## CERTIFICATE CHAIN

I certificati usati dai web-server non sono mai firmati dalle CA principali, dette ROOT CERTIFICATION AUTHORITY, bensì sono firmate da CA intermedie che sono "delegate" dalle ROOT CA per ragioni di scalabilità.

In genere questa catena è a 2 livelli. Le CA intermedie si devono a loro volta accreditare presso una CA ROOT, la quale FIRMA LA CHIAVE PUBBLICA della CA INTERMEDIA.

Gli user agent devono quindi conservare nel loro registro delle firme affidabili solo le chiavi pubbliche delle (pochissime) CA ROOT per verificare l'intera catena.

Una ROOT CA è definita SOLO IN BASE A POLICY STANDARD.

Non esiste un ente centrale che "accredita" le CA ma ogni user-agent definisce la propria politica per la selezione delle ROOT CA.

---

## TIPI DI CERTIFICATION AUTHORITIES

A seconda del contesto in cui operano, le CA sono di vari tipi:

I) PUBBLICHE: Aziende che rilasciano certificati per clienti che lo richiedono, per web-server pubblici

II) INTERNE: Rilasciano certificati per web-server pubblici ma solo a determinati siti. Es: In passato grandi aziende hanno una CA intermedia interna solo per i loro siti.

III) PRIVATE: Utilizzate solo nella rete locale privata per firmare certificati ad uso interno.

Queste CA prendono anche il nome di CERTIFICATI AUTOFIRMATI (SELF-SIGNED)

---

## STANDARD x509

È lo standard mondiale per i certificati web. Contempla vari campi, alcuni dei quali fondamentali per il funzionamento del protocollo mentre altri sono metadati di supporto.

Uno dei campi più importanti è il DISTINGUISHED NAME (DN), ovvero il nome univoco che identifica il web-server.

I certificati x509 possono essere codificati in vari modi, generalmente sono in CODIFICATI IN BASE 64 ed hanno estensione .pem

Per gestire in modo completo i certificati x509 è possibile usare OpenSSL che è il tool più diffuso e potente disponibile. Tramite OpenSSL è anche possibile generare certificati autofirmati.

Alcuni dei metadati nei certificati x509 sono INFORMAZIONI LEGALI, ovvero informazioni che NON HANNO VALORE ai fini del protocollo di autenticazione (ad esempio il nome dell'azienda titolare del sito web.)

Nei certificati AUTOFIRMATI i campi ISSUER (autorità che rilascia il certificato) e SUBJECT (entità per cui è rilasciato il certificato) SONO UGUALI. Questo accade solo nei certificati privati e nei CERTIFICATI DELLE ROOT CA. Le Root CA hanno certificati autofirmati che vengono distribuiti direttamente agli user-agent per validare le altri

certificate chain. La differenza quindi è che i certificati self-signed delle CA sono distribuiti agli user-agent in modo separato, non durante la connessione al server.

I certificati hanno anche un TIPO che identifica il loro scopo, come ad esempio: autenticare END ENTITY (siti web), autenticare EMAIL, autenticare CERT ENTITY (certificati rilasciati a CA intermedie per validare le loro chiavi di firma).

È anche possibile usare un certificato autofirmato per costruire una CA PRIVATA. Poiché è necessario distribuire manualmente su ogni user-agent i certificati autofirmati, distribuire 1 solo certificato è più semplice (quello della CA) e usarlo poi per autenticare tutti i server di interesse nella rete aziendale. In questo modo viene sostanzialmente istituita una CERTIFICATE CHAIN PRIVATA totalmente isolata da quelle pubbliche.

---

OTTENIMENTO DI UN CERTIFICATO

Per ottenere un certificato valido da una CA bisogna svolgere i seguenti passaggi:

I) Generare un KEY PAIR valido

II) Creare una CERTIFICATE SIGN REQUEST (o CSR), ovvero un oggetto che la rappresenta la richiesta di rilascio di un certificato.

III) Invio della CSR alla CA ed eventuale pagamento

IV) Generazione del certificato. La chiave privata del server generata al punto I va salvata in modo molto sicuro. La scelta migliore è salvarla in hardware.

# Tipi di Certificato

I) Certificati DOMAIN VALIDATED (DV): la CA ha semplicemente controllato che il server che richiede il certificato abbia effettivamente il controllo del dominio da certificare.

II) Certificati ORGANIZATION VALIDATED (OV): la CA verifica anche le informazioni legali fornite dal richiedente, come il nome della società, compagnia, le attività, ecc.

III) Certificati EXTENDED VALIDATION (EV):

- Vengono richieste e verificate informazioni legali aggiuntive
- Richiedono pratiche di sicurezza aggiuntive (uso di certi protocolli o di certe pratiche)
- In genere mostrano sulla barra del browser il nome legale dell'azienda.

In generale i certificati OV non vengono usati, si preferiscono gli EV se i DV non sono sufficienti.

I certificati EV non danno maggiori garanzie tecniche, semplicemente forniscono garanzie sull'identità legale del titolare del certificato.

# REVOCA DEI CERTIFICATI DSC

Qualunque certificato emesso ha una data di scadenza e le informazioni contenute in un DSC non possono essere modificate senza invalidare la firma.

L'operazione di revoca consiste nel RENDERE NON VALIDO UN CERTIFICATO PRIMA DELLA SCADENZA NATURALE.

La data revoca può essere fatta in 2 modi:

I) Tramite una CERTIFICATE REVOCATION LIST (CRL), ovvero un file che contiene tutti gli ID dei certificati revocati.

II) Tramite protocollo OCSP (ONLINE CERTIFICATE STATUS PROTOCOL), ovvero un protocollo che consente di fare verifiche dinamiche su file CRL remoti.

Il vantaggio di OCSP è quello di evitare di trasmettere le CRL (magari di grandi dimensioni).

I certificati possono essere solo revocati solo DA CHI LI HA EMESSI. Inoltre, le CA devono mettono a disposizione le API OCSP che permettono di validare le CRL relative ai propri certificati.

2 grandi PROBLEMI DI PKI sono:

I) Il sistema di revoca è facilmente attaccabile (vedi dopo)

II) Tutto si basa su dei SINGLE POINT OF TRUST, ovvero su alcune CA considerate totalmente fidate (se vengono violate è un casino).

## PROBLEMI DELL'ARCHITETTURA DI REVOCA DSC

I) Nel caso di furto della chiave privata del WEB SERVER, l'attaccante può fare un attacco MAN IN THE MIDDLE poiché il certificato risulta valido.
In questo caso il gestore del sito attaccato richiede alla CA di revocare il certificato, mettendo nella propria CRL il certificato da invalidare.
Tutta questa infrastruttura si scontra con il fatto che lo user-agent deve consultare il servizio OCSP volontariamente per controllare la CRL.
In caso di attacco MAN IN THE MIDDLE l'attaccante deve violare il mezzo di comunicazione e di conseguenza può facilmente intercettare anche la comunicazione con l'OCSP, facendo uno STALE ATTACK, ovvero fornire una risposta VALIDA ottenuta prima dell'inserimento in CRL del certificato rubato.
In questo caso lo user-agent riceve una risposta valida (firmata dalla CA) che segnala NON REVOCATO il certificato rubato, quindi il meccanismo CRL fallisce.

II) Un altro scenario è quello in cui non è possibile ottenere una risposta dell'OCSP, quindi non è possibile fare uno stale attack. In questo caso l'attaccante potrebbe semplicemente bloccare BLOCCARE IL SERVIZIO OCSB tramite un attacco DoS. Lo user-agent potrebbe considerare non valido il certificato a priori se non riesce a contattare l'OCSP

III) Un ulteriore scenario è quello di un attacco DoS (DENIAL OF SERVICE) semplice, volto a bloccare solo il servizio OCSP.
Se lo user-agent usa la politica del considerare non valido un DSC che non può validare questo blocca l'accesso a tutti i servizi

che hanno certificati firmati da quella CA. Da notare che in questo caso l'attaccante non ha bisogno di violare la chiave privata del server e non è quindi necessario revocare il certificato.

Nonostante questi problemi il protocollo basato sulle CRL è l'unica soluzione per revocare i DSC.

Il protocollo OCSP descritto prima è di tipo PULL (recupero informazione solo quando serve, ovvero quando voglio verificare un DSC).

Potremmo tentare di risolvere cambiando approccio, usando quindi un protocollo PUSH (il servizio OCSP invia a tutti i client un aggiornamento ogni volta che la CRL viene aggiornata). In questo modo non è lo user-agent che controlla spesso ma l'infrastruttura che si auto-regola.

In questo modo è molto più difficile fare attacchi DoS.

I problemi di questo paradigma sono:

I) Ogni client deve memorizzare le CRL

II) Chi si occupa della propagazione? → Solitamente le società che producono browser.

In realtà per minimizzare il lavoro gli user agent fanno il push solo delle revoche critiche (DSC rubati) mentre continuano a fare richieste pull per tutti gli altri casi, assumendo come validi i certificati non presenti nella propria CRL locale in caso di DoS.

La propagazione dei componenti CRL importanti è affidata a servizi implementati dai produttori di browser.

In generale, visto quanto è difficile revocarli, è preferibile assegnare durate più corte per i DSC (attualmente MAX 1 ANNO).

# Problemi legati alle Certification Authorities

I problemi legati alle CA variano in base al tipo di CA.

## I) CA pubbliche

Il problema fondamentale è un problema di FIDUCIA, ovvero come possiamo essere certi che le CA rilascino certificati solo ai legittimi richiedenti e non a terzi? Se ciò non accade la CA sta rilasciando DSC FALSI e sta COMPROMETTENDO L'INTERA INFRASTRUTTURA PKI.

## II) CA private:

In questo caso non è un problema di FIDUCIA ma di SICUREZZA, dobbiamo quindi garantire la sicurezza della nostra CA autoformata.

## Problema di fiducia (CA pubbliche):

Ci sono vari approcci a questo problema:

I) AD-HOC: contromisure prese dall'utente per limitare il numero di CA ritenute affidabili (blacklist, whitelist ...).

II) PROTOCOLLARE: soluzioni integrate all'interno del protocollo che usa il certificato (ad esempio HTTPS)
La tecnica più utilizzata (impostata) per contrastare gli attacchi a PKI condotti da governi è HTTP PINNING che si basa su TOFU e sul fatto che, la prima volta, la connessione avviene su rete sicura, per poi difendersi da sostituzioni malevole dei DSC

III) ESTENSIONE DELLA PKI: introdotta per migliorare l'HTTP-PINNING che è molto rigido e si basa su TOFU.
La migliaia si chiama CERTIFICATE TRANSPARENCY

CERTIFICATE TRANSPARENCY (CT).

Ci si vuole difendere da man in the middle portato avanti con DSC falsi ma validi erogati da CA diverse (spesso sono attacchi governativi).

Dal 2018 è obbligatorio per tutti i tipi di certificati.

Oltre a CA, proprietari dei DSC (gestori siti) e user-agent si aggiungono nuovi attori di controllo per diminuire il potere delle CA.

Si introducono quindi dei SERVIZI DI LOG che tengono traccia di TUTTI I CERTIFICATI EROGATE DA QUALUNQUE CA. Questi log sono PUBBLICAMENTE ACCESSIBILI. Le CA quindi rimangono TTP ma è possibile sorvegliare il loro lavoro senza imporre dei vincoli alle CA stesse. Se una CA eroga certificati falsi è facile vederlo e prendere delle contromisure.

Gli user-agent come prima cosa richiedono metadati aggiuntivi che DIMOSTRINO l'inserzione del DSC dentro ad alcuni log fidati. Esistono infatti AUTORITÀ DI LOG FIDATE ed i metadati che attestano la registrazione sono firmati da queste entità.

Per registrare il DSC la CA invia un PRE-CERTIFICATE al LOG SERVER (il PRE-CERTIFICATE è il DSC NON FIRMATO) il quale lo registra e risponde con un SIGNED CERTIFICATE TIMESTAMP (SCT), ossia la conferma autenticata richiesta dallo user-agent.

→ Gli SCT vengono inseriti nel certificato e poi firma il DSC.

→ In questo modo tutto è verificabile. Attualmente sono richiesti almeno 2 SCT per ragioni di ridondanza.

→ I client devono inoltre conoscere la lista dei servizi di log verificati

Questa infrastruttura consente solo di controllare a POSTERIORI chi ha emesso un DSC, quindi è possibile individuare le CA disoneste e revocarne l'autorità.

Gli SCT possono essere distribuiti tramite DSC ma anche tramite TLS (la CA non deve fare niente) o tramite OCSP (approccio bilanciato a livello di impegno tra CA e client).

Le info delle ai servizi di log sono salvati in STRUTTURE DATI VERIFICABILI. Sono quindi strutture che consentono al client che interagisce con i servizi di log stessi di rendersene conto (ad esempio log eliminati, risposte sbagliate alle API, ecc.)

→ In questo modo è virtualmente impossibile rompere la catena della fiducia in PKI / Introdurre certificate transparency.

NOTA: Gli user-agent NON CONTATTANO I LOG SERVER quando vogliono verificare DSC, si limitano a controllare la presenza degli SCT e la firma di ognuno di loro.

→ Il controllo vero e proprio sui log server è affidato a 2 tipi di attori:

I) CT MONITOR: enti che effettuano controlli ad ampio spettro su tutti i log server per verificare il buon funzionamento di internet (es Google, Microsoft...)

II) CT AUDITORS: enti che effettuano verifiche mirate su determinate informazioni (es aziende che vogliono controllare che il proprio certificato non venga violato).

---

## APPROCCIO IBRIDO: CA PRIVATE AS-A-SERVICE

È possibile utilizzare dei servizi commerciali per gestire una architettura PKI PRIVATA senza dover effettivamente gestire l'architettura.

In questo modo i certificati sono scollegati dalla PKI pubblica e pertanto i certificati NON possono da certificate transparency.

Questo approccio è utilizzato soprattutto da grandi aziende per gestire la propria rete interna. Molti servizi cloud come AWS, AZURE, GOOGLE CLOUD PLATAFORM e altri forniscono simili servizi per le macchine il cloud.

---

## ESTENSIONE MULTILIVELLO CA PRIVATA

Nel caso si voglia sviluppare una CA privata, così come per le CA pubbliche, la gerarchia migliore per garantire la sicurezza del sistema è a 2 livelli:

I) ROOT CA: firma solo i certificati delle altre CA, le chiavi vengono custodite offline in modo molto sicuro

II) INTERMEDIATE CA: firma effettivamente i DSC per i domini.

Una estensione RARAMENTE USATA ma possibile è quella di avere MULTIPLE CA INTERMEDIE (di pari livello) per LIMITARE L'AMBITO A UN SOTTODOMINIO (SCOPING).

In questo modo ogni CA firma solo un sottogruppo di domini, rendendo la gestione di quei servizi autonoma rispetto agli altri gruppi.

Ad esempio:

I) ROOT CA → UNIMORE.IT

II)
- MIDDLE 1 → DIEF.UNIMORE.IT (& sottodomini)
- MIDLE 2 → DEMI.UNIMORE.IT (& sottodomini)

È possibile estendere il certificato rilasciato dalla ROOT CA con un campo NAME CONSTRAINT che rimanda la CA intermedia a firmare solo certi sottodomini (altrimenti i client non riconoscono il certificato).

Questo è possibile SOLO nelle CA PRIVATE. Questo sistema limita i danni in caso di violazione perché limita il campo d'azione degli attaccanti.

# CROSS-PROTOCOL / CROSS-SERVICE AUTOMATIC VERIFICATION

Esistono dei protocolli che consentono di verificare automaticamente (machine to machine) l'identità di un server per poi rilasciare un certificato.

Il protocollo ACME / LETSENCRYPT si basa su alcuni SERVIZI PUBBLICI SEMPRE DISPONIBILI i quali eseguono dei controlli tramite RICHIESTE AUTOMATICHE al dominio indicato nella richiesta di certificato.

Il certificato è di tipo DV e la validazione si basa su protocollo PROOF OF POSSESSION, il richiedente deve DIMOSTRARE di avere il CONTROLLO sul dominio da certificare.

Il protocollo a livello comando è di tipo CHALLENGE / RESPONSE: il server lancia una sfida tramite una richiesta e valuta la risposta data dal server.