Torino Technologies Group

ICT POWER.IT

Incontro TTG 13 dicembre 2018

Public Key Infrastructure Windows e Let's Encrypt

Ermanno Goletto

MVP Cloud and Datacenter Management

@ermannog

www.devadmin.it

Roberto Massa

MVP Cloud and Datacenter Management

@robi_massa

massarobi.wordpress.com

Agenda

ICT POWER.IT

- Architettura di una PKI
- Installazione PKI a due livelli in Windows Server 2016
- Utilizzo di Let's Encrypt in Windows Server 2016

Torino Technologies Group

ICT POWER.IT

Architettura di una PKI

Public Key Infrastructure Windows e Let's Encrypt

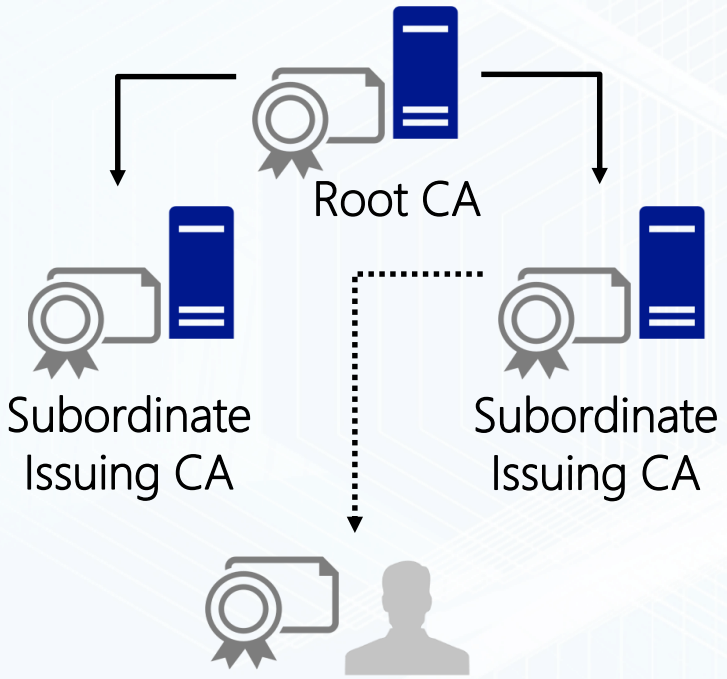
Public Key Infrastructure (PKI)

Una *Infrastruttura a Chiave Pubblica*, in *inglese Public Key Infrastructure (PKI)*, è un *insieme di processi e mezzi tecnologici* che consentono a terze parti fidate di *emettere*, *verificare chiavi pubbliche* (certificati digitali), e di *associarle ad un titolare*

Una infrastruttura PKI è *strutturata gerarchicamente da più CA e al vertice si trova una Root CA* che *certifica le Subordinate CA e firma il proprio certificato* (self-signed), mentre le *Subordinate CA rilasciano certificati per usi specifici* (Issuing CA)

Una *Certification Authority (CA)* è un *soggetto terzo di fiducia* (trusted third party), pubblico o privato, *abilitato ad emettere un certificato digitale*

La *CA ha anche il compito di revocare i certificati* e di pubblicare un elenco di revoche dei certificati (CRL)



La CA *utilizza la crittografia a doppia chiave, o asimmetrica*, in cui una delle due chiavi detta *Chiave Pubblica* viene resa pubblica all'interno del certificato, mentre la seconda detta *Chiave Privata*, univocamente correlata con la prima, rimane segreta e associata al titolare

Tipo CA in ambiente Windows

- Una *CA Stand-alone non richiede l'integrazione con Active Directory*, ma *più complessa da amministrare* in quanto non consente l'utilizzo dei modelli di certificato
- Una *CA Enterprise deve essere integrate in Active Directory*, ma *la gestione è più semplice* grazie ai modelli di certificato

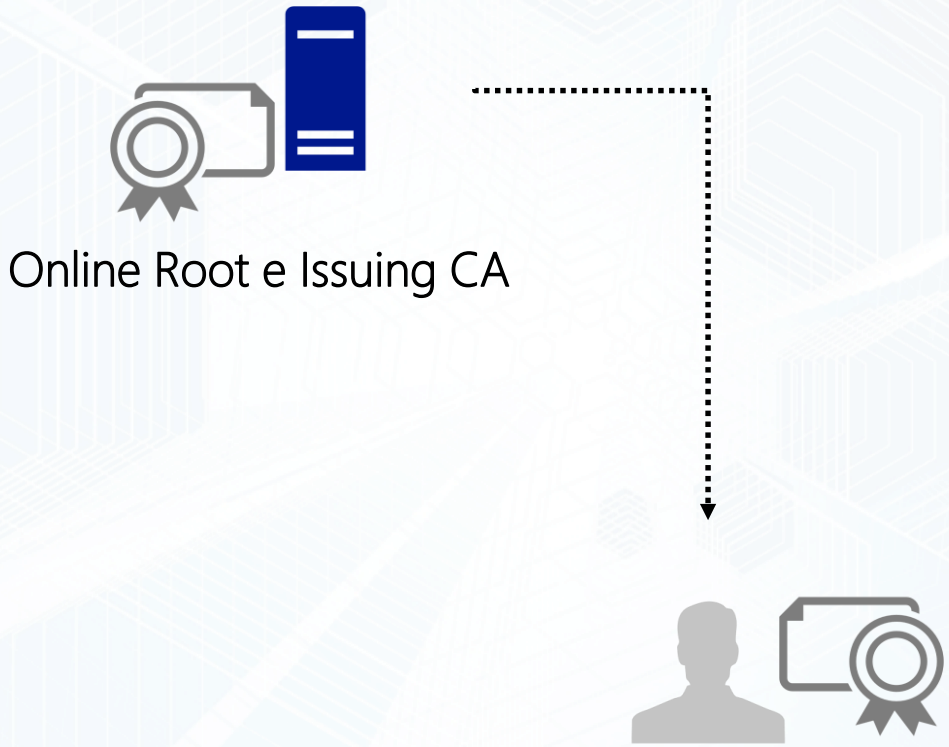
La CA dispone di un certificato con il quale sono firmati tutti i certificati emessi agli utenti e quindi *deve essere messa in sicurezza*

Certification Authority Guidance: https://technet.microsoft.com/en-us/library/hh831574(v=ws.11).aspx

Gerarchia PKI a livello singolo

Single/One-Tier Hierarchy

- *Singola CA* che è sia root CA che issuing CA
- *Non è mai raccomanda in produzione* perché la compromissione della CA corrisponde alla compromissione dell'intera PKI
- Dal momento che la CA è sempre online è *più suscettibile alla compromissione*
- Se la CA è compromessa *non è possibile revocarla*

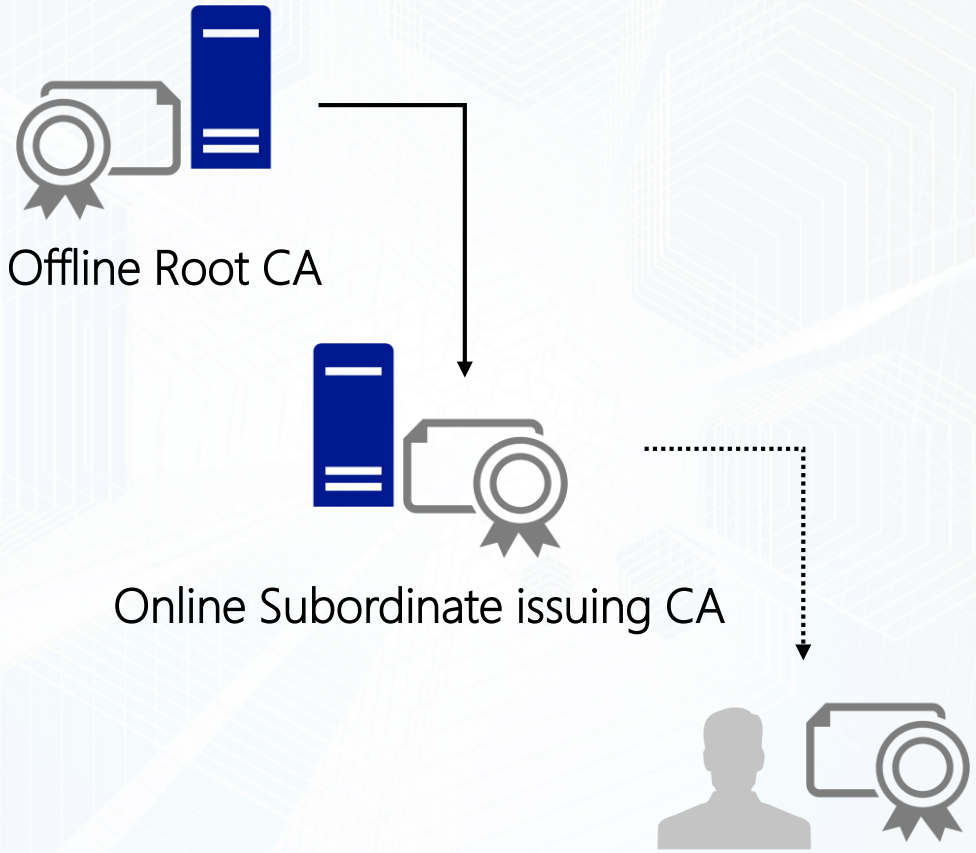


Planning a CA Hierarchy: https://technet.microsoft.com/en-us/library/dn786436(v=ws.11).aspx

Gerarchia PKI a due livelli

Two-Tier Hierarchy

- *Root CA offline* per garantire una *migliore protezione della chiave privata* da tentativi di compromissione
- *Subordinate issuing CA online* per *aumentare il livello di sicurezza tramite la separazione dei ruoli della Root CA e della Issuing CA*
- Offre *maggiore scalabilità e flessibilità*, grazie alla *possibilità di usare più Subordinate issuing CA*, ad esempio in aree geografiche diverse o per livelli di sicurezza differenti
- Comporta una *gestione maggiore* in *quanto la Root CA deve essere portata online per la firma della CRL*
- *Soddisfa le esigenze della maggior parte delle aziende*
- *Microsoft IT* ha adottato una gerarchia di CA a due livelli per la propria infrastruttura PKI interna

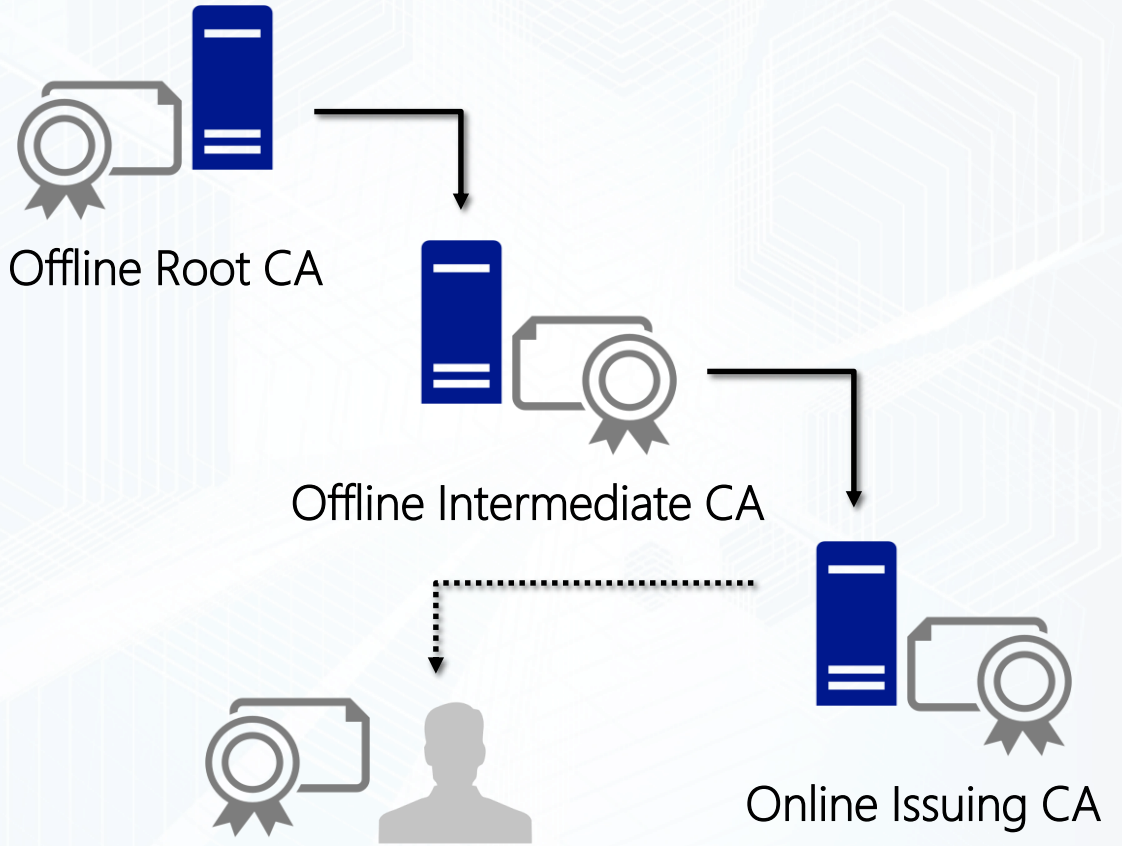


Planning a CA Hierarchy: https://technet.microsoft.com/en-us/library/dn786436(v=ws.11).aspx

Gerarchia PKI a tre livelli

Three-Tier Hierarchy

- *Root CA offline* per garantire una *migliore protezione della chiave privata* da tentativi di compromissione
- *Intermediate CA utilizzabile come policy CA*, per esempio, *per rilasciare certificati all'Issuing CA che è configurata per rilasciare solo alcuni tipi di certificati* (ad esempio potrebbero non rilasciare certificati che richiedono a un utente di presentarsi di persona)
- Le *Intermediate CA consentono la revoca di Issuing CA in caso compromissione*, per questo motivo le Intermediate CA devono essere mantenute offline
- L'aggiunta di un secondo livello offre *maggiore sicurezza, scalabilità e flessibilità*, ma *aumenta la gestione e i costi e diminuisce le prestazioni* della creazione di una catena di certificati nei client
- *Non è consigliata a meno che non si intenda implementare policy amministrative sul rilascio dei certificati*
- *Let's Encrypt* è basata su una gerarchia di CA a tre livelli



Planning a CA Hierarchy: https://technet.microsoft.com/en-us/library/dn786436(v=ws.11).aspx

Bad & good practices

Enterprise Root CA su un DC Online
Aumento della gestione e del restore del DC, la Root CA è online e quindi più esposta alle compromissioni

Enterprise Root CA Online
La Root CA è online e quindi più esposta alle compromissioni

Enterprise Root CA Offline
Non è raccomandata, si può incorrere in difficoltà amministrative e malfunzionamenti

One-tier hierarchy
Non è raccomandata perché la compromissione della CA corrisponde alla compromissione dell'intera PKI

Stand-alone Offline Root CA
Aumenta la sicurezza riducendo il rischio di compromissione

Two-tier CA hierarchy
Buon compromesso tra sicurezza, semplicità e scalabilità della PKI

Three-tier CA hierarchy
Architettura consigliata se è necessario implementare policy per il rilascio di certificati

Hardware Security Module (HSM)
Store hardware sicuro per le chiavi private della CA indipendente dall'OS

Public Key Infrastructure Design Guidance: http://social.technet.microsoft.com/wiki/contents/articles/2901.pki-design-guidance.aspx
Securing Public Key Infrastructure (PKI): https://technet.microsoft.com/en-us/library/dn786443(v=ws.11).aspx

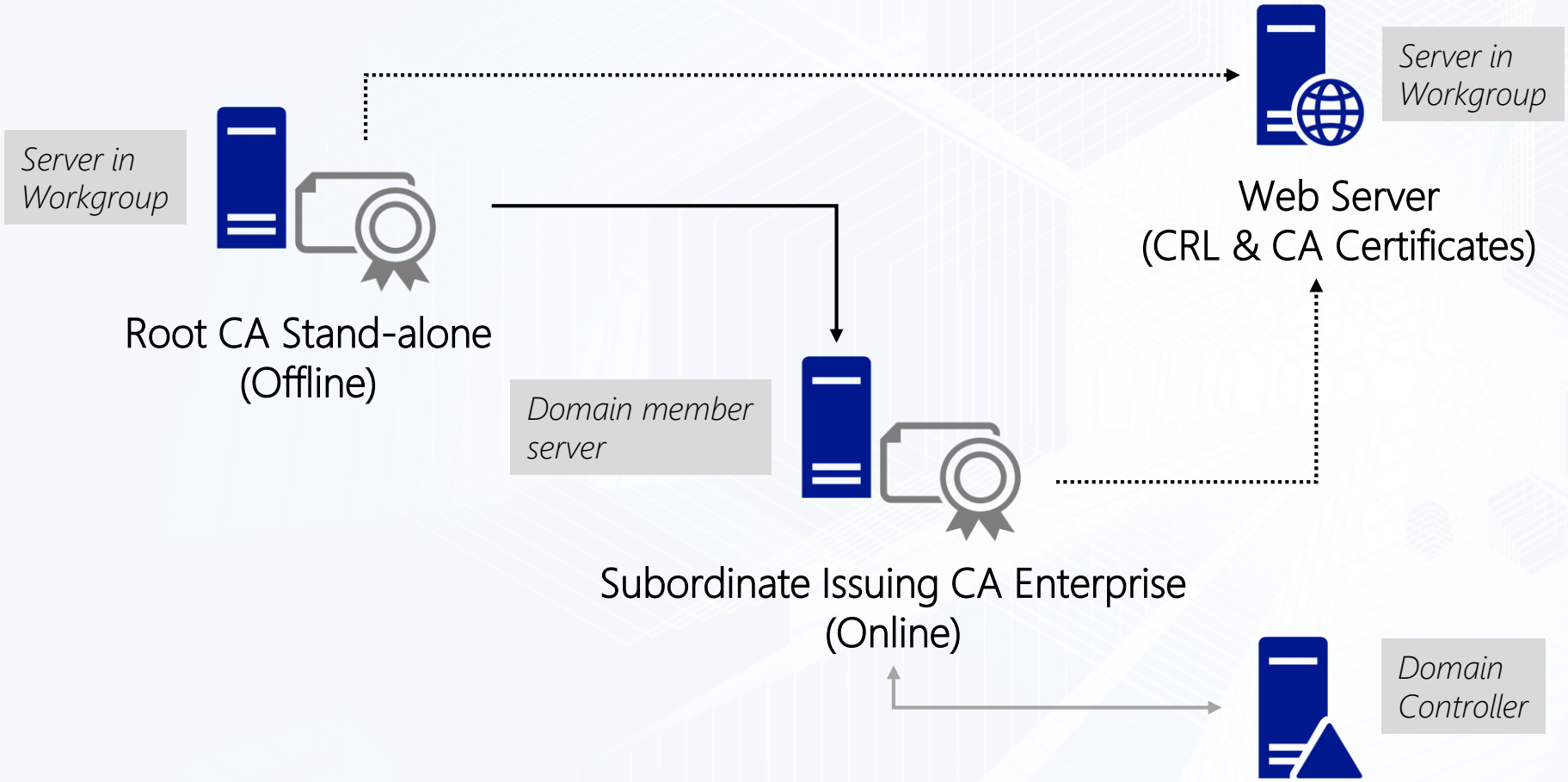
Torino Technologies Group

ICT POWER.IT

Installazione PKI a due livelli in Windows Server 2016

Public Key Infrastructure Windows e Let's Encrypt

Architettura di una PKI a due livelli



Server Web

Utilizzo

Pubblicare la CRL e i certificati delle CA necessari nei casi in cui i certificati distribuiti non contengano la catena dei certificati

Configurazione

- *Impostazione del server in Workgroup* e *aggiunta del ruolo Web Server (IIS)*
- Configurazione di un *record DNS di tipo A relativo all'Hostname*
- Configurazione di un *record DNS di tipo CNAME per la pubblicazione della CRL* per evitare che l'URL della CRL faccia riferimento all'hostname del server:
  - *Per consentire la configurazione dell'alta disponibilità* della CRL
  - *Per ragioni di sicurezza* in modo da evitare di diffondere informazioni sull'infrastruttura interna
- *Creazione di una cartella dedicata a contenere le CRL e i certificati delle CA in disco dedicato* per ragioni di sicurezza
- *Creazione di un account locale* che non sia membro di alcun gruppo locale a cui condividere tale cartella in scrittura, che verrà *utilizzato dalle CA per caricare le CRL e i certificati delle CA* in modo che il server web li possa pubblicare
- *Configurazione in IIS una Virtual Directory* nel Default Web Site *che punta alla cartella della CRL e dei certificati delle CA con accesso anonimo in sola lettura*
- *Configurazione delle Regole di Filtro per la Virtual Directory per consentire il double escaping* in modo da evitare che il carattere "+" contenuto nel URI delle delta CRL venga bloccato

Per pubblicare la CRL e il Certificato della CA non è possibile utilizzare HTTPS

https://www.ictpower.it/sistemi-operativi/deploy-pki-in-windows-server-2016-parte-1-architettura-di-una-pki-two-tier.htm

Demo

Torino
Technologies
Group

ICT POWER.IT

Installazione e configurazione Server Web

Installazione Root CA Offline

Architettura

- *Mantenuta offline*, quindi è *preferibile che non sia integrata in Active Directory* e quindi dovrà essere *Stand-alone*
- *Per sicurezza non dovrebbe essere connessa alla rete* neppure quando viene avviata saltuariamente per installare gli aggiornamenti automatici, ma attestata su una rete separata che consenta la comunicazione con l'infrastruttura solo per permettere l'aggiornamento della CRL e del certificato della CA sul server web

Configurazione

- *Impostazione del server in Workgroup* e aggiunta del *ruolo Active Directory Certificate Services* selezionando il *servizio Certification Autority*
- Configurazione della *CA di tipo Root* specificado *impostazioni per la creazione della chiave privata che garantiscano il maggior livello di sicurezza* implementabile nell'infrastruttura informatica (xes: lunghezza chiave 4096 e Algoritmo hash per firma certificati SHA512) con validità congrua al rischio di compromissione per brute force (xes 20 anni)
- Per sicurezza *il nome della CA non deve fare riferimento all'hostname e il database dei certificati risiederà in un disco separato*
- *Configurazione Estensioni per impostare i CRL* (Certificate Revocation List) *Distribution Points* (CDP) che *indicano ai client dove trovare le CRL aggiornate firmate dalla CA*, tali informazioni sono anche inserite nei certificati generati
- *Configurazione Estensioni per impostare gli Access Information Access* (AIA) *Distribution Points* (CDP) che *indicano ai client dove trovare le CRL aggiornate firmate dalla CA*, tali informazioni sono anche inserite nei certificati generati

https://www.ictpower.it/sistemi-operativi/deploy-pki-in-windows-server-2016-parte-2-installazione-e-configurazione-di-una-root-ca-offline.htm

Configurazione Root CA Offline

Configurazione CRL

- Impostare l'*intervallo di pubblicazione della CRL* ad un *periodo superiore a quello a cui si prevede di avviare periodicamente la Root CA Offline* che normalmente sarà mantenuta spenta salvo quando vengono installati per gli aggiornamenti del sistema
- Ipotizzando un *avvio settimanalmente della Root CA per installare gli aggiornamenti* un *intervallo di pubblicazione adeguato* potrebbe essere di *4 settimane*
- *Avvio pubblicazione della CRL* tramite la voce del menu contestuale All Tasks\Pubblish dei Certificati revocati, oppure mediante il comando: *certutil –CRL*
- Verificare che la CLR sia stata memorizzata in *%SystemRoot%\System32\CertSrv\CertEnroll*

Configurazione della validità dei certificati emessi

- *Per default la validità dei certificati emessi da una CA Stand-alone è di un anno*, ma *modificabile tramite certutil* che modifica le impostazioni relative nel registry
- *Ipotizzando che la durata del certificato della Root CA sia di 20 anni la validità dei certificati emessi può essere impostata a 10 anni* prevedendo quindi di rinnovarli ogni 10 anni mediante il comando *certutil -setreg ca\ValidityPeriodUnits "10"* che imposta la registry key *HKLM\System\CurrentControlSet\Configuration\CAName\ValidityPeriodUnitis*

Al termine della configurazione riavviare il servizio della CA (certsvc)

https://www.ictpower.it/sistemi-operativi/deploy-pki-in-windows-server-2016-parte-2-installazione-e-configurazione-di-una-root-ca-offline.htm

Pubblicazione CRL Root CA Offline

Copia della CRL e del certificato della CA sul Web Server

- *L'accesso al Web Server da parte della Root CA Offline avviene tramite un account locale creato sul Web Server* che ha privilegi di lettura e scrittura su una share che punta alla cartella sul Server Web che conterrà le CRL e i certificati delle CA
- *Memorizzare sulla Root CA l'account che verrà utilizzato per accedere alle share sul Web Server*
- Creare un'*operazione schedulata per aggiornare la CRL e copiare sul Web Server la CRL e il certificato CA della Root CA*

```
REM Aggiornamento della CRL
certutil –CRL

REM della CRL e il certificato CA della Root CA sul Web Server
copy /Y “%SytemRoot%\System32\CertSrv\CertEnroll\*.crl” \\webserver.domain.ext\CertEnrollShare
copy /Y “%SytemRoot%\System32\CertSrv\CertEnroll\*.crt” \\webserver.domain.ext\CertEnrollShare
```

https://www.ictpower.it/sistemi-operativi/deploy-pki-in-windows-server-2016-parte-2-installazione-e-configurazione-di-una-root-ca-offline.htm

Demo

Torino Technologies Group

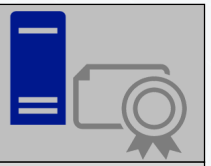
ICT POWER.IT

Installazione e configurazione Root CA Offline

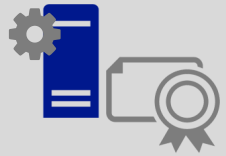
Installazione Subordinate CA

Architettura

Sarà di tipo *Enterprise* per essere *utilizzata in modo integrato con Active Directory* e *utilizzare i modelli di certificato*



Configurazione



- Impostazione del server come *membro del dominio* e aggiunta del *ruolo Active Directory Certificate Services* selezionando il *servizio Certification Autority* utilizzando le credenziali di un *account membro del gruppo Enterprise Admin*
- Configurazione della *CA di tipo Subordinate* specificando delle *impostazioni per la creazione della chiave privata che garantiscano il maggior livello di sicurezza* implementabile nell'infrastruttura informatica (xes: lunghezza chiave 4096 e Algoritmo hash per firma certificati SHA512)

- Per sicurezza *il nome della CA non deve fare riferimento all'hostname* e *il database dei certificati risiederà in un disco separato*
- Creazione di una *richiesta di certificato per il certificato della Subordinate CA* che dovrà poi essere *emesso dalla Root CA, esportato in formato PKCS #7* (.P7B) e *installato sulla Subordinate CA*

- La durata del certificato della CA sarà pari alla durata dei certificati emessi dalla Root CA (10 anni in base alla configurazioni di esempio per la Root CA)
- *Configurazione Estensioni per impostare i CRL* (Certificate Revocation List) *Distribution Points* (CDP) che *indicano ai client dove trovare le CRL aggiornate firmate dalla CA*, tali informazioni sono anche inserite nei certificati generati

- *Configurazione Estensioni per impostare gli Access Information Access* (AIA) *Distribution Points* (CDP) che *indicano ai client dove trovare le CRL aggiornate firmate dalla CA*, tali informazioni sono anche inserite nei certificati generati

https://www.ictpower.it/sistemi-operativi/deploy-pki-in-windows-server-2016-parte-3-installazione-subordinate-ca.htm

Configurazione Subordinate CA

Configurazione CRL

- *Avvio pubblicazione della CRL* tramite la voce del menu contestuale All Tasks\Pubblish dei Certificati revocati, oppure mediante il comando: *certutil –CRL*
- Verificare che la CLR sia stata memorizzata in *%SystemRoot%\System32\CertSrv\CertEnroll*

Configurazione della validità dei certificati emessi

- *Per default la validità dei certificati emessi da una CA Enterprise è di due anni*, ma *modificabile tramite certutil* che modifica le impostazioni relative nel registry
- *Ipotizzando che la durata del certificato della Subordinate CA sia di 10 anni la validità dei certificati emessi può essere impostata a 5 anni* prevedendo quindi di rinnovarli ogni 5 anni mediante il comando *certutil -setreg ca\ValidityPeriodUnits "5"* che imposta la registry key *HKLM\System\CurrentControlSet\Configuration\CAName\ValidityPeriodUnitis*
- *Verificare che la PKI funzioni correttamente*, uno strumento utile per eseguire tale verifica è il tool *pkiview.msc*

Al termine della configurazione riavviare il servizio della CA (certsvc)

https://www.ictpower.it/sistemi-operativi/deploy-pki-in-windows-server-2016-parte-3-installazione-subordinate-ca.htm

Pubblicazione CRL Subordinate CA

Copia della CRL e del certificato della CA sul Web Server

- *L'accesso al Web Server da parte della Subordinate CA avviene tramite un account locale creato sul Web Server* che ha privilegi di lettura e scrittura su una share che punta alla cartella sul Server Web che conterrà le CRL e i certificati delle CA
- *Memorizzare sulla Subordinate CA l'account che verrà utilizzato per accedere alle share sul Web Server*
- Creare un'*operazione schedulata per aggiornare la CRL e copiare sul Web Server la CRL e il certificato CA della Subordinate CA*

```
REM Aggiornamento della CRL
certutil –CRL

REM della CRL e il certificato CA della Subordinate CA sul Web Server
copy /Y “%SytemRoot%\System32\CertSrv\CertEnroll\*.crl” \\webserver.domain.ext\CertEnrollShare
copy /Y “%SytemRoot%\System32\CertSrv\CertEnroll\*.crt” \\webserver.domain.ext\CertEnrollShare
```

https://www.ictpower.it/sistemi-operativi/deploy-pki-in-windows-server-2016-parte-3-installazione-subordinate-ca.htm

Demo

Torino
Technologies
Group

ICT POWER.IT

Installazione e configurazione Subordinate CA

Gestione PKI con PowerShell

Deploy Active Directory Certificate Service

ServerManager Module Cmdlets (12 CmdLets)

Windows Server 2012 e succ.

Configurazione CRL, AIA e OCSP

- *Add-CACrlDistributionPoint, Get-CACrlDistributionPoint, Remove-CACrlDistributionPoint*
- *Add-CAAuthorityInformationAccess, Get-CAAuthorityInformationAccess, Remove-CAAuthorityInformationAccess*

Windows Server 2012 e succ.

Configurazione certificate template

- *Add-CATemplate, Remove-CATemplate*

Windows Server 2012 e succ.

Gestione ciclo vita certificati

PKI Client Cmdlets (17 CmdLets)

Windows Server 2012 e succ.

AD CS Deployment Cmdlets: https://technet.microsoft.com/en-us/library/hh848387(v=wps.620).aspx
AD CS Administration Cmdlets: https://technet.microsoft.com/en-us/library/hh848365(v=wps.630).aspx
PKI Client Cmdlets: https://technet.microsoft.com/en-us/library/hh848636(v=wps.620).aspx

Backup and Restore

- *Backup-CARoleService*
- *Restore-CARoleService*

Windows Server 2012R2 e succ.

Verifica TPM

- *Confirm-CAEndorsementKeyInfo*

Windows Server 2012R2 e succ.

Automazione certutil

E' possibile utilizzare PowerShell per automatizzare certutil, ad esempio per creare un report dei certificati scaduti

https://blogs.technet.microsoft.com/poshchap/2016/01/01/powershell-and-certutil-exe/

CA Backup and Restore:
https://technet.microsoft.com/en-us/library/dn535774(v=ws.11).aspx

Torino Technologies Group

ICT POWER.IT

Utilizzo di Let's Encrypt in Windows Server 2016

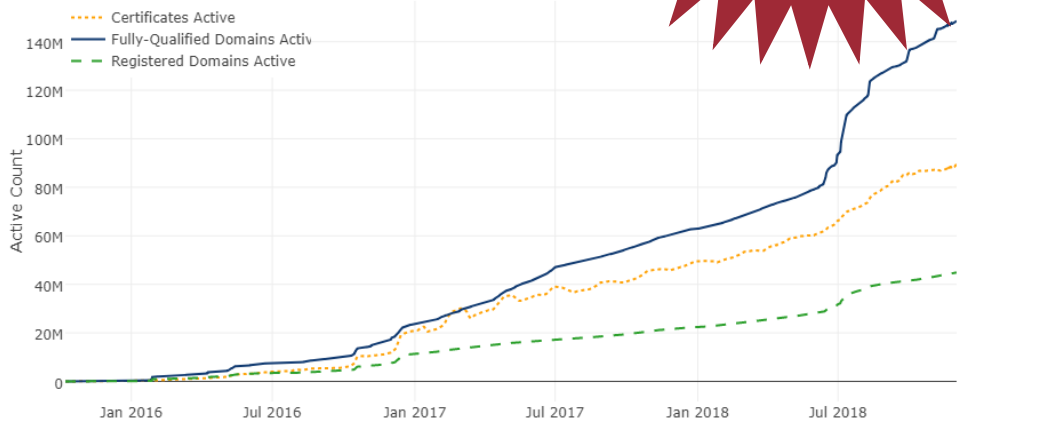
Public Key Infrastructure Windows e Let's Encrypt

Let's Encrypt Overview

Servizio fornito da IRSG
Free e nonprofit, finanziata con donazioni e nominal fee applicate agli Integrators (hosting providers)

Caratteristiche, funzionalità e limiti di utilizzo

- Global Certificate Authority (CA) che *automatizza gratuitamente* la creazione, la validazione, il rilascio e il rinnovo di certificati X.509 per il protocollo *SSL/TLS* con validità *90 giorni, rinnovabili dopo 60 giorni*
- Basata sul *protocollo ACME* (Automated Certificate Management Environment)
- Offre certificati *Domain Validation (DV)*, certificati *SAN (Subject Alternative Names)* e certificati *wildcard* (supportati da marzo 2018 come parte dell'ACME v2)
- Non offre certificati *Organization Validation (OV)* o *Extended Validation (EV)*
- Non offre certificati per *Email Encryption* e *Code Signing*
- *Principali limiti di utilizzo:*
  - *Certificati per dominio*: 50 a settimana
  - *Nomi per singolo certificato* (SAN): 100
  - *Certificati duplicati* (per lo stesso domino): 5 a settimana
  - *Il rinnovo dei certificati sono esentati dai limiti di rilascio*
  - *Validazioni fallite*: 5 per per account, per hostname, per ora
  - *Limite di richieste*: 20 al secondo
  - *Accounts per indirizzo IP*: 10 per 3 ore
  - *Autorizzazioni pendenti per account*: 300



Let's Encrypt Growth: https://letsencrypt.org/stats/

Browser supportati	Firefox, Chrome, Internet Explorer, Edge, Safari, Silk
OS supportati	Windows XP SP3 e succ., Android, Debian, Ubuntu, Blackberry, PS4, Jolla Sailfish, Kindle
Software supportato	NSS Library, Java
Software da supportare	Apache HTTP Server Project (httpd)
Software non supportato	Windows Live Mail (2012 mail client)

Rate Limits: https://letsencrypt.org/docs/rate-limits/ Status: https://letsencrypt.status.io/ Certificate Compatibility: https://letsencrypt.org/docs/certificate-compatibility Certificate Search: https://crt.sh/

mozilla Akamai CISCO EFF OVH chrome IdenTrust FORD FOUNDATION Internet Society facebook ALA American Library Association shopify cyon ...and more!!

Architettura e requisiti di Let's Encrypt

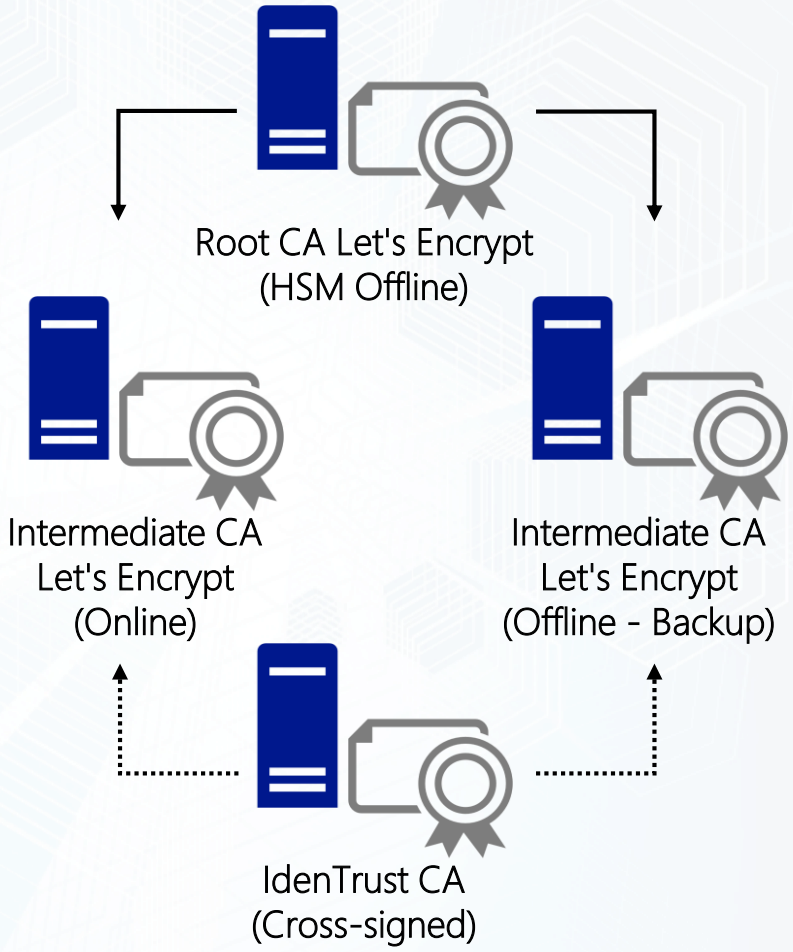
Architettura

- Let's Encrypt ha generato un *certificato root RSA* che si trovava su un *hardware security module* (HSM) poi *messo offline*
- *Il certificato root è stato utilizzato per firmare due certificati intermedi*, uno usato per la firma dei certificati e uno offline per backup
- *I due certificati intermedi sono anche cross-signed da parte di IdenTrust* in modo che i certificati generati siano considerati attendibili anche se nel caso in cui il certificato root non sia considerato attendibile dal browser

Requisiti

- Il *certificato DST Root X3 di IdenTrust* deve essere incluso nell'elenco dei certificati attendibili
- Supporto a *SHA-2 (Secure Hash Algorithm 2)* utilizzato da tutti i certificati di Let's Encrypt

Chain of Trust: https://letsencrypt.org/certificates/

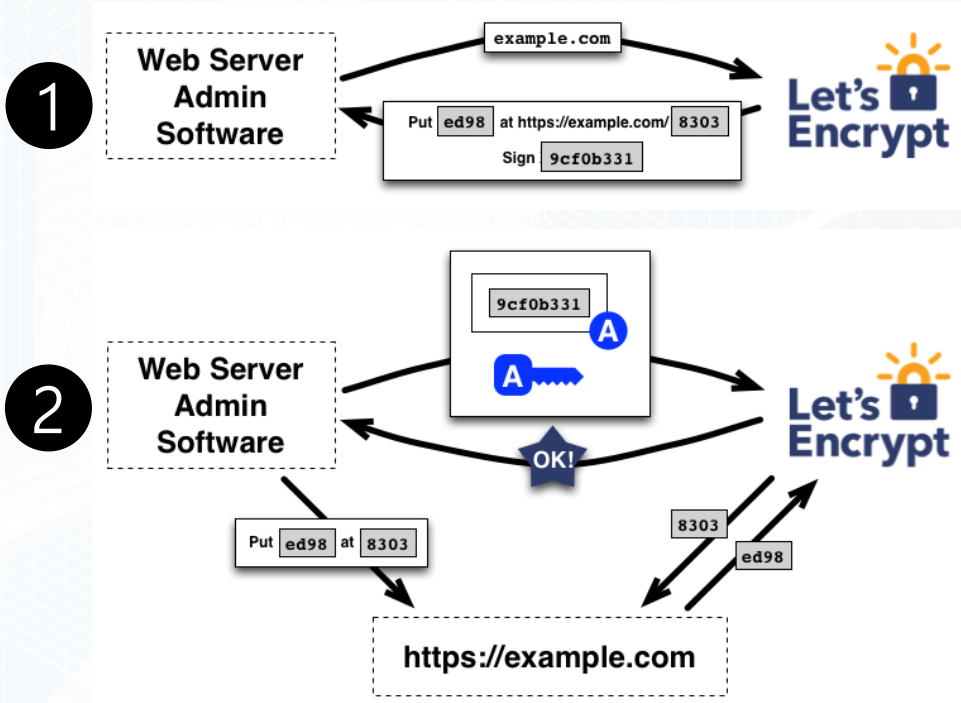


Protocollo ACME: Validazione dominio

Procedura

- Occorre utilizzare un *Agent per la gestione dei certificati* sul server web
- L'Agent deve *creare la coppia di chiavi pubblica e privata* da autorizzare
- L'Agent deve *dare prova alla CA che il server web controlla il dominio* creando una *risorsa HTTP in un URI prestabilito dalla CA* o creando un *record DNS prestabilito dalla CA* nel dominio
- L'Agent deve *dare prova alla CA che controlla le chiavi pubblica e privata* firmando un file prestabilito dalla CA
- Se le verifiche hanno esito positivo l'Agent identificato dalla chiave pubblica è autorizzato a eseguire la gestione del certificato per il dominio
- La *chiave privata è sempre generata e gestita sul proprio server* e non dalla CA Let's Encrypt, è possibile anche utilizzare una chiave privata esistente o una Certificate Signing Request (CSR)
- Dopo aver eseguito la procedura di validazione del dominio l'*autorizzazione è messa in cache per 30 giorni* e riutilizzata senza rieseguire la validazione

How It Works: https://letsencrypt.org/how-it-works/ FAQ: https://letsencrypt.org/docs/faq/



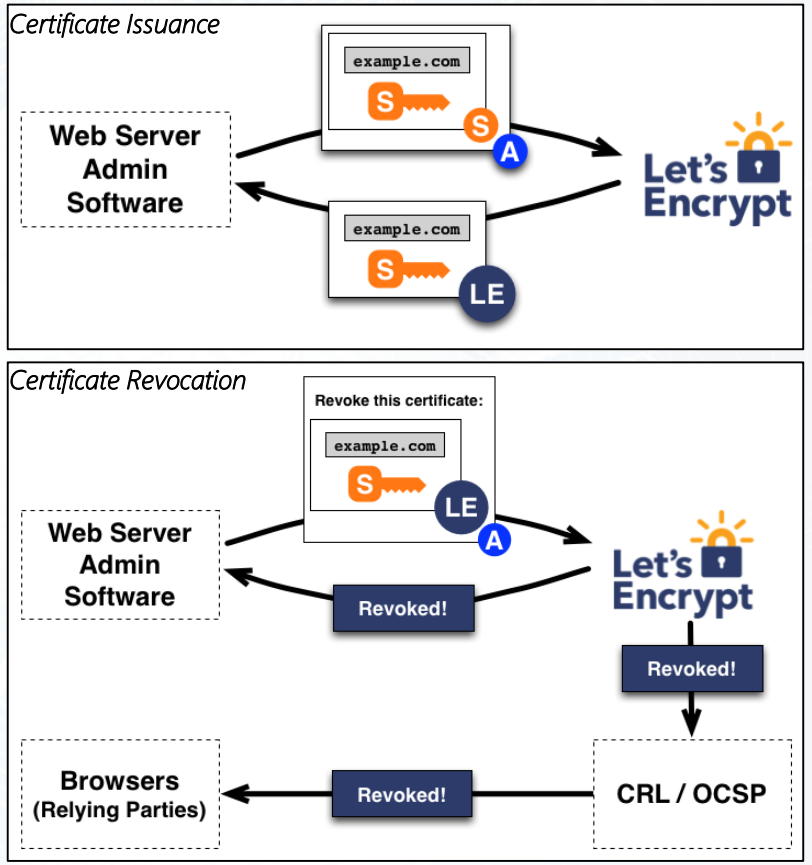
Protocollo ACME: Gestione certificato

Richiesta e rinnovo di un certificato

- L'Agent deve aver ottenuto una *coppia di chiavi pubblica e privata autorizzata* tramite la procedura di validazione del dominio
- L'Agent genera una *PKCS#10 Certificate Signing Request (CSR)* che richiede alla CA la generazione di un certificato *specificando la chiava Pubblica*, include nella CSR una *firma della chiave privata corrispondente alla chiave pubblica* e *firma la CSR* per dare prova alla CA di essere autorizzato per il dominio
- La *CA controlla le due firme* e se valide *emette un certificato per il dominio con la chiave Pubblica del CSR* e lo *restituisce all'Agent*

Revoca di un certificato

- L'Agent *firma la richiesta di revoca con la coppia di chiavi Pubblica e Privata* autorizzata per il dominio
- La *CA controlla che la richiesta sia autorizzata* e quindi *pubblica le informazioni di revoca* sui normali canali di revoca: *CRL* (Certificate Revocation List) e *OCSP* (Online Certificate Status Protocol)



How It Works: https://letsencrypt.org/how-it-works/

ACME v2

Supporto a certificati wildcard

La generazione di certificati wildcard è pienamente supportata dal 13 marzo 2018

ACME v2 sarà uno standard IETF

L'ACME v2 è stato presentato all'IETF (https://tools.ietf.org/html/draft-ietf-acme-acme-02) per diventare uno standard ed essere utilizzato da altre CA (L'ACME v1 era invece una specifica ben documentata)

Miglioramenti tecnici

ACME v2 apporta alcuni miglioramenti tecnici che permetteranno una miglior gestione

ACME v1 ancora utilizzabile

Non è ancora stata impostata una end-of-life date per l'ACME v1, ma è raccomanda la migrazione appena il client utilizzato lo consente

How It Works: https://letsencrypt.org/2017/06/14/acme-v2-api.html

Client ACME

Mantenere aggiornato il Client ACME

Gestione tramite accesso alla Shell

- *Client:* Bash, Browser, C, Clojure, Docker, Go, HAProxy, Java, Microsoft Azure, nginx, Node.js, OpenShift, Perl, PHP, Python, Ruby, Rust, Windows / IIS
- *Librerie:* Go, Java, Node.js, Perl, PHP, Python, Ruby, Rust, Windows

ACME Client Implementations: https://letsencrypt.org/docs/client-options/

Certbot

- Client raccomandato
- OS: UNIX-like
- Ver: 0.29.1 (6 dic 2018)

https://certbot.eff.org/

ACMESharp (diventerà ACMESharp Core)

- Libreria .NET
- PowerShell client
- Ver: 0.9.1 (26 ott 2017)

GitHub: ebekker/ACMESharp

Windows ACME Simple

- Client .NET command line
- Basato su ACMESharp
- Ver: v1.9.12.2 (29 nov 2018)

GitHub: PKISharp/win-acme

Staging Environment

Ambiente di test con rate limits meno stringenti (30.000 certificati per dominio a settimana e 30.000 certificati duplicati a settimana)

https://letsencrypt.org/docs/staging-environment/

Implementazione di un client ACME

- Disponibili *indicazioni da rispettare e sorgenti* (GitHub: letsencryp)
- Let's Encrypt, Web PKI, le API del protocollo ACME sono in continua evoluzione, occorre *gestire gli aggiornamenti di tutti i servizi utilizzati* da Let's Encrypt

ACME Protocol Updates: https://letsencrypt.org/docs/acme-protocol-updates/

Gestione senza accesso alla Shell

- *Utilizzare il supporto built-in support dell'hosting provider* (metodo consigliato)
- *Utilizzare un Client* ACME che supporta la modalità manuale (Certbot, letsencrypt-win-simple) per eseguire l'upload sul website di un file specifico e dare prova alla CA di avere il controllo, *occorre ripetere la procedura ad ogni rinnovo*

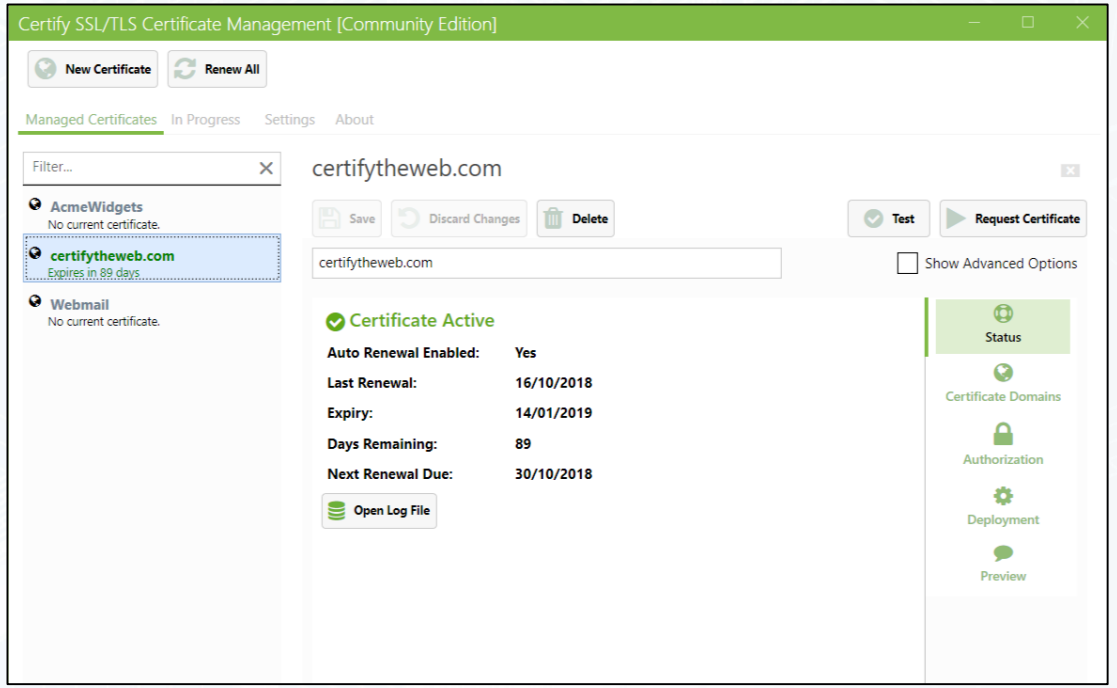
Web Hosting who support Lets Encrypt: https://community.letsencrypt.org/t/web-hosting-who-support-lets-encrypt/6920

Certify the web

SSL Certificate Manager for Windows

- *GUI Open Source basata su ACMESharp (per il supporto di ACME v1) e Certes (per il supporto di ACME v2)* sviluppata da Webprofusion Pty Ltd (GitHub: webprofusion/certify)
- Consente l'installazione e il rinnovo automatico di certificati Let's Encrypt su IIS
- *Applicazione gratuita fino a 5 siti per server*, per poter gestire un numero maggiore di siti o per supportarne lo sviluppo occorre acquistare una *upgrade key*
- L'ultima release è la *V4.0.12 rilasciata il 4 dic 2018*
- Requisiti minimi: *Windows Server 2008 R2 SP1* o successive con *.Net 4.6.2 superiore*

http://certify.webprofusion.com/



Demo

Torino Technologies Group

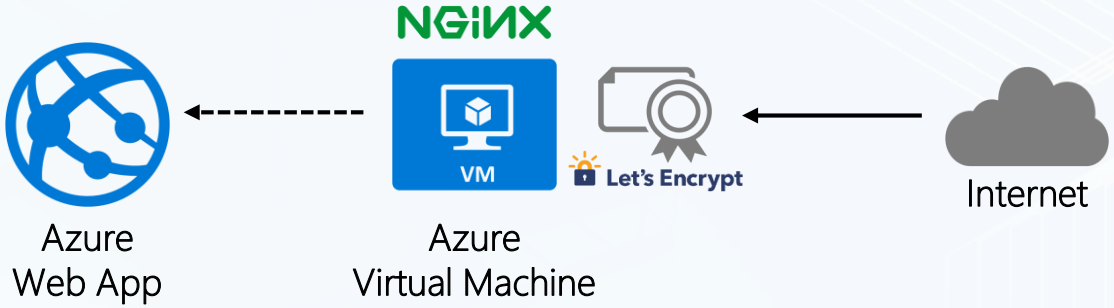
ICT POWER.IT

Utilizzo di WACS (Windows ACME Simple) e CERTBOT

Utilizzo di Let's Encrypt in Azure

Metodo 1: Reverse proxy

- Tramite un *client Acme* come CertBot (in ambiente Linux) o Windows ACME Simple (in ambiente Windows) *si assegna al reverse proxy un certificato generato da Let's Encrypt*
- Come reverse proxy è possibile utilizzare *Nginx disponibile sia in ambiente Linux che Windows*
- Tramite un reverse proxy come Nginx *è possibile mappare anche un Azure Web Site Free con un nome dns differente*

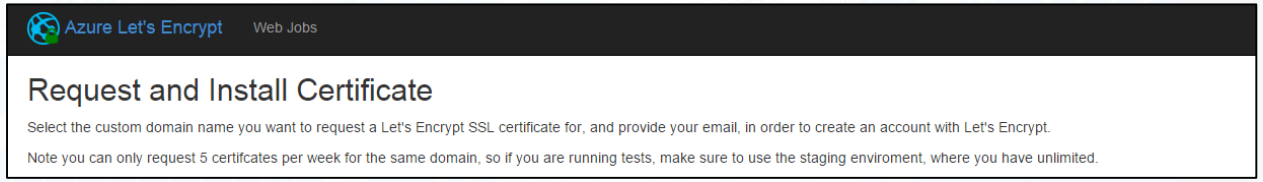


http://www.fhtino.it/blog/showpost.aspx?id=4ec32bde08304ba6a8f332bef671973b

Let's Encrypt Certificates on the Azure Application Gateway
https://allthingscloud.info/2018/08/01/lets-encrypt-certificates-on-the-azure-application-gateway/

Metodo 2: Site extension

- *Utilizzare la site extension Let's Encrypt Site Extension* che permette l'installazione e la configurazione di certificati SSL rilasciati da Let's Encrypt
  - Let's Encrypt Site Extension *non è supportata da Microsoft ed è al momento in beta* (l'ultima versione è la 0.8.8 del 13 novembre 2018),
  - La versione 0.8.5 è *pubblicata nella Azure Site Extension gallery*
- *letsencrypt-webapp-renewer console web per il rinnovo dei certificati* basata su Azure Web App Site Extension



GitHub: sjkp/letsencrypt-siteextension [0.8.8 13 nov 2018]
GitHub: ohadschn/letsencrypt-webapp-renewer [0.8.5.1 30 dec 2017]

https://blogs.msdn.microsoft.com/mihansen/2018/01/25/azure-web-app-with-lets-encrypt-certificate-powershell-automation/

https://www.troyhunt.com/everything-you-need-to-know-about-loading-a-free-lets-encrypt-certificate-into-an-azure-website/

DNS CAA Resource Record

DNS Certification Authority Authorization Resource Record

- **Gennaio 2013**: definizione dei record di risorsa DNS CCA per *consentire di specificare una più Certification Authorities* (CAs) *autorizzate ad emettere certificati per il dominio* e ridurre il rischio di sicurezza relativo all'emissione di un certificato non richiesto (RFC 6844 - https://tools.ietf.org/html/rfc6844)
- **8 settembre 2017**: *il controllo dei record CAA diventa mandatorio per le CAs* come parte del processo di rilascio dei certificati (CA/Browser Forum - Ballot 187), ma non è obbligatorio configurare i record CAA
- **16 novembre 2017**: Azure DNS supporta i Certificate Authority Authorization (CAA) Records
- **Let's Encrypt**: *controlla i record CAA, ma non supporta il "tree-climbing"* che prevede il controllo anche dei domini padre (tale controllo è stato rimosso dall'erratum 5065)

"As part of the issuance process, the CA must check for a CAA record for each dNSName in the subjectAltName extension of the certificate to be issued, according to the procedure in RFC 6844, following the processing instructions set down in RFC 6844 for any records found. If the CA issues, they must do so within the TTL of the CAA record, or 8 hours, whichever is greater."

DNS CAA Resource Record: https://www.devadmin.it/2017/11/27/dns-caa-resource-record/

Torino Technologies Group

ICT POWER
IL POTERE DELLA TECNOLOGIA

Question & Answer