Let’s encrypt! Certificati SSL per tutti!

Nicola Corti

Gruppo Utenti Linux Pisa

03 Febbraio 2016

Introduction

Cosa è Let's encrypt?

Let's encrypt è una **Certification Authority (CA)** che rilascia certificati SSL/TLS a titolo **gratuito**.

- Dal 5 Dicembre 2015 si trova in **Public Beta**.
- Ad oggi ha rilasciato più di **480 k** certificati.
- Fa dell'**automazione** il suo punto di forza.
- **https://letsencrypt.org/**

Cosa è Let's encrypt?

Let's encrypt è una **Certification Authority (CA)** che rilascia certificati SSL/TLS a titolo **gratuito**.

- Dal 5 Dicembre 2015 si trova in **Public Beta**.
- Ad oggi ha rilasciato più di **480 k** certificati.
- Fa dell'**automazione** il suo punto di forza.
- **https://letsencrypt.org/**

Cosa è Let's encrypt?

Let's encrypt è una **Certification Authority (CA)** che rilascia certificati SSL/TLS a titolo **gratuito**.

- Dal 5 Dicembre 2015 si trova in **Public Beta**.
- Ad oggi ha rilasciato più di **480 k** certificati.
- Fa dell'**automazione** il suo punto di forza.
- **https://letsencrypt.org/**

Cosa è Let's encrypt?

Let's encrypt è una **Certification Authority (CA)** che rilascia certificati SSL/TLS a titolo **gratuito**.

- Dal 5 Dicembre 2015 si trova in **Public Beta**.
- Ad oggi ha rilasciato più di **480 k** certificati.
- Fa dell'**automazione** il suo punto di forza.
- **https://letsencrypt.org/**

Cosa è Let's encrypt?

Let's encrypt è una **Certification Authority (CA)** che rilascia certificati SSL/TLS a titolo **gratuito**.

- Dal 5 Dicembre 2015 si trova in **Public Beta**.
- Ad oggi ha rilasciato più di **480 k** certificati.
- Fa dell'**automazione** il suo punto di forza.
- **https://letsencrypt.org/**

Sponsors

Chi sta supportando il progetto?

Platinum

Gold

History

2012 Iniziato il progetto dentro **Mozilla**

11-2014 Let's Encrypt **annunciato pubblicamente**

01-2015 Protocollo **ACME** sottoposto all'**IETF** per essere standardizzato.

04-2015 **ISRG** (Internet Security Research Group) e la **Linux Foundation** si uniscono al progetto

09-2015 Rilasciato il primo certificato per **helloworld.letsencrypt.org**

10-2015 Il certificato intermedio di L.E. diventa ***cross-signed*** grazie a IdenTrust. (*Let's Encrypt is trusted!*)

12-2015 **Public beta!**

History

2012 Iniziato il progetto dentro **Mozilla**

11-2014 Let's Encrypt **annunciato pubblicamente**

01-2015 Protocollo **ACME** sottoposto all'**IETF** per essere standardizzato.

04-2015 **ISRG** (Internet Security Research Group) e la **Linux Foundation** si uniscono al progetto

09-2015 Rilasciato il primo certificato per **helloworld.letsencrypt.org**

10-2015 Il certificato intermedio di L.E. diventa ***cross-signed*** grazie a IdenTrust. (*Let's Encrypt is trusted!*)

12-2015 **Public beta!**

History

2012 Iniziato il progetto dentro **Mozilla**

11-2014 Let's Encrypt **annunciato pubblicamente**

01-2015 Protocollo **ACME** sottoposto all'**IETF** per essere standardizzato.

04-2015 **ISRG** (Internet Security Research Group) e la **Linux Foundation** si uniscono al progetto

09-2015 Rilasciato il primo certificato per **helloworld.letsencrypt.org**

10-2015 Il certificato intermedio di L.E. diventa ***cross-signed*** grazie a IdenTrust. (*Let's Encrypt is trusted!*)

12-2015 **Public beta!**

History

2012 Iniziato il progetto dentro **Mozilla**

11-2014 Let's Encrypt **annunciato pubblicamente**

01-2015 Protocollo **ACME** sottoposto all'**IETF** per essere standardizzato.

04-2015 **ISRG** (Internet Security Research Group) e la **Linux Foundation** si uniscono al progetto

09-2015 Rilasciato il primo certificato per **helloworld.letsencrypt.org**

10-2015 Il certificato intermedio di L.E. diventa ***cross-signed*** grazie a IdenTrust. (*Let's Encrypt is trusted!*)

12-2015 **Public beta!**

History

2012 Iniziato il progetto dentro **Mozilla**

11-2014 Let's Encrypt **annunciato pubblicamente**

01-2015 Protocollo **ACME** sottoposto all'**IETF** per essere standardizzato.

04-2015 **ISRG** (Internet Security Research Group) e la **Linux Foundation** si uniscono al progetto

09-2015 Rilasciato il primo certificato per **helloworld.letsencrypt.org**

10-2015 Il certificato intermedio di L.E. diventa ***cross-signed*** grazie a IdenTrust. (*Let's Encrypt is trusted!*)

12-2015 **Public beta!**

History

2012 Iniziato il progetto dentro **Mozilla**

11-2014 Let's Encrypt **annunciato pubblicamente**

01-2015 Protocollo **ACME** sottoposto all'**IETF** per essere standardizzato.

04-2015 **ISRG** (Internet Security Research Group) e la **Linux Foundation** si uniscono al progetto

09-2015 Rilasciato il primo certificato per **helloworld.letsencrypt.org**

10-2015 Il certificato intermedio di L.E. diventa ***cross-signed*** grazie a IdenTrust. (*Let's Encrypt is trusted!*)

12-2015 **Public beta!**

History

2012 Iniziato il progetto dentro **Mozilla**

11-2014 Let's Encrypt **annunciato pubblicamente**

01-2015 Protocollo **ACME** sottoposto all'**IETF** per essere standardizzato.

04-2015 **ISRG** (Internet Security Research Group) e la **Linux Foundation** si uniscono al progetto

09-2015 Rilasciato il primo certificato per **helloworld.letsencrypt.org**

10-2015 Il certificato intermedio di L.E. diventa ***cross-signed*** grazie a IdenTrust. (*Let's Encrypt is trusted!*)

12-2015 **Public beta!**

Why?

- **Free**
- **Automatico**
- **Open**
- **Sicuro**
- **Trasparente**

Server

Boulder

Architettura

Il sistema di Let's Encrypt si basa su **3 Componenti**: un **server**, un **client** e il **protocollo** che permette a server e client di comunicare.

Il server si chiama **Boulder** ed è scritto interamente in **Go**. Si occupa di gestire tutte le procedure per il **rilascio** e **revoca** dei certificati.

Di fatto si tratta di un server HTTP che espone un'interfaccia **RESTful**

https://github.com/letsencrypt/boulder

Boulder

Architettura

Il sistema di Let's Encrypt si basa su **3 Componenti**: un **server**, un **client** e il **protocollo** che permette a server e client di comunicare.

Il server si chiama **Boulder** ed è scritto interamente in **Go**. Si occupa di gestire tutte le procedure per il **rilascio** e **revoca** dei certificati.

Di fatto si tratta di un server HTTP che espone un'interfaccia **RESTful**

https://github.com/letsencrypt/boulder

Boulder

Architettura

Il sistema di Let's Encrypt si basa su **3 Componenti**: un **server**, un **client** e il **protocollo** che permette a server e client di comunicare.

Il server si chiama **Boulder** ed è scritto interamente in **Go**. Si occupa di gestire tutte le procedure per il **rilascio** e **revoca** dei certificati.

Di fatto si tratta di un server HTTP che espone un'interfaccia **RESTful**

https://github.com/letsencrypt/boulder

Boulder

Architettura

Il sistema di Let's Encrypt si basa su **3 Componenti**: un **server**, un **client** e il **protocollo** che permette a server e client di comunicare.

Il server si chiama **Boulder** ed è scritto interamente in **Go**. Si occupa di gestire tutte le procedure per il **rilascio** e **revoca** dei certificati.

Di fatto si tratta di un server HTTP che espone un'interfaccia **RESTful**

https://github.com/letsencrypt/boulder

Client

letsencrypt

Il client si chiama (semplicemente) **letsencrypt** ed è scritto interamente in **Python**. Si occupa di farvi interagire con il server remoto e **gestisce i vostri certificati**.

- Installabile tramite il pacchetto **letsencrypt** (solo debian *sid/stretch*).
- Installabile tramite **clone** del repository.

https://github.com/letsencrypt/letsencrypt

letsencrypt

Il client si chiama (semplicemente) **letsencrypt** ed è scritto interamente in **Python**. Si occupa di farvi interagire con il server remoto e **gestisce i vostri certificati**.

- Installabile tramite il pacchetto **letsencrypt** (solo debian *sid/stretch*).
- Installabile tramite **clone** del repository.

https://github.com/letsencrypt/letsencrypt

letsencrypt

Il client si chiama (semplicemente) **letsencrypt** ed è scritto interamente in **Python**. Si occupa di farvi interagire con il server remoto e **gestisce i vostri certificati**.

- Installabile tramite il pacchetto **letsencrypt** (solo debian *sid/stretch*).
- Installabile tramite **clone** del repository.

https://github.com/letsencrypt/letsencrypt

Plugins

Lo scopo principale del client è quello di **semplificare e automatizzare** tutto il processo di autenticazione e di creazione del certificato.

Per questo motivo sono stati sviluppati dei plugin per il setup automatico dei certificati ottenuti sui principali web browser: **apache** ed **nginx**.

Plugins

Lo scopo principale del client è quello di **semplificare e automatizzare** tutto il processo di autenticazione e di creazione del certificato.

Per questo motivo sono stati sviluppati dei plugin per il setup automatico dei certificati ottenuti sui principali web browser: **apache** ed **nginx**.

Protocol

ACME

Il protocollo che utilizzano client e server per interagire si chiama **Automated Certificate Management Environment (ACME)**.

ACME si base sullo scambio di file **JSON** firmati (anche detti **JWS, Json Web Signature**). Questi documenti contengono le richieste inviate dal client e le risposte ottenute dal server.

Lo scambio di questi documenti deve avvenire attraverso **HTTPS**.

ACME

Il protocollo che utilizzano client e server per interagire si chiama **Automated Certificate Management Environment (ACME)**.

ACME si base sullo scambio di file **JSON** firmati (anche detti **JWS, Json Web Signature**). Questi documenti contengono le richieste inviate dal client e le risposte ottenute dal server.

Lo scambio di questi documenti deve avvenire attraverso **HTTPS**.

ACME

Il protocollo che utilizzano client e server per interagire si chiama **Automated Certificate Management Environment (ACME)**.

ACME si base sullo scambio di file **JSON** firmati (anche detti **JWS, Json Web Signature**). Questi documenti contengono le richieste inviate dal client e le risposte ottenute dal server.

Lo scambio di questi documenti deve avvenire attraverso **HTTPS**.

ACME

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 22, 2016

R. Barnes
Mozilla
J. Hoffman-Andrews
EFF
J. Kasten
University of Michigan
July 21, 2015

Automatic Certificate Management Environment (ACME)

draft-barnes-acme-04

Abstract

Certificates in the Web's X.509 PKI (PKIX) are used for a number of purposes, the most significant of which is the authentication of domain names. Thus, certificate authorities in the Web PKI are trusted to verify that an applicant for a certificate legitimately represents the domain name(s) in the certificate. Today, this verification is done through a collection of ad hoc mechanisms. This document describes a protocol that a certificate authority (CA) and an applicant can use to automate the process of verification and certificate issuance. The protocol also provides facilities for other certificate management functions, such as certificate revocation.

https://github.com/letsencrypt/acme-spec

ACME

Il protocollo ACME ci permette di eseguire tre operazioni:

1. **Provare** che noi siamo i responsabili del dominio *example.com*
2. **Ottenere** un certificato per il dominio *example.com*
3. **Revocare** o **rinnovare** un certificato per il dominio *example.com*

ACME

Il protocollo ACME ci permette di eseguire tre operazioni:

1. **Provare** che noi siamo i responsabili del dominio *example.com*
2. **Ottenere** un certificato per il dominio *example.com*
3. **Revocare** o **rinnovare** un certificato per il dominio *example.com*

ACME

Il protocollo ACME ci permette di eseguire tre operazioni:

1. **Provare** che noi siamo i responsabili del dominio *example.com*
2. **Ottenere** un certificato per il dominio *example.com*
3. **Revocare** o **rinnovare** un certificato per il dominio *example.com*

ACME

Il protocollo ACME ci permette di eseguire tre operazioni:

1. **Provare** che noi siamo i responsabili del dominio *example.com*
2. **Ottenere** un certificato per il dominio *example.com*
3. **Revocare** o **rinnovare** un certificato per il dominio *example.com*

Security

Le interazioni fra client e server sono cifrate tramite una coppia di **chiavi asimmetriche** (una **privata** ed una **pubblica**) che viene generata durante il primo avvio del client.

Il server ci permette di **provare** che siamo gli amministratori del dominio desirato, inviandoci una o più **challenge** che dobbiamo risolvere.

Ogni interazione con il server è inoltre corredata da un **nonce** che permette di evitare attacchi di tipo **Replay**.

Security

Le interazioni fra client e server sono cifrate tramite una coppia di **chiavi asimmetriche** (una **privata** ed una **pubblica**) che viene generata durante il primo avvio del client.

Il server ci permette di **provare** che siamo gli amministratori del dominio desirato, inviandoci una o più **challenge** che dobbiamo risolvere.

Ogni interazione con il server è inoltre corredata da un **nonce** che permette di evitare attacchi di tipo **Replay**.

Security

Le interazioni fra client e server sono cifrate tramite una coppia di **chiavi asimmetriche** (una **privata** ed una **pubblica**) che viene generata durante il primo avvio del client.

Il server ci permette di **provare** che siamo gli amministratori del dominio desirato, inviandoci una o più **challenge** che dobbiamo risolvere.

Ogni interazione con il server è inoltre corredata da un **nonce** che permette di evitare attacchi di tipo **Replay**.

Challenges

Il server può, a sua discrezione, inviare una o più challenges.

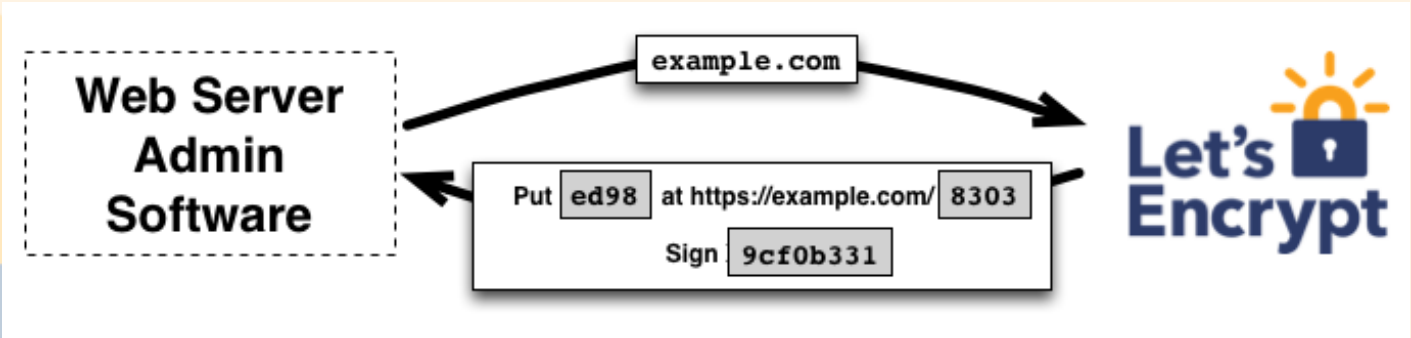
Tipo	Descrizione
Simple HTTP	Si chiede di posizionare un **file** all'interno del proprio webserver contenente un *token* specifico (Accettato sia HTTP che HTTPS).
DNS	Si chiede di inserire un **record TXT** all'interno del proprio DNS.
Proof of possession	Si chiede di **firmare un documento** utilizzando una chiave che il server può ricondurre al client.
Domain Validation with Server Name Indication	Si chiede di **configurare un server TLS** ad uno specifico indirizzo IP (tramite un record A nel DNS).

Challenges

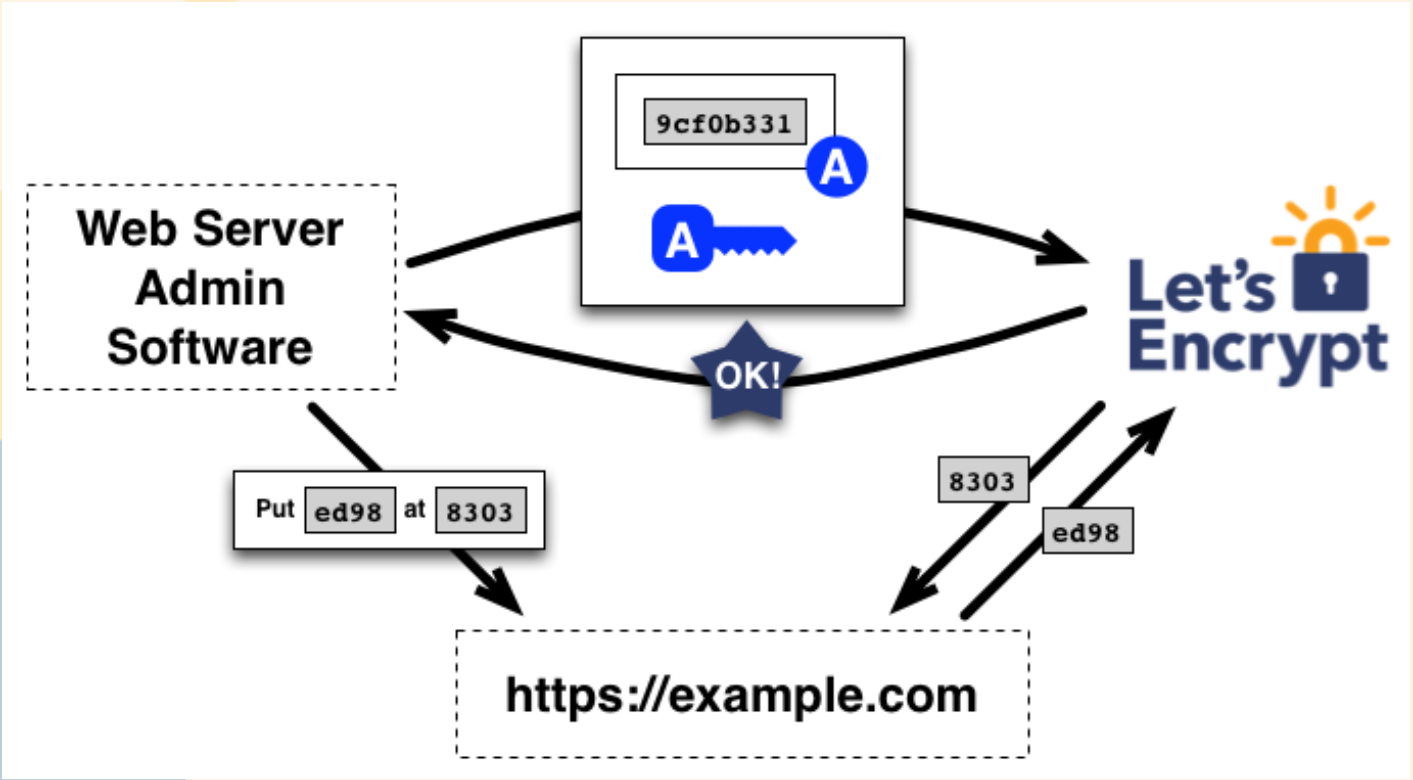
Il server può, a sua discrezione, inviare una o più challenges.

Tipo	Descrizione
Simple HTTP	Si chiede di posizionare un **file** all'interno del proprio webserver contenente un *token* specifico (Accettato sia HTTP che HTTPS).
DNS	Si chiede di inserire un **record TXT** all'interno del proprio DNS.
Proof of possession	Si chiede di **firmare un documento** utilizzando una chiave che il server può ricondurre al client.
Domain Validation with Server Name Indication	Si chiede di **configurare un server TLS** ad uno specifico indirizzo IP (tramite un record A nel DNS).

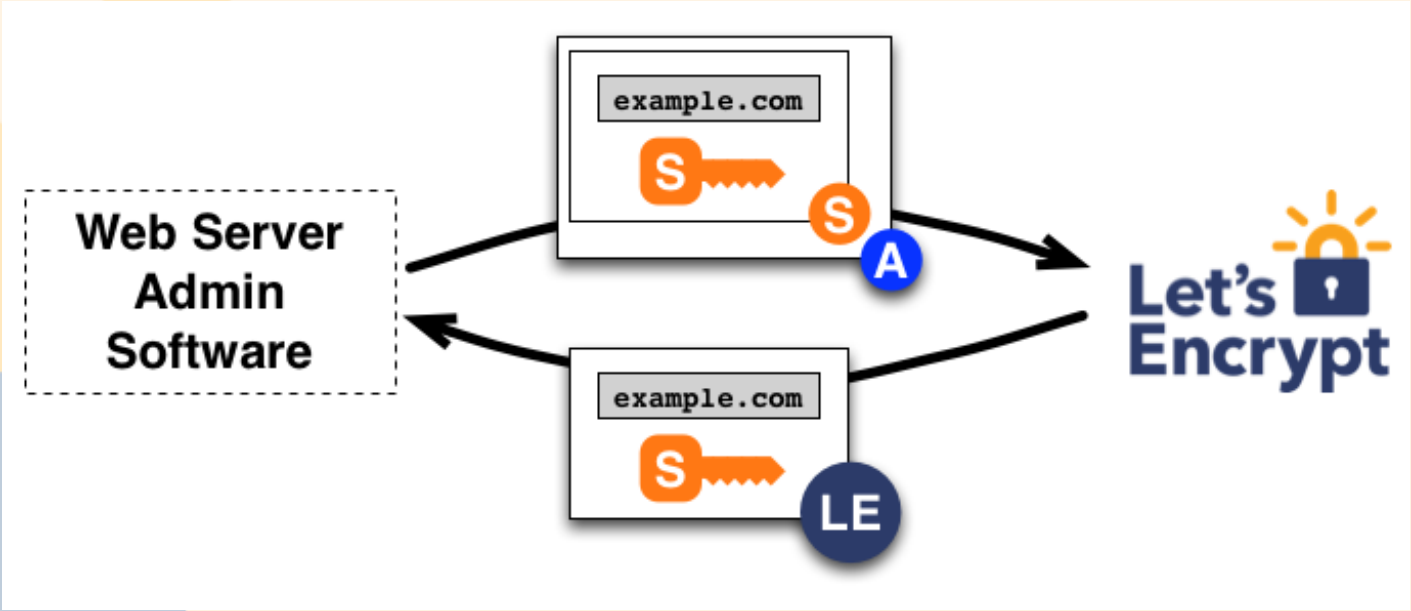
Domain Validation



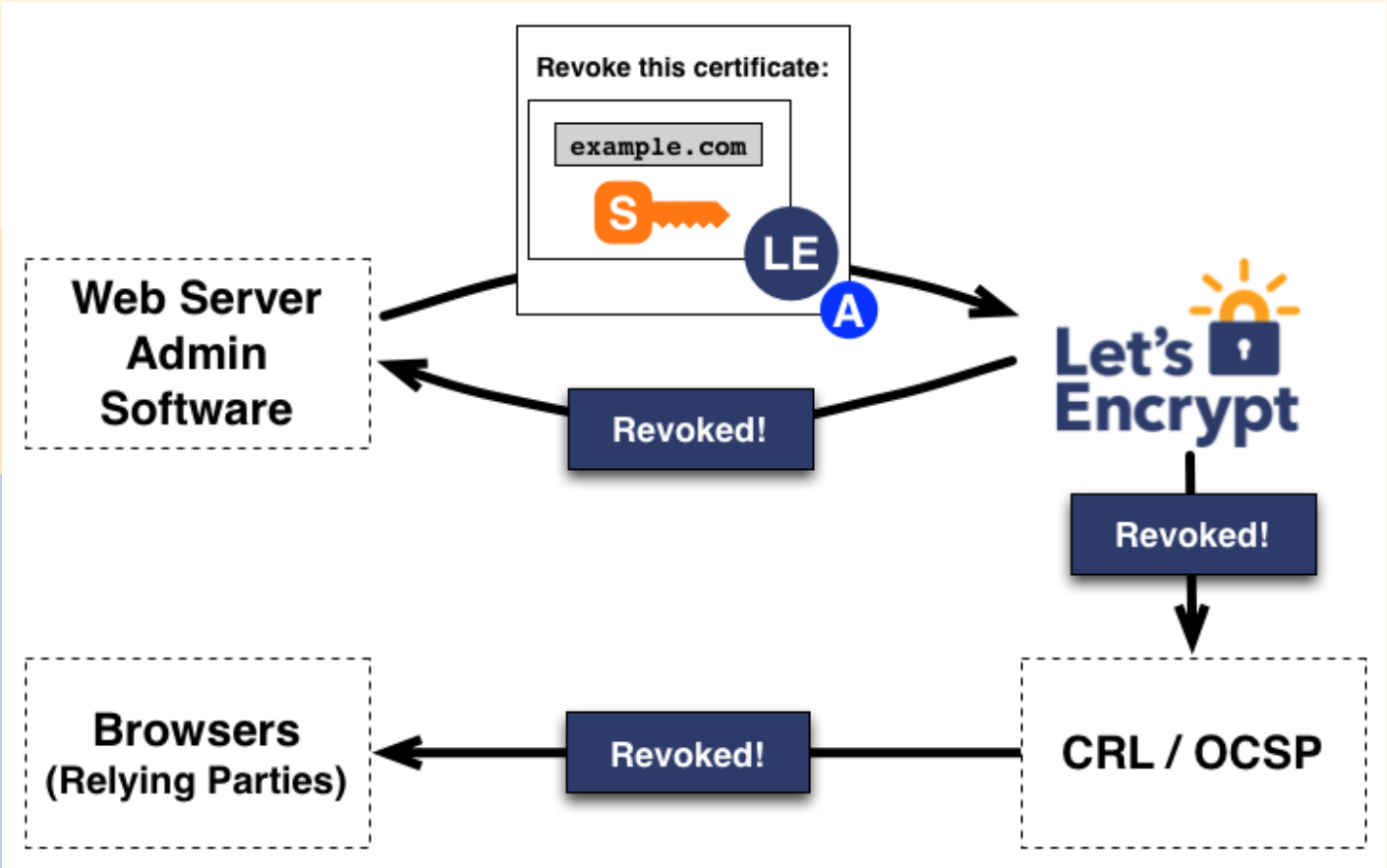
Domain Validation



Certificate Issuance



Certificate Revocation



Certificates

DV Certificates

DV

I certificati rilasciati sono di tipo **Domain Validated**. Provano cioè che si è effettivamente i proprietari di un determinato dominio.

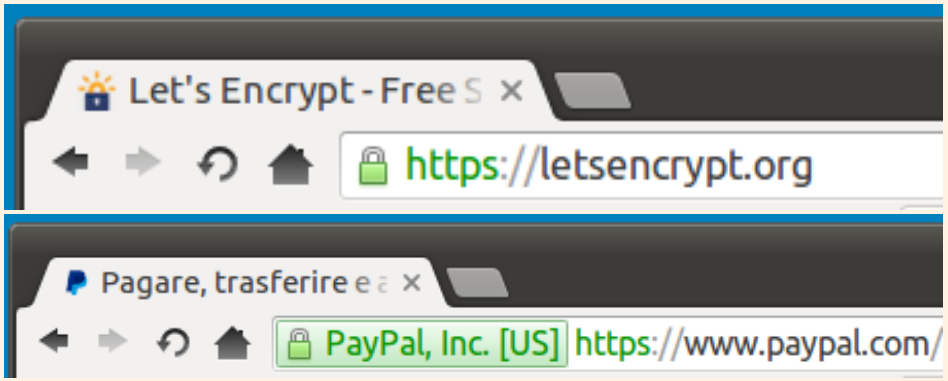
Certificati di tipo **Organization Validation** oppure **Extended Validation** richiederebbero verifiche esplicite del soggetto che fa richiesta del certificato.

DV Certificates

DV

I certificati rilasciati sono di tipo **Domain Validated**. Provano cioè che si è effettivamente i proprietari di un determinato dominio.

Certificati di tipo **Organization Validation** oppure **Extended Validation** richiederebbero verifiche esplicite del soggetto che fa richiesta del certificato.

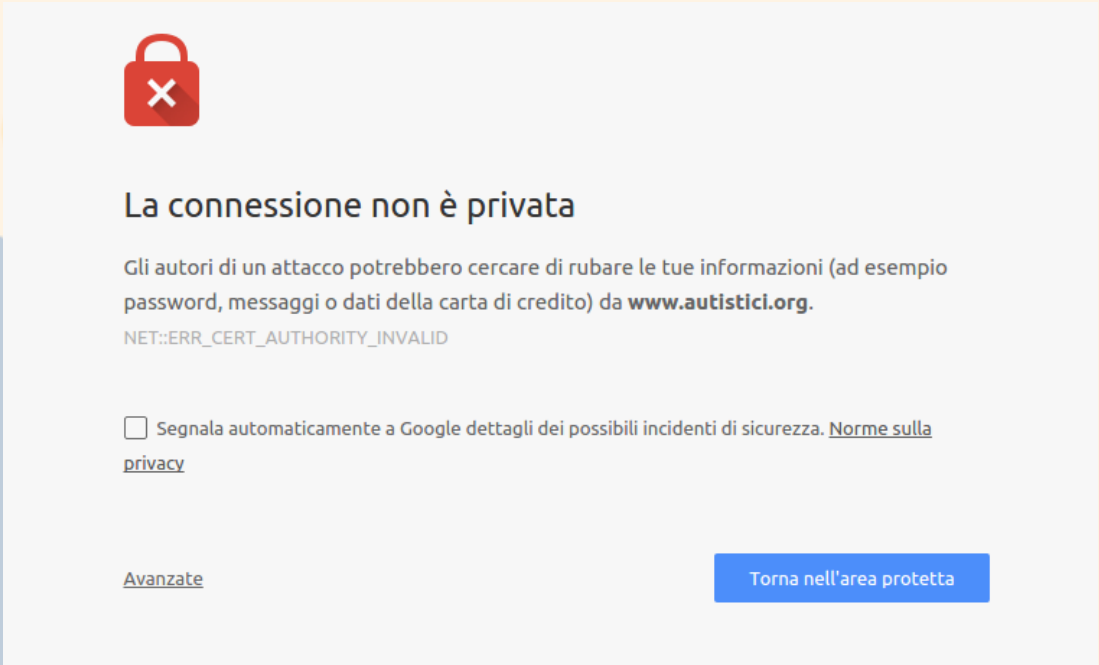


Cross Signing

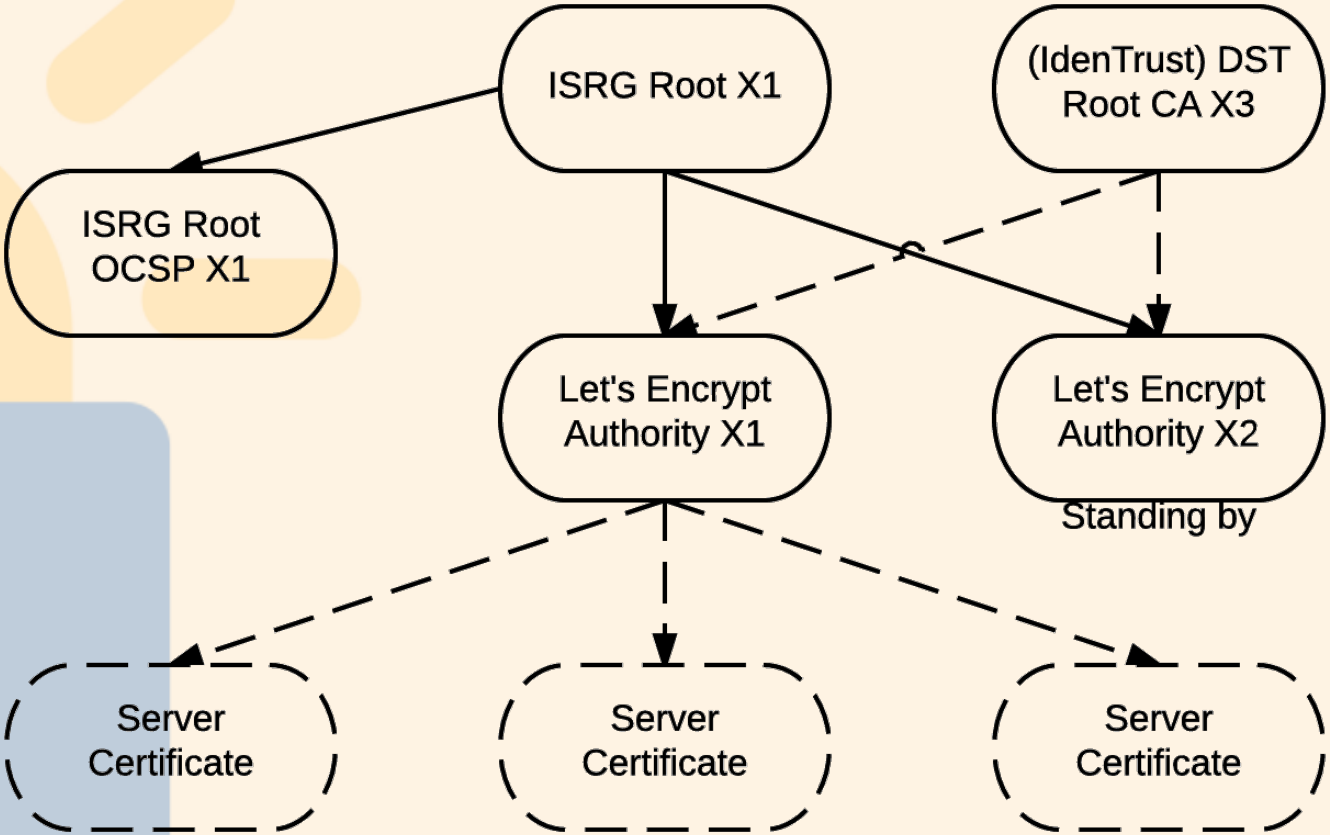
Tutti i certificati rilasciati sono *Cross-signed* da parte di **IdenTrust**. Questo permette ai certificati di L.E. di essere riconosciuti dalla maggioranza dei browser.

Cross Signing

Tutti i certificati rilasciati sono *Cross-signed* da parte di **IdenTrust**. Questo permette ai certificati di L.E. di essere riconosciuti dalla maggioranza dei browser.



Cross Signing



Validity

I certificati hanno una validità di **90 giorni** dopo i quali scadono, e non sono considerati più validi dai browser.

90-days validity

Questa non è una novità nel mondo della sicurezza. **Rinnovare i certificati frequentemente** è una buona abitudine e permette di **limitare i danni** di eventuali certificati compromessi.

Riceveremo delle mail di *remind* che ci ricordano della scadenza. È possibile automatizzare il processo di rinnovo tramite dei task di **cron**.

Validity

I certificati hanno una validità di **90 giorni** dopo i quali scadono, e non sono considerati più validi dai browser.

90-days validity

Questa non è una novità nel mondo della sicurezza. **Rinnovare i certificati frequentemente** è una buona abitudine e permette di **limitare i danni** di eventuali certificati compromessi.

Riceveremo delle mail di *remind* che ci ricordano della scadenza. È possibile automatizzare il processo di rinnovo tramite dei task di **cron**.

Validity

I certificati hanno una validità di **90 giorni** dopo i quali scadono, e non sono considerati più validi dai browser.

90-days validity

Questa non è una novità nel mondo della sicurezza. **Rinnovare i certificati frequentemente** è una buona abitudine e permette di **limitare i danni** di eventuali certificati compromessi.

Riceveremo delle mail di *remind* che ci ricordano della scadenza. È possibile automatizzare il processo di rinnovo tramite dei task di **cron**.

Setup

Requirements

Per eseguire il client è necessario:

- Essere su un sistema **Unix like**.
- Avere **Python 2.6** o **2.7**.
- Avere accesso di **root** sul proprio sistema.

Inoltre, il plugin di autoconfigurazione di **apache**, funziona solamente con sistemi Debian based: **Ubuntu 12.04+** e **Debian 7+**

Requirements

Per eseguire il client è necessario:

- Essere su un sistema **Unix like**.
- Avere **Python 2.6** o **2.7**.
- Avere accesso di **root** sul proprio sistema.

Inoltre, il plugin di autoconfigurazione di **apache**, funziona solamente con sistemi Debian based: **Ubuntu 12.04+** e **Debian 7+**

Download

Per prima cosa è necessario scaricare il client **letsencrypt**.

Per Debian *sid/stretch*:

```
$ sudo apt-get install letsencrypt
$ letsencrypt --help
```

Per tutti gli altri:

```
$ git clone https://github.com/letsencrypt/letsencrypt
$ cd letsencrypt
$ ./letsencrypt-auto --help
```

Da ora in poi ci riferiremo ai comandi con **letsencrypt-auto** assumendo che si scelga di clonare il repository.

Download

Per prima cosa è necessario scaricare il client **letsencrypt**.

Per Debian *sid/stretch*:

```
$ sudo apt-get install letsencrypt
$ letsencrypt --help
```

Per tutti gli altri:

```
$ git clone https://github.com/letsencrypt/letsencrypt
$ cd letsencrypt
$ ./letsencrypt-auto --help
```

Da ora in poi ci riferiremo ai comandi con **letsencrypt-auto** assumendo che si scelga di clonare il repository.

Download

Per prima cosa è necessario scaricare il client **letsencrypt**.

Per Debian *sid/stretch*:

```
$ sudo apt-get install letsencrypt
$ letsencrypt --help
```

Per tutti gli altri:

```
$ git clone https://github.com/letsencrypt/letsencrypt
$ cd letsencrypt
$ ./letsencrypt-auto --help
```

Da ora in poi ci riferiremo ai comandi con **letsencrypt-auto** assumendo che si scelga di clonare il repository.

Download

Per prima cosa è necessario scaricare il client **letsencrypt**.

Per Debian *sid/stretch*:

```
$ sudo apt-get install letsencrypt
$ letsencrypt --help
```

Per tutti gli altri:

```
$ git clone https://github.com/letsencrypt/letsencrypt
$ cd letsencrypt
$ ./letsencrypt-auto --help
```

Da ora in poi ci riferiremo ai comandi con **letsencrypt-auto** assumendo che si scelga di clonare il repository.

Run

Eseguire il client è molto semplice. È sufficiente invocare il comando:

```
$ ./letsencrypt-auto
```

E verremo guidati durante il processo di issuing dei certificati.

Apache

Se si vuole configurare **Apache** per utilizzare i certificati che vengono generati, è possibile utilizzare il comando

```
$ ./letsencrypt-auto --apache -d example.com -d www.example.com
```

Indicando con **--apache** che si vuole configurare Apache e con **-d** i vari domini per cui si sta richiedendo un certificato.

Contacts

Durante la prima interazione ci verrà chiesta la nostra **email** per i contatti e ci verrà chiesto di accettare i **Terms of service**.

È possibile saltare questi passaggi tramite i seguenti flag

```
$ ./letsencrypt-auto --email admin@example.com --agree-tos
```

Contacts

Durante la prima interazione ci verrà chiesta la nostra **email** per i contatti e ci verrà chiesto di accettare i **Terms of service**.

È possibile saltare questi passaggi tramite i seguenti flag

```
$ ./letsencrypt-auto --email admin@example.com --agree-tos
```

Plugins

Plugin	A	I	Descrizione
Apache	Y	Y	Richiede ed installa automaticamente i certificati su Apache 2.4 (Debian based)
Standalone	Y	N	Avvia un webserver *standalone* sulla porta 80/443
Standalone	Y	N	Ottiene il certificato andando a scrivere all'interno della *root* di web server già attivo
Manual	Y	N	Fornisce i comandi da eseguire per ottenere un certificato su un'altro web server
Nginx	Y	Y	Richiede ed installa automaticamente i certificati su Nginx (sperimentale)

certonly

È possibile utilizzare i plugin con il supporto per l'*authentication* (Y nella colonna A) per ottenere **solamente i certificati** senza effettuare l'installazione degli stessi.

È sufficiente indicare l'opzione **certonly** nella linea di comando.

Standalone example

Ad esempio tramite il comando

```
$ ./letsencrypt-auto --standalone-supported-challenges \
 http-01 certonly -d example.com
```

Si avvia un server standalone sulla porta 80 e si otterrà il certificato per il dominio example.com

certonly

È possibile utilizzare i plugin con il supporto per l'*authentication* (Y nella colonna A) per ottenere **solamente i certificati** senza effettuare l'installazione degli stessi.

È sufficiente indicare l'opzione **certonly** nella linea di comando.

Standalone example

Ad esempio tramite il comando

```
$ ./letsencrypt-auto --standalone-supported-challenges \
 http-01 certonly -d example.com
```

Si avvia un server standalone sulla porta 80 e si otterrà il certificato per il dominio example.com

/etc/letsencrypt

Il client andrà a salvare i certificati e le chiavi dentro la cartella **/etc/letsencrypt**.

Dentro questa cartella andranno a finire sia i **certificati** che le **chiavi pubbliche/private**. È quindi **FONDAMENTALE** fare un backup della cartella e mantenerla in un **luogo sicuro**.

I certificati si troveranno dentro la cartella **/etc/letsencrypt/live/example.com/**, sono link simbolici che verranno aggiornati ad ogni rinnovo dei certificati.

/etc/letsencrypt

Il client andrà a salvare i certificati e le chiavi dentro la cartella **/etc/letsencrypt**.

Dentro questa cartella andranno a finire sia i **certificati** che le **chiavi pubbliche/private**. È quindi **FONDAMENTALE** fare un backup della cartella e mantenerla in un **luogo sicuro**.

I certificati si troveranno dentro la cartella **/etc/letsencrypt/live/example.com/**, sono link simbolici che verranno aggiornati ad ogni rinnovo dei certificati.

/etc/letsencrypt

I file che andremo a trovare sono:

privkey.pem Chiave privata del certificato. Da non condividere MAI!

cert.pem Certificato del server (quello che viene inviato al client).

chain.pem Elenco dei certificati intermedi delle C.A. collegati al certificato.

fullchain.pem Elenco dei certificati intermedi delle C.A. collegati al certificato, più il certificato stesso.

Revoke

Per revocare un certificato è sufficiente invocare l'opzione **revoke**.

```
$ ./.letsencrypt-auto revoke --cert-path example-cert.pem
```

Revoke

Per revocare un certificato è sufficiente invocare l'opzione **revoke**.

```
$ /.letsencrypt-auto revoke --cert-path example-cert.pem
```

Renewal

L'operazione di rinnovo è molto semplice, e può essere effettuata semplicemente invocando il comando **letsencrypt**, senza parametri.

È inoltre possibile impostare flag quali **--renew-by-default** per effettuare il rinnovo senza nessuna interazione da parte dell'utente.

In questo modo è possibile schedulare un task su **cron** per il rinnovo automatico dei certificati prima dei 90 giorni.

Renewal

L'operazione di rinnovo è molto semplice, e può essere effettuata semplicemente invocando il comando **letsencrypt**, senza parametri.

È inoltre possibile impostare flag quali **--renew-by-default** per effettuare il rinnovo senza nessuna interazione da parte dell'utente.

In questo modo è possibile schedulare un task su **cron** per il rinnovo automatico dei certificati prima dei 90 giorni.

Renewal

L'operazione di rinnovo è molto semplice, e può essere effettuata semplicemente invocando il comando **letsencrypt**, senza parametri.

È inoltre possibile impostare flag quali **--renew-by-default** per effettuare il rinnovo senza nessuna interazione da parte dell'utente.

In questo modo è possibile schedulare un task su **cron** per il rinnovo automatico dei certificati prima dei 90 giorni.

Update

È importante tenere il client aggiornato, in quanto si tratta di una beta pubblica.

Per Debian *sid/stretch*:

```
$ sudo apt-get update && sudo apt-get upgrade
```

Per tutti gli altri:

```
$ cd letsencrypt
$ git pull
```

Update

È importante tenere il client aggiornato, in quanto si tratta di una beta pubblica.

Per Debian *sid/stretch*:

```
$ sudo apt-get update && sudo apt-get upgrade
```

Per tutti gli altri:

```
$ cd letsencrypt
$ git pull
```

Update

È importante tenere il client aggiornato, in quanto si tratta di una beta pubblica.

Per Debian *sid/stretch*:

```
$ sudo apt-get update && sudo apt-get upgrade
```

Per tutti gli altri:

```
$ cd letsencrypt
$ git pull
```

General usage

Nella stragrande maggioranza dei casi, sarà sufficiente invocare il comando **letsencrypt** e farsi guidare dall'interfaccia grafica da terminale.

Let's Encrypt ci guiderà durante tutti i processi di generazione/rinnovo di ogni certificato. Sarà sufficiente rispondere alle domande che ci vengono chieste.

Troppi parametri da ricordare renderebbero poco snello il processo automatico di letsencrypt, ma al tempo stesso permettono molta **flessibilità** e permettono di includere letsencrypt dentro **script**.

General usage

Nella stragrande maggioranza dei casi, sarà sufficiente invocare il comando **letsencrypt** e farsi guidare dall'interfaccia grafica da terminale.

Let's Encrypt ci guiderà durante tutti i processi di generazione/rinnovo di ogni certificato. Sarà sufficiente rispondere alle domande che ci vengono chieste.

Troppi parametri da ricordare renderebbero poco snello il processo automatico di letsencrypt, ma al tempo stesso permettono molta **flessibilità** e permettono di includere letsencrypt dentro **script**.

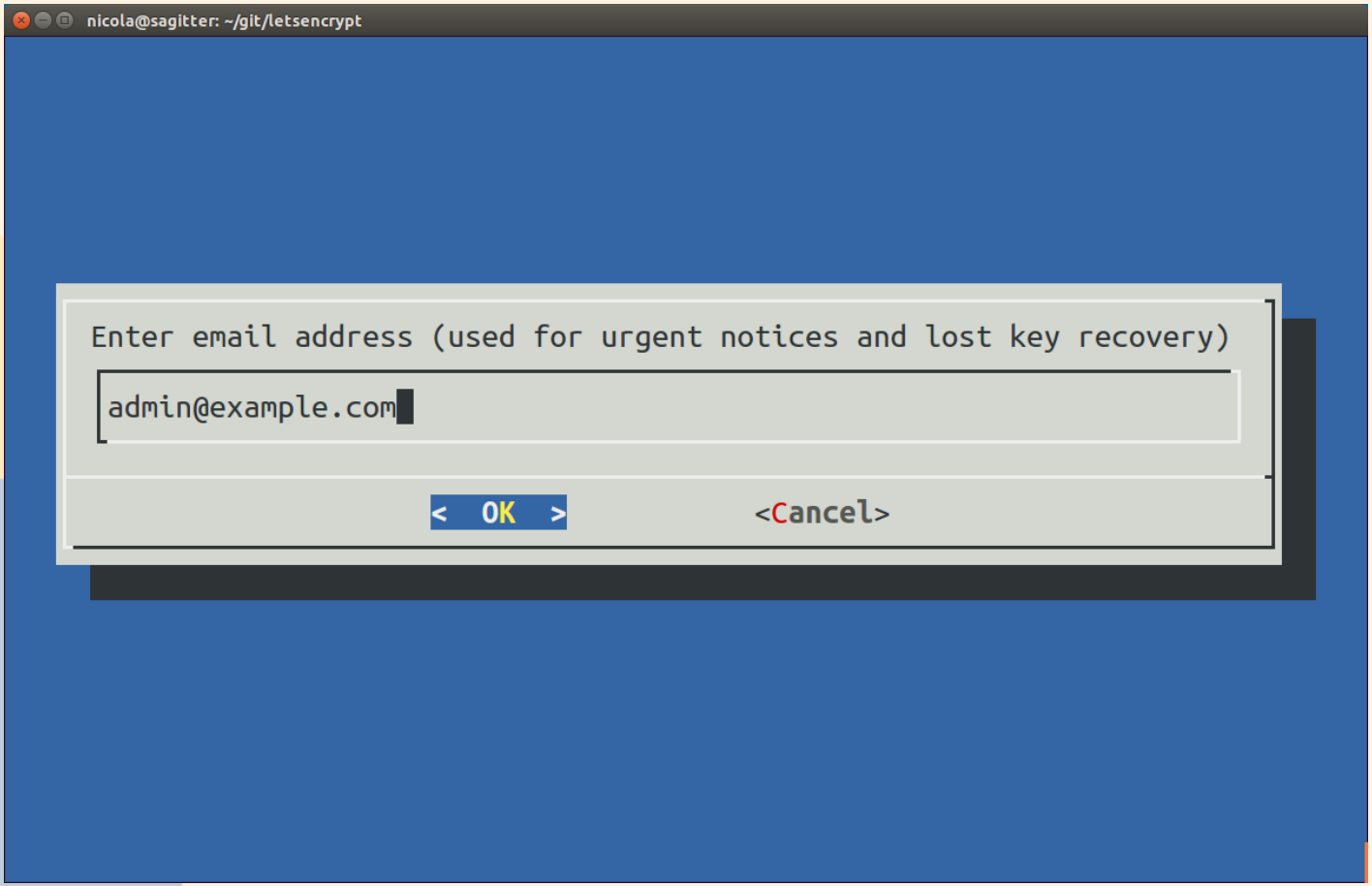
General usage

Nella stragrande maggioranza dei casi, sarà sufficiente invocare il comando **letsencrypt** e farsi guidare dall'interfaccia grafica da terminale.

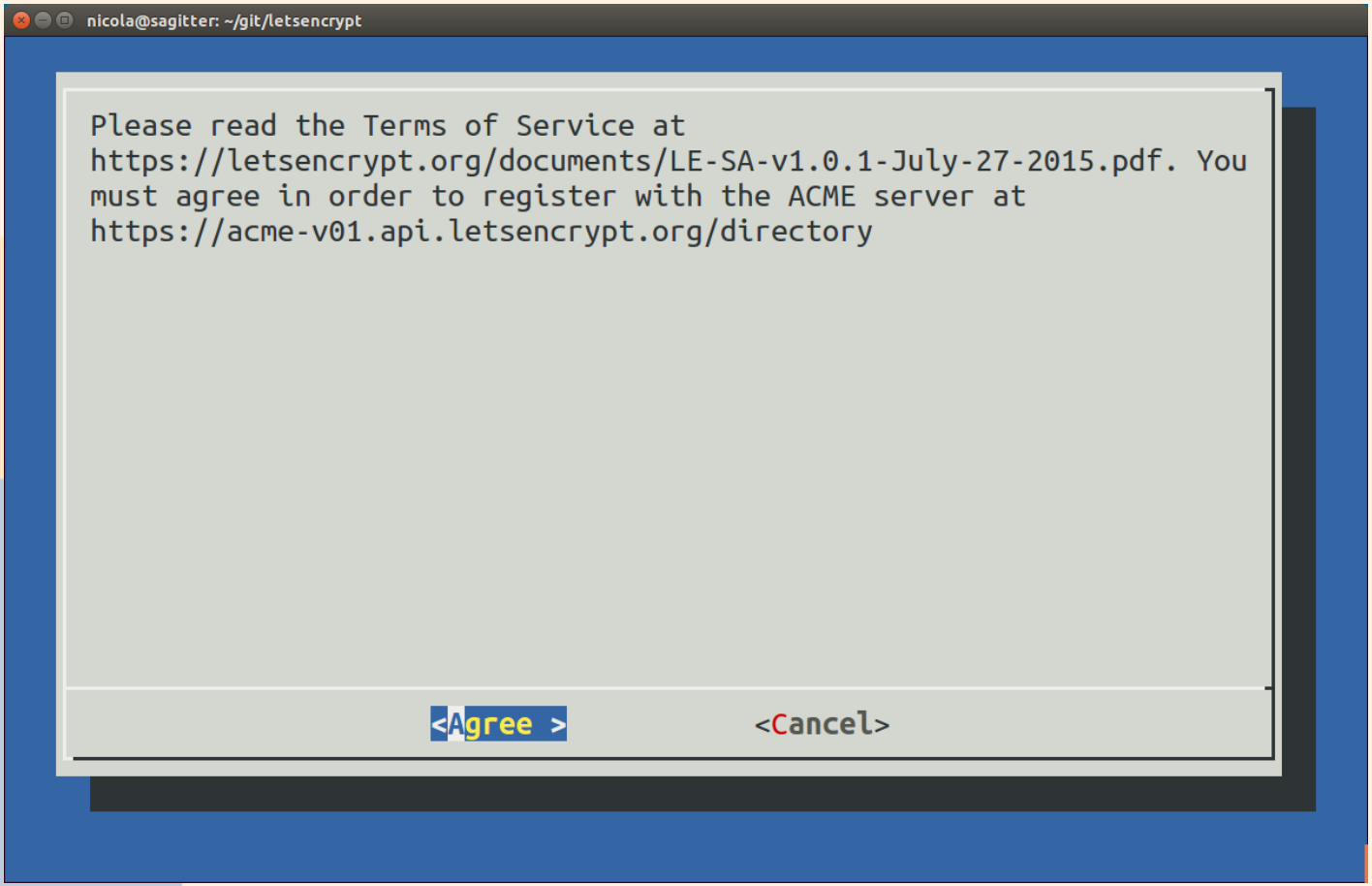
Let's Encrypt ci guiderà durante tutti i processi di generazione/rinnovo di ogni certificato. Sarà sufficiente rispondere alle domande che ci vengono chieste.

Troppi parametri da ricordare renderebbero poco snello il processo automatico di letsencrypt, ma al tempo stesso permettono molta **flessibilità** e permettono di includere letsencrypt dentro **script**.

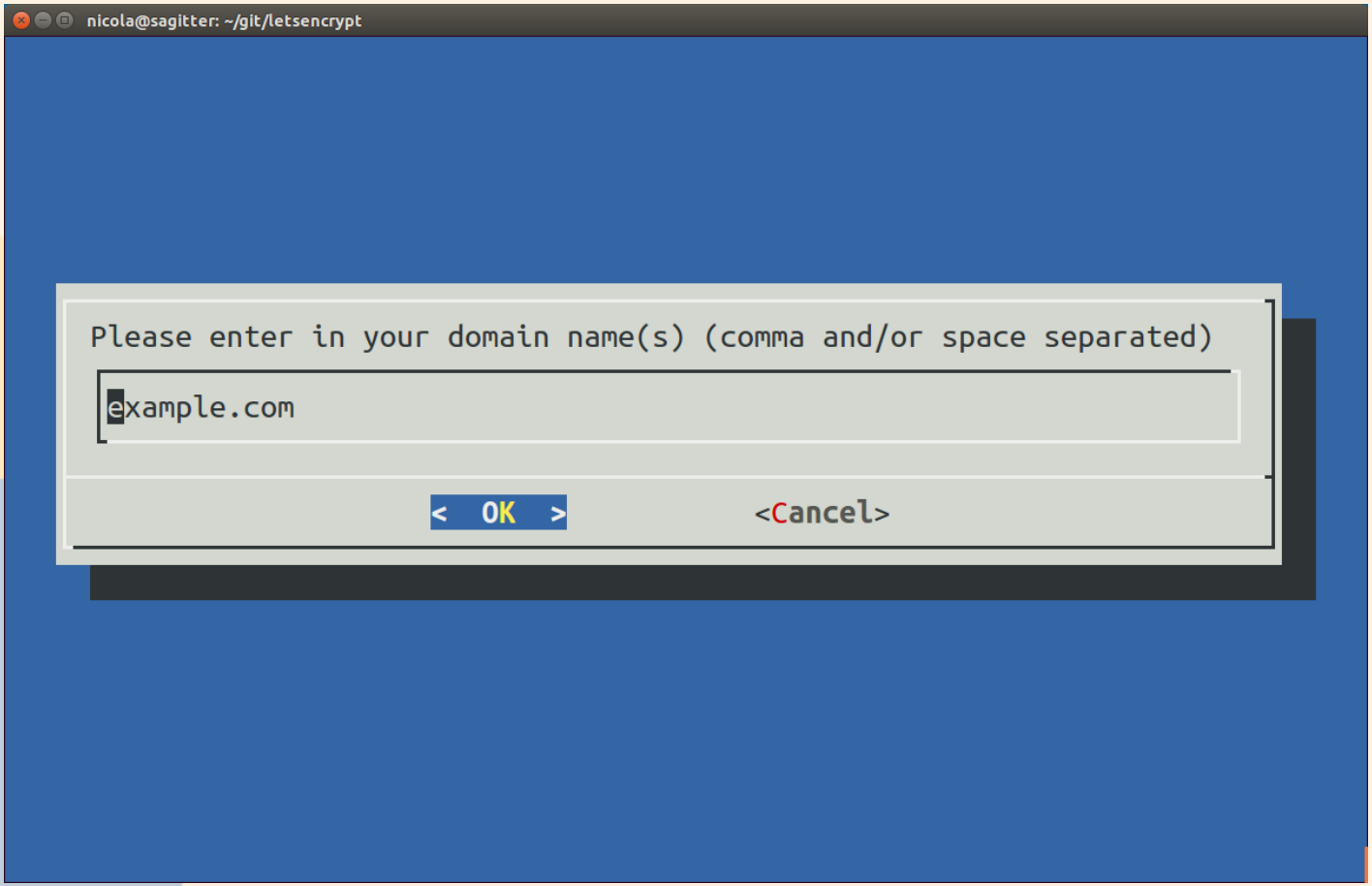
Screens


nicola@sagitter: ~/git/letsencrypt
Enter email address (used for urgent notices and lost key recovery)
admin@example.com
< OK >
<Cancel>

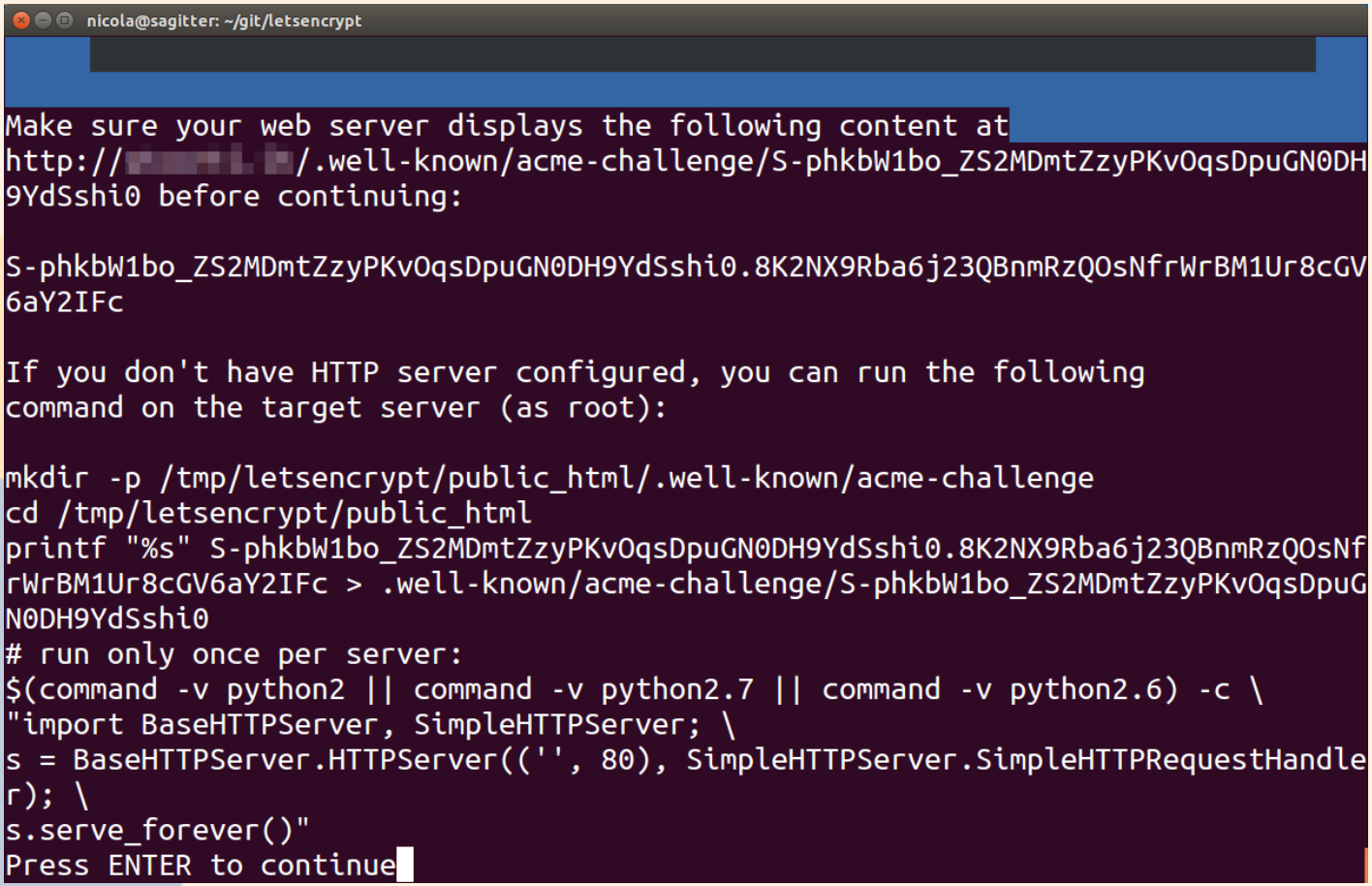
Screens


nicola@sagitter: ~/git/letsencrypt
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.0.1-July-27-2015.pdf. You
must agree in order to register with the ACME server at
https://acme-v01.api.letsencrypt.org/directory
<Agree >
<Cancel>

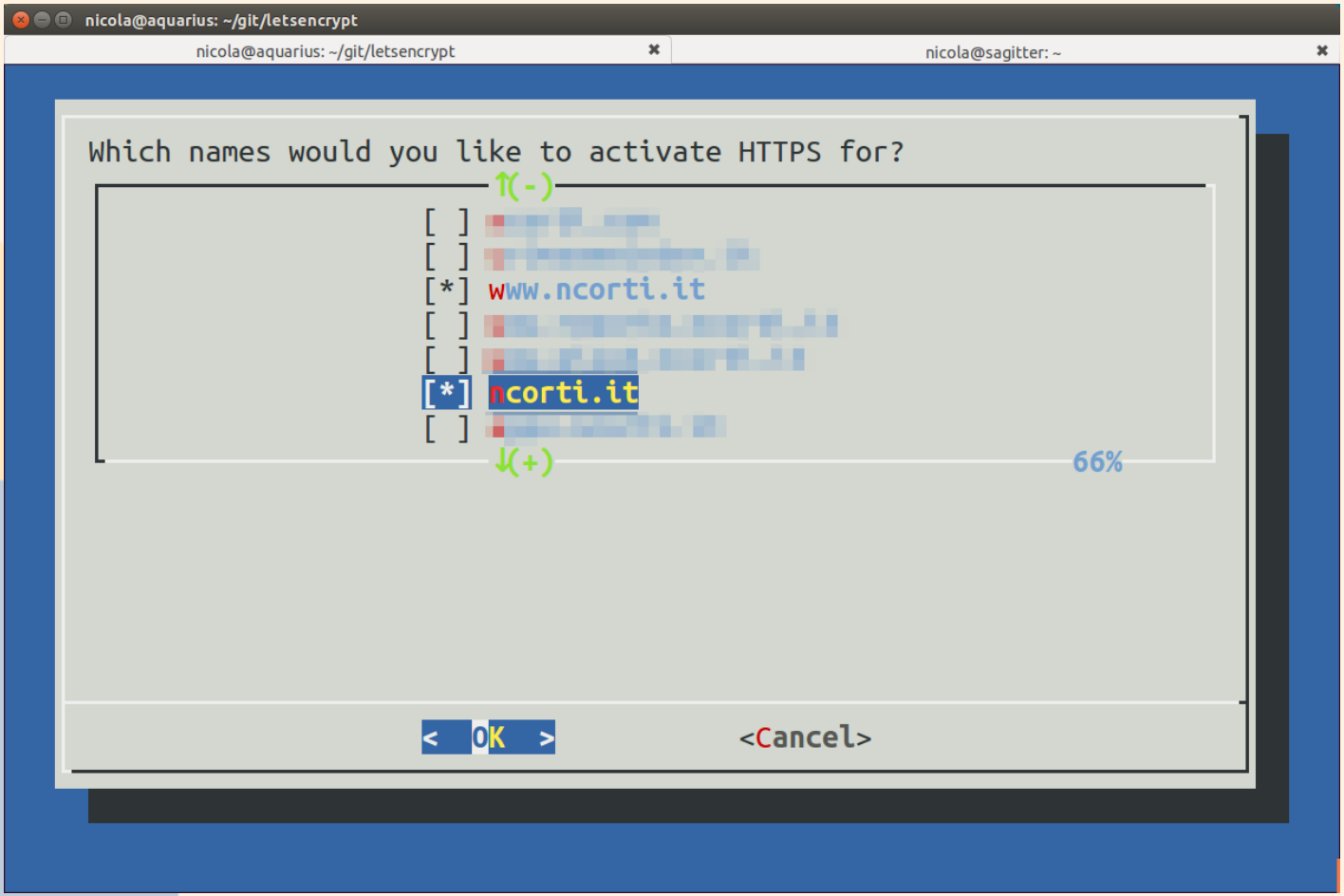
Screens


nicola@sagitter: ~/git/letsencrypt
Please enter in your domain name(s) (comma and/or space separated)
example.com
< OK >
<Cancel>

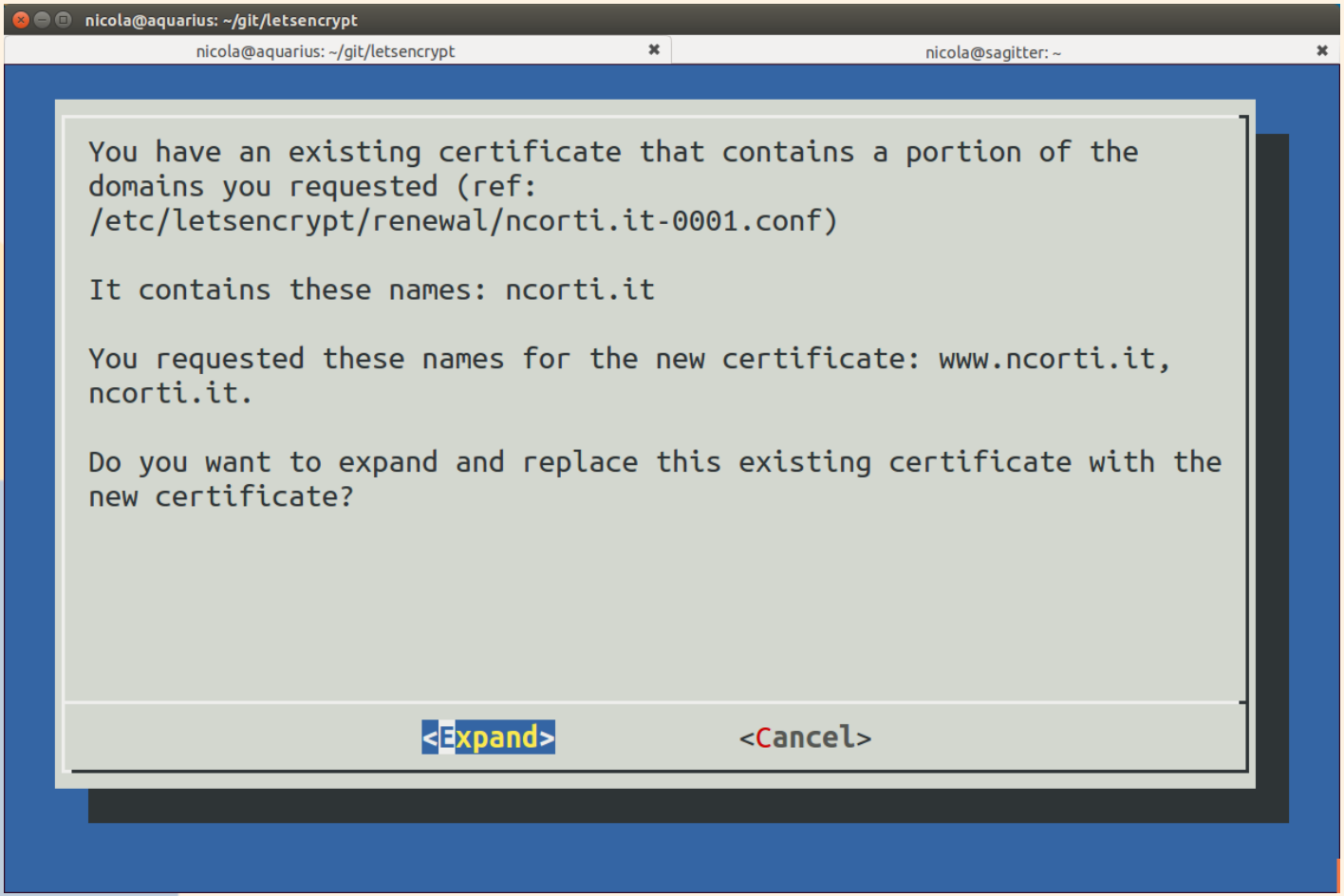
Screens


nicola@sagitter: ~/git/letsencrypt
Make sure your web server displays the following content at
http://[redacted]/.well-known/acme-challenge/S-phkbW1bo_ZS2MDmtZzyPKvOqsDpuGN0DH
9YdSshi0 before continuing:
S-phkbW1bo_ZS2MDmtZzyPKvOqsDpuGN0DH9YdSshi0.8K2NX9Rba6j23QBnmRzQOsNfrWrBM1Ur8cGV
6aY2IFc
If you don't have HTTP server configured, you can run the following
command on the target server (as root):
mkdir -p /tmp/letsencrypt/public_html/.well-known/acme-challenge
cd /tmp/letsencrypt/public_html
printf "%s" S-phkbW1bo_ZS2MDmtZzyPKvOqsDpuGN0DH9YdSshi0.8K2NX9Rba6j23QBnmRzQOsNf
rWrBM1Ur8cGV6aY2IFc > .well-known/acme-challenge/S-phkbW1bo_ZS2MDmtZzyPKvOqsDpuG
N0DH9YdSshi0
run only once per server:
$(command -v python2 || command -v python2.7 || command -v python2.6) -c \
"import BaseHTTPServer, SimpleHTTPServer; \
s = BaseHTTPServer.HTTPServer(('', 80), SimpleHTTPServer.SimpleHTTPRequestHandle
r); \
s.serve_forever()"
Press ENTER to continue

Screens


nicola@aquarius: ~/git/letsencrypt
nicola@aquarius: ~/git/letsencrypt
nicola@sagitter: ~
Which names would you like to activate HTTPS for?
↑(-)
[ ]
[ ]
[*] www.ncorti.it
[ ]
[ ]
[*] ncorti.it
[ ]
↓(+)
66%
< OK >
<Cancel>

Screens


nicola@aquarius: ~/git/letsencrypt
nicola@aquarius: ~/git/letsencrypt
nicola@sagitter: ~
You have an existing certificate that contains a portion of the domains you requested (ref: /etc/letsencrypt/renewal/ncorti.it-0001.conf)
It contains these names: ncorti.it
You requested these names for the new certificate: www.ncorti.it, ncorti.it.
Do you want to expand and replace this existing certificate with the new certificate?
<Expand>
<Cancel>

Conclusions

Drawbacks

- Nessun supporto per **Organization Validation** o **Extended Validation**, troppo complessi da automatizzare.
- Nessun supporto per le **wildcards** (*.example.com), forse in futuro.
- Challenges solo tramite HTTP (public beta), supporto DNS ancora assente.

Drawbacks

- Nessun supporto per **Organization Validation** o **Extended Validation**, troppo complessi da automatizzare.
- Nessun supporto per le **wildcards** (*.example.com), forse in futuro.
- Challenges solo tramite HTTP (public beta), supporto DNS ancora assente.

Drawbacks

- Nessun supporto per **Organization Validation** o **Extended Validation**, troppo complessi da automatizzare.
- Nessun supporto per le **wildcards** (*.example.com), forse in futuro.
- Challenges solo tramite HTTP (public beta), supporto DNS ancora assente.

Resources

- How it works **https://letsencrypt.org/howitworks/**
- Tech details **https://letsencrypt.org/howitworks/technology/**
- Read the docs **https://letsencrypt.readthedocs.org/**
- Community board **https://community.letsencrypt.org/**
- Code **https://github.com/letsencrypt/**
- Mailing lists
  - Client **https://groups.google.com/a/letsencrypt.org/forum/#!forum/client-dev**
  - Server **https://groups.google.com/a/letsencrypt.org/forum/#!forum/ca-dev**
  - ACME (IETF) **https://www.ietf.org/mailman/listinfo/acme**

Domande...?

ncorti.com

@cortinico

@cortinico