Autenticazione nei sistemi distribuiti

Lezione 16 di Sicurezza dei sistemi informatici 1

Docente: Giuseppe Scollo

Università di Catania, sede di Comiso (RG)
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Studi in Informatica applicata, AA 2008-9

Indice

problemi salienti

in che cosa differisce l'autenticazione in un sistema distribuito da quella tradizionale di un utente di un sistema operativo?

- il **canale di comunicazione: non** si può assumere che sia **sicuro**
- l'**utente:** non necessariamente una persona, più in generale un processo, **soggetto** di richieste di
  - **accesso a oggetti** per determinate operazioni su essi
  - **comunicazione** autenticata con altri **soggetti**

problemi di autenticazione che derivano da tali differenze:

- definizione di **identità** dei soggetti
- autenticazione basata solo su *"cosa conosce"*
- autenticazione come **servizio**, che fornisce meccanismi di identificazione protetti, possibilmente basato su *trusted third-party* (TTP)

la protezione crittografica è di largo impiego nelle soluzioni di questi problemi, ma la sicurezza che ne consegue **non** dipende solo dalla sua (in)violabilità

autenticazione remota

l'autenticazione basata su password associata a un identificatore di utente è frequente nelle interazioni *client-server* anche in sistemi distribuiti

ad es., accesso HTTP autorizzato a una risorsa protetta

vulnerabilità: la password è un **segreto condiviso** da *client* e *server*, la cui trasmissione in chiaro la espone al rischio di cattura da *password sniffers*

- ***Basic Access Authentication*** : trasmissione in chiaro (in codice base64)
- ***DigestAccess Authentication*** : uso di una funzione di *hash* crittografico h (e.g. MD5) in un protocollo *challenge-response* fra *server* e *client* in cui
  - il *server* reagisce alla richiesta di accesso a risorsa protetta inviando al *client* un identificatore di *realm* di protezione $\mathbf{r}$ e un ***nonce*** $\mathbf{n}$ (usato solo una volta)
  - il *client* risponde alla sfida inviando in chiaro l'identificatore di utente $\mathbf{u}$, il *nonce* $\mathbf{n}$ e il ***digest*** della richiesta: $h(h(\mathbf{u} \| \mathbf{r} \| \mathbf{p}) \| \mathbf{n} \| h(\mathbf{m} \| \mathbf{i}))$, dove
    - $\mathbf{p}$ è la password di $\mathbf{u}$
    - $\mathbf{m}$ è il metodo HTTP di accesso
    - $\mathbf{i}$ è l'URI della risorsa protetta

autenticazione e scambio di chiavi

l'autenticazione ha spesso luogo nella fase di formazione di una sessione di accesso del soggetto a un oggetto, o di comunicazione fra soggetti, come parte dello scambio di chiavi, tuttavia...

autenticazione e scambio di chiavi sono concetti diversi, non sempre associati

terminologia:

peer entity authentication (ISO 7498-2, 1989): concomitante alla formazione di una sessione

entity authentication (ISO/IEC 9798-1, 1991): meccanismo svincolato dalla formazione di una sessione, permette la *dead peer detection*

autenticazione **unilaterale** oppure **mutua**

key establishment, responsabilità: *key transport* oppure *key agreement*

key establishment, garanzie:

- *key authentication*
- *key confirmation*
- *explicit key authentication* : entrambe le precedenti

STS: Diffie-Hellman rivisitato

un esempio di *key agreement* è il protocollo di Diffie-Hellman

tuttavia esso non dà alcuna garanzia di autenticità del partner della comunicazione!

un intruso che abbia accesso al canale di comunicazione può portare l'attacco detto ***"man in the middle"***, presentandosi a ciascun partner come se fosse l'altro

il protocollo ***station-to-station* (STS)** (Diffie, van Oorschot & Wiener, 1992), estende quello di Diffie-Hellman con un meccanismo di mutua *explicit key authentication*, usando a tal fine (senza tuttavia prefissarli):

- un algoritmo di crittografia e
- un algoritmo di firma digitale s

sia $K = g^{ab} \bmod p$ la chiave Diffie-Hellman che si stabilisce, e S_a, S_b le rispettive chiavi di firma dei partner A, B; lo scambio STS consta dei seguenti passi (exp. mod p è intesa):

1. $A \rightarrow B$: g^a
2. $B \rightarrow A$: $g^b, e_K(s_{S_b}(g^b, g^a))$
3. $A \rightarrow B$: $e_K(s_{S_a}(g^a, g^b))$

autenticazione in Kerberos (1)

il sistema Kerberos (Miller, Neuman, Schiller & Saltzer, 1987) ha origine dal protocollo di (Needham & Schroeder, 1978), di *key transport*

autenticazione di utenti a servizi in un sistema distribuito, fornita da un server centrale di autenticazione (KAS) che distribuisce chiavi di sessione

il KAS condivide con ciascun utente A o servizio B un chiave segreta di lungo termine, risp. K_{as}, K_{bs}, ha un database che associa le identità di utenti e servizi a tali chiavi, e genera una chiave di sessione K_{ab} di durata limitata L per ciascuna sessione di A con B, alla quale associa un ***ticket*** $t_B = e_{K_{bs}}(K_{ab},A,L)$

in prima approssimazione, lo schema del protocollo è il seguente, dove n_a è un *nonce* generato da A e T_a è un *timestamp* riferito al clock di A:

1. $A \rightarrow S$: A, B, n_a
2. $S \rightarrow A$: $e_{K_{as}}(K_{ab},n_a,L,B), t_B$
3. $A \rightarrow B$: $t_B, e_{K_{ab}}(A,T_a)$
4. $B \rightarrow A$: $e_{K_{ab}}(T_a)$

autenticazione in Kerberos (2)

una migliore approssimazione al funzionamento effettivo del protocollo si ha considerando la distribuzione dei *ticket* di accesso ai servizi come un servizio essa stessa, fornito da appositi *ticket-granting server* (TGS), ai quali l'utente si autentica via KAS, che genera:

la chiave di sessione $K_{a,tgs}$ fra A e TGS

il *ticket-granting ticket* (TGT) $t_{A,TGS} = e_{K_{tgs}}(K_{a,tgs},A,L_2)$

dove: K_{tgs} è una chiave condivisa da KAS e TGS

la durata L_2 della sessione da stabilirsi fra A e B è generalmente distinta dalla durata L_1 della sessione fra A e TGS

A genera *nonce* e *timestamp* distinti n_a, n'_a, T_a, T'_a, per le due sessioni in gioco

lo schema del protocollo fra utente A, KAS, TGS e server B è il seguente:

1. $A \rightarrow KAS$: A, TGS, n_a
2. $KAS \rightarrow A$: $e_{K_{as}}(K_{a,tgs},n_a,L_1,TGS), t_{A,TGS}$
3. $A \rightarrow TGS$: $t_{A,TGS}, e_{K_{a,tgs}}(A,T_a), B, n'_a$
4. $TGS \rightarrow A$: $e_{K_{a,tgs}}(K_{ab},n'_a,L_2,B), t_B$
5. $A \rightarrow B$: $t_B, e_{K_{ab}}(A,T'_a)$
6. $B \rightarrow A$: $e_{K_{ab}}(T'_a)$

problemi di sicurezza con Kerberos

non tutti i problemi di sicurezza dipendono dalla crittografia, come testimoniano i seguenti aspetti della sicurezza di Kerberos:

- sincronizzazione dei clock: richiede, fra l'altro, protezione dei clock da attacchi, e questa può a sua volta richiedere autenticazione...
- il controllo dei *timestamp* deve ammettere una tolleranza non nulla; se troppo larga, questa costituisce una vulnerabilità ad attacchi di *replay* da intrusi che vengono in possesso di una chiave passata
- la disponibilità dei servizi dipende da quella dei server di autenticazione
- attacchi alle password, dalle quali sono generate le chiavi lungo termine, sono possibili (Wu, 1999)
- chiavi e ticket sono memorizzati nelle macchine client; la sicurezza di Kerberos dipende anche da quella di queste ultime...
- la richiesta iniziale non è autenticata; un intruso che chiede autenticazione ottiene un ticket in ritorno... : vulnerabilità ad attacchi alla disponibilità dei servizi, e alla raccolta di dati per criptoanalisi

riferimenti

Diffie, van Oorschot & Wiener (1992) :
Authentication and authenticated key exchanges,
Design, Codes and Cryptography **2**, 107-125, 1992.
http://www.springerlink.com/content/p18377m548230848

Needham & Schroeder (1978) :
Using encryption for authentication in large networks of computers,
Communications of the ACM **21**(12), 993-999, 1978.
http://portal.acm.org/citation.cfm?doid=359657.359659

Miller, Neuman, Schiller & Saltzer (1987) :
Section E.2.1: Kerberos authentication and authorization system,
MIT Project Athena, Technical Report, Cambridge, MA, 1987.
http://clifford.neuman.name/publications

Wu (1999) :
A real-world analysis of Kerberos password security,
Proc. 1999 Network and Distributed System Security Symp., Internet Society, February 1999.
http://www.isoc.org/isoc/conferences/ndss/99/proceedings/papers/wu.pdf