CAPITOUL
Commandes OpenSSL

UNIVERSITÉ TOULOUSE 1 CAPITOLE

Virginie.girou@ut-capitole.fr

Commandes OpenSSL

- Définition
- Versions
- Commandes Standard
  - Procédure de demande de certificat
  - Affichage de certificat
  - Chiffrement de clé privée
  - Conversion de format
  - Correspondance clé privée/certificat
  - Suivi d’une connexion ssl
- Vulnérabilités OpenSSL
- Liens utiles

Définition

- " OpenSSL est un **utilitaire cryptographique** qui implémente les protocoles réseau SSL et TLS ainsi que les standards cryptographiques liés dont ils ont besoin.
- C'est un **outil de ligne de commande** pour utiliser les différentes **fonctions cryptographiques** de la **librairie crypto** d'OpenSSL à partir du shell. Il peut être utilisé pour :
  - Création de paramètres des clefs RSA, DH et DSA
  - Création de certificats X.509, CSRs et CRLs
  - Calcul de signature de messages
  - Chiffrement et Déchiffrement
  - Tests SSL/TLS client et server
  - Gestion de mail S/MIME signé ou chiffrés "

Définition

- Syntaxe des commandes :

Openssl <commande standard/signature/chiffrement> <options/arguments>

- " Man openssl " liste les 3 types de commande et les arguments " Man <commande> " liste les options de chaque commande

Versions

- Etat des versions :
  - 11 juin 2015 openssl-0.9.8zg (SHA1) (fin support : 31/12/2015)
  - 11 juin 2015 openssl-1.0.0s (SHA1) (fin support : 31/12/2015)
  - 9 juillet 2015 openssl-1.0.1p (SHA256) (fin support : 31/12/2016)
  - 9 juillet 2015 openssl-1.0.2d (SHA256) (fin support : 31/12/2019)
- Signification de l'évolution des versions :
  - Modification mineure avec ajout de nouvelles fonctionnalités
  - Correction de bug et mise à jour sécurité (pas de nouvelle fonctionnalité)

Versions

- Obtenir des informations sur la version :

```
#openssl version -a

OpenSSL 1.0.1e-fips 11 Feb 2013
built on: Mon Jun 29 12:45:07 UTC 2015
platform: linux-x86_64
options:  bn(64,64) md2(int) rc4(8x,char) des(idx,cisc,16,int) idea(int)
blowfish(idx)
compiler: gcc -fPIC -DOPENSSL_PIC -DZLIB -DOPENSSL_THREADS -D_REENTRANT
-DDSO_DLFCN -DHAVE_DLFCN_H -DKRB5_MIT -m64 -DL_ENDIAN -DTERMIO -Wall -O2
-g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-
strong --param=ssp-buffer-size=4 -grecord-gcc-switches   -m64 -
mtune=generic -Wa,--noexecstack -DPURIFY -DOPENSSL_IA32_SSE2 -
DOPENSSL_BN_ASM_MONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -
DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DMD5_ASM -DAES_ASM -DVPAES_ASM -
DBSAES_ASM -DWHIRLPOOL_ASM -DGHASH_ASM
OPENSSLDIR: "/etc/pki/tls"
```

Commandes Standard : Procédure de demande de certificat

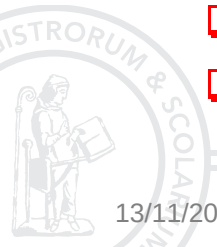
- Créer une demande de certificat (.csr ou .p10)

```
#openssl req -new -newkey rsa:2048 -nodes -out
demande_certif.csr -keyout cle_privee.key -subj
"/C=FR/ST=Midi Pyrénées/L=Toulouse/O=Université Toulouse
1 Capitole/CN=dsi.ut-capitole.fr"
```

Génération du biclé :

- **demande_certif.csr** : identification du demandeur (CN, O, L, S, C) clé publique, signature
- **cle_privee.key** : clé privée

- Comprendre les options :
  - req : traitement de type demande de CSR
  - new : nouvelle requête
  - newkey rsa:2048 : clé de type rsa de taille 2048 bits
  - nodes : la clé privée n'est pas chiffrée
  - subj : contient l'identification du demandeur



Commandes Standard : Procédure de demande de certificat

- Dépôt de la demande sur le site de l'Autorité de Certification (AC)
  - Copie du contenu du fichier généré demande_certif.csr
  - Choix de quelques paramètres (SHA2, durée de validité, …)

Changement de fournisseur de service RENATER depuis le 1er juillet 2015 :
- Digicert remplace Comodo
- SHA1 n'est pas conforme au RGS : utilliser du SHA2

- Génération par l'AC du certificat final
  - certif.crt : certificat serveur
  - certifCa.crt : certificat de l'Autorité de Certification

Ou

  - certif.p7b (format pkcs7) : contient les 2 fichiers

Commandes Standard : Affichage de certificat

- Affichage des certificats :

#openssl x509 -text -in certif.crt –noout (affichage complet)

Version, Numéro de série, algorithme de signature, émetteur, validité, identification du propriétaire, clé publique, CRLs, signature (sha1, sha2, …)

- Comprendre les options :
  - x509 = certif ; rsa = clé privée ; req = demande certif
  - -text : affichage en mode texte du contenu du certificat
  - -noout : évite l'affichage du certificat chiffré en plus du texte
  - -date, -subject, -issuer, -serial : filtres d'affichages sur le contenu du certificat
  - -nameopt + : -oneline, -multiline : option d'affichage de la sortie

Commandes Standard : Affichage de certificat

- Quelques exemples de personnalisation d'affichage :

```
#openssl x509 -in dsi_ut-capitole_fr.crt -dates
-subject -issuer -nameopt multiline –noout
```

```
notBefore=Oct  1 00:00:00 2015 GMT
notAfter=Oct  5 12:00:00 2017 GMT
subject=
    countryName               = FR
    localityName              = TOULOUSE
    organizationName          = Universite Toulouse 1 Capitole
    commonName                = dsi.ut-capitole.fr
issuer=
    countryName               = NL
    stateOrProvinceName       = Noord-Holland
    localityName              = Amsterdam
    organizationName          = TERENA
    commonName                = TERENA SSL High Assurance CA 3
```

```
#openssl x509 -in dsi_ut-capitole_fr.crt -dates
-subject -issuer -nameopt oneline -noout
```

```
notBefore=Oct  1 00:00:00 2015 GMT
notAfter=Oct  5 12:00:00 2017 GMT
subject= C = FR, L = TOULOUSE, O = Universite Toulouse 1 Capitole, CN = dsi.ut-capitole.fr
issuer= C = NL, ST = Noord-Holland, L = Amsterdam, O = TERENA, CN = TERENA SSL High Assurance CA 3
```

Commandes Standard : Chiffrement de clé privée

- Chiffrer sa clé privée avec une passphrase :

```
# openssl rsa -in cle_privee.pem -des3 -out
cle_privee_chiffree.pem
```

  - rsa : traitement de données rsa
  - -des3 : chiffrement

- Déchiffrer sa clé privée :

```
# openssl rsa -in cle_privee_chiffree.pem
 -out cle_privee.pem
```

La passphrase de chiffrement/déchiffrement est demandée à l'issue de chaque commande

Commandes Standard : Conversion de format

- Conversion de format pem vers p12 sécurisé

```
#openssl pkcs12 -export -inkey cle_privee.key -in
certif.pem –certfile cacert.pem -out resultat.p12
```

  - export : création d'un p12
  - inkey : clé privée
  - in : certificat
  - certfile : certificat de l'autorité de certification
  - out : fichier résultat

Une passphrase de chiffrement est demandée

- Inversement extraction d'un p12

```
#openssl pkcs12 –in resultat.p12 –out resultat.pem
```

Ou bien séparement :

```
#openssl pkcs12 -in resultat.p12 -nocerts -out
cle_privee.key
#openssl pkcs12 -in resultat.p12 –clcerts/-cacerts
-nokeys -out certif.pem/certifCa.pem
```

Commandes Standard : Conversion de format

- Conversion de format pkcs7 vers pem

```
#openssl pkcs7 -print_certs -in certif.p7b -out resultat.pem
```

  - -print_certs : affichage du certificat et des sections subject et issuer
  - Extraction manuelle par copier/coller des 2 certificats

- Conversion d'encodage PEM vers DER

```
#openssl x509 -in certif.crt -inform pem -outform der -out certif.der
```

- DER vers PEM

```
#openssl x509 -in certif.crt -inform der -outform pem -out certif.pem
```

  - -outform :spécifie le format de sortie
  - -inform : spécifie le format d'entrée

Commandes Standard : Correspondance clé privée/certificat

- Correspondance clé privée/certificat (crypto RSA) par comparaison des modulo

```
#openssl x509 -noout -modulus -in certif.crt
Modulus=D93C7B0F0B2AD2187EBDA67F9C7B316927B4EDBEAF062010B1C2044C6A4305A566975238EF33AD499F92840A4B
268A7D5089CDDFD7103591290A34E9BF3A7DB5D9476CAF2D761C406D5581292FE17A41D734157AAAC439A50942BDF85E70
86921824B8E668DE80A435BC68EE3FFA0B98B4881A1857C4F9BCCA5D3B3D9647B129C7E60084625E1DB2FCC4AC9113DC9A
BC10691245103617B97E7027111A8466D8A6D87A0ADA0408F639ED5F98A12AA7924AFBFDF63146D054880B18E214435E07
FD0DB09A7B4C22903B33E40FD34F844B001E457C1D352DABD440DA2FE080434439FCD2A1673567B5ADD7A7EDEB83B99754
D87275A3EF2B2AD3615CB9AAD41A4B
```

- La comparaison d'un condensé (hash) md5 ou sha1 est plus facile

```
#openssl x509 -noout -modulus -in certif.crt |openssl md5
(stdin)= c8b0f9bebe6e0a5643d3cacbf4454fce
#openssl rsa -noout -modulus -in cle_privee.key |openssl md5
(stdin)= c8b0f9bebe6e0a5643d3cacbf4454fce
```

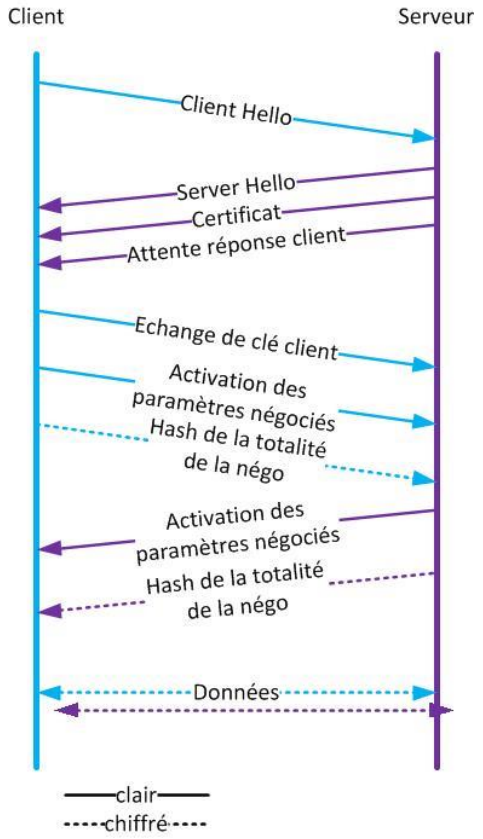
  - `Openssl md5` est une commande de type **signature**

Commandes Standard : Suivi d'une connexion ssl

- Test de connexion ssl

#openssl s_client -connect localhost:443 –state

```
CONNECTED(00000003)
SSL_connect:before/connect initialization
SSL_connect:SSLv2/v3 write client hello A
SSL_connect:SSLv3 read server hello A
SSL_connect:SSLv3 read server certificate A
SSL_connect:SSLv3 read server key exchange A
SSL_connect:SSLv3 read server done A
SSL_connect:SSLv3 write client key exchange A
SSL_connect:SSLv3 write change cipher spec A
SSL_connect:SSLv3 write finished A
SSL_connect:SSLv3 flush data
SSL_connect:SSLv3 read server session ticket A
SSL_connect:SSLv3 read finished A
```



Vulnérabilités OpenSSL

- Vulnérabilité Poodle :
  - Sslv3 et v2 doivent être désactivés sur les serveurs
  - Test de présence de sslv3 :

#openssl s_client -connect mon_site:443 -ssl3

```
SSL routines:SSL3_READ_BYTES:sslv3 alert handshake failure
```

- Failles Freak et Logjam :
  - Bannir les cipher EXPORT_RSA et DHE_EXPORT
  - Test de présence des fonctions EXPORT :

#openssl s_client -connect mon_site:443 -cipher EXPORT

```
SSL routines:SSL23_GET_ SERVER_HELLO:sslv3 alert handshake failure
```

  - Utiliser au moins un groupe DH 1024-bits et le protocole ECDH sur les serveurs et navigateurs
  - Génération d'un nouveau groupe DH :

#openssl dhparam -out dhparams.pem 2048

Vérification de sécurité

- Vulnérabilité du SHA1 :
  - Vérification de la version de la signature du certificat :

#openssl x509 -text -in certif.crt -noout

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            03:eb:14:f7:ec:53:80:22:15:bf:9a:fc:f1:81:cf:10
    Signature Algorithm: sha256WithRSAEncryption
```

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            2e:7e:a9:28:10:95:eb:dd:f5:bd:83:29:38:e9:9a:6d
    Signature Algorithm: sha1WithRSAEncryption
```

Liens utiles

- Site officiel OpenSSL : https://www.openssl.org/
- Site Digicert :
  - génération de requête : https://www.digicert.com/easy-csr/openssl.htm
  - page d'accueil : https://services.renater.fr/tcs/index
- Test de vulnérabilité :
  - Poodle https://www.poodletest.com/
  - Sha1
    - https://www.digicert.com/sha1-sunset/
    - https://tls.imirhil.fr/
- Procédure contre la faille logjam :
  - https://weakdh.org/sysadmin.html

Questions