

Prof3ta Publications



*"TRUST NO ONE"
A Certificate Revocation
Survey*

*Università degli Studi di Catania
Facoltà di Ingegneria
Corso di Laurea Specialistica in
Ingegneria Informatica*

*Docente: Orazio Tomarchio
Studente: Roberto Aloï*

Linux: Be Root
Windows: Re Boot
(Anonymous Programmer)

Indice

Indice	3
Prefazione	5
1 La revoca di un certificato digitale	7
1.1 Una Panoramica Generale	7
1.2 Le Ragioni della Revoca	8
1.3 Progettazione di un Metodo di Revoca	9
2 Certificate Revocation List	11
2.1 CRL in Breve	11
2.2 Un po' di Storia	11
2.3 Funzionamento delle CRL	12
2.4 Pro e Contro delle CRL	12
2.5 Un Caso Eclatante	13
3 Online Certificate Status Protocol	15
3.1 OCSP in breve	15
3.2 Funzionamento di OCSP	15
3.3 Pro e Contro di OCSP	16
4 Ulteriori Meccanismi di Revoca	18
5 Minacce ai Meccanismi di Revoca	19
5.1 Denial of Service	19
5.2 Intrusione	20
5.3 Aggiornamento della risposta di validità	21
5.4 Riproduzione	22
6 Conclusioni	24
A CR in Pratica: Mozilla Firefox	25

INDICE 4

Bibliografia 28

Prefazione

In un mondo in cui, giorno dopo giorno, l'interazione e la comunicazione stanno muovendo il proprio baricentro da un universo di tipo analogico e tradizionale ad uno digitale e futurista, nascono spontanee alcune necessità di sicurezza sino a pochi anni fa inconcepibili.

In una realtà in cui *workstations* prendono il posto di persone e nodi di rete rimpiazzano *umani intermediari*, ci si trova dinanzi alla possibilità di relazionarsi, di trattare con entità perfettamente ignote, sconosciute.

Il principio basilare da seguire in questi casi è sempre il solito, canuto, eterno:

Trust No One

ovvero

Non fidarti di nessuno

In realtà di qualcuno, alla fine, ci si dovrà pur fidare, pena l'impossibilità di eseguire qualunque tipo di transazione, l'isolamento.

Ci stiamo riferendo alle cosiddette *Certification Authorities* (CA), fidejutorie autorità dedite alla generazione ed alla gestione dei cosiddetti *certificati digitali*.

Abbiamo già visto come i certificati digitali costituiscano una strategia in grado di consentire a due o più entità di fidarsi (*to trust*) l'una dell'altra.

Spesso, però, capita che le informazioni attestate all'interno di un determinato certificato non siano *definitive*, ma in qualche modo *temporanee* (basti pensare all'utilizzo di un indirizzo email, che può variare ripetutamente anche all'interno di un breve arco di tempo). Ecco allora nascere il concetto di *scadenza* del certificato stesso.

Talvolta può avvenire anche che per svariate ragioni (che spaziano dalla possibilità che i dati relativi al certificato stesso siano variati a quella che la chiave privata sia stata smarrita e/o rubata) si riveli necessario *annullare la validità* di un dato certificato *prima* della sua effettiva data di scadenza.

In questo caso si parla proprio di *processo di revoca del certificato*.

Anche se può suonare ambiguo, *la revoca di un certificato* si ottiene mediante l'utilizzo di un *certificato di revoca*, che può essere predisposto sia dalla stessa persona che si costruisce le chiavi, che dall'autorità di certificazione che si occupa di rilasciare i certificati.

In questo elaborato, verrà fornita una panoramica dei principali metodi di revoca dei certificati esistenti, dei quali verranno analizzati pro e contro.

In particolare:

Il **Capitolo 1** affronta in maniera del tutto generale il problema del processo di revoca di un certificato, analizzandone ragioni, problematiche e possibilità, indipendentemente dal tipo di approccio pratico utilizzato.

Il **Capitolo 2** è dedicato alle *Certificate Revocation Lists* (o CRL), il meccanismo probabilmente più diffuso nell'ambito dei processi di revoca dei certificati digitali.

Il **Capitolo 3** introduce l'alternativa più significativa alle CRL: l'*Online Certificate Status Protocol*, in tutte le sue varianti.

Il **Capitolo 4** presenta, in maniera molto concisa, gli altri possibili candidati come meccanismi di revoca di un certificato digitale.

Il **Capitolo 5** costituisce un'analisi sistematica delle principali minacce ai meccanismi di revoca visti nei precedenti capitoli e delle vulnerabilità più significative di questi ultimi.

Il **Capitolo 6** riporta le conclusioni relative agli argomenti trattati.

L'**Appendice A** contiene un'analisi delle possibilità offerte nell'ambito dei meccanismi di revoca dei certificati digitali da parte di uno dei browsers web attualmente più diffusi: Mozilla Firefox.

Capitolo 1

La revoca di un certificato digitale

Già nella prefazione abbiamo accennato ad alcuni importanti concetti, primi fra tutti quello di *fiducia* (to *trust*) in una *entità informatica* e quello di *Certification Authority* (CA).

Abbiamo visto come siano proprio le CA ad assumere il ruolo fondamentale di *garanti* nell'ambito di una transazione informatica, essendo proprio le CA deputate alla gestione dei certificati digitali.

In virtù di tutte queste considerazioni, è facile intuire come il livello di fiducia riponibile nel proprio *interlocutore* debba necessariamente coincidere con quello riposto nella rispettiva Certification Authority che, attraverso la sua *security policy*, definirà il periodo di attendibilità di un certificato.

1.1 Una Panoramica Generale

Solitamente, il periodo di validità di un certificato varia da un paio di mesi a due anni. Tuttavia, in alcune circostanze, esso necessita di un'operazione di *revoca*, essendo la sua validità terminata prima del previsto.

Il meccanismo di revoca costituisce una delle fasi più delicate nell'ambito della gestione dei certificati da parte di una Certification Authority.

L'utente interessato deve necessariamente essere messo a conoscenza dei tempi e delle modalità di revoca di un certificato, così come deve essere tempestivamente informato nel caso in cui ciò accada.

Il meccanismo stesso di revoca deve essere chiaro, veloce, efficace e sicuro.

L'idea di base è parecchio semplice: l'utente, durante un normale processo di verifica della validità di un dato certificato dovrà effettuare, tra gli altri, un test volto a scoprire se il certificato in questione sia stato o meno revocato.

Solitamente, ciò avviene mediante l'invio, da parte dello stesso utente, di una richiesta ad una CA (in quest'occasione spesso riferita con il termine

di *directory*), la quale dovrà essere in grado, a partire da un numero seriale contenuto nel corpo della richiesta ed identificante, in maniera univoca, un dato certificato, di fornire una valida *risposta* all'utente.

Tale risposta dovrà contenere tutte le informazioni necessarie (i.e. un numero di serie, lo stato, la data e le regioni della revoca) perché l'utente possa stabilire se fidarsi o meno del dato certificato.

1.2 Le Ragioni della Revoca

Pare opportuno, prima di imbattersi nell'analisi e nello studio dei più diffusi meccanismi di revoca di un certificato, capire appieno quali eventi possano rendere necessaria la revoca di un certificato.

- **Chiave compromessa**

La chiave privata dell'utente o della CA potrebbe essere stata compromessa o potrebbero comunque esistere ragioni che spingano a pensare ciò. La chiave potrebbe essere stata violata o rubata.

- **Cambio di affiliazione**

Alcune delle informazioni presenti nel certificato potrebbero non essere più valide.

- **Cessazione delle operazioni**

Il certificato potrebbe non essere più necessario per lo scopo per cui era stato generato.

- **Algoritmo compromesso**

L'algoritmo utilizzato per la firma potrebbe essere stato violato (a causa, ad esempio, di imprevisti sviluppi nella teoria algoritmica o di un improvviso aumento delle capacità computazionali disponibili).

- **Perdita o compromissione di *security token*, *password* o *PIN***

Tali dati potrebbero essere stati persi (o danneggiati) dal soggetto del certificato.

- **Cambio di chiave**

La chiave potrebbe non essere più adatta al suo scopo.

- **Cambio della politica di sicurezza**

La CA potrebbe cambiare politica di sicurezza o comunque non essere in grado di supportare più la vecchia politica predefinita.

1.3 Progettazione di un Metodo di Revoca

Nel corso dei capitoli 2, 3 e 4 ci preoccuperemo di affrontare quelle che sono le principali tecniche utilizzate in pratica nell'ambito dei processi di revoca di un certificato.

Possono, tuttavia, essere identificati una serie di principi e di problematiche di base che esulano dalla singola implementazione e che possono essere sintetizzati in una serie di importanti linee guida.

Nella fase iniziale della progettazione di un metodo di revoca di un certificato, notevole attenzione dovrà essere riservata alla scelta di:

- **Way of Control**

Sono fundamentalmente due i *metodi di controllo* possibili: uno di tipo *offline* ed uno di tipo *online*. Talvolta è possibile individuare una sorta di compromesso, di *trade-off*, tra le due possibilità.

Nel primo caso la validità di un certificato viene *pre-calcolata* da una Certification Authority e quindi distribuita al richiedente mediante l'utilizzo di una *non-trusted directory*.

Nel secondo caso, al contrario, l'informazione richiesta viene fornita *online* direttamente dalla CA, ovvero da una *trusted directory*. Il calcolo della validità del certificato in questione viene effettuato proprio in occasione della richiesta da parte dell'utente che, quindi, ha a disposizione informazioni sempre aggiornate.

- **Kind of List**

Durante il controllo di validità di un certificato è possibile ricorrere a due tipologie fondamentali di *liste*: *liste negative (black lists)* e *liste positive (white lists)*.

Esse corrispondono alla scelta di memorizzare (controllare), rispettivamente, una lista dei certificati revocati o una lista dei certificati validi. Anche in questo caso si ha la possibilità di utilizzare soluzioni *ibride*.

- **Way of Providing Evidence**

Il concetto di base qui è molto semplice: si parla di *direct evidence* nel caso in cui il certificato da analizzare venisse individuato direttamente all'interno di una delle due liste sopra menzionate ed una *categorizzazione* del suddetto tra *valid certificate* e *not-valid certificate* fosse immediatamente possibile. Qualora, invece, tale certificato non venisse trovato su una delle liste e, di conseguenza, si dovesse *assumere* di individuarlo nella lista *complementare*, ci si riferisce alla cosiddetta *indirect evidence*.

Chiaramente il *modo di testimoniare* risulterà fortemente dipendente dal *tipo di lista* selezionato in precedenza.

- **Information Distribution**

Il problema della *distribuzione delle informazioni* si riferisce, infine, alla tipologia dello scambio di informazioni tra le diverse entità in gioco, che potrà avvenire in modalità *push* (la CA che invia le informazioni ai clients) o *pull* (sono i singoli clients che le richiedono).

Tali problematiche, fondamentali nell'ambito del processo di progettazione di un *sicuro* metodo di revoca dei certificati, appariranno più chiare nel seguito, quando ci dedicheremo allo studio sistematico dei principali meccanismi di revoca esistenti.

Capitolo 2

Certificate Revocation List

2.1 CRL in Breve

Si definisce Certification Revocation List (o, in breve, CRL) una lista *affidabile* di certificati (più precisamente, un'elenco dei loro numeri seriali) che sono stati, per un motivo o per un altro, *revocati* e, dunque, non possono più essere considerati validi.

Essa contiene, inoltre, alcuni certificati che si trovano nel cosiddetto stato di *hold*, ovvero di *attesa*.

Si tratta, nello specifico, di certificati *revocati* in maniera non definitiva a causa di *dubbi* sulla attuale validità di un certificato (ad esempio, dovuta al dubbio che la chiave privata sia stata rubata).

Nel corso della trattazione non sottolineeremo più tale distinzione tra certificati *revoked* ed *hold*, ma ci limiteremo a trattarli come un gruppo unico.

2.2 Un po' di Storia

Le *Certificate Revocation Lists* (o, in breve, *CRL*) sono state introdotte insieme ai certificati X.509 nel 1988 dall'ITU-T (*l'International Telecommunication Union*, sino ad allora meglio nota come *CCITT*).

Con l'arrivo della seconda edizione della *X.509-Recommandation* del 1993 è stata introdotta, poi, anche una versione *improved* delle CRL, da parte di ITU-T ed ISO/IEC.

2.3 Funzionamento delle CRL

Una CRL può essere vista, per quanto detto prima, come una *black list* che fornisce una *evidence* di tipo *indiretto*: un utente cerca nella CRL un dato certificato e, non trovandolo, lo marchia come *valido*.

Essa, inoltre, si basa su un meccanismo di tipo *offline*, essendo la CRL stessa generata solitamente a predefiniti intervalli di tempo (e, opzionalmente, in occasione della revoca di un dato certificato).

La CRL è sempre gestita dalla CA che gestisce i corrispondenti certificati, per ovvi motivi.

Durante il loro (spesso breve) ciclo di vita, le CRL possono essere consultate da un'applicazione *PKI-Enabled* per verificare la validità del certificato di una controparte prima di un suo effettivo utilizzo.

Ogni CRL conterrà, allora, oltre al numero di serie del certificato revocato ed alla data della sua revoca, anche la data di generazione della CRL stessa e quella del prossimo aggiornamento previsto. In alcuni casi, è possibile che sia presente persino il *motivo* della revoca.

Infine, per prevenire attacchi informatici basati su *spoofing* o di tipo *denial-of-service*, le CRL vengono *firmate* dalla stessa CA mediante *firma digitale*.

Sarà dunque necessario andare a recuperare il certificato della stessa CA (che sarà disponibile all'interno di una *directory* - spesso pubblica -) prima di potersi *fidare* della stessa CRL.

Saranno, dunque, gli utenti a richiedere (lavorando la lista in *pull mode*) la validazione di un dato certificato, ricevendo come risposta l'intera CRL.

Una volta verificata, come già detto, l'autenticità della CRL stessa, saranno in grado di stabilire se fidarsi o meno del certificato in questione, basandosi proprio sul contenuto della CRL.

2.4 Pro e Contro delle CRL

Le CRL, come è facile intuire, costituiscono un meccanismo *semplice* (probabilmente il più semplice da concepire) per la gestione del meccanismo di revoca dei certificati e ciò spiega il loro diffusissimo uso.

Tuttavia, poiché il periodo di validità dei certificati è solitamente molto lungo ed enorme è il numero di utenti, le CRL finiscono con l'assumere dimensioni *immense*.

Ciò si traduce in un'inevitabile gigantesca mole di dati che debbono essere trasmessi nel corso della procedura di verifica della validità di un certificato.

E' proprio questo il motivo che portò, dunque, alla definizione delle cosiddette *Delta-CRLs*, liste contenenti solo le differenze tra una CRL e la sua successiva, in grado di migliorare notevolmente l'efficienza del sistema.

Ma vi è anche un altro problema, relativo al metodo di controllo *offline* che abbiamo sin qui visto caratterizzare le CRL.

Indipendentemente da dove e come lo stato un certificato sia mantenuto, il principio di base da seguire è quello di controllare sempre la validità di un certificato prima di ogni suo utilizzo. Pena la validazione di un certificato non più valido.

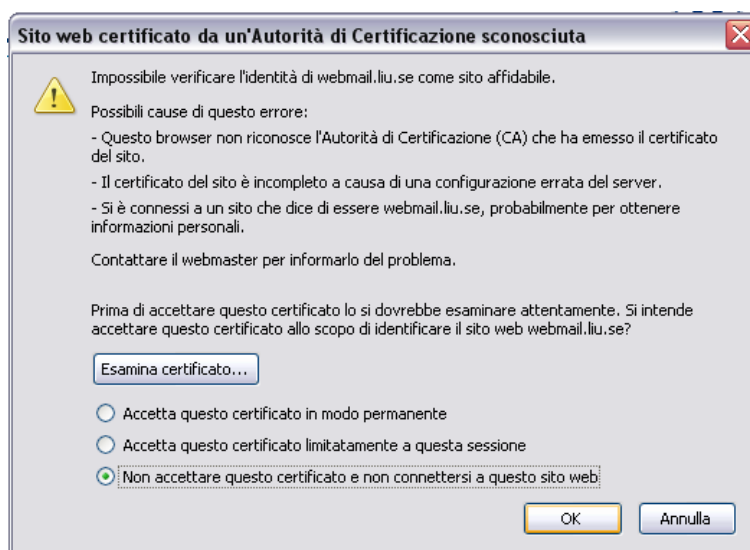


Figura 2.1: Cosa scegliereste?

Ciò comporta la necessità di un accesso alle CRLs (ovvero accesso ad Internet, nel caso di un PKI) ogni qualvolta un PKI debba essere utilizzato e, conseguentemente, la vanificazione di uno dei vantaggi principali del PKI rispetto ai protocolli di crittografia simmetrica. Non a caso i certificati erano stati etichettati come *self-authenticating certificates*.

2.5 Un Caso Eclatante

Credo che il metodo migliore per verificare la robustezza e le eventuali debolezze di un sistema di sicurezza sia quello di andare a studiare casi pratici, nello specifico violazioni realmente accadute.

E per quanto concerne le CRL, non può non ritornare alla mente il caso eclatante che ha visto come vittima niente poco di meno che mamma *Micro-*

soft e quei non proprio pochi milioni di utenti che ne utilizzano i prodotti in tutto il mondo.

Tra i lontani (ma non troppo!) 29 e 30 gennaio del 2001, la CA *VeriSign, Inc.* rilasciò, infatti, due certificati ad un individuo che si spacciò per un dipendente della *Microsoft Corporation*.

Ogni software firmato da quei due certificati appariva come legittimamente firmato da Microsoft che, invece, ne era completamente all'oscuro.

Il bollettino Microsoft in cui venne reso pubblico il misfatto venne rilasciato il 22 marzo dello stesso anno, quasi 2 mesi dopo l'accaduto.

Per di più, in quel caso, pur avendo la CA, una volta scoperto l'accaduto, inserito i due certificati in una opportuna CRL, ogni singolo utente fu costretto ad installare una patch sulla propria macchina, in quanto Internet Explorer (l'allora più diffuso browser web) non controllava le *revoche* in maniera automatica.

D'altra parte, né IE, né gli altri browser sarebbero stati in grado di cercare informazioni riguardanti un'eventuale revoca di tali certificati, non essendo stata inclusa, all'interno degli stessi certificati, alcuna informazione a riguardo da parte della stessa autorità di certificazione (i.e. La VeriSign, Inc.).

Si noti che questo *piccolo* inconveniente mise l'ignoto *cracker* nelle condizione di eseguire qualunque tipo di codice sulle macchine degli utenti che avessero approvato l'esecuzione di quel pacchetto software *firmato Microsoft!*.

Capitolo 3

Online Certificate Status Protocol

3.1 OCSP in breve

L'*Online Certificate Status Protocol* (in breve *OCSP*) costituisce probabilmente l'alternativa più significativa alle CRLs nell'ambito del processo di revoca di un certificato.

Introdotta dall'*IETF* (*Internet Engineering Task Force*), è stato presentato come un protocollo in grado di “condurre verifiche in tempo reale, risparmiando tempo e denaro, e fornendo alle attività di e-business un sistema più rapido, semplice e affidabile per la validazione dei certificati digitali rispetto a quello offerto dal tradizionale scaricamento ed elaborazione delle CRL”.

Pur essendo stato progettato esplicitamente per i certificati X.509, esso può lavorare anche con altri tipi di certificato.

Esso può, inoltre, essere utilizzato *congiuntamente* alle CRL oltre che come alternativa a queste ultime, potendo le informazioni riguardanti il *metodo* di verifica da utilizzare essere incluse all'interno del certificato X.509 stesso (negli *extension fields*, per essere precisi).

3.2 Funzionamento di OCSP

L'architettura del protocollo è di tipo *client-server*, dove da una parte vi sono i singoli *clients* (*requesters*) e dall'altra vi sta il *server* (*responder*).

Saranno i *clients* a generare richieste di tipo *OCSP Request*, per controllare la validità di uno o più certificati.

A quel punto il *server* risponderà mediante una *OCSP Response*, contenente un cosiddetto *messaggio di validità* (*good*, *revoked* oppure *unknown*), indicante proprio lo stato del certificato richiesto.

Si rifletta sul fatto che l'identificativo *good* potrebbe assumere, in un simile contesto, tre differenti significati:

- Il certificato non è stato revocato;
- Il certificato potrebbe non essere stato ancora gestito;
- L'istante in cui la risposta è stata prodotta non si trova all'interno del periodo di validità del certificato.

Alla stessa maniera, l'identificativo *revoked* potrebbe riferirsi sia ad un certificato realmente revocato o ad uno in stato di *hold*.

Infine, l'identificativo *unknown* sta a significare che il server non è in possesso di informazioni significative riguardanti lo specifico certificato richiesto.

Insieme al *messaggio di validità* è, poi, solitamente presente un *intervallo di validità* a specificare l'istante effettivo in cui lo stato indicato nella risposta è considerato *corretto*.

Infine, è possibile rintracciare all'interno della risposta stessa un'informazione riguardante futuri aggiornamenti sullo stato del certificato in questione o ulteriori informazioni.

La *OCSP Response* dovrebbe essere firmata digitalmente o dal server o dalla CA.

Il formato delle *Requests* e delle *Responses* può essere differente, a causa del protocollo di trasmissione utilizzato (ad esempio HTTP o LDAP).

In realtà, oggi si parla sempre più di una sorta di variante del modello appena visto (oggi meglio noto come *T-OCSP*, ovvero *Traditional-OCSP*): il *D-OCSP*, ovvero il *Distributed-OCSP*.

Rispetto al precedente modello, quest'ultimosi basa sul principio di tenere separati i dati sensibili e le applicazioni da proteggere dalle procedure di fornitura dello stato di validità dei certificati degli utenti collegati.

Le due architetture relative a T-OCSP e D-OCSP sono illustrate di seguito:

3.3 Pro e Contro di OCSP

OCSP ha alcuni evidenti vantaggi rispetto alle CRL:

- Elimina la necessità per i clients di scaricare e analizzare le liste di revoca;
- Rende più efficiente l'utilizzo della banda, dal momento che un messaggio OCSP ha una dimensione trascurabile rispetto alle CRL;

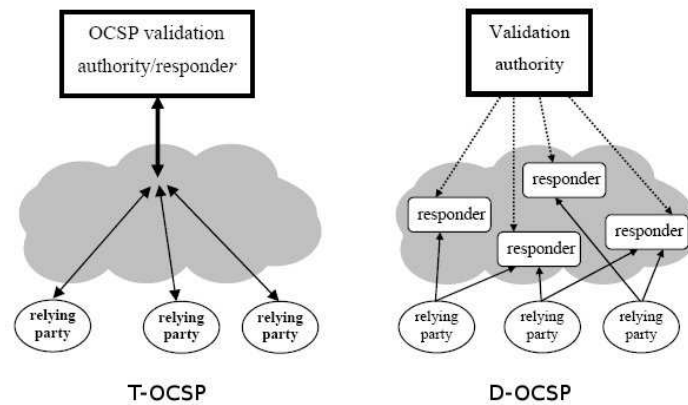


Figura 3.1: Architetture T-OCSP e D-OCSP a confronto.

- Supporta una catena fidata di OCSP richiesta tra i vari *responder*. Questo permette ai clienti di comunicare con un *responder* fidato per interrogare un altro *responder*.

Inoltre, grazie al suo meccanismo di funzionamento, esso fornisce informazioni riguardanti i certificati in maniera più tempestiva rispetto a qualunque altro meccanismo di revoca.

D'altra parte, un suo utilizzo implica la necessità di accesso ai server (ancora una volta si legga in termini di *connessione ad Internet*) ogni qualvolta sia necessario effettuare un controllo sull'autenticità di un dato certificato.

Infine, è necessaria un'attenta analisi dei meccanismi di *caching* dei vari *HTTP-Browsers*, che potrebbero rendere l'intero sistema inaffidabile.

Capitolo 4

Ulteriori Meccanismi di Revoca

Nella nostra trattazione ci siamo limitati ad analizzare i due meccanismi di revoca attualmente più diffusi (e, probabilmente, didatticamente più significativi).

Ciò non toglie che nel corso degli anni si sia sviluppato un buon numero di tecniche alternative che stravolgono o meno le due sin qui studiate e che trovano più o meno larga diffusione nelle applicazioni odierne.

Il *CRS* (*Certificate Revocation System*) sviluppato dall'italiano Silvio Micali nel 1995 si basava, ad esempio sull'utilizzo di firme *online/offline* e di liste ibride *black & white* (In medium stat virtus?).

Ancora, i *CRTs* (*Certificate Revocation Trees*) introdotti da Paul Kocher nel 1998 si soffermavano sul bisogno di *efficienza* del sistema di revoca e sfruttavano per lo scopo quella insita negli *hash trees*.

Capitolo 5

Minacce ai Meccanismi di Revoca

A questo punto della nostra trattazione, ci sembra quanto meno opportuno dedicare lo spazio che segue all'analisi delle principali vulnerabilità dei sistemi sin qui studiati, ai possibili attacchi cui potrebbero essere sottoposti ed alle eventuali contromisure adottabili.

In particolare, nel processo di valutazione del livello di sicurezza fornita dai tre sistemi di convalida sin qui analizzati, focalizzeremo la nostra attenzione sui seguenti *attacchi*:

- Attacchi di tipo DoS (Denial of Service)
- Intrusione
- Aggiornamento della risposta di validità
- Riproduzione

5.1 Denial of Service

Con l'espressione *attacco di tipo Denial of Service* ci si riferisce a tutta quella categoria di attacchi volti ad impedire il corretto funzionamento di un qualche sistema.

Questa tipologia di attacco, in grado di colpire praticamente chiunque utilizzi un dato servizio, è stata adottata molto di frequente, specie in tempi recenti.

Chiaramente, i sistemi maggiormente soggetti a questo tipo di attacchi sono risultati essere i sistemi maggiormente centralizzati e più *lenti* nell'eseguire una singola transazione.

Nel caso specifico dei servizi di convalida dei certificati, si potrebbe verificare che un *attaccante* (o un gruppo di *attaccanti*) sovraccarichi il servizio stesso di validazione attraverso un massiccio numero di false richieste di validazione.

Si noti come lo stesso effetto potrebbe essere provocato, in maniera più generale, da un reale imprevedibile numero di contemporanee richieste di verifica o, ancora, dalla perdita di accesso al server o ai servers fornitori del servizio di validazione dei certificati, provocando, di fatto, un autentico DoS per tutti gli utenti.

Tutti e tre i sistemi di validazione sin qui analizzati (CRL, T-OCSP e D-OCSP) sono vulnerabili a questo tipo di attacco.

In particolare, il metodo CRL risulta il più vulnerabile da questo punto di vista a causa dei grossi files CRL da scaricare (Per l'americano *Department of Defense* tali files assumono dimensione pari a 5-6 MB!) in occasione di ogni richiesta.

Lo stesso metodo T-OCSP è altamente suscettibile di attacchi di tipo Denial of Service, essendo ogni risposta firmata digitalmente in tempo reale; processo, quest'ultimo, spesso molto lungo.

La migliore soluzione sembra, dunque, essere offerta proprio dal metodo D-OCSP, il meno vulnerabile a questo tipo di attacchi in virtù della sua architettura di tipo fortemente distribuito e dei tempi di generazione delle singole risposte molto più rapidi rispetto a quelli degli altri due meccanismi di revoca.

5.2 Intrusione

Con il termine "intrusione" ci si riferisce ad un deliberato tentativo di penetrare in un sistema informatico, magari con lo scopo di sfruttare o compromettere dati o informazioni sensibili (i.e., revocando certificati validi e/o rendendo validi certificati revocati).

Chiaramente, ogni sistema connesso in rete è potenzialmente soggetto ad un attacco di tipo intrusivo.

Nel caso specifico, il sistema T-OCSP risulta quello più esposto al rischio di una intrusione, essendo l'unico a consentire traffico in ingresso al server, al contrario di CRL e D-OCSP.

Le contromisure standard anti-intrusione (quale, ad esempio, l'utilizzo di un firewall) possono in questo caso essere adottate.

5.3 Aggiornamento della risposta di validità

Un meccanismo di revoca dei certificati è efficace in maniera direttamente proporzionale al grado di aggiornamento delle risposte di validità fornite.

Chiaramente, la garanzia di un aggiornamento in termini assoluti non sarà in alcun modo possibile, esistendo comunque una qualche *finestra di vulnerabilità* più o meno ampia, direttamente dipendente dalla tempestività degli aggiornamenti stessi delle risposte di validità.

In linea generale, un aggiornamento dovrebbe scattare nel momento esatto della ricezione, da parte di una Certification Authority, di un nuovo elemento di revoca.

Un approccio di tale tipo, tuttavia, risulta totalmente impraticabile nella maggior parte dei casi.

Lo stesso Department of Defense, ad esempio, ha un numero di utenti pari a circa 4,4 milioni ed un tasso di revoca dei certificati pari al 17% (uno ogni 2 minuti!).

Non essendo possibile rilasciare un nuovo CRL ogni 2 minuti, l'attuale policy del DoD prevede di rilasciare un nuovo CRL ogni 24 ore, valido per le successive 96 ore.

In questo caso, la finestra di vulnerabilità del sistema oscillerà proprio tra le 24 e le 96 ore, a seconda dell'istante di ricezione, da parte dell'utente, della versione aggiornata della CRL.

Anche in questo caso, tutti e tre i sistemi di revoca visti risultano vulnerabili ad un aggiornamento non tempestivo delle risposte di validità.

Si noti come tale aggiornamento, in entrambi i casi OCSP Tradizionale e Distribuito, non sia determinato dall'istante in cui la risposta è firmata, ma da quello in cui la Certification Authority ha rilasciato un elemento di revoca via CRL.

Quanto appena detto è illustrato nella figura 5.1.

Chiaramente, il rischio principale dovuto ad attacchi basati sulla mancata tempestività dei servizi di aggiornamento delle liste dei certificati revocati è rappresentata dalla possibilità che un certificato revocato sia ancora considerato valido, finché il dato di revoca non sia regolarmente ricevuto dalla relativa applicazione.

La minaccia più significativa, in questi termini, è associata ai certificati revocati per dimissioni o termine dell'incarico di un legittimo possessore, che potrebbe continuare ad autenticarsi al sistema, sfruttandone le risorse.

Per quanto riguarda i certificati revocati a causa di una potenziale compromissione, al contrario, essi risultano difficili da sfruttare, richiedendo spesso l'accesso alle chiavi private una qualche autenticazione da parte dell'utente.

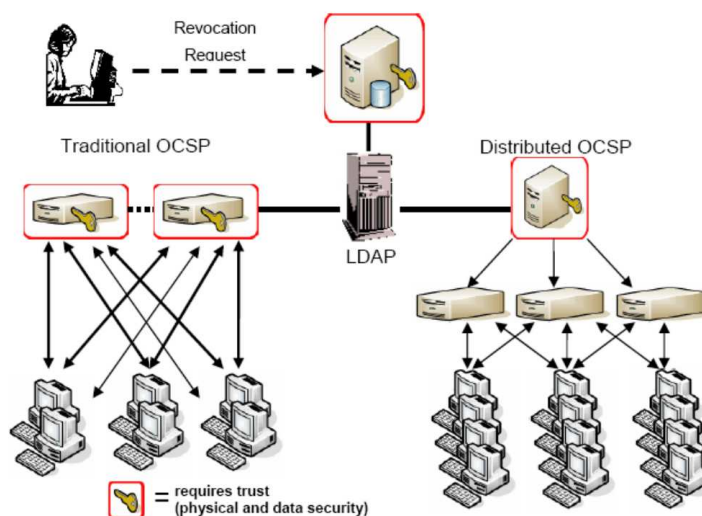


Figura 5.1: Aggiornamento della risposta di validità ed OCSP.

L'unica valida contromisura appare, dunque, quella di proibire in una qualche maniera l'accesso alle chiavi private da parte dell'*utente dimesso*.

5.4 Riproduzione

Con il termine di *riproduzione* ci si riferisce al processo di registrazione di un legittimo messaggio da parte di una persona che intende utilizzarlo senza autorizzazione.

Si consideri, ad esempio, lo scenario in cui un *attaccante* riesca ad *intercettare* le versioni aggiornate di una CRL inviate ad un utente e che le sostituisca con delle *copie* di una vecchia CRL nella quale un certificato ormai revocato risulti ancora perfettamente valido (Alternativamente, potrebbe essere *clonata* la risposta di stato del certificato nel caso di OCSP).

Lo sfruttamento di tale vulnerabilità risulta tecnicamente molto complesso, ma non del tutto impossibile.

Innanzitutto, l'attaccante dovrà essere a conoscenza del certificato in fase di revoca (ipotesi estremamente *pesante* nel caso di violazioni occasionali).

La preparazione alla violazione del certificato dovrà, quindi, partire in concomitanza con il rilascio dell'ultima CRL in cui il certificato in questione risulti ancora valido.

L'attaccante dovrà, a questo punto, inviare una richiesta all'utenza che

tenta di violare utilizzando il certificato valido (dovrà, dunque, possederne anche la chiave privata).

Ciò comporterà l'emissione di una richiesta di validità da parte dell'utente, la cui risposta sarà captata e memorizzata.

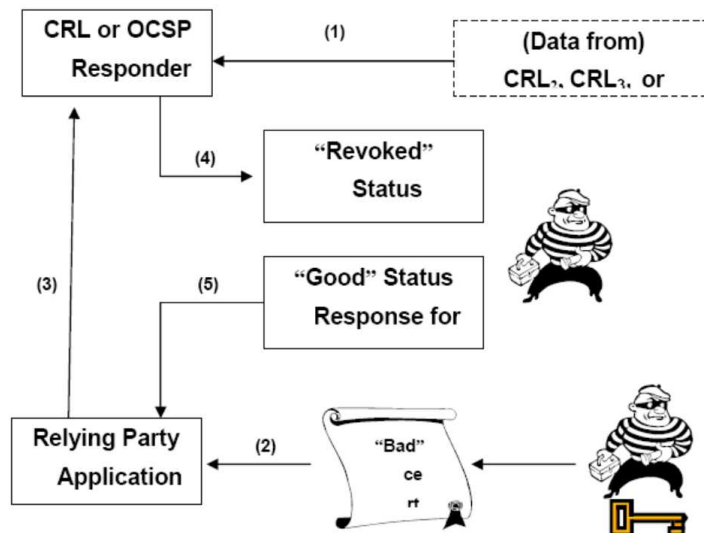


Figura 5.2: Un attacco di tipo *riproduzione*.

Si noti come non tutti i certificati siano vulnerabili a questo tipo di attacco, ma solo quelli revocati durante il periodo di validità di un dato CRL.

Si potrebbe allora pensare di ridurre il periodo di validità delle CRL al fine di ridurre il correlato periodo di vulnerabilità, con le conseguenze già viste in precedenza.

Nel metodo T-OCSP la minaccia di riproduzione può essere virtualmente eliminata mediante l'utilizzo di un numero casuale. L'utente potrebbe inserire tale elemento ("nonce") all'interno della richiesta inviata al responder che potrebbe, a sua volta, includere lo stesso dato (o una sua eventuale elaborazione) all'interno della risposta fornita.

Tale stratagemma, se da una parte risolve il problema, dall'altra incrementa i tempi di elaborazione delle singole transazioni, aumentando la possibilità di DoS ed eliminando l'uso di risposte "nascoste", critiche per un sistema che serve un grande numero di utenti.

D'altra parte, lo stesso stratagemma si rivelerebbe inutile nel caso in cui l'attaccante fosse in grado di "editare" al volo la risposta del reale responder, inserendo in essa il parametro "casuale" desiderato.

Capitolo 6

Conclusioni

Cominciamo con l'affermare che *non esiste*, almeno in linea generale, un certificato *migliore* degli altri, ma che l'utilizzo di un meccanismo di revoca piuttosto che di un altro dovrà dipendere dallo scopo primario prefissato.

Come sempre in questi casi, il *costo* di una soluzione rispetto ad un'altra andrà considerato prepotentemente.

Si noti come, in questo caso, il costo non sia per forza legato, ad esempio, alla quantità di dati trasmessa ma anche a specifiche quali la disponibilità tempestiva di dati o il grado di sicurezza da fornire al sistema da gestire.

L'utilizzare meccanismi di verifica di tipo *offline* potrebbe tradursi in costi molto bassi da una parte, ma in tempi di aggiornamento molto lunghi ed un grado di sicurezza insufficiente dall'altra.

Ulteriori più sottili considerazioni andrebbero, poi, effettuate nel caso in cui i meccanismi di revoca dovessero finire con l'interessare i cosiddetti *storage equipment*, quali *smart card* o *security tokens*.

La conoscenza dei vari meccanismi di revoca non è ancora molto diffusa ed alternative efficaci ed efficienti sono tuttora al vaglio degli studiosi.

Uno dei problemi fondamentali da affrontare consiste nel riuscire ad integrare i nuovi eventuali meccanismi con i certificati attuali, talmente diffusi da non poter essere sostituiti facilmente.

E probabilmente questo bisogno di backward-compatibility finirà con il ritardare sempre più nuove soluzioni in questo ambito.

Appendice A

CR in Pratica: Mozilla Firefox

Prima di concludere questo lavoro, vorremmo analizzare un po' più a fondo le possibilità offerte nell'ambito dei meccanismi di revoca dei certificati da parte di uno dei browser web attualmente più efficienti e diffusi: Mozilla Firefox.

In particolare, ne è stata analizzata la versione 1.5.0.8.



Figura A.1: Mozilla Firefox 1.5.0.8.

Attraverso il menu Strumenti -> Opzioni -> Avanzate è possibile accedere alla sezione riservata alle impostazioni relative alle procedure di gestione dei certificati.

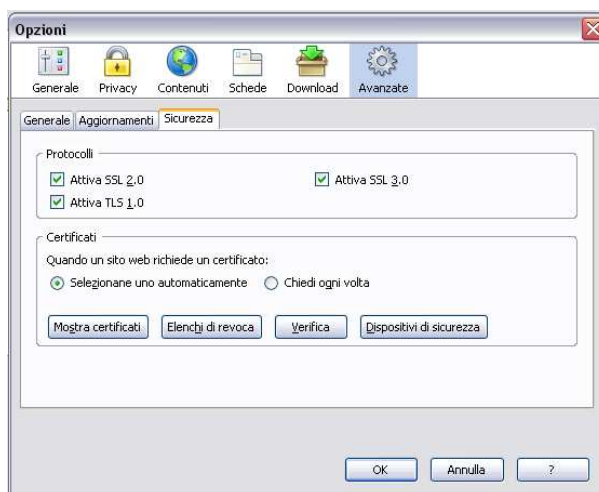


Figura A.2: La schermata *Opzioni Avanzate di Sicurezza* di Firefox.

Un semplice click sulla voce “Elenchi di revoca” consente di accedere ad una schermata di gestione delle CRL.

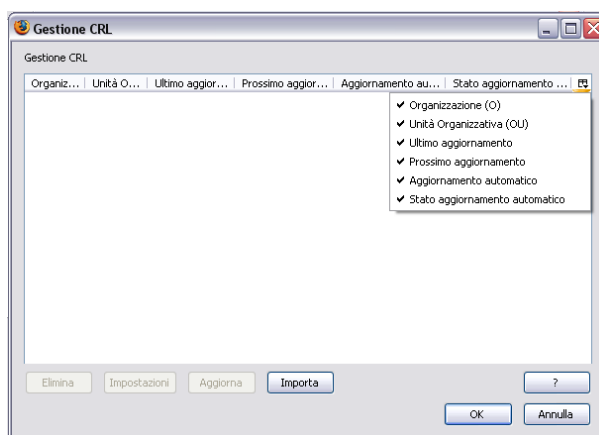


Figura A.3: Gestione delle CRL in Firefox.

Attraverso tale schermata è possibile accedere ad informazioni importanti relative alle singole CRL (nome dell’organizzazione ed unità organizzativa, ultimo aggiornamento, aggiornamento successivo, livello dell’automazione del relativo aggiornamento).

E’ anche possibile importare uno o più elenchi dei certificati di revoca.



Figura A.4: Importazione di una CRL in Firefox.

Tra le altre possibilità offerte dal browser, anche quella di impostare un eventuale utilizzo di OCSP per le operazioni di verifica dei certificati.

E' possibile evitare l'utilizzo di OCSP, utilizzare tale meccanismo per verificare solo i certificati che specificano un indirizzo ad un servizio OCSP o selezionare un firmatario ed un indirizzo per OCSP da un ricco elenco fornito.

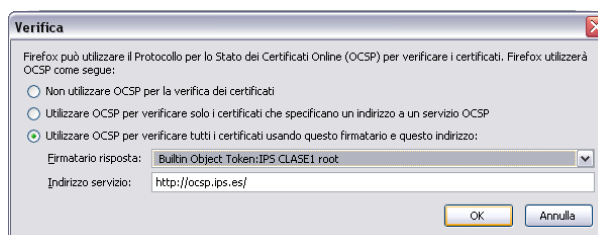


Figura A.5: OCSP e Mozilla Firefox.

Infine, sono disponibili una serie di *application-specific plugins* gratuiti per Mozilla Firefox relativi alla gestione dei certificati digitali in diversi ambiti. Noi abbiamo provato il *DoD Plugin*, utile per chi è interessato ad utilizzare i servizi offerti dall'*American Department of Defense*.

Bibliografia

- [1] D.Giacomini,
Appunti di Informatica Libera, Cap. 272.
- [2] P.Wohlmacher,
Digital Certificates: A Survey of Revocation Methods.
- [3] AA.VV.,
CRL @ Wikipedia, the Free Encyclopedia
- [4] AA.VV.,
OCSP @ Wikipedia, the Free Encyclopedia
- [5] N.A.,
<http://www.cert.org/advisories/CA-2001-04.html>
- [6] S.Even, O.Goldreich, S.Micali,
On-line/Off-line Digital Signing, Proc. of Crypto 89, pp. 263-275.
- [7] P.Kocher,
A Quick Introduction to Certificate Revocation Trees
- [8] N.A.,
<http://www.corestreet.com/about/library/international/vulnerabilita-it-v1.pdf>