ESAME - Sicurezza e Affidabilità

Fabio Ferrario
@fefabo

2023/2024

Indice

Capitolo 1

Crittografia

1.0.1 Crittografia Simmetrica

Nella crittografia simmetrica si utilizza una sola chiave, sia per cifrare che per decifrare un messaggio: Il mittente cifra il messaggio con la chiave k e lo invia al destinatario, che dovrà decifrare il messaggio con la stessa chiave k

$$\text{decrypt}(\text{encrypt}(\text{message},k))$$

Numero di Chiavi Ogni coppia di utenti ha bisogno di una chiave segreta per potersi scambiare messagi, quindi il numero di chiavi necessario è $\frac{N(N-1)}{2}$, con N numero di utenti.

1.0.2 Crittografia Asimmetrica

Nella crittografia asimmetrica ogni utente genera una coppia di chiavi (legate matematicamente), una pubblica e una privata. La chiave pubblica è accessibile in chiaro a tutti, quella privata invece rimane segreta e conosciuta solo a chi l'ha generata. Queste chiavi sono fatte in modo che se una viene utilizzata per cifrare un messaggio esso potrà essere decifrato solo con l'altra. Quindi se devo inviare un messaggio a un destinatario lo cifrerò con la sua chiave pubblica, di modo che soltanto esso potrà decifrarlo con la sua chiave privata.

Numero di Chiavi Nella crittografia asimmetrica è necessaria una coppia di chiavi per ogni utente, quindi $2N$ chiavi.

1.0.3 Firme Digitali

Le Firme Digitali sono utilizzate per verificare la provenienza e l'integrità di un messaggio.
Firmare un messaggio intero è computazionalmente molto costoso, quindi generalmente si effettua prima un digest (MD5) del messaggio e poi si usa RSA per crittare il digest. Il messaggio viene quindi inviato al destinatario insieme al digest crittato, che verificherà l'identità del mittente decrittando il digest con la chiave pubblica dello stesso.
In questo modo avrà sia conferma del mittente, sia la conferma che il messaggio è integro verificando a sua volta con il digest del messaggio inviato.
Quindi i passaggi sono:

1. Il mittente crea il digest del messaggio da inviare.
2. Cifra poi il digest del messaggio con la sua chiave privata ottenendo così la firma digitale.
3. Infine invia il messaggio insieme alla firma.
4. Il destinatario, ricevuti il messaggio e la firma, decifra la firma con la chiave pubblica del mittente ottenendo così il messaggio e il relativo digest.
5. Per controllare l'autenticità del messaggio genera a sua volta un digest di esso e lo confronta con il digest ricevuto.

1.0.4 Domande

- Si spieghino le differenze tra Crittografia Simmetrica e Asimmetrica

- Si descriva la relazione fra numero di chiavi e utenti per la crittografia simmetrica e asimmetrica

- Si spieghi il funzionamento della crittografia a **chiave pubblica**, indicando in particolare come si può usarla per implementare **firme digitali**

- Si descriva come si possono combinare crittografia a chiave asimmetrica e algoritmi di Message Digest per ottenere meccanismi efficienti di **firma digitale**

- Perché la crittografia DES non é piú considerata sicura? Quali modi di suo utilizzo potrebbero migliorare la sicurezza?

Capitolo 2

Sistemi Operativi

2.0.1 DAC

Nel modello DAC (Discretionary Access Control) il proprietario di una risorsa ne concede l'accesso ad altri utenti a sua discrezione. è quindi il proprietario della risorsa che decide ogni singolo utente che può accedere ad essa.

esempi Un esempio è google documenti, in cui il proprietario decide di dare accesso a una sua risorsa a un utente singolarmente.

2.0.2 MAC

Nel modello MAC (Mandatory Access Control) il sistema impone un modello che limita e controlla la discrezionalità degli utenti nell'assegnare i diritti di accesso alle risorse.

esempi Un esempio è la sicurezza multi livello militare, in cui ad ogni utente e risorsa è assegnato un livello di segretezza.

2.0.3 Access Control List

Una Access Control List è una lista dove, data ogni risorsa, possiamo inserire chi ha diritto e che diritti ha (Lettura, scrittura, esecuzione). Una ACL ci permette di risparmiare spazio rispetta alla tabella completa, è facile determinare la lista degli utenti che hanno un diritto specifico su una risorsa condivisa ed è facile revocare/modificare l'accesso di un utente ad una risorsa.

Funzionamento Per una risorsa la ACL contiene una lista di utenti o di gruppi e i rispettivi diritti (ownership, read, write, execute).

ACL in Unix Unix utilizza un modello semplificato della ACL che utilizza solo 9 bit di protezione (per file). Vengono definiti i permesi soltanto per L'owner, un gruppo definito, e il resto del mondo. Per ognuno di essi abbiamo 3 bit (R,W,X) che definiscono i rispettivi permessi.

Questo tipo di ACL è molto più piccola ma permette una granularità molto inferiore.

2.0.4 Principi di Mediazione Completa e Privilegi Minimi

Mediazione Completa Ogni tentativo di accesso deve essere controllato.

2.0.5 Bell-LaPadula

Il modello Bell-LaPadula è un modello MAC multilivello che nasce in ambito militare per garantire la confidenzialità dei dati. In questo tipo di modello gli utenti e le risorse sono classificati secondo dei livelli di sicurezza, in modo che un utente possa accedere ad un dato solo se il suo livello è maggiore o uguale a quello della risorsa.

Bell-LaPadula implementa due proprietà:

- No Read Up: Un soggetto non può leggere oggetti di livello più alto. (Simple Security)
- No Write Down: Un soggetto non può scrivere oggetti di livello più basso (impedendogli di trasferire documenti del suo livello a livelli più bassi). (Confinement)

Bell-Lapadula può essere esteso con **compartimenti**, in cui ad ogni risorsa e utente è assegnato un compartimento, e un utente può accedere ad una risorsa sse il suo livello è maggiore o uguale a quello della risorsa e se fanno parte dello stesso compartimento.

Autenticazione Challenge-Response

Nei protocolli Challenge-Response la Password è una funzione matematica F(x) che l'utente sa calcolare. Per l'autenticazione il sistema propone un valore X e l'untente deve rispondere con il valore corretto $F(X)$.

C-R con chiave Simmetrica Nell'autenticazione C-R con chiave simmetrica, client e server condividono una chiave segreta K.
In questo caso avremo che:

- Challenge: Il sistema passa all'utente un messaggio.

- Response: L'utente codifica il messaggio con la chiave segreta K.
- Autenticazione: Se il messaggio può essere decodificato con successo.

In questo caso la Chiave non viene mai mandata sulla rete.

Nonce La challenge viene cambiata di volta in volta, altrimenti basterebbe intercettre il messaggio cifrato per poter accedere.

2.0.6 Domande

Si descriva la differenza tra **DAC** e **MAC**, facendo un esempio di una politica di controllo degli accessi per ognuno dei due tipi

Si descriva il funzionamento di una Access Control List

Si descriva il funzionamento di una Access Control List basata su 9bit come quella di Unix/Linux

Si descrivano i concetti di Mediazione Completa e Principio dei Privilegi Minimi

Descrivere il modello Bell-LaPadula completo

Si spieghi come la crittografia simmetrica possa essere usata per implementare un algoritmo di autenticazione **challenge-response**

Si descriva Challenge-Response con chiave Asimmetrica

Capitolo 3

Sicurezza nelle Applicazioni

3.0.1 Attacco SQL-Injection

Funzionamento L'attacco SQL Injection consiste nell'insetimento di una query SQL attraverso l'input utente al fine di modificare l'esecuzione di comandi SQL parametrici.
Un attacco eseguito con successo può:

- Leggere/Modificare dati sensibili
- Eseguire operazioni di amministrazione del database
- Recuperare il contenuto di un file presente nel filesystem del dbms.

SQL Injection si potrebbe manifestare in tutti i programmi che fanno uso di un DBMS.

Contromisure Le contromisure per questo tipo di attacco sono:

- Encoding sull'input: Trasformare i caratteri potenzialmente terminatori in una certa codifica (<→ <).
- Validazione dell'input: Accettarlo o rifiutarlo in base ad una white/black list per permettere di evitare input potenzialmente dannosi.

3.0.2 Attacco Buffer Overflow

Funzionamento Un attacco di tipo Buffer-Overflow, dovuto alla mancanza di controli OutOfBound, consiste nello scrivere in un buffer uscendo intenzionalmente dai limiti dello stesso, andando a scrivere in aree di memoria non assegnategli adiacenti. Questo permette di modificare vaari valori, tra cui variabili locali e l'activation record (che contiene il return address) di una funzione.

In genere questo tipo di attacco cerca di sovrascrivere il return aress delle funzioni in esecuzione, in modo da farlo puntare ad una funzione malevola inserita dall'attaccante nel buffer stesso.

ShellCode Il codice malevolo solitamente è uno shellcode, ovvero un programma che tramite una chiamata di sistema permette di sostituire il processo corrente con uno nuovo (una shell) ed effettuare una syscall per ottenere privilegi Kernel.

Difficoltà Le due difficoltà principali nell'uso di questo exploit sono:

- Modifica del Base Pointer, se esso viene modificato e presenta un valore inconsistente, il programma termina.
- Bisogna indovinare la posizione esatta del return address.

Contromisure Le principali contromisure sono:

- Modifica degli indirizzi di memoria virtuale tra un'esecuzione e l'altra.
- Introduzione di un Canary Value, che cambia ad ogni esecuzione e viene inserito dopo il base pointer.