Let’s encrypt! Certificati SSL per tutti!

Nicola Corti

Gruppo Utenti Linux Pisa

02 Febbraio 2016

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**

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

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**.

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*

ACME

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**.

ACME

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**.

ACME

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.

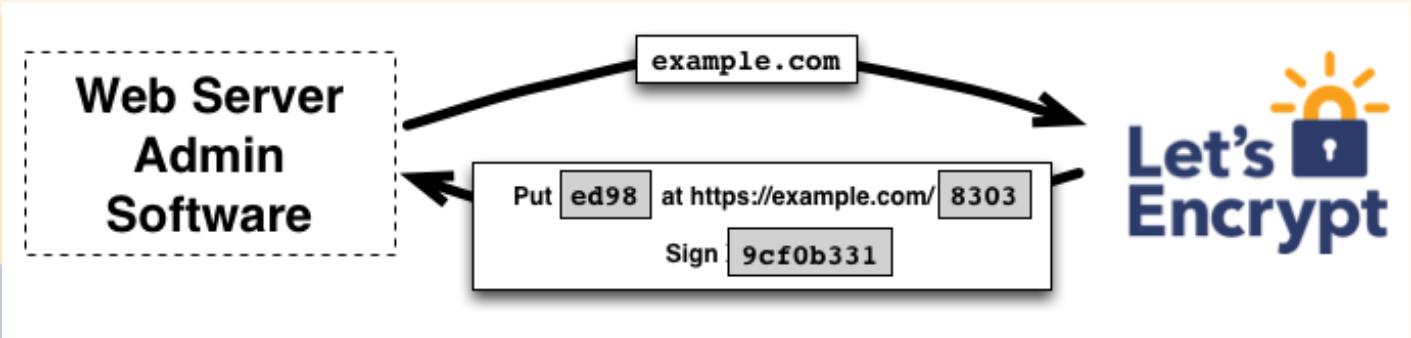
Tipo	Descrizione
Simple HTTP	Si chiede di posizionare un **file** all'interno del proprio webserver contenente un *token* specifico (Accettato sia HTTP che HTTPS).
DNS	Si chiede di inserire un **record TXT** all'interno del proprio DNS.
Proof of possession	Si chiede di **firmare un documento** utilizzando 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

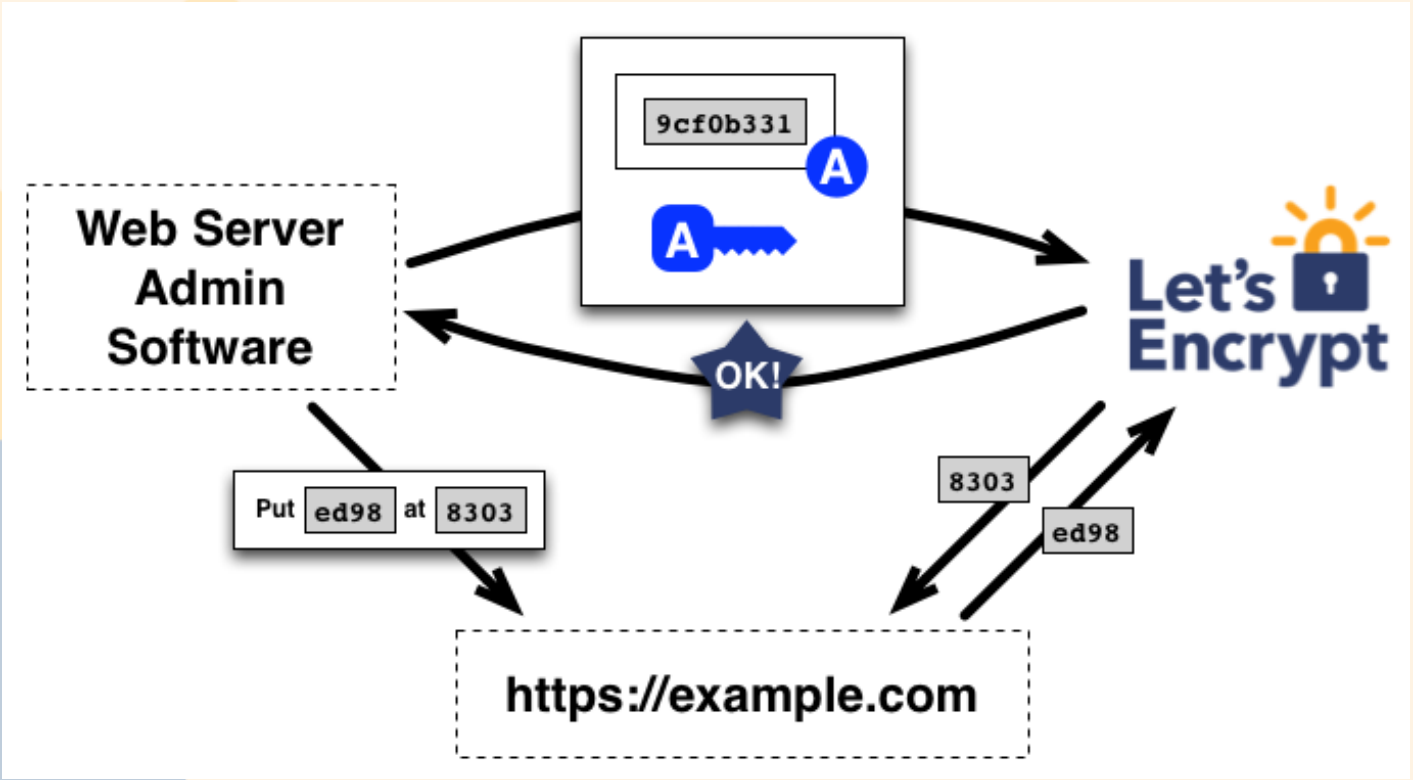
Il server può, a sua discrezione, inviare una o più challenges.

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).

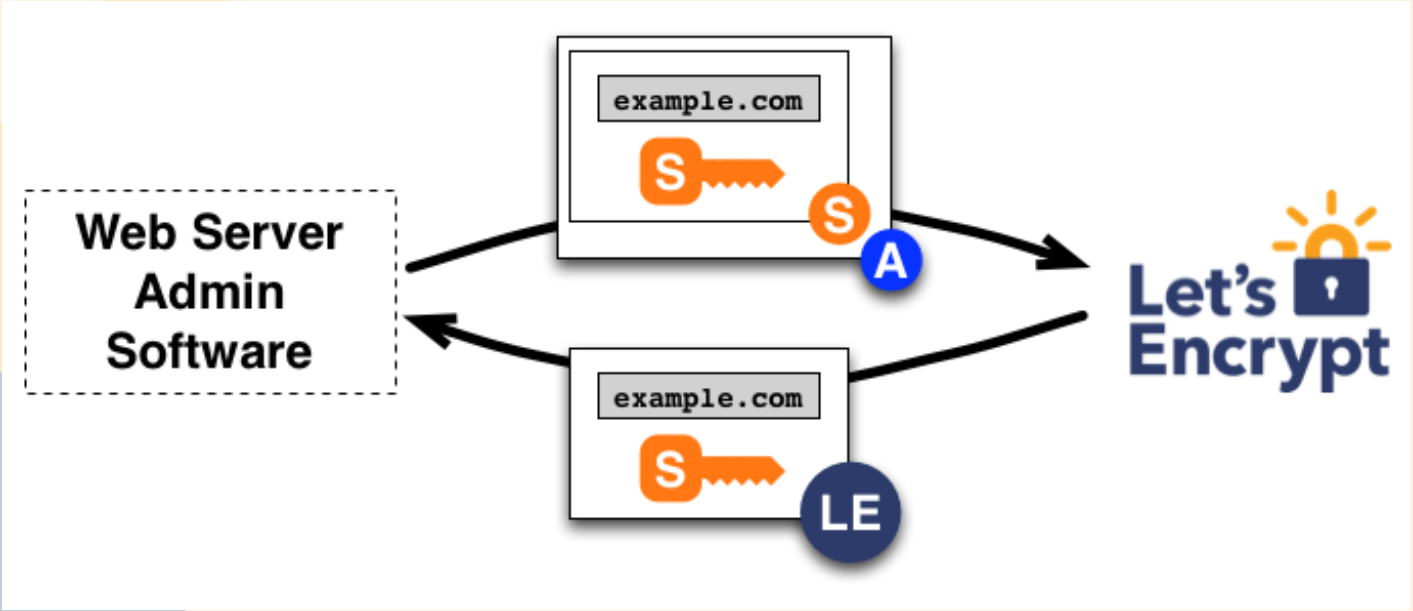
Domain Validation



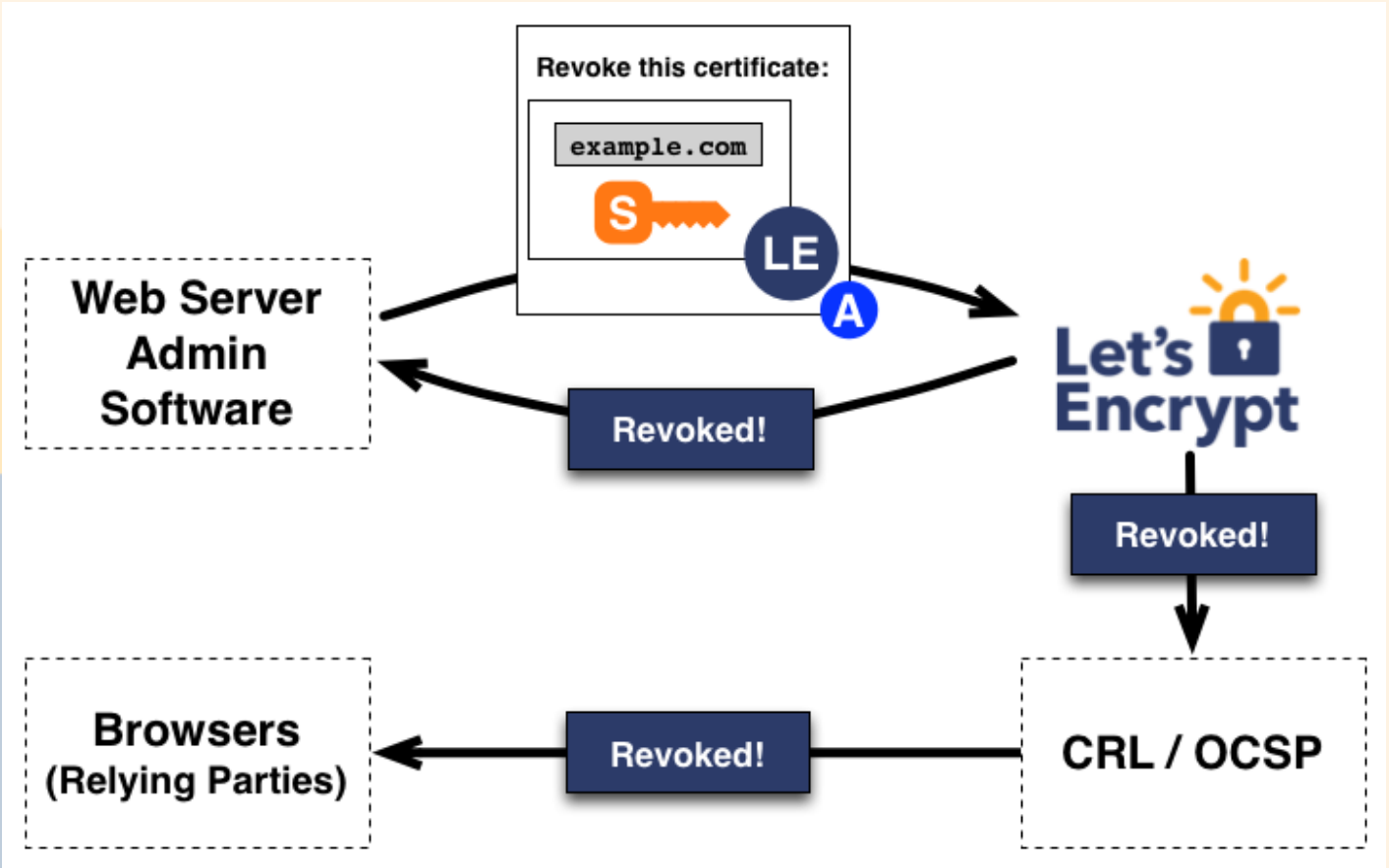
Domain Validation



Certificate Issuance



Certificate Revocation



DV Certificates

DV

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

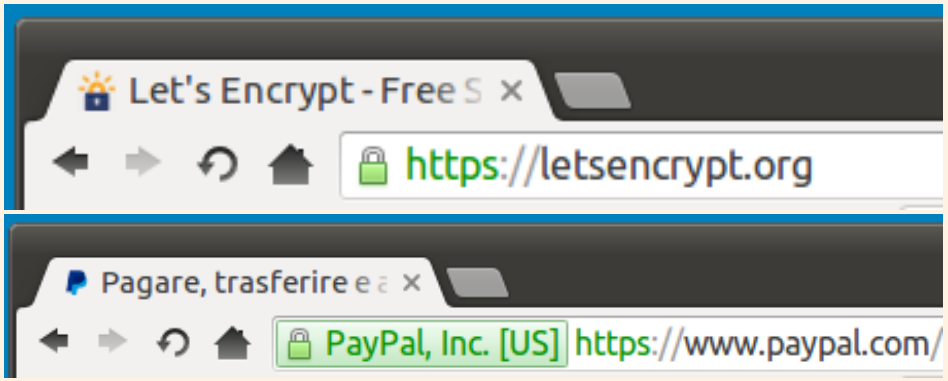
Certificati di tipo **Organization Validation** oppure **Extended Validation** richiederebbero verifiche esplicite del soggetto che fa richiesta del certificato.

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.

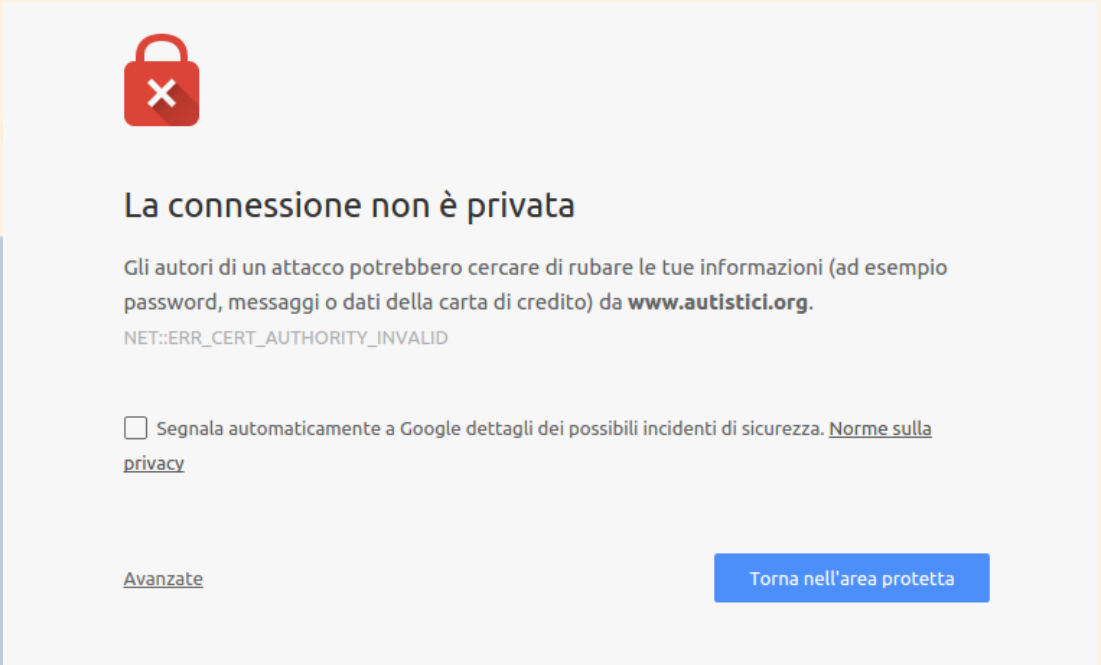


Cross Signing

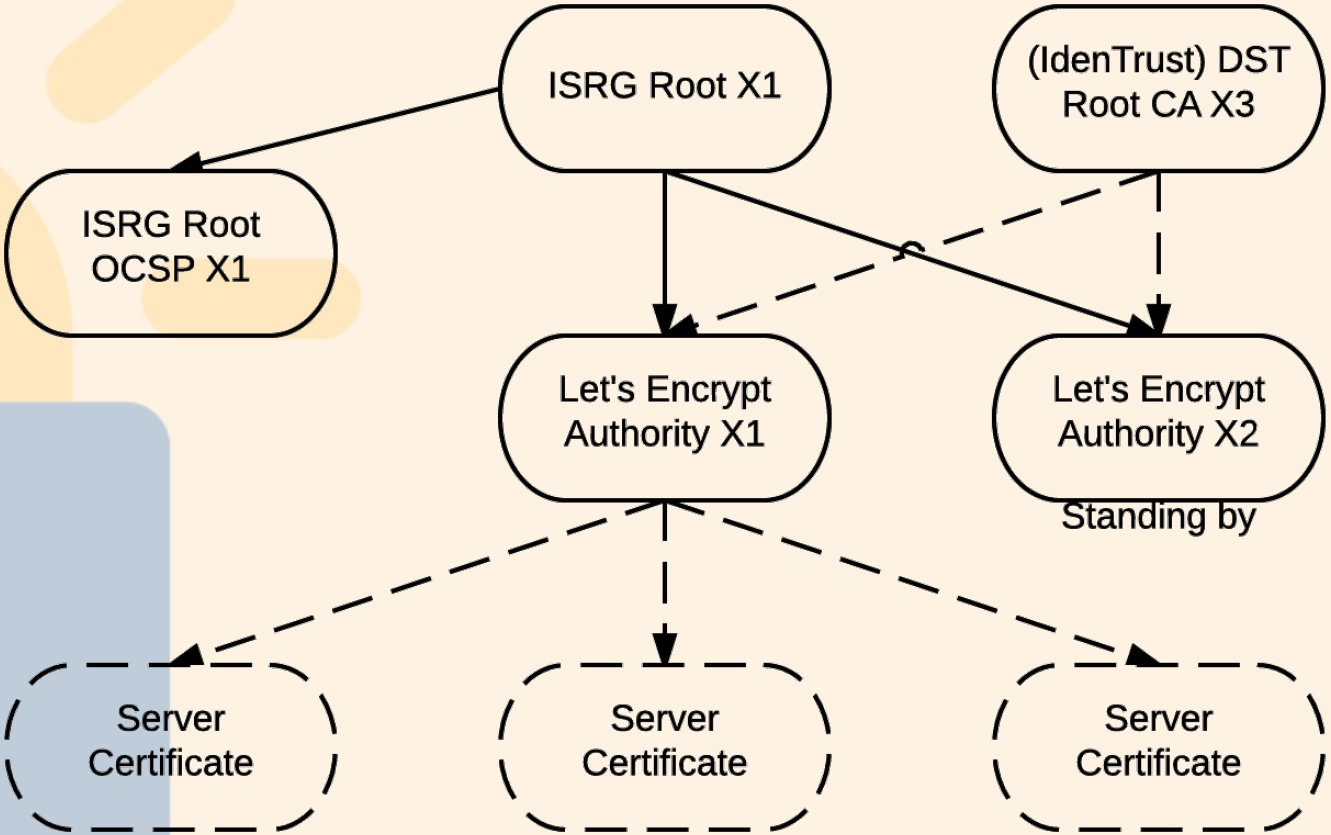
Tutti i certificati rilasciati sono *Cross-signed* da parte di **IdenTrust**. Questo permette ai certificati di L.E. di essere riconosciuti dalla maggioranza dei browser.

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**.

Domande...?

Slides realizzate da:
Nicola Corti - corti.nico [at] gmail [dot] com

Slides realizzate con LaTeX Beamer.