EBook Gratuito

APPRENDIMENTO

openssl

Free unaffiliated eBook created from
Stack Overflow contributors.

#openssl

Sommario

Di

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: openssl

It is an unofficial and free openssl ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official openssl.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con openssl

Osservazioni

Questa sezione fornisce una panoramica di ciò che openssl è, e perché uno sviluppatore potrebbe voler usarlo.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di openssl e collegarsi agli argomenti correlati. Poiché la documentazione di openssl è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Versioni

pubblicazione	Data
1.1.0e	2017/02/16
1.1.0d	2017/01/26
1.1.0c	2016/11/10
1.1.0b	2016/09/26
1.1.0a	2016/09/22
1.1.0	2016/08/25
1.0.2k	2017/01/26
1.0.2j	2016/09/26
1.0.2i	2016/09/22
1.0.2h	2016/05/03
1.0.2g	2016/03/01

Examples

Installazione o configurazione

OpenSSL è un progetto open source che fornisce un toolkit robusto, di livello commerciale e completo per i protocolli Transport Layer Security (TLS) e Secure Sockets Layer (SSL). È anche una libreria di crittografia generica.

Il toolkit OpenSSL è concesso in licenza in base ad una licenza in stile Apache, il che significa che

sei libero di ottenerlo e usarlo per scopi commerciali e non commerciali soggetti a semplici condizioni di licenza.

Costruisci e installa openssl su sistemi Linux / Unix

Panoramica

Queste istruzioni sono per l'acquisizione, la costruzione e l'installazione di openssl dalla sorgente. Openssl è solitamente incluso anche nei gestori di pacchetti.

risorse

https://github.com/openssl/openssl

dipendenze

- rendere
- perl 5
- gcc / clang
- idiota

Le dipendenze possono essere installate tramite un gestore di pacchetti come apt, dnf o brew.

passi

```
$ cd ~/path/to/projects
$ git clone https://github.com/openssl/openssl.git
$ cd openssl
$ ./config
$ make
$ make test
$ sudo make install
```

Di default, openssl verrà installato su / usr / local.

Verificare

```
$ openssl --version
```

Ora hai una build predefinita di openssl installata sul tuo computer.

(De-) Inizializzazione della libreria openssl

Panoramica

Openssl consiste di 2 librerie: `libcrypto` e `libssl` . Prima che l'API openssl possa essere utilizzata in un'applicazione, è previsto che vengano eseguite le procedure di inizializzazione obbligatorie. Una volta che l'applicazione viene eseguita con il lavoro relativo a openssl, è previsto che pulisca le risorse allocate.

Il codice qui sotto completa l'inizializzazione, tuttavia, lo sviluppatore è libero di inizializzare solo le cose di openssl a cui è interessato.

Inizializza libcrypto

```
ERR_load_crypto_strings();
OpenSSL_add_all_algorithms();
OPENSSL_config(NULL); // Load default configuration (e.g. openssl.conf)
```

Inizializza libssl

```
OPENSSL_init_ssl(0, NULL);
```

Deinitialize

```
CONF_modules_unload(1);
EVP_cleanup();
CRYPTO_cleanup_all_ex_data();
ERR_remove_state();
ERR_free_strings();
```

Esegui OpenSSL su Windows senza installare

Questo workaround ci ha aiutato moltissimo nel mio lavoro (supporto tecnico), abbiamo creato un semplice file batch che potevamo eseguire da qualsiasi luogo (non disponevamo delle autorizzazioni per installare l'exe effettivo). Questa soluzione alternativa eseguirà OpenSSL e aprirà la cartella bin per te (poiché in questo caso verranno salvati tutti i file creati o modificati).

Come impostare:

1. Scarica i binari OpenSSL [qui] [1]. (Si noti che questo è confermato per funzionare con la versione 0.9.8h.)
2. Copia questo codice in un file denominato StartOpenSSL.bat. Salva questo in un luogo a tua scelta. Può essere eseguito da qualsiasi luogo.

```
@echo off
title OpenSSL

cd\openssl\bin
```

```
if exist "C:\openssl\share\openssl.cnf" (

set OPENSSL_CONF=c:/openssl/share/openssl.cnf
start explorer.exe c:\openssl\bin

echo Welcome to OpenSSL

openssl

) else (

echo Error: openssl.cnf was not found
echo File openssl.cnf needs to be present in c:\openssl\share
pause

)

exit
```

3. Una volta scaricati i binari di OpenSSL, estraili nell'unità C in una cartella denominata OpenSSL. (Il percorso deve essere C: \ OpenSSL). Non spostare alcun contenuto delle cartelle in giro, semplicemente estrailo nella cartella.

4. Sei pronto per usare OpenSSL. Questa è una soluzione eccellente per gli utenti Windows che non dispongono dei privilegi per installarlo in quanto non richiede autorizzazioni. Basta eseguire il file bat da prima facendo doppio clic su di esso. [1]: http://gnuwin32.sourceforge.net/packages/openssl.htm

Esempi di comandi OpenSSL

Ispeziona il certificato SSL

```
openssl x509 -in server.crt -noout -text
```

Genera la chiave del server

```
openssl genrsa -out server.key 2048
```

Genera CSR

```
openssl req -out server.csr -key server.key -new
```

Leggi Iniziare con openssl online: https://riptutorial.com/it/openssl/topic/2695/iniziare-con-openssl

Capitolo 2: chiavi

Sintassi

- EVP_PKEY * EVP_PKEY_new (void);
- RSA * RSA_new (vuoto);
- int RSA_generate_key_ex (RSA * rsa, bit int, BIGNUM * e, BN_GENCB * cb);
- int EVP_PKEY_assign_RSA (tasto EVP_PKEY *, tasto RSA *);
- int PEM_write_PrivateKey (FILE * fp, EVP_PKEY * x, const EVP_CIPHER * enc, unsigned char * kstr, int klen, pem_password_cb * cb, void * u);
- int PEM_write_bio_PrivateKey (BIO * bp, EVP_PKEY * x, const EVP_CIPHER * enc, unsigned char * kstr, int klen, pem_password_cb * cb, void * u);
- EVP_PKEY * PEM_read_PrivateKey (FILE * fp, EVP_PKEY ** x, pem_password_cb * cb, void * u);
- EVP_PKEY * PEM_read_bio_PrivateKey (BIO * bp, EVP_PKEY ** x, pem_password_cb * cb, void * u);
- void EVP_PKEY_free (chiave EVP_PKEY *);

Examples

Genera chiave RSA

Per generare una chiave RSA, `EVP_PKEY` deve prima essere allocato con `EVP_PKEY_new` :

```
EVP_PKEY *pkey;
pkey = EVP_PKEY_new();
```

È necessario anche un esponente per la chiave, che richiederà l'assegnazione di un `BIGNUM` con `BN_new` e quindi l'assegnazione con `BN_set_word` :

```
BIGNUM *bn;
bn = BN_new();
BN_set_word(bn, RSA_F4);
```

Per generare la chiave, crea un nuovo `RSA` con `RSA_new` e chiama `RSA_generate_key_ex` :

```
RSA *rsa;
rsa = RSA_new();
RSA_generate_key_ex(
    rsa,  /* pointer to the RSA structure */
    2048, /* number of bits for the key - 2048 is a good value */
    bn,   /* exponent allocated earlier */
    NULL, /* callback - can be NULL if progress isn't needed */
);
```

Per assegnare la chiave appena generata alla struttura `EVP_PKEY` , chiama `EVP_PKEY_assign_RSA` :

```
EVP_PKEY_assign_RSA(pkey, rsa);
```

La struttura `RSA` verrà automaticamente liberata quando la struttura `EVP_PKEY` viene liberata. Questo è fatto con `EVP_PKEY_free` :

```
EVP_PKEY_free(pkey);
```

Salva chiave privata

Un `EVP_PKEY` può essere salvato direttamente su disco in diversi formati. `PEM_write_PrivateKey` viene utilizzato per salvare `EVP_PKEY` in un formato PEM:

```
FILE *f;
f = fopen("key.pem", "wb");
PEM_write_PrivateKey(
    f,                  /* use the FILE* that was opened */
    pkey,               /* EVP_PKEY structure */
    EVP_des_ede3_cbc(), /* default cipher for encrypting the key on disk */
    "replace_me",       /* passphrase required for decrypting the key on disk */
    10,                 /* length of the passphrase string */
    NULL,               /* callback for requesting a password */
    NULL                /* data to pass to the callback */
);
```

Per salvare una chiave privata in un `BIO` , utilizzare `PEM_write_bio_PrivateKey` :

```
BIO *bio;
bio = BIO_new(BIO_s_mem());
PEM_write_bio_PrivateKey(
    bio,                /* BIO to write the private key to */
    pkey,               /* EVP_PKEY structure */
    EVP_des_ede3_cbc(), /* default cipher for encrypting the key on disk */
    "replace_me",       /* passphrase required for decrypting the key on disk */
    10,                 /* length of the passphrase string */
    NULL,               /* callback for requesting a password */
    NULL                /* data to pass to the callback */
);
```

Carica chiave privata

Per caricare una chiave privata direttamente dal disco, utilizzare la funzione `PEM_read_PrivateKey` :

```
FILE *f;
EVP_PKEY *pkey;
f = fopen("key.pem", "rb");
PEM_read_PrivateKey(
    f,     /* use the FILE* that was opened */
    &pkey, /* pointer to EVP_PKEY structure */
    NULL,  /* password callback - can be NULL */
    NULL   /* parameter passed to callback or password if callback is NULL */
);
```

Per caricare una chiave privata da un `BIO` , utilizzare `PEM_read_bio_PrivateKey` :

```
BIO *bio;
bio = BIO_new_mem_buf((void *)input, input_len);
PEM_read_bio_PrivateKey(
    bio,   /* BIO to read the private key from */
    &pkey, /* pointer to EVP_PKEY structure */
    NULL,  /* password callback - can be NULL */
    NULL   /* parameter passed to callback or password if callback is NULL */
);
```

Leggi chiavi online: https://riptutorial.com/it/openssl/topic/4760/chiavi

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con openssl	Camille G., Community, Josip Ivic, Michael, Rex Linder, Roland Bär, tysonite, user, vaibhav magar
2	chiavi	Nathan Osman, tysonite