

Let's Encrypt! Certificati SSL per tutti!

Nicola Corti

Gruppo Utenti Linux Pisa



03 Febbraio 2016

[Intro](#)

[Sponsors](#)

[History](#)

[Why?](#)

[Server](#)

[Client](#)

[Protocol](#)

[Security](#)

[Challenges](#)

[The protocol](#)

[Cert revocation](#)

[Certs](#)

[DV](#)

[Cross Signing](#)

[Setup](#)

[Requirements](#)

[Download](#)

[Run](#)

[Plugins](#)

[Revoke](#)

[Renewal](#)

[Update](#)

[Screens](#)

[Conclusions](#)

[Drawbacks](#)

[Resources](#)



Introduction

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

Cosa è Let's Encrypt?



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/>

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

Chi sta supportando il progetto?

Platinum

mozilla



OVH.com

Gold



facebook

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

History

Let's Encrypt!
Certificati SSL per
tutti!

Nicola Corti

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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**

**Let's Encrypt!
Certificati SSL per
tutti!**

Nicola Corti

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources



Server

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 HTTPS che espone un'interfaccia **RESTful**

🔗 <https://github.com/letsencrypt/boulder>

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources



Client

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>

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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 server: **apache** ed **nginx**.



Let's Encrypt!
Certificati SSL per
tutti!

Nicola Corti

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

Protocol

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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

ACME si basa 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 necessariamente attraverso **HTTPS**.

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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>

[Intro](#)[Sponsors](#)
[History](#)
[Why?](#)[Server](#)[Client](#)[Protocol](#)[Security](#)
[Challenges](#)
[The protocol](#)
[Cert revocation](#)[Certs](#)[DV](#)
[Cross Signing](#)[Setup](#)[Requirements](#)
[Download](#)
[Run](#)
[Plugins](#)
[Revoke](#)
[Renewal](#)
[Update](#)
[Screens](#)[Conclusions](#)[Drawbacks](#)
[Resources](#)

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*

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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 desiderato, 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**.

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

Challenges

Let's Encrypt!
Certificati SSL per
tutti!

Nicola Corti

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 <i>token</i> 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). |

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

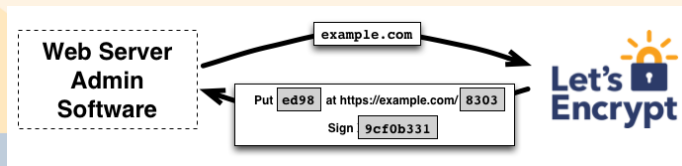
Drawbacks

Resources

Domain Validation

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti



Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

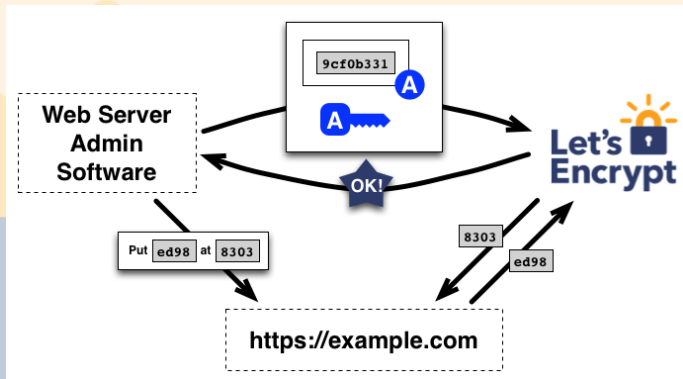
Drawbacks

Resources

Domain Validation

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti



Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

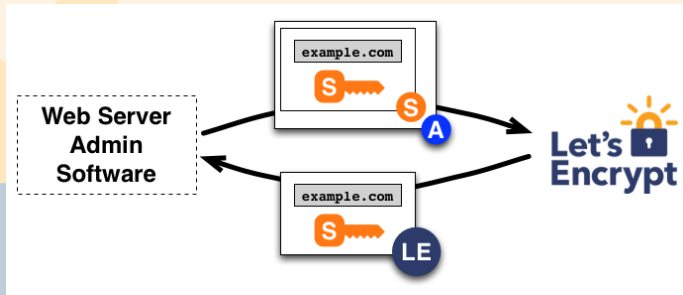
Drawbacks

Resources

Certificate Issuance

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti



Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

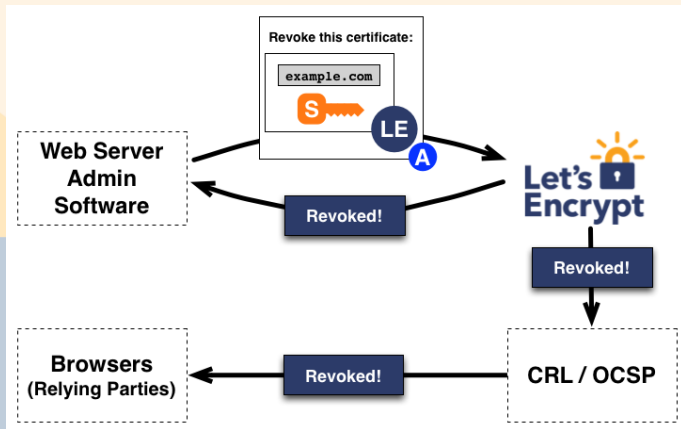
Drawbacks

Resources

Certificate Revocation

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti



Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources



Certificates

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

DV Certificates

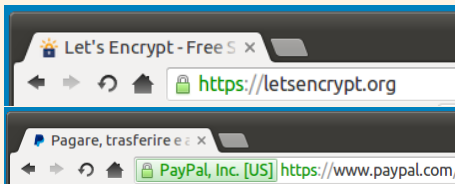
Let's Encrypt!
Certificati SSL per
tutti!

Nicola Corti

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.



Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

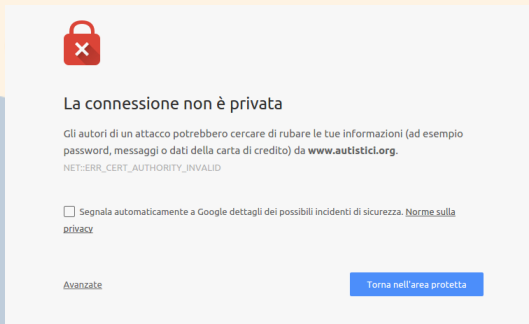
Drawbacks

Resources

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.

In questo modo possiamo evitare errori quali:



Let's Encrypt!
Certificati SSL per
tutti!

Nicola Corti

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

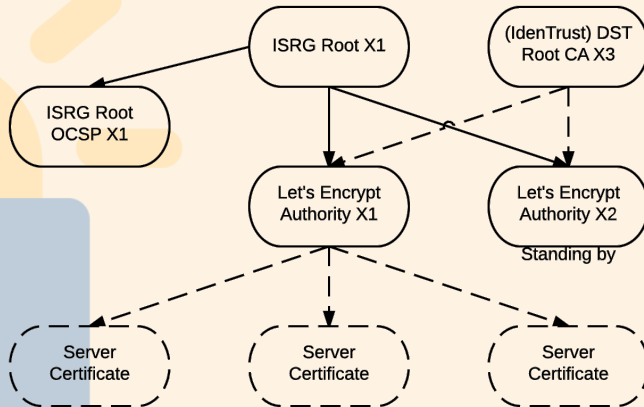
Screens

Conclusions

Drawbacks

Resources

Cross Signing



Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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**.

Let's Encrypt!
Certificati SSL per
tutti!

Nicola Corti

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

Setup

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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 OS 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.

Run

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

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

E verremo guidati durante il processo di issuing dei certificati.

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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.

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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
```

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

Plugins

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti

| 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 <i>standalone</i> sulla porta 80/443 |
| Webroot | Y | N | Ottiene il certificato andando a scrivere all'interno della <i>root</i> 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) |

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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

[Intro](#)[Sponsors](#)[History](#)[Why?](#)[Server](#)[Client](#)[Protocol](#)[Security](#)[Challenges](#)[The protocol](#)[Cert revocation](#)[Certs](#)[DV](#)[Cross Signing](#)[Setup](#)[Requirements](#)[Download](#)[Run](#)[Plugins](#)[Revoke](#)[Renewal](#)[Update](#)[Screens](#)[Conclusions](#)[Drawbacks](#)[Resources](#)

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.

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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.

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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

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

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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 schedare un task su **cron** per il rinnovo automatico dei certificati prima dei 90 giorni.

Let's Encrypt!
Certificati SSL per
tutti!

Nicola Corti

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

Update

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

Per Debian *sid/stretch*:

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

Per tutti gli altri:

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

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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 seguirà 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**.

Intro

- Sponsors
- History
- Why?

Server

Client

Protocol

- Security
- Challenges
- The protocol
- Cert revocation

Certs

- DV
- Cross Signing

Setup

- Requirements
- Download
- Run
- Plugins
- Revoke
- Renewal
- Update
- Screens

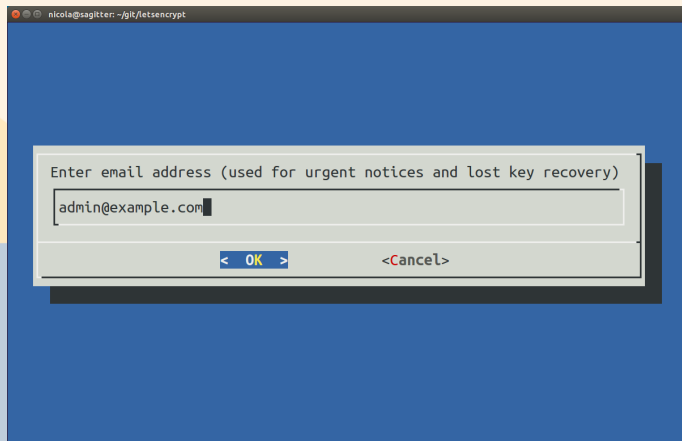
Conclusions

- Drawbacks
- Resources

Screens

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti



Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

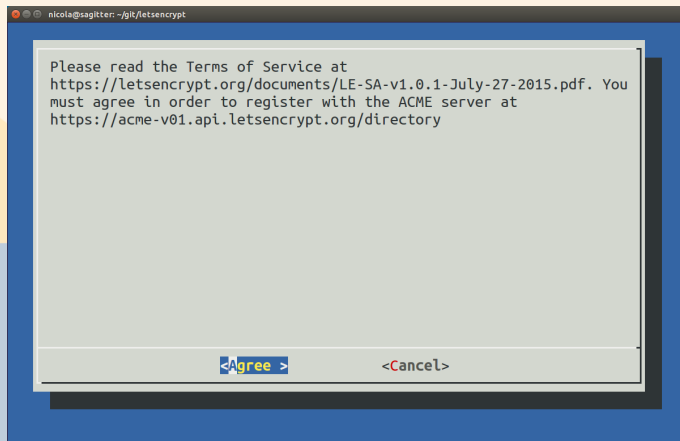
Drawbacks

Resources

Screens

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti



Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

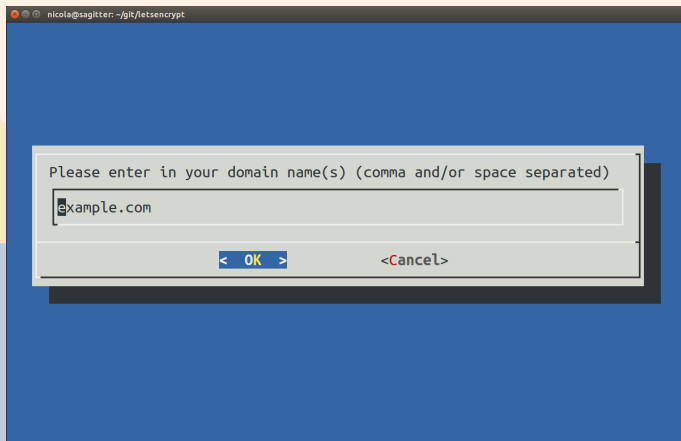
Drawbacks

Resources

Screens

**Let's Encrypt!
Certificati SSL per
tutti!**

Nicola Corti



Intro

[Sponsors](#)

[History](#)

[Why?](#)

Server

Client

Protocol

[Security](#)

[Challenges](#)

[The protocol](#)

[Cert revocation](#)

Certs

[DV](#)

[Cross Signing](#)

Setup

[Requirements](#)

[Download](#)

[Run](#)

[Plugins](#)

[Revoke](#)

[Renewal](#)

[Update](#)

[Screens](#)

Conclusions

[Drawbacks](#)

[Resources](#)

Screens

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti

```
nicola@sagitter: ~/git/letsencrypt

Make sure your web server displays the following content at
http://[redacted].well-known/acme-challenge/S-phkbw1bo_ZS2MDmtZzyPKv0qsDpuGN0DH9YdSshi0 before continuing:

S-phkbw1bo_ZS2MDmtZzyPKv0qsDpuGN0DH9YdSshi0.8K2NX9Rba6j23QBnmRzQ0sNfrWrBM1Ur8cGV6aY2IFc

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_ZS2MDmtZzyPKv0qsDpuGN0DH9YdSshi0.8K2NX9Rba6j23QBnmRzQ0sNfrWrBM1Ur8cGV6aY2IFc > .well-known/acme-challenge/S-phkbw1bo_ZS2MDmtZzyPKv0qsDpuGN0DH9YdSshi0
# 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.SimpleHTTPRequestHandler; \
s.serve_forever()"
Press ENTER to continue
```

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

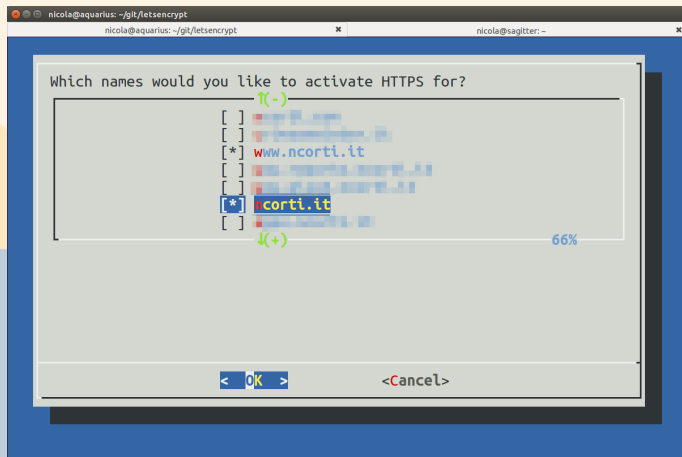
Drawbacks

Resources

Screens

Let's Encrypt!
Certificati SSL per tutti!

Nicola Corti

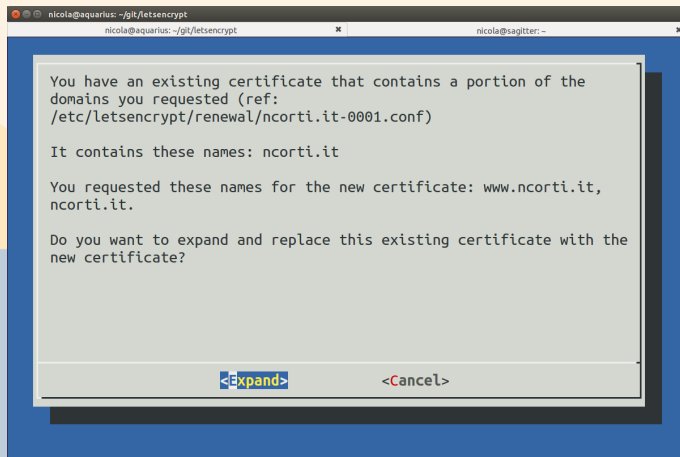


Setup

Screens

**Let's Encrypt!
Certificati SSL per
tutti!**

Nicola Corti



Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

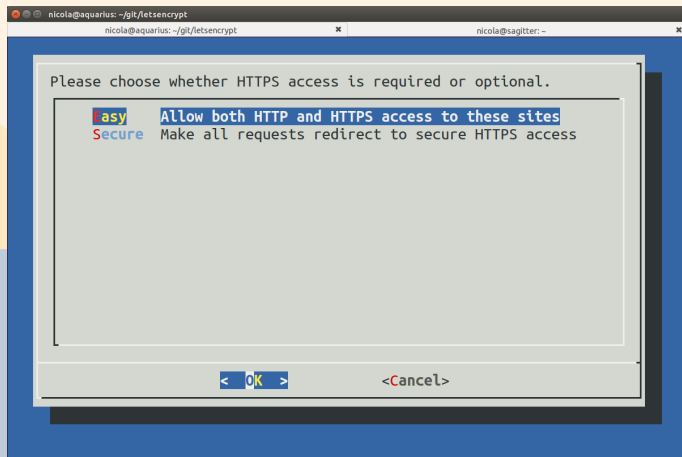
Drawbacks

Resources

Screens

Let's Encrypt!
Certificati SSL per
tutti!

Nicola Corti



Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

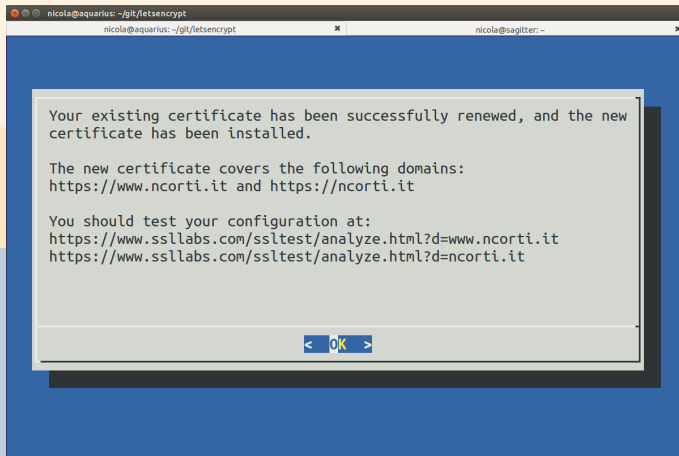
Drawbacks

Resources

Screens

**Let's Encrypt!
Certificati SSL per
tutti!**

Nicola Corti



Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources



Conclusions

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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.

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources

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**

Slides realizzate con \LaTeX Beamer.

La seguente presentazione è rilasciata sotto licenza

Creative Commons - Attributions, Non Commercial, Share-alike.



Sorgenti su **<https://github.com/cortinico/gulp-letsencrypt>**

Intro

Sponsors

History

Why?

Server

Client

Protocol

Security

Challenges

The protocol

Cert revocation

Certs

DV

Cross Signing

Setup

Requirements

Download

Run

Plugins

Revoke

Renewal

Update

Screens

Conclusions

Drawbacks

Resources