OpenSSL - ts

Outil de Time Stamping Authority (TSA) client/serveur

ts est un outil de base TSA client/serveur comme spécifié dans la RFC3161. Un TSA peut être une partie d'une PKI et son rôle est de fournir une preuve à long terme de l'existence d'un horodatage avant une date particulière. Brève description du protocole :

1. Le client TSA calcule un hash 'one-way' pour un fichier de données et l'envoie au TSA.
2. Le TSA attache la date et l'heure courante au hash reçu, signe le tout et renvoie ce token au client.
3. Le client TSA reçoit le token et vérifie la signature. Il vérifie également si le token contient le même hash qu'il avait envoyé au TSA.

Génération de requête et Time Stamp

-query Permet de générer une requête avec les options suivantes :

-rand file :file... Fichier contenant les données pour le générateur pseudo-aléatoire, plusieurs fichiers peuvent être utilisés

-config configfile Fichier de configuration à utiliser. Seul la section OID est utilisée (remplace la variable OPENSSL_CONF)

-data file_to_hash Fichier de donnée pour lequel la requête doit être crée.

-digest digest_bytes Permet de spécifier l'empreinte du message sans le fichier de donnée, au format hexadécimal

-md2|-md4|-md5|-sha|-sha1|-mdc2|-ripemd160|... Message digest à appliquer au fichier de données

-policy object_id Stratégie que le client s'attend à ce que le TSA utilise pour créer le time stamp. Sous forme OID numérique ou nom IOD.

-no_nonce Aucun nonce n'est utilisé, sinon un nombre pseudo-aléatoire 64 bits est inclus dans la requête. nonce est recommandé.

-cert Le TSA doit inclure son certificat de signature dans la réponse

-in request.tsq Spécifie une précédente requête time stamp en DER à afficher en sortie

-out request.tsq Nom du fichier de sortie pour écrire la requête.

-text Affiche la sortie compréhensible au lieu du DER.

Génération d'une réponse Time Stamp

-reply Créer une réponse time stamp, incluant le status de la réponse et le token lui-même, avec les options suivantes :

-config configfile Fichier de configuration à utiliser. (remplace la variableOPENSSL_CONF)

-section tsa_section Nom de la section contenant les paramètres pour la génération de la réponse

-queryfile request.tsq Nom du fichier contenant la requête timestamp encodée en DER

-passin password_src Source du mot de passe pour la clé provée du TSA

-signer tsa_cert.pem Certificat du signataire du TSA au format PEM. Doit avoir eactement un extension d'utilisation de clé 'timeStamping' et doit être critique

-inkey private.pem clé privée du signataire du TSA au format PEM

-chain certs_file.pem Certification au format PEM à inclure dans la répnose en plus du certificat u signataire

-policy object_id Stratégie par défaut à utiliser pour la réponse à moins que le client en demande une. (nom ou OID)

-in response.tsr Spécifie une réponse timestamp créé précédemment ou un tocken timestamp au format DER qui sera écrit dns le fichier de sortie. Ne requière pas de requête. Permet d'examiner le contenu d'une réponse

-token_in avec -in, indique que l'entrée est un otken timestamp DER (ContentInfo) au lieu d'une réponse (TimeStampResp)

-out response.tsr Fichier de sortie

-token_out La sortie est un timestamp token (ContentInfo) au lieu d'une réponse (TimeStampResp)

-text Sort au format human-readable au lieu d'un DER

-engine id ts va tenter d'obtenir une référence fonctionnelle du moteur spécifié.

Vérification de réponse Time Stamp

-verify permet de vérifier une réponse ou un jeton, il accèpte les options suivante :

-data file_to_hash La réponse ou le token doit être vérifié avec. Ce fihcier est hashé avec le message digest spécifié dans le token.

-digest digest_bytes La réponse ou le token doit être vérifié avec le message digest

-queryfile request.tsq La requête originale en DER

-in response.tsr La réponse qui doit être vérifiée au format DER

-token_in avec -in, indique que l'entrée est un token DER (ContentInfo) au lie d'une réponse (TimeStampResp)

-CApath trusted_cert_path Répertoire contenant Les certificats de CA du client

-CAfile trusted_certs.pem Le nom du fichier contenant un jeu de certificats PEM de ca

-untrusted cert_file.pem Définis des certificats non-trustés PEM qui peuvent être nécessaire pour créer une chaîne de certificat

Options de configuration

tsa section, default_tsa Section principale. Spécifie le nom d'une autre section qui contient toutes les autres options pour la commande -reply

oid_file voir ca

oid_section voir ca

RANDFILE voir ca

serial Voir le nom du fichier qui contient le numéro de série héxa de la dernière réponse timestamp. il est incrémenté de 1

crypto_device Moteur openssl par défaut pour tous les algorithmes disponibles

signer_cert PEM certificat de signature TSA (idem à -signer)

certs Fichier contenant les certificats PEM à inclure dans la réponse (idem à -chain)

signer_key Clé privée du TSA en PEM

default_policy Stratégie par défaut (idem à -policy)

other_policies Liste de stratégies accèptés par le TSA

digests Liste des algorithmes de messages digest accéptès par le TSA

accuracy Source de précision du temps de TSA en secondes, millisecondes et microsecondes

clock_precision_digits Nombre maximum de chiffre représentant la fraction de secondes à inclure dans le champs time (max : 6)

ordering à yes, la réponse générée par ce TSA peut toujour être ordonnée, même si la différence de temps entre 2 réponses est inférieurs à la précision du temps

tsa_name à yes, le sujet du TSA doit est inclus dans le champs name de la réponse

ess_cert_id_chain les objets SignedData créés par le TSA contiennent toujours l'id du certificat du signataire dans l'attribut signed (RC2634)

Exemples

Créé une requêtes pour design.txt avec SHA-1 sans nonce ni stratégie ni certificat dans la réponse :

openssl ts -query -data design1.txt -no_nonce -out design1.tsq
Similaire en spécifiant le digest :
openssl ts -query -digest b7e5d3f93198b38379852f2c04e78d73abdd0f4b -no_nonce -out design1.tsq
Afficher le contenu des requêtes précédentes :
openssl ts -query -in design1.tsq -text
Créer une requête qui inclus MD5 de design2.txt, le certificat du signataire et nonce et spécifie la stratégie :
openssl ts -query -data design2.txt -md5 -policy tsa_policy1 -cert -out design2.tsq
Créer une réponse :
openssl ts -reply -queryfile design1.tsq -inkey tsakey.pem -signer tsacert.pem -out design1.tsr
idem en utilisant les paramètres dans un fichierde configuration :
openssl ts -reply -queryfile design1.tsq -out design1.tsr
Afficher la réponse :
openssl ts -reply -in design1.tsr -text
Créer un token :
openssl ts -reply -queryfile design1.tsq -out design1_token.der -token_out
Afficher un token
openssl ts -reply -in design1_token.der -token_in -text -token_out
Extraire le token d'un réponse :
openssl ts -reply -in design1.tsr -out design1_token.der -token_out
Ajouter le status 'granted à un token en crént une réponse valide :
openssl ts -reply -in design1_token.der -token_in -out design1.tsr
Vérifier une réponse avec la requête :
openssl ts -verify -queryfile design1.tsq -in design1.tsr -CAfile cacert.pem -untrusted tsacert.pem
Vérifier une réponse qui inclus la chaîne de certificat :
openssl ts -verify -queryfile design2.tsq -in design2.tsr -CAfile cacert.pem
Pour vérifier un token avec le fichier originel :
openssl ts -verify -data design2.txt -in design2.tsr -CAfile cacert.pem
Pour vérifier un token avec une empreinte :
openssl ts -verify -digest b7e5d3f93198b38379852f2c04e78d73abdd0f4b -in design2.tsr -CAfile cacert.pem