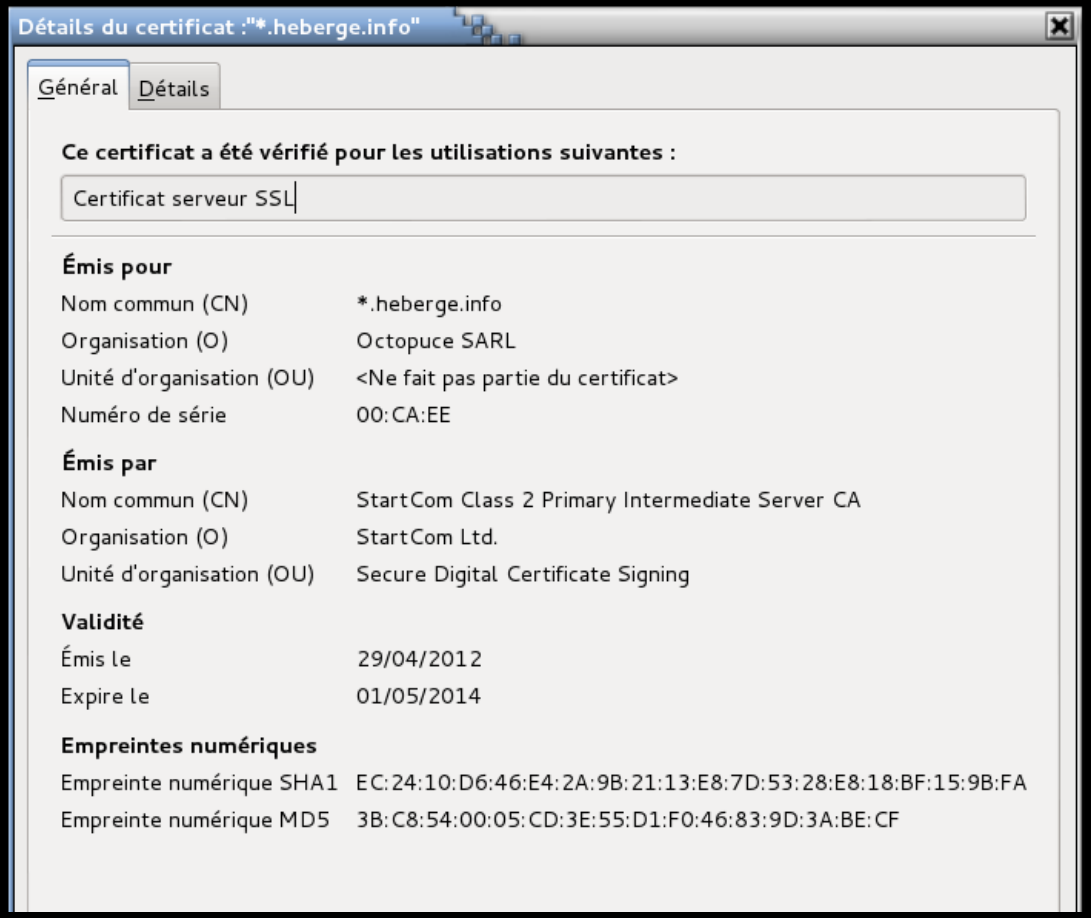
- https://brassens.heberge.info/form.php
- On voit l'échange TLS et les données chiffrées
- L'identité est assurée par un certificat SSL

4. Exemple de session TLS



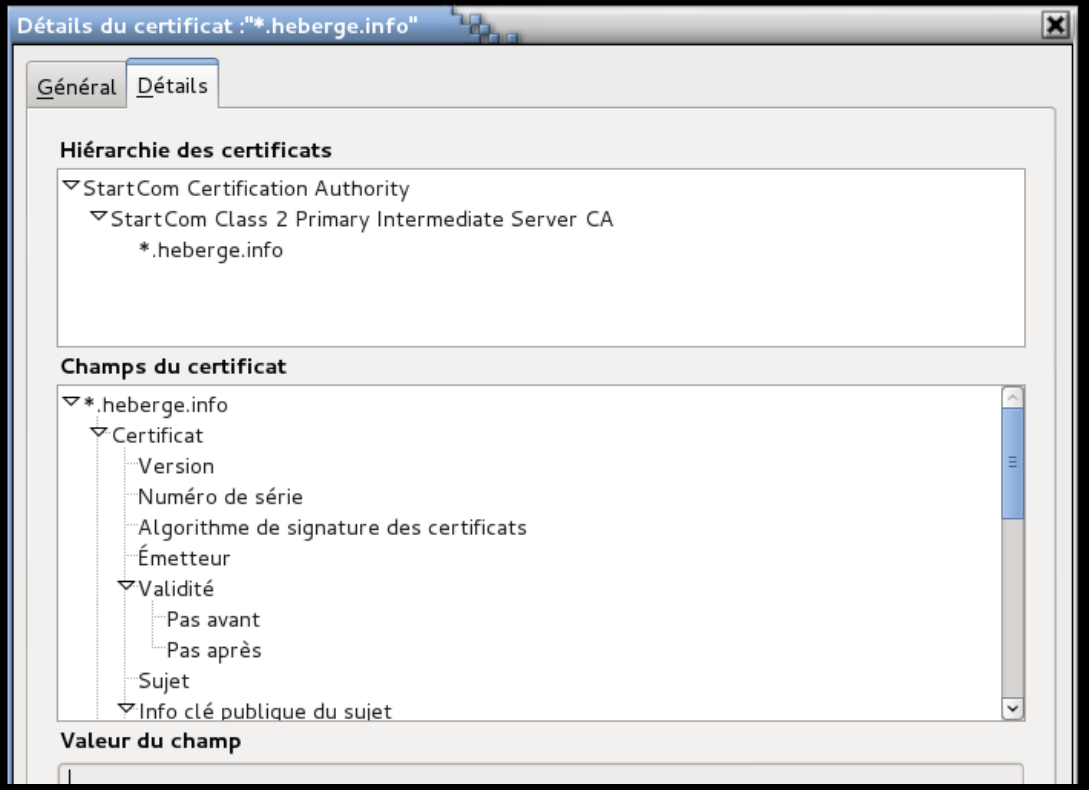
5. Le certificat SSL/TLS à la norme X.509

- Le certificat précise le nom DNS
- Il indique des dates de validité
- Et le signataire
- Le navigateur affiche l'empreinte
- Et l'usage autorisé (ici serveur SSL)



5. Le certificat SSL/TLS à la norme X.509

- Le certificat précise le nom DNS
- Il indique des dates de validité
- Et le signataire
- Le navigateur affiche l'empreinte
- Et l'usage autorisé (ici serveur SSL)



6. Les autorités de certifications (CA)

- Émettent un document décrivant leur processus
- Disposent d'une clé privée dont la clé publique est dans les logiciels clients
- Signent un ou plusieurs *Registration Authority* (RA)
- Qui signent à leur tour des requêtes de certificats spécifiques

- Les rôles possibles : SSL, SMIME, Software Signing, Timestamping, EV …
- Les niveaux de vérification : class1, class2, EV.

7. Les normes techniques, X.509, PEM, PKCS

- X.500 est une norme d'annuaire de l'ITU (branche telecom de l'ONU)
- Pas ou peu utilisée, LDAP en est la version légère
- X.509 est la sous-norme décrivant les certificats

- On utilise aussi les normes PKCS#1 à 15, mises au point par RSA
- Dont certaines ont été portées en RFC
- Qui décrivent les formats de stockage et de transport des objets cryptographiques

- Tout cela fut ajouté à Netscape Navigator via une librairie nommée NSS (*Network Security Services*). L'autre librairie libre connue étant OpenSSL

8. Les objets en jeu dans SSL/TLS

- Clé RSA
- Requête de certificat
- Certificat
- Certificat de révocation

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQI5U2RQAlQ52YCAggA
MBQGCCqGSIb3DQMHBAgYgMlI73qXVwSCBMjofM2xKsB6a37xwIcOQa+gahixAUNn
JJifXkkzOKx9zHGKZ2nXdSn2WaVZTKLJJiPcxZf8Cqjewjr/PUL/zFnOBBCwy2Bw
S8sOnR9PmmkrugM1uxePJNJ5GQKsjx6cmNcD3+ejh/qx2lRne5OLuKUroGQZJK4Z
aX4=
-----END ENCRYPTED PRIVATE KEY-----

-----BEGIN CERTIFICATE REQUEST-----
MIICVzCCAT8CAQAwEjEQMA4GA1UEAxMHdGVzdC5mcjCCASIwDQYJKoZIhvcNAQEB
BQADggEPADCCAQoCggEBALanq4mkWlWwE7GDVOQTu9JBf0QNSGyDFgswf5Dt6/EM
okqKGOu6eARkMzhESOAT61qp5tMiDyn/J003RpFTOLLz3fY+scBtarFsko1aMgo3
AF60Z4tHDkbXKREoo6Nr3tcShtN6soajFfswAv5SYWcGvTTE1bhIzS1Yc6vVONJd
luYK7fgfneu7tPxGCowkzogcxdPoe4mg5ipL
-----END CERTIFICATE REQUEST-----
```

```
-----BEGIN CERTIFICATE-----
MIIGajCCBVKgAwIBAgIDC9DOMA0GCSqGSIb3DQEBBQUAMIGMMQswCQYDVQQGEwJJ
TDEWMBQGA1UEChMNU3RhcnRDb20gTHRkLjErMCkGA1UECxMiU2VjdXJlIERpZ2l0
YWwgQ2VydGlmaWNhdGUgU2lnbmluZzE4MDYGA1UEAxMvU3RhcnRDb20gQ2xhc3Mg
+fiPdrsaTI2drdj+8peZb6h5VJDNRQNFbMGl/jSr7+8BStI59ddFwAKqkdQxuohI
qLwfQ8ggwFL98XIjAbNPo3wVm7JOOwUCpkCTqwjx891XWNbeeg6djNNYWDw3fl8D
PNAhEgWWO6fnKAdNY2I=
-----END CERTIFICATE-----

-----BEGIN X509 CRL-----
MIJy2jCCccIwDQYJKoZIhvcNAQEFBQAwgYwxCzAJBgNVBAYTAklMMRYwFAYDVQQK
Ew1TdGFydENvbSBMdGQuMSswKQYDVQQLEyJTZWN1cmUgRGlnaXRhbCBDZXJ0aWZp
Y2FOZSBTaWduaW5nMTgwNgYDVQQDEy9TdGFydENvbSBDbGFzcyAyIFByaW1hcnkg
GkXllImV2lLMiccn2PvaWQfRUvF7GEVA2bSnfpvomSBx6J3ViWb9blLnJjS3sV5g
1rZ3hbbmf1IBU7CrYxRJd4jzobc9mH/YPC9FnviwsuKAe5v+SyLXHLSRtsOzoMKQ
mJr5IWREgsvsVkIpAFsy3PeiBBC3JRpErgY/HTH
-----END X509 CRL-----
```

9. En pratique : génération d'un certificat X.509

- Utilisation de la libssl / Openssl
- Fichier de configuration minimal >
- Lignes de commandes v

```
HOME                    = .
RANDFILE                = $ENV::HOME/.rnd
oid_section             = new_oids
[ new_oids ]
[ req ]
default_bits            = 2048
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
string_mask             = nombstr

[ req_distinguished_name ]
commonName                      = Votre nom de domaine
commonName_max                  = 256
emailAddress                    = Votre Adresse Email
emailAddress_max                = 256
[ req_attributes ]
```

```
$ openssl req -new -config openssl.cnf
Generating a 2048 bit RSA private key
.....................................................................
.....................................................+++
.........................................+++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Votre nom de domaine []:example.com
Votre Adresse Email []:
-----BEGIN CERTIFICATE REQUEST-----
MIICWzCCAUMCAQAwFjEUMBIGA1UEAxMLZXhhbXBsZS5jb20wggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQChfAR9Ta9+99e0IM3tsONSXV1mGZfp+LMoTl48
IY4LayF/xGEuL+VhwNq6XlfZRbwwZwu4zP7aNI2tdxukWKCgs84yRPG+i27cJsCj
ltZH2Zf16NuKy0dXjsqgO0ccCkyXBfDbLHr1Y+6bmUOVy8qCMn78yZ7yPj8Star7
08HhX3s/6shQFlQD0sT+SY7Z9VSt1YnbyA19q3kA2Pr3oVumRTac9d/fvhl/aZlU
7hwWq1b/zIXOpCyIsJPllCVb1N8mp75vRbLIdehCKI+9cPIrLM1fwuKO4sVdv5mD
wzfkPpau52wxOhg41WoCsWxX6G1elHrxyBzzuJnXiw==
-----END CERTIFICATE REQUEST-----
```

9. En pratique : génération d'un certificat X.509

- Déprotection de la clé RSA
- Récupération du certificat
- Et du certificat chaîné (intermédiaire / RA)

```
$ openssl rsa -in privkey.pem -out example.com.key
Enter pass phrase for privkey.pem:
writing RSA key
$ cat example.com.key
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAoXwEfU2vfvfXtCDN7bDjUl1dZhmX6fizKE5ePBmb9AK1n5pN
rlNQYVJv5rPa8yQQfN/0oXSXw3aQbSTCcghT1as2M78uz62R5Or5kBK/hIfa1lF1
dibuHxIA+tMhDOK6e5Lh14UVfsM0VUAO7DqQ7R6/spx4Fp1VUD4syy2w6p3jrUpr
nZT0KNJ4de5kIZ6c7zF7uAknNPpOUwq+HjNH9p8KoVqlL8f0uH/nhQ==
-----END RSA PRIVATE KEY-----
```

- À la fin, on obtient les fichiers suivants :

.key (clé RSA) .csr (requête de certificat)

.crt (Certificat X.509) .chain (Certificat de l'intermédiaire)

.crt+chain et .all (les .crt .key .chain mis bout à bout, utile pour certains logiciels)

10. Exemples de tunnel SSL via openssl

Exemple de requête HTTP :

```
telnet 91.194.60.2 80
GET /form.php HTTP/1.0
Host : brassens.heberge.info
```

Exemple de requête HTTPS :

```
openssl s_client -host 91.194.60.2 -port 443
  -CAfile /etc/ssl/certs/ca-certificates.crt
...
GET /form.php HTTP/1.0
Host : brassens.heberge.info
```

On peut aussi montrer STARTTLS :
Exemple de dialogue SMTP :

```
telnet 91.194.60.2 25
```

Exemple de dialogue SMTP + STARTTLS :

```
openssl s_client -host 91.194.60.2 -port 25
  -CAfile /etc/ssl/certs/ca-certificates.crt -starttls smtp
```

11. De bons outils autour de SSL/TLS

• LibSSL / OpenSSL	openssl.org
• SSL Observatory	eff.org/observatory
• HTTPS Everywhere	eff.org/https-everywhere
• SSL Labs	ssllabs.com
• Tcpdump	tcpdump.org
• Dsniff (webmitm)	monkey.org/~dugsong/dsniff/

12. Les attaques possibles sur SSL/TLS

- MD5 considered harmful — win.tue.nl/hashclash/rogue-ca/
- MITM (volontaire ou frauduleux)
- CRIME — bit.ly/Q8061M (ssl labs)
- BEAST — bortzmeyer.org/beast-tls.html
- Rogue Certificate
- Rogue Autority
- Sécurité logicielle (affaiblir, trouer, divulguer les clés, choisir les ellipses ;))
- Out-of-band (écoutons E. Snowden & B. Schneier ;))

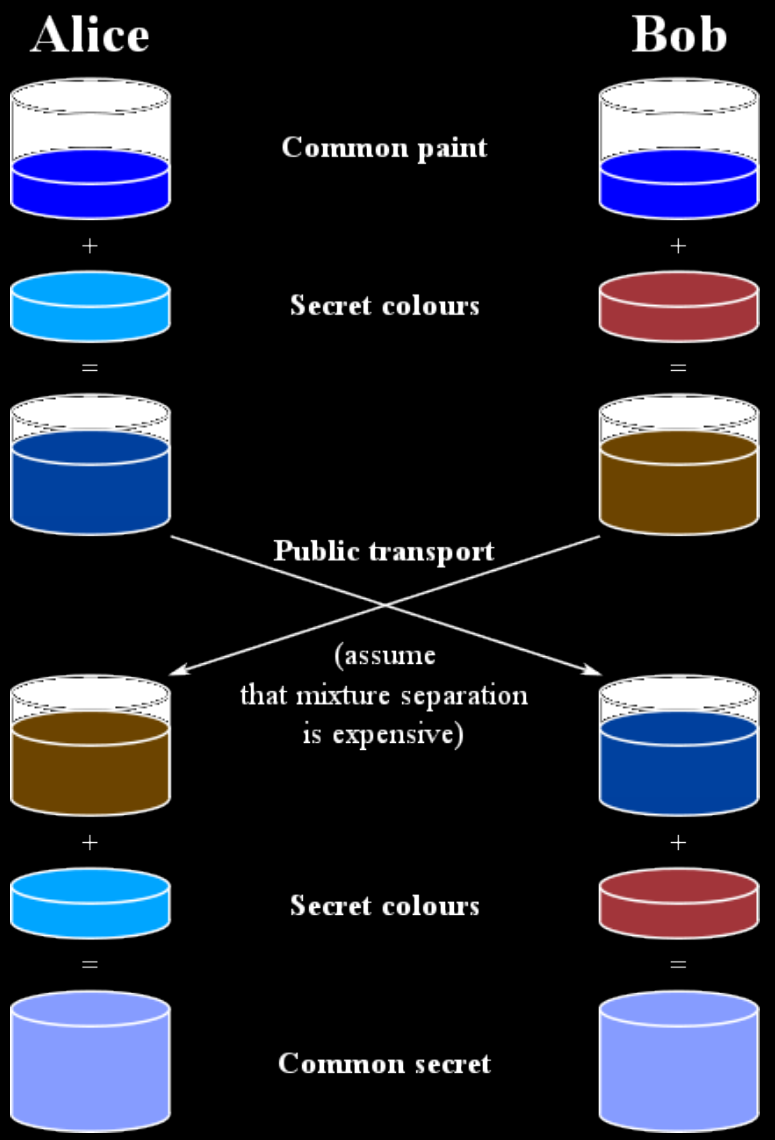
« *Encryption works. Properly implemented strong crypto systems are one of the few things that you can rely on.* »
« *Unfortunately, endpoint security is so terrifically weak that NSA can frequently find ways around it.* »
schneier.com/blog/archives/2013/09/how_to_remain_s.html

13. principes techniques évolués

- PFS (« « Perfect » » Forward Secrecy)
- Choix des algorithmes de chiffrement (dans OpenSSL)
- SNI (Server Name Indication)
- subjectAltName (certificats multi domaines)
- Wildcards
- Certificats EV
- Certificats Clients
- S/Mime
- OCSP
- TLSA/DANE & DNSSEC

13. principes techniques évolués

- PFS (« « Perfect » » Forward Secrecy)
  exemple Diffie Hellmann
  elliptic curves …



13. principes techniques évolués

- Choix des algorithmes de chiffrement (dans OpenSSL)

Keygen	Asymmetric	Symmetric	Chaining	Signature
RSA	RSA	AES128/256	CBC	SHA256
DHE	DSA	CAMELLIA	GCM	SHA384
ECDHE		3DES	EBC	MD5
...	...	...	...	...

ECDHE-RSA-AES256-GCM-SHA512 (256 bits)

DES-CBC-SHA (56 bits)

14. Installation et configuration de SSL/TLS pour vos logiciels préférés

- Apache (2.4)
- Nginx
- Lighttpd
- Proftpd
- Prosody
- Postfix
- Dovecot
- ...

14. Installation et configuration de SSL/TLS

Apache2.4 :

```
<VirtualHost _default_:443>
  ServerAdmin webmaster@octopuce.fr
  DocumentRoot /var/www

  SSLCertificateFile /etc/ssl/private/octopuce.fr.crt
  SSLCertificateChainFile  /etc/ssl/private/octopuce.fr.chain
  SSLCertificateKeyFile /etc/ssl/private/octopuce.fr.key

  SSLEngine on
  SSLProtocol +TLSv1.2 +TLSv1.1 +TLSv1
  SSLCompression off
  SSLHonorCipherOrder on
  # SSLCipherSuite ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-
AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-RC4-SHA:ECDHE-RSA-AES256-
SHA:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-RSA-AES128-GCM-
SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA:RC4-
SHA:AES256-GCM-SHA384:AES256-SHA256:CAMELLIA256-SHA:ECDHE-RSA-AES128-SHA:AES128-
GCM-SHA256:AES128-SHA256:AES128-SHA:CAMELLIA128-SHA
  SSLCipherSuite ALL:!aNULL:!eNULL:!LOW:!EXP:!RC4:!3DES:+HIGH:+MEDIUM
  Header set Strict-Transport-Security "max-age=2678400"
  ...
</VirtualHost>
```

14. Installation et configuration de SSL/TLS

Nginx :

```
server {
     listen 443;
     ssl on;
     ssl_certificate /etc/ssl/private/octopuce.fr.crt+chain;
     ssl_certificate_key /etc/ssl/private/octopuce.fr.key;

     ssl_session_timeout 5m;

     ssl_prefer_server_ciphers on;
     ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
#    ssl_ciphers ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-
AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-RC4-SHA:ECDHE-RSA-AES256-
SHA:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-RSA-AES128-GCM-
SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA:RC4-
SHA:AES256-GCM-SHA384:AES256-SHA256:CAMELLIA256-SHA:ECDHE-RSA-AES128-SHA:AES128-
GCM-SHA256:AES128-SHA256:AES128-SHA:CAMELLIA128-SHA;
     ssl_ciphers ALL:!aNULL:!eNULL:!LOW:!EXP:!RC4:!3DES:+HIGH:+MEDIUM;

     ssl_dhparam /etc/ssl/private/dh2048.pem;

     add_header Strict-Transport-Security max-age=2678400;
}

openssl dhparam -out /etc/ssl/private/dh2048.pem -outform PEM -2 2048
```

14. Installation et configuration de SSL/TLS

Lighttpd :

```
$SERVER["socket"] == "0.0.0.0:443" {
      ssl.engine  = "enable"
      ssl.pemfile = "/etc/ssl/private/octopuce.fr.key+crt"
      ssl.ca-file = "/etc/ssl/private/octopuce.fr.chain"

#     ssl.cipher-list = "ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-
SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-RC4-
SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-RSA-
AES128-GCM-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA:DHE-RSA-AES128-
SHA:RC4-SHA:AES256-GCM-SHA384:AES256-SHA256:CAMELLIA256-SHA:ECDHE-RSA-AES128-
SHA:AES128-GCM-SHA256:AES128-SHA256:AES128-SHA:CAMELLIA128-SHA"
      ssl.cipher-list = "ALL:!aNULL:!eNULL:!LOW:!EXP:!RC4:!3DES:+HIGH:+MEDIUM"
      ssl.honor-cipher-order = "enable"
setenv.add-response-header  = ( "Strict-Transport-Security" => "max-age=2678400")
}
```

14. Installation et configuration de SSL/TLS

Prosody :

```
ssl = {
      key = "/etc/ssl/private/octopuce.fr.key";
      certificate = "/etc/ssl/private/octopuce.fr.crt+chain";
}
-- Only allow encrypted streams? Encryption is already used when
-- available. These options will cause Prosody to deny connections that
-- are not encrypted. Note that some servers do not support s2s
-- encryption or have it disabled, including gmail.com and Google Apps
-- domains.
c2s_require_encryption = true
s2s_require_encryption = true
```

+ copier le bloc SSL = { } dans tout VirtualHost. **IMPORTANT**

14. Installation et configuration de SSL/TLS

Proftpd :

```
<IfModule mod_tls.c>
  TLSEngine                               on
  TLSLog                                  /var/log/proftpd/tls.log

  TLSProtocol                             TLSv1

  TLSCipherSuite            ALL:!aNULL:!eNULL:!LOW:!EXP:!RC4:!3DES:+HIGH:+MEDIUM
  TLSRSACertificateFile                   /etc/ssl/private/octopuce.fr.crt+chain
  TLSRSACertificateKeyFile                /etc/ssl/private/octopuce.fr.key

  TLSRenegotiate        required off
  TLSOptions            NoCertRequest EnableDiags NoSessionReuseRequired
  TLSVerifyClient       off
  TLSRequired           on # facultatif...
</IfModule>
```

14. Installation et configuration de SSL/TLS

Postfix :

```
# TLS parameters
smtpd_tls_cert_file = /etc/ssl/private/octopuce.fr.crt+chain
smtpd_tls_dcert_file = $smtpd_tls_cert_file
smtpd_tls_key_file = /etc/ssl/private/octopuce.fr.key
smtpd_tls_dkey_file = $smtpd_tls_key_file
smtpd_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
smtpd_tls_mandatory_ciphers = high
smtpd_tls_mandatory_exclude_ciphers = aNULL, MD5
smtpd_tls_dh1024_param_file = /etc/ssl/private/dh2048.pem

smtpd_use_tls = yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtpd_tls_protocols = !SSLv2!SSLv3
smtpd_tls_received_header = yes

smtp_tls_cert_file = $smtpd_tls_cert_file
smtp_tls_dcert_file = $smtpd_tls_dcert_file
smtp_tls_key_file = $smtpd_tls_key_file
smtp_tls_dkey_file = $smtpd_tls_dkey_file
smtp_tls_CAfile = $smtpd_tls_CAfile
smtp_tls_mandatory_ciphers = $smtpd_tls_mandatory_ciphers
smtp_tls_mandatory_exclude_ciphers = $smtpd_tls_mandatory_exclude_ciphers

smtp_use_tls = yes
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtp_tls_protocols = !SSLv2!SSLv3
smtp_tls_secure_cert_match = nexthop, dot-nexthop
```

Voir http://www.postfix.org/TLS_README.html pour plus de détails.

14. Installation et configuration de SSL/TLS

Dovecot :

```
ssl = yes

ssl_cert = </etc/ssl/private/octopuce.fr.crt+chain
ssl_key = </etc/ssl/private/octopuce.fr.key

ssl_protocols = !SSLv2:!SSLv3
ssl_cipher_list = ALL:!aNULL:!eNULL:!LOW:!EXP:!RC4:!3DES:+HIGH:+MEDIUM
```

des questions ?

mailto : benjamin@sonntag.fr
xmpp : benjamin@mailfr.com
pgp : 0x586073E6

et… surfez couvert ;)

Edward Snowden :

« Encryption works. Properly implemented strong crypto systems are one of the few things that you can rely on. »

« Unfortunately, endpoint security is so terrifically weak that NSA can frequently find ways around it. »