Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
PREMIER MINISTRE

S . G . D . S . N
Agence nationale de la sécurité des systèmes d'information
CERT-FR

Paris, le 01 février 2016
N° CERTFR-2016-ACT-005

Affaire suivie par :
CERT-FR

BULLETIN D'ACTUALITÉ

Objet : Bulletin d'actualité CERTFR-2016-ACT-005

1 - Bonnes pratiques pour les certificats

Les usages de la cryptographie à clés asymétriques sont multiples, mais le plus visible est l'utilisation de TLS pour la communication sécurisée entre ordinateurs (HTTPS sur Internet, mais également LDAPs vers un annuaire ou SSH pour de l'exécution de commandes à distance). Il existe également d'autres usages comme la signature de binaires, la messagerie sécurisée (signature ou chiffrement de messages), le chiffrement de fichier (par exemple "Encrypting File System" de Microsoft) ou l'ouverture de session par carte à puce. Ce premier article se concentre sur le principe de la cryptographie à clé asymétrique et son usage pour la communication sécurisée.

Le contenu d'un certificat

Un certificat électronique (aussi appelé certificat numérique) est une structure signée contenant une clé publique, des informations d'identités (nom, adresse électronique, etc), une validité temporelle. D'autres informations optionnelles peuvent être ajoutées. La définition complète des champs présents dans un certificat est disponible dans la RFC 5280. Ce certificat est associé à un utilisateur ou à un ordinateur qui est le seul à détenir la clé privée associée. Il est possible de référencer différents noms dans un certificat. Pour cela, on peut utiliser un caractère de remplacement dans le champ "Objet" (*.example.org par exemple) ou spécifier plusieurs noms dans le champ "Autre nom de l'objet" (www.example.org et web.example.intra).

Validité d'un certificat

La validité d'un certificat est définie par les conditions suivantes :

- Le certificat n'a pas été modifié. Pour cela, le client calcule le hash du certificat puis déchiffre celui stocké dans le certificat lui-même avec la clé publique de l'autorité émettrice et enfin compare les deux valeurs qui doivent être égales.
- Le certificat est bien dans sa période de validité (entre la date d'émission et la date d'expiration).
- Le certificat n'a pas été révoqué par l'autorité émettrice (il n'est pas référencé dans la liste de révocation en cours).
- Il existe une chaîne de certificats qui remonte jusqu'à une autorité racine de confiance et tous les certificats de la chaîne respectent les 3 critères ci-dessus.
- Il contient une indication d'usage conforme à celui qui en est fait (par exemple "authentification du serveur" pour une connexion TLS)

Pour vérifier manuellement la validité d'un certificat sous Microsoft Windows, l'outil certutil.exe peut être utilisé sur le fichier du certificat. Exemple :

```
C:\textbackslash >certutil.exe -verify example.org.cer
Emetteur:
CN=DigiCert SHA2 High Assurance Server CA
OU=www.digicert.com
```

```
O=DigiCert Inc
C=US
Hachage du nom (sha1) : cf26f518fac97e8f8cb342e01c2f6a109e8e5f0a
Hachage du nom (md5) : f24c7558789938cd4b21602c2286ddb3
Objet:
CN=www.example.org
OU=Technology
O=Internet Corporation for Assigned Names and Numbers
L=Los Angeles
S=California
C=US
Hachage du nom (sha1) : 13388c8838c5b3c606c85c1d45aa37ea8a419f26
Hachage du nom (md5) : 388adb6fe8a7ad427fc7980f0a3cf24a
Numero de serie du certificat : 0e64c5fbc236ade14b172aeb41c78cb0
[...]
CertContext[0][0]: dwInfoStatus=102 dwErrorStatus=0
Issuer: CN=DigiCert SHA2 High Assurance Server CA, OU=www.digicert.com, O=DigiCert Inc,
NotBefore: 03/11/2015 01:00
NotAfter: 28/11/2018 13:00
Subject: CN=www.example.org, OU=Technology, O=Internet Corporation for Assigned Names an
Serial: 0e64c5fbc236ade14b172aeb41c78cb0
SubjectAltName: Nom DNS=www.example.org, Nom DNS=example.com, Nom DNS=example.edu, Nom D
[...]
CertContext[0][1]: dwInfoStatus=102 dwErrorStatus=0
Issuer: CN=DigiCert High Assurance EV Root CA, OU=www.digicert.com, O=DigiCert Inc, C=US
NotBefore: 22/10/2013 13:00
NotAfter: 22/10/2028 13:00
Subject: CN=DigiCert SHA2 High Assurance Server CA, OU=www.digicert.com, O=DigiCert Inc,
Serial: 04e1e7a4dc5cf2f36dc02b42b85d159f
[...]
CertContext[0][2]: dwInfoStatus=10a dwErrorStatus=0
Issuer: CN=DigiCert High Assurance EV Root CA, OU=www.digicert.com, O=DigiCert Inc, C=US
NotBefore: 10/11/2006 01:00
NotAfter: 10/11/2031 01:00
Subject: CN=DigiCert High Assurance EV Root CA, OU=www.digicert.com, O=DigiCert Inc, C=U
Serial: 02ac5c266a0b409b8f0b79f2ae462577
[...]
Le certificat est un certificat d'entite de fin
Verification de revocation du certificat feuille reussie
CertUtil: -verify La commande s'est terminee correctement.
```

Clé publique, clé privée

La robustesse d'une protection par cryptographie asymétrique repose principalement sur la confidentialité de la clé privée associée. Il convient donc de sécuriser fortement cette information. Si elle venait à être compromise, une personne malveillante pourrait l'utiliser de façon illégitime. La cryptographie asymétrique est utilisée pour :

- garantir l'intégrité de données (par exemple, signature de code) ou de flux (signature SMB) ;
- garantir la confidentialité de données (chiffrement de fichiers) ou de flux (connexion HTTPS) ;
- garantir l'auteur de données (signature de messages électroniques).

La clé publique associée est stockée dans le certificat.

Les algorithmes de condensat et l'obsolescence de SHA1

La signature de données repose sur l'aspect intégrité des données. Elle permet à un utilisateur de s'assurer que les données n'ont pas été modifiées depuis leur émission par son auteur.

En pratique, la totalité des données n'est pas signée : l'auteur calcule un condensat et le signe avec sa clé privée. Ce procédé limite la quantité de données à traiter et donc accélère l'opération.

Lors de la vérification de la signature, le client calcule le condensat selon la même méthode et déchiffre le condensat présent dans le fichier avec la clé publique de l'éditeur. Si les deux valeurs coïncident, la signature est valide. Bien sûr, le certificat utilisé pour signer doit être lui-même valide.

Il convient donc de s'assurer que la méthode de calcul du condensat est parfaitement fiable. A l'heure actuelle, les algorithmes suivants sont utilisés :

- MD5 (Message-Digest Algorithm 5, 128 bits) qui ne doit plus être utilisé en raison de failles permettant de créer des condensats identiques pour des données différentes.
- SHA-1 (Secure Hash Algorithm 1), dont le retrait est en cours (voir ci-dessous)
- SHA-2 (SHA-256, SHA-384 ou SHA-512 bits au choix).

SHA-1 est une fonction publiée par le NIST, dont la première version date de 1995. Plusieurs publications depuis 2005 font craindre que SHA-1 soit cassé. Des chercheurs seraient parvenus à créer deux condensats identiques à partir de données différentes. Ainsi, le NIST déconseille l'utilisation de SHA-1 depuis mars 2006 et une décision du CA/B Forum du 16 octobre 2014 préconise le retrait de SHA-1 d'ici fin 2016 (voir le bulletin d'actualité CERTFR-2015-ACT-018 du 4 mai 2015).

En conséquence, Microsoft, Mozilla, Google et d'autres grands éditeurs ont décidé de bloquer les certificats SHA-1 utilisés par TLS à partir du 1er janvier 2017.

Le Référentiel Général de Sécurité précise les algorithmes cryptographiques recommandés par l'ANSSI. Pour l'algorithme de condensat pour les certificats utilisés (ceux des autorités de certification et ceux émis pour les entités feuilles), il convient d'utiliser au minimum SHA256.

SSL et TLS

La sécurisation de l'accès à un site Internet par l'utilisation de certificat et du protocole HTTPS (SSL - Secure Sockets Layers - ou son successeur TLS - Transport Layer Security) a deux objectifs :

- Garantir l'identité du serveur auquel l'utilisateur se connecte.
- Assurer la confidentialité et l'intégrité des échanges entre le client et le serveur.

Validité Lors de l'utilisation de HTTPS le navigateur Internet vérifie la validité du certificat présenté par le serveur. Il s'assure ensuite que le nom du site interrogé est bien présent dans le champ "Objet" ou "Autre nom de l'objet" du certificat que le serveur a envoyé. Il est donc nécessaire que le certificat soit valide, non seulement sur le serveur Internet, mais également sur les ordinateurs clients.

Pour des serveurs Internet publics (exposés à l'extérieur du réseau d'entreprise), il faut s'assurer que le certificat du serveur pourra bien être validé par les clients externes. Cela nécessite d'une part que le certificat de l'autorité racine de certification soit considéré comme digne de confiance et d'autre part que la chaîne de certificats puisse bien être construite. Enfin, il faut que les listes de révocation des différentes autorités présentes dans la chaîne puissent bien être atteintes. En raison de ces contraintes, l'utilisation de certificats publics (émis par une autorité reconnue par le navigateur) est généralement préférée.

Chiffrement symétrique des données En raison de la charge processeur importante nécessaire pour la signature de données avec la cryptographie asymétrique (50 à 100 fois plus que pour la cryptographie symétrique), celle-ci ne peut pas être utilisée pour du chiffrement de grand volume d'information.

C'est un chiffrement symétrique qui est utilisé (la même clé est utilisée pour chiffrer et déchiffrer les données) et la transmission de cette clé est protégée par le chiffrement asymétrique. En pratique le client génère une clé symétrique et la transmet au site Internet en la chiffrant avec la clé publique du serveur. Cette clé symétrique est aussi appelée clé de session car sa durée de vie est limitée à la session TLS courante, ce qui correspond généralement au téléchargement d'une seule page.

Suites cryptographiques Une suite cryptographique décrit les algorithmes

- D'authentification (du serveur)
- D'échange de clé
- De chiffrement des données
- De protection en intégrité des données

Traditionnellement, authentification et échange de clés sont regroupés d'une part et les chiffrement et intégrité des données d'autre part. Exemples : TLS_RSA_WITH_RC4_128_MD5 indique :

- RSA pour le chiffrement de l'échange de clé et l'authentification implicite

– RC4_128 pour le chiffrement des données
– HMAC-MD5 pour l'intégrité des données

TLS_DHE_RSA_WITH_AES_128_CBC_SHA indique :

– Diffe-Hellman pour l'échange de clé
– RSA pour la signature de cet échange
– AES_128_CBC pour le chiffrement des données
– HMAC-SHA1 pour l'intégrité des données

Le client propose une liste de suites dans le message ClientHello d'initialisation de la connexion TLS selon ses capacités. Le serveur choisit parmi cette liste le protocole le plus sécurisé supporté par les deux parties. Il convient de s'assurer d'utiliser des suites dont la sécurité présente une robustesse acceptable. On trouvera une liste des suites dans le document "SSL/TLS: état des lieux et recommandations" en référence.

La vérification des suites TLS supportées par un site Internet peut être faite avec l'outil sslscan. Par exemple :

```
C:\>SSLScan.exe --no-failed --http www.example.org
                      Version 1.8.2-win
           http://www.titania.co.uk
           Copyright Ian Ventura-Whiting 2009
       Compiled against OpenSSL 0.9.8m 25 Feb 2010

 Testing SSL server www.example.org on port 443

Supported Server Cipher(s):
  Accepted  TLSv1  256 bits  AES256-SHA
  Accepted  TLSv1  128 bits  AES128-SHA
  Accepted  TLSv1  168 bits  DES-CBC3-SHA

Prefered Server Cipher(s):
  TLSv1  256 bits  AES256-SHA
```

Version de SSL/TLS SSL a connu plusieurs versions. La version 1, spécifiée en 1994, n'a jamais été mise en oeuvre. La version 2, dont l'implémentation commence en 1995 a été bannie en 2001. La version 3, la dernière, date de 1996 et présente des failles connues. Son successeur, TLS en est à la version 1.2. La version 1.0 date de 1999. La version 1.1 a été publiée en 2006, la version 1.2, en 2008. La version 1.3 est actuellement à l'état de brouillon et n'est donc pas implémentée.

Pour s'assurer de la sécurité maximale, SSL v2 et v3 ne doivent pas être utilisés. Et TLS 1.1 doit être préféré. À noter que des voix s'élèvent même à l'IETF pour marquer également TLS 1.0 (voir TLS 1.1) comme obsolète.

Épinglage de certificat Cette extension de HTTP intitulée "certificate pinning" en anglais est une fonctionnalité de sécurité qui impose que la chaîne de validité d'un certificat feuille contienne le certificat d'une autorité donnée. Par exemple, que la chaîne de validité du certificat "www.google.fr" contienne bien le certificat pour "Google Internet Authority G2".

Une autre solution consiste à associer une clé publique cryptographique avec un certain serveur Internet. On parle alors d'épinglage de clé publique ou "public key pinning". Dans ce cas le navigateur mémorise la clé publique associée au certificat du site Internet lors de la première connexion. Ainsi si une autre clé est présentée lors d'une connexion ultérieure celle-ci sera refusée.

Ces mécanismes permettent d'éviter des attaques exploitant des certificats contrefaits. Bien que l'épinglage de certificat ne soit pas une solution idéale, notamment à cause de la complexité induite par la création ou mise à jour des règles, il reste un moyen simple et efficace pour remédier à des faiblesses de l'infrastructure de gestions de certificats basée sur les autorités de certification.

Références

– sslscan :
http://sourceforge.net/projects/sslscan/
– Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
https://tools.ietf.org/html/rfc5280

- Politique d'obsolescence de SHA-1, Bulletin d'actualité CERTFR-2015-ACT-018
  http://www.cert.ssi.gouv.fr/site/CERTFR-2015-ACT-018/CERTFR-2015-ACT-018.html
- Bulletin d'actualité CERTFR-2015-ACT-042
  http://www.cert.ssi.gouv.fr/site/CERTFR-2015-ACT-042/CERTFR-2015-ACT-042.html
- NIST's Policy on Hash Functions
  http://csrc.nist.gov/groups/ST/hash/policy.html
- Windows Enforcement of Authenticode Code Signing and Timestamping
  http://social.technet.microsoft.com/wiki/contents/articles/32288.windows-enforcement-of-authenticode-code-signing-and-timestamping.aspx
- Phasing Out Certificates with SHA-1 based Signature Algorithms
  https://blog.mozilla.org/security/2014/09/23/phasing-out-certificates-with-sha-1-based-signature-algorithms/
- Continuing to Phase Out SHA-1 Certificates
  https://blog.mozilla.org/security/2015/10/20/continuing-to-phase-out-sha-1-certificates/
- Gradually Sunsetting SHA-1
  http://blog.chromium.org/2014/09/gradually-sunsetting-sha-1.html
- SSL/TLS: état des lieux et recommandations
  http://www.ssi.gouv.fr/uploads/IMG/pdf/SSL_TLS_etat_des_lieux_et_recommandations.pdf
- SSL/TLS, 3 ans plus tard
  http://www.ssi.gouv.fr/uploads/2015/06/SSTIC2015-Article-ssltls_soa_reloaded-levillain_cObDbqp.pdf
- Bulletin d'actualité CERTFR-2015-ACT-004
  http://cert.ssi.gouv.fr/site/CERTFR-2015-ACT-004/index.html
- Référentiel Général de Sécurité
  http://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_Corps_du_texte.pdf
- Annexe A1 - Règles relatives à la mise en oeuvre des fonctions de sécurité basées sur l'emploi de certificats électroniques
  http://www.ssi.gouv.fr/uploads/20144/11/RGS_v-2-0_A1.pdf
- Annexe A2 - Politique de Certification Type "certificats électronique de personne"
  http://www.ssi.gouv.fr/uploads/20144/11/RGS_v-2-0_A2.pdf
- Annexe A3 - Politique de Certification Type "certificats électronique de services applicatifs"
  http://www.ssi.gouv.fr/uploads/20144/11/RGS_v-2-0_A3.pdf

2 - Sortie d'EMET 5.5

Le 29 janvier 2016, Microsoft a mis à disposition l'outil Enhanced Mitigation Experience Toolkit (EMET) en version 5.5 finale, après une phase de test initiée en octobre dernier. Cet outil, compatible avec les versions supportées de Windows, permet de se protéger contre certaines techniques communément utilisées pour l'exploitation de vulnérabilités. Son utilisation s'inscrit dans une logique de défense en profondeur, en complément d'autres protections telles que le déploiement d'antivirus ou l'installation de correctifs de sécurité.

Parmi les nouveautés apportées par la version 5.5, il est possible de citer :

- la compatibilité avec Windows 10 ;
- l'amélioration de la configuration de certaines protections par GPO ;
- l'amélioration de la gestion des protections via la base de registre, afin de pouvoir utiliser les outils existants pour configurer EMET par GPO ;
- une amélioration des performances de la protection EAF/EAF+ ;
- le support de la protection Untrusted font sous Windows 10.

Le CERT-FR recommande donc de déployer la dernière version d'EMET sur les systèmes d'informations, après une phase de qualification pour identifier et corriger les éventuels problèmes de compatibilité. De plus, le CERT-FR conseille de suivre les recommandations du guide [3] de l'ANSSI au sujet du déploiement et la configuration centralisée d'EMET.

Documentation

- 1 Description d'EMET :
  https://technet.microsoft.com/en-us/security/jj653751

– 2 Téléchargement d'EMET 5.5 :
https://www.microsoft.com/en-us/download/details.aspx?id=50766
– 3 Guide Déploiement et configuration centralisés d'EMET de l'ANSSI :
http://www.ssi.gouv.fr/emet

3 - Rappel des avis émis

Dans la période du 25 au 31 janvier 2016, le CERT-FR a émis les publications suivantes :

– CERTFR-2016-AVI-035 : Vulnérabilité dans Cisco APIC-EM
– CERTFR-2016-AVI-036 : Vulnérabilité dans Cisco Unified Contact Center Express
– CERTFR-2016-AVI-037 : Multiples vulnérabilités dans Ruby On Rails
– CERTFR-2016-AVI-038 : Multiples vulnérabilités dans Mozilla Firefox
– CERTFR-2016-AVI-039 : Multiples vulnérabilités dans Nginx
– CERTFR-2016-AVI-040 : Multiples vulnérabilités dans les produits Cisco
– CERTFR-2016-AVI-041 : Multiples vulnérabilités dans OpenSSL
– CERTFR-2016-AVI-042 : Multiples vulnérabilités dans les produits Cisco
– CERTFR-2016-AVI-043 : Vulnérabilité dans Huawei E5186

Gestion détaillée du document

01 février 2016 version initiale.