Sicurezza dei sistemi e delle reti[1]

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

a.a. 2015/16

Sicurezza delle reti
Monga
IPsec
TLS/SSL
A livello di trasporto
Sicurezza perimetrale
Tipologie di firewall
Stateless filtering
Stateful filtering
Deep packet inspection

Lezione VI: IPsec

La suite TCP/IP

Sicurezza delle reti
Monga
IPsec
TLS/SSL
A livello di trasporto
Sicurezza perimetrale
Tipologie di firewall
Stateless filtering
Stateful filtering
Deep packet inspection

La suite TCP/IP non è progettata con particolari misure di difesa per la confidenzialità o integrità dei dati dalle manomissioni.

- Lo scenario di riferimento: nodi per lo piú cooperativi (accademici)
- e qualcuno sostiene che NSA fu contraria all'inserimento di tecniche crittografiche in una rete pubblica

IPsec

Sicurezza delle reti
Monga
IPsec
TLS/SSL
A livello di trasporto
Sicurezza perimetrale
Tipologie di firewall
Stateless filtering
Stateful filtering
Deep packet inspection

IPsec specifica come **crittare**, **autenticare** e **scambiare chiavi** con IP.

- Basato su IP (in maniera differente IPv4 e IPv6)
- Obbligatorio supportarlo per gli stack IPv6, facoltativo in IPv4

Servizi di sicurezza offerti

- Controllo dell'accesso alla comunicazione
- Autenticazione dell'origine dei dati
- Integrità dei dati
- Confidenzialità dei dati
- Protezione da *replay*

Protocolli IPsec

Si tratta in realtà di piú specifiche protocollari

- Authentication Header (AH) per l'autenticazione e integrità del datagramma
- Encapsulating Security Payload (ESP) per la confidenzialità

Entrambi presuppongono una **Security Association** (SA), per lo scambio di credenziali.

AH

- Serve per autenticare l'origine del pacchetto e l'integrità dei campi immutabili.
- Un security parameter index identifica la **SA**
- Identifica replay di pacchetti con una tecnica "sliding window" e un contatore che per essere inizializzato necessita una nuova **SA**

Sliding window

Il nodo destinazione tiene un array di $SW[1:w] = 0$ elementi per ogni SA

1. Primo datagramma contatore n: $SW[w] = n$
2. Datagramma contatore i

   $n - w + 1 \leq i \leq n \wedge OK(sig)$ controlla se $SW[i + w - n] > 0$ (replay!), altrimenti $SW[i + w - n] = i$

   $i \leq n - w$ vecchio

   $i > n \wedge OK(sig)$ sposta la finestra

ESP

- Serve per crittare il contenuto dei pacchetti
- Un security parameter index identifica la **security association**
- Due modalità
  1. transport protocolli superiori vengono crittati end-to-end
  2. tunnel i pacchetti IPsec contengono (crittati) pacchetti IP

Security association

Ogni conversazione IPsec è abbinata ad una Security association (SA) frutto di una negoziazione dei parametri di sicurezza e delle credenziali.

- IP destinazione
- Una SA per AH e una per ESP
- Statiche o dinamiche (ISAKMP: Internet Security Association Key Management Protocol, IKE: Internet Key Exchange)

problemi IPsec

- La configurazione dei firewall per permettere i protocolli IPsec non è banale
- Ogni volta che una comunicazione comporta la manipolazione dei pacchetti IP (proxy e NAT) occorre adottare misure speciali, con successive security association.

Riassumendo

- IPsec introduce autenticazione, integrità e confidenzialità
- Protezione da *replay*
- Necessita di un certo overhead amministrativo e computazionale

TLS/SSL

Un'altra possibilità è introdurre misure di sicurezza sopra il livello di trasporto TCP.

- 1993–1995, Netscape rilascia un **Secure Socket Layer** SSL (2.0) pensato per proteggere la navigazione web.
- SSL 3.0, standardizzato da IETF come TLS **Transport Layer Security**

Obiettivi di sicurezza

- cifratura end-to-end
- protezione dell'integrità
- autenticazione **del server** (il client rimane anonimo)
- efficienza adeguata alle connessioni HTTP, brevi e stateless

sessione TLS

I nodi mantengono lo stato della sessione per gestire la cifratura del traffico.

- TLS handshake protocol
- TLS record layer
- una sessione può gestire piú connessioni per ridurre l'overhead

TLS handshake

1. C richiede la connessione, elencando quali cipher suite (CS) conosce
2. S sceglie CS compatibile e spedisce un digital certificate (DC) firmato da una CA
3. C controlla DC e manda criptata una chiave di sessione (K) random

Integrazione con le applicazioni

Tre strategie:

1. Creare un nuovo servizio (es. SSH2)
2. Aggiungere TLS ad un servizio noto (es. HTTPS)
3. Estendere un servizio noto affinché usi TLS (es. ESMTP)

Riassumendo

- TLS permette cifratura e autenticazione dei server (tramite CA) a livello di trasporto
- La gestione delle sessioni è progettata per essere efficiente in presenza di connessioni ripetute
- Molto diffuso perché facile da integrare nelle applicazioni

Livello di trasporto

IPsec e TLS possono essere piuttosto penalizzanti dal punto di vista delle prestazioni (Dal punto di vista delle performance del server, TLS può arrivare ad essere fino a 82 volte piú lento di una connessione TCP).
tcpcrypt è una proposta recente (2010) piú efficiente (3 volte piú lento di TCP)

Altri limiti di TLS

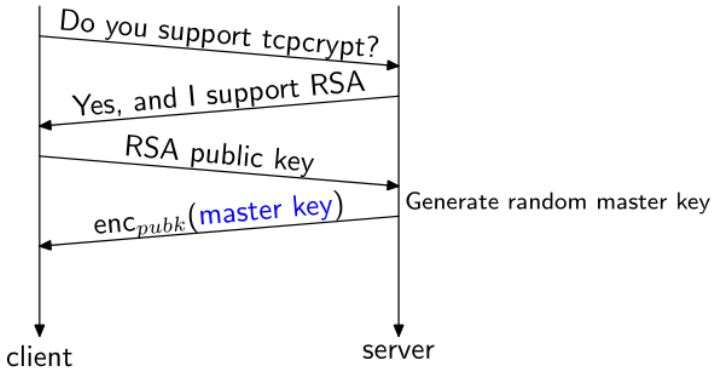
La cifratura dipende dall'autenticazione del server, a sua volta garantita dall'autorità certificatrice.

- Se l'autenticazione è falsa, la cifratura non è molto utile (ma l'overhead rimane)

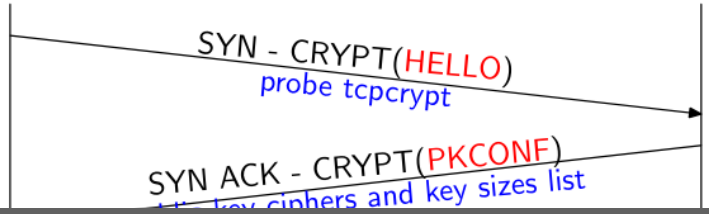
tcpcrypt

- Estensione di TCP
- Il carico computazionale crittografico è per lo piú spostato sui client
- **Opportunistic encryption**: attiva solo se supportata da entrambi (attenzione agli attacchi attivi!)

tcpcrypt handshake



36 volte piú veloce di TLS



Autenticazione

Non c'è autenticazione del server con CA come nel caso di TLS, ma un **session ID** probabilisticamente unico (anche quando uno dei nodi è malevolo).
Un segreto condiviso k può essere usato cosí

1. $C \rightarrow S : HASH(k, C|SessionID)$
2. $S \rightarrow C : HASH(k, S|SessionID)$

Se anche S è malevolo (e k non generabile da un dizionario), non potrà riusare k (non estraibile da $HASH(k, C|SessionID)$) né $HASH(k, C|SessionID)$ perché il $SessionID$ sarà diverso.

Riassumendo

- tcpcrypt è un'estensione di TCP, che permette di cifrare il livello di trasporti
- è molto piú efficiente di TLS perché il carico crittografico è per lo piú spostato sui client
- Il Session ID permette di costruire protocolli di autenticazione a livello applicativo

Sicurezza perimetrale

- Poiché in Internet è una rete di reti (locali) si parla di protezione del perimetro di sottorete.
- Abbiamo già visto che l'assunzione è `locale == trusted`.
- I firewall vengono usati per definire località parzialmente diverse da quelle imposte dai mezzi trasmissivi (LAN).

Sicurezza perimetrale

Firewall

(*parete tagliafuoco*) è un dispositivo che:

- è al confine fra due reti A e B
- tutto il traffico tra A e B (e viceversa) **deve** passare attraverso di esso
- filtra il traffico secondo una precisa **politica d'accesso** (policy)

Compito dei firewall

Il compito dei firewall è stabilire quale traffico ha accesso alla rete (*policy*) e non controllare che il traffico permesso non faccia danni (*control*, intrusion detection).

Cosa sono i firewall

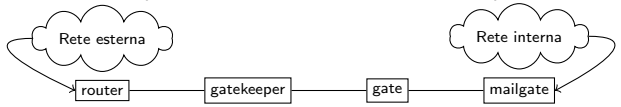
Tipicamente sono realizzati come

- Forwarding gateway
- Filtering router
- Proxy

E stabiliscono politiche (regole) ai vari livelli dello stack TCP/IP

Firewall a vari livelli

I primi firewall (Mogul, 1989 e Ranum, 1992) e



- Gatekeeper proxy applicativo: raccoglie le richieste applicative (Telnet, FTP, SMTP, ...) dall'interno e le manda verso l'esterno
- Gate filtra il traffico

Riassumendo

I firewall

- sono al confine fra due reti
- filtrano il traffico secondo una precisa **politica d'accesso** (policy)
- servono per definire zone di traffico trusted parzialmente diverse da quelle imposte dalle LAN.

Livelli firewall

In generale si possono avere firewall

- a livello applicativo (application gateway, proxy)
- a livello di trasporto (circuit gateway)
- a livello rete (packet filter)

Livelli firewall

- Esistono anche ibridi: dynamic packet filter agiscono a livello rete e trasporto (e talvolta anche applicativo).
- Possono essere realizzati via software o hardware (piú veloci, ma piú costosi e meno flessibili nelle configurazioni).

Stateless filtering

È il metodo piú semplice e piú comune

Stateless filtering

Ogni pacchetto (o comando protocollare, se a livello applicativo) è valutato in isolamento, senza tenere traccia di quelli precedenti

ACL

In pratica si tratta di avere una Access Control List (ACL) che *filtra* i pacchetti o le richieste, uno alla volta

int addr	int port	ext addr	ext port	action
*	*	a.b.c.d	*	block
192.168.2.3	110	*	110	allow

Digressione: ACL

Una ACL fissa la politica d'accesso: espressa in maniera compatta (e comprensibile). Come va interpretato *il silenzio* dell'ACL?

default deny Vietato tutto ciò che non è **esplicitamente** permesso

default permit Permesso tutto ciò che non è **esplicitamente** vietato

Default deny

Normalmente l'ACL è una serie di regole che vengono esaminate dalla prima all'ultima, quindi se l'ultima regola è equivalente a

int addr	int port	ext addr	ext port	action
*	*	*	*	block

si ha *default deny*

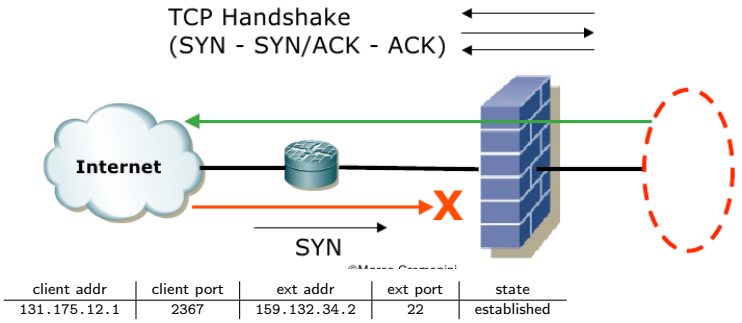
Stateful filtering

Stateful filtering

Si tiene traccia di uno *stato* del sistema e il filtraggio avviene sulla **storia** dei pacchetti o delle richieste.

Allo scopo occorre mantenere una tabella delle connessioni

Stateful filtering



client addr	client port	ext addr	ext port	state
131.175.12.1	2367	159.132.34.2	22	established

Deep packet inspection

Firewall stateful che operano filtraggio applicativo analizzando il **contenuto** dei pacchetti vengono talvolta detti deep packet filters.

- Analisi del traffico applicativo, la cui liceità va valutata caso per caso
- Generalmente basati su pattern matching di stringhe

Riassumendo

I firewall si differenziano per

- il livello a cui agiscono
- il tipo di regole di filtraggio
  - stateless
  - stateful
  - "deep packet inspection"