Ottimizzazione Combinatoria
Appunti

Giovanni "Qua' Qua' dancer" Palma e Alex Basta

Contents

Chapter 1

Introduzione

Soccia che bononia

Chapter 2

Introduzione alla sicurezza di rete

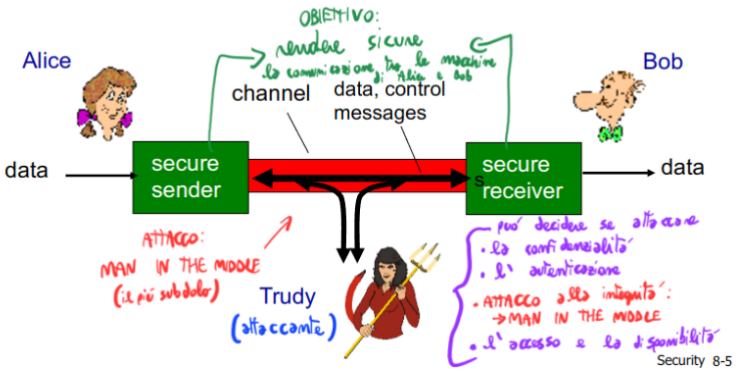
Definition 2.0.1: Sicurezza di rete

La sicurezza di rete è l'insieme delle strategie, tecnologie e protocolli utilizzati per proteggere le reti di calcolatori da accessi non autorizzati, attacchi, perdite di dati e altre minacce informatiche

In pratica consiste nel creare un'architettura di difesa e prevenzione. Questi sono in modo riassuntivo i principali obbiettivi della sicurezza di rete:

- **Confidenzialità**: Solo il mittente e il destinatario devono poter leggere il messaggio. Questo si ottiene con la crittografia
  - Il mittente crittografa (encrypts) il messaggio.
  - Il destinatario decrittografa (decrypts) il messaggio
- **Autenticazione**: Processo per verificare l'identità di un utente o un dispositivo, fondamentale per garantire accessi sicuri (es. Password, certificati digitali, autenticazione a due fattori (2FA))
- **Integrità dei messaggi**: Garantisce che i dati non siano stati alterati durante la trasmissione (es. Password, certificati digitali, autenticazione a due fattori (2FA))
- **Accesso e Disponibilità**: I servizi devono essere accessibili agli utenti autorizzati. Minacce:
  - Attacchi DoS/DDoS che bloccano il servizio.
  - Gestione dei permessi per prevenire accessi non autorizzati

Si introduce il concetto di attacco con il tipico esempio di Alice e Bob che vogliono comunicare e Trudy, l'attaccante, che può intercettare, elliminare o aggiungere messaggi durante la comunicazione dei due amamnti carucci (Morbidelli - Morigi)



Vabbuò passiamo alla sezione sucessiva:

2.1 Principi di crittografa

Partiamo dalla defnizione

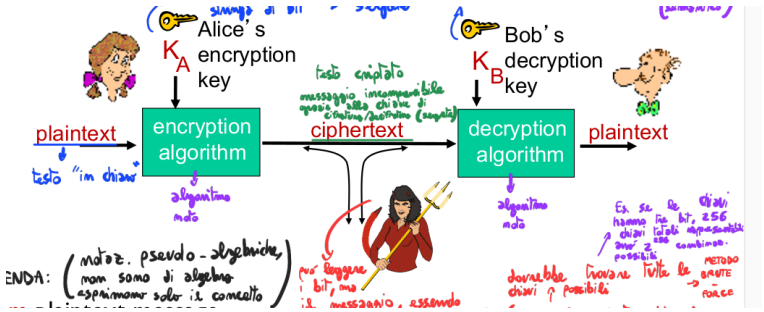
Definition 2.1.1: crittografa

si **crittografa** la disciplina che studia le tecniche per proteggere le informazioni trasformandole in un formato illeggibile per chi non è autorizzato, consentendo solo ai destinatari legittimi di decifrarle.

Nelle reti di comunicazione, i dati trasmessi possono essere intercettati da chiunque abbia accesso al canale di comunicazione, rendendo le informazioni vulnerabili a lettura, modifica o attacchi malevoli. Per proteggere la riservatezza e l'integrità dei dati, si utilizza, quindi, la crittografia, una tecnica che consente di trasformare il testo in chiaro, detto *plaintext*, in un formato incomprensibile chiamato testo cifrato o *ciphertext*, attraverso l'applicazione di un algoritmo di cifratura.
L'obiettivo della crittografia è garantire che, anche se un malintenzionato intercettasse il messaggio durante la trasmissione, non sarebbe in grado di comprenderne il contenuto senza la conoscenza della chiave segreta. Solo il destinatario legittimo, in possesso della chiave corretta, può applicare un algoritmo di decifratura per convertire il testo cifrato nuovamente in testo leggibile.
Va sottolineato che gli algoritmi di cifratura e decifratura sono generalmente pubblici e noti a tutti. Tuttavia, la sicurezza della crittografia si basa sulla segretezza della cosiddetta *chiave crittografica*, ovvero una sequenza di bit utilizzata all'interno di un algoritmo di cifratura per trasformare un messaggio in un formato sicuro e, successivamente, per riconvertirlo nel suo stato originale. Questa chiave viene generata all'inizio della comunicazione tra mittente e destinatario e loro e solo loro ne sono a conoscenza.
Adesso un'introduzione all'algebra della crittografia
Si ha:

- m: plaintext
- $K_A(m)$: ciphertext, encrypted with key K_A
- $m = K_B(K_A(m))$: plaintext ripristinato grazie alla chiave K_B

Si noti la seguente immagine (con appunti bonziani):



2.1.1 Attaccare uno schema di crittografia

Gli attacchi crittografici mirano a violare la sicurezza di un sistema di cifratura. Ne esistono diversi tipi:

1. **Cipher-text only attack**, l'attaccante possiede solo il testo cifrato a disposizione e per decifrare il messaggio ha due approcci possibile:
   - **Forza bruta**: prova tutte le chiavi possibili
   - **Analisi statistica**: cerca pattern ripetuti nei dati cifrati
2. **known-plaintext attack**: Trudy possiede sia il testo cifrato che il corrispondente testo in chiaro. Con queste informazioni, cerca di scoprire la chiave di cifratura o il meccanismo utilizzato, così da poter decifrare altri messaggi cifrati dallo stesso sistema.

Example 2.1.1

Se Trudy sa che nel testo in chiaro compare la parola "hello" e trova nel messaggio cifrato la sequenza "JGRRG", può iniziare a costruire un dizionario di corrispondenze tra lettere:

- $h \rightarrow J$
- $e \rightarrow G$
- $l \rightarrow R$
- $o \rightarrow G$

Note:

un buon metodo per evitare questi attacchi è rendere lo spazio delle chiavi il più vasto possibile e utilizzare algoritmi sicuri che resistano alle analisi statistiche

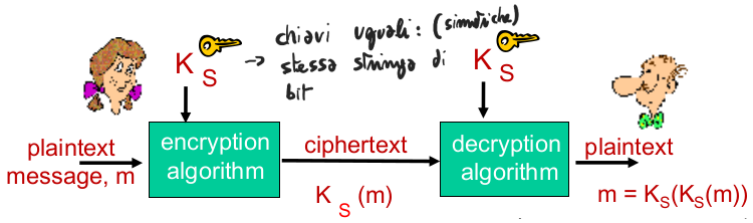
2.1.2 Crittografia a chiave simmetrica

Adesso entriamo nel vivo della difesa cazzo

Definition 2.1.2: Crittografia a chiave simmetrica

mittente e destinatario condividono la stessa chiave segreta K_S per cifrare e decifrare i messaggi

I pratica il mittente (Alice) cirfra il plaintext con l'algoritmo di cifratura usando la chiave $K_S$ottenendo il ciphertext, Bob riceve il ciphertext e lo decifra usando lo stesso algoritmo e la chiave K_S, recuperando il messaggio originale



Per implementare la chiave simmetrica esistono diverse tecniche

Cifrario a sostituzione

Definition 2.1.3: Cirfrario a sostituizione

Un **cifrario a sostituzione** è una tecnica di cifratura in cui ogni lettera del testo in chiaro viene sostituita con un'altra lettera secondo un mapping predefinito

Cifrario monolitico Tra i vari tipi di cifrario a sostruzioni vi è il cifrario monolitico dove ogni lettera dell'alfabeto viene sostituita da un'altra lettera fissa

Example 2.1.2 (Cifrario monolitico)

- Alfabeto in chiaro: abcdefghijklmnopqrstuvwxyz
- Alfabeto cifrato: mnbvcxzasdfghjklpoiuytrewq
- Messaggio originale: bob. i love you. alice
- Messaggio cifrato: nkn. s gktc wky. mgsbc

La chiave di cifratura è una funzione di permutazione ad un insieme di 26 lettere ad un altro insieme di 26 lettere
Tuttavia è poco sicuro perché mantiene la frequenza delle lettere originali e lo rende particolarmente vulnerabile alle analisi delle frequenze

Schema ciclico Per rendere la cifratura più sicura, si può usare più cifrari di sostituzione e applicarli in un ciclo predefinito
La chiave di cifratura quindi è un insieme di n cifrari sostituzione $(M_1, M_2, \ldots, M_n)$ ed un modello cicilico su come usarli. Ogni lettera del plaintext quindi viene cifrata con un cifrario diverso seguendo il ciclo

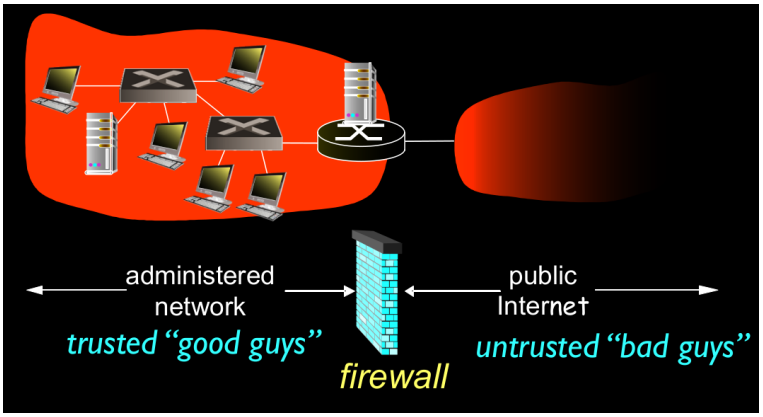
2.2 Firewall e IDS

Innanzi definiamo che cos'è un *Firewall*

Definition 2.2.1: Firewall

Si definisce **Firewall** un sistema di sicurezza che isola una rete interna di un'organizzazione da internet, controllando e filtrando il traffico di rete in entrata e uscita in base a regole di sicurezza definite dette di `ACCESS` o `DENIED`

L'idea del Firewall è che l'esterno di una rete è composta dai cosiddetti "cattivi ragazzi" che la mamma non vi raccomanderebbe come compagni d'uscita, mentre l'interno della rete è composta da "bravi ragazzi" di cui fidarsi, lo scopo del sistema è, quindi, separare i buoni dai cattivi



Il Firewall si rivela utile per principalmente tre motivi:

- **Prevenzione degli attacchi Denial of Service (Dos)**: tipologia di attacco informatico che mira a rendere inaccessibili o indisponibili i servizi di una rete ad utenti legittimi

  Un esempio è il **SYN flooding** in cui un attaccante invia molte richieste di connessione false, esaurendo le risorse del server sovraccaricandolo e impedendo connessioni legittime

- **Protezione dei dati interni da accessi non autorizzati**: ovvero impedisce che gli attaccanti possona *modificare o rubare dati sensibili*

  Un esempio classico è un attaccante che sostituisce il sito web di un'organizzazione con un contenuto malevolo.

- **Accesso selettivo e autorizzato alla rete interna**: Permette l'accesso solo a utenti o dispositivi autenticati, migliorando la sicurezza

Esistono tre tipi di tipoligie di Firewall che approfonoidremo nel dettaglio:

- **Stateless Packet Filter**: Controlla ogni pacchetto singolarmente, senza tenere traccia delle connessioni
- **Statefull Packet Filter**: Tiene traccia dello stato delle connessioni (es. richieste e risposte)
- **Application Gateway** (proxy Firewall): controlla il traffico a livello applicativo (es. HTTP, FTP, e.mail), filtrando così le informazioni che all'interno dei protocolli del livello (ad esempio il contenuto di una mail)

Si notino nel dettaglio

2.2.1 Stateless Packet Filtering

Definition 2.2.2: Stateless Packet Filtering

Lo **Stateless Packet Filtering** è una tecnica di sicurezza di rete in cui un firewall esamina ogni pacchetto individualmente, senza tener conto delle connessioni stabilite in precedenza. Questo significa che ogni pacchetto è valutato isolatamente sulla base di un insieme di regole predefinite

Di solito un firewall con filtraggio stateless è implementato su un router che collega una rete interna a internet. Il router analizza ogni pacchetto in ingresso e in uscita e decide se bloccarlo o lasciarlo in base a regole definite nelle cosiddette "**white list**" (pacchetti che possono passare) e o "**black list**" (lista di pacchetti da bloccare)
Il firewall prende decisioni basandosi su parametri del pacchetto, tra cui:

- **IP** del mittente e destinatario
- **Numero di porta TCP/UDP** del mittente e destinatario
- **Tipo di messaggio ICMP** bloccando, ad esempio, attacchi di scansione provenienti dall'esterno
- **bit SYN e ACK nei pacchetti TCP**: Il firewall può, ad esempio, bloccare pacchetti con SYN in entrata per impedire connessioni indesiderate dall'esterno

Qui degli esempietti:

Example 2.2.1 (Bloccare tutti i pacchetti con protocollo UDP o Telnet)

- **Regola**: blocca i pacchetti in entrata e in uscita se il protocollo IP è 17 (UDP) e se con la porta di origine o desitinazione è 23
- **Risultato**: Tutto il traffico UDP e telnet vengono bloccati

Example 2.2.2 (Bloccare pacchetti TCP in ingresso con ACK=0)

- **Regola**: blocca tutti i pacchetti TCP in ingresso se il bit ACK = 0
- **Risultato**:Le connessioni in entrata non possono essere iniziate da un host esterno verso la rete interna e le connessioni in uscita funzionano normalmente, perché il traffico di ritorno (che ha ACK=1) è consentito.

Riporto qui una tabella con altri esempi:

Politica	**Impostazione Firewall**
Nessun accesso Web esterno.	Elimina tutti i pacchetti in uscita verso qualsiasi indirizzo IP, porta 80
Nessuna connessione TCP in entrata, tranne quelle per il server Web pubblico dell'istituzione.	Elimina tutti i pacchetti TCP SYN in entrata verso qualsiasi IP tranne 130.207.244.203, porta 80
Impedisci alle Web-radio di consumare la larghezza di banda disponibile.	Elimina tutti i pacchetti UDP in entrata - tranne DNS e broadcast del router.
Impedisci alla tua rete di essere utilizzata per un attacco DoS smurf.	Elimina tutti i pacchetti ICMP diretti a un indirizzo di "broadcast" (ad esempio, 130.207.255.255).
Impedisci alla tua rete di essere tracciata	Elimina tutto il traffico ICMP TTL scaduto in uscita

Access control list

Definition 2.2.3: Liste di controllo d'accesso

Le **ACL** sono tabelle di regole applicate, *con priorità dall'alto verso il basso*, ai pacchetti in arrivo per decidere se consentire (allow) o bloccare (deny) il traffico di rete

Queste tabelle sono il vero e proprio cuore pulsante del Firewall e indicano quale pacchetto può passare e chi no

Example 2.2.3 (ACL)

azione	indirizzo sorgente	indirizzo destinazione	protocollo	porta sorgente	p
allow	222.22/16	outside of 222.22/16	TCP	> 1023	8
allow	outside of 222.22/16	222.22/16	TCP	80	>
allow	222.22/16	outside of 222.22/16	UDP	> 1023	5
allow	outside of 222.22/16	222.22/16	UDP	53	>
deny	all	all	all	all	a

Criticità

Un serio problema dei Firewall Stateless che potrebbero lasciare passare pacchetti che non hanno senso nel contesto di una connessione. Esempio:

- Un pacchetto con destinazione porta 80 (HTTP) e ACK=1 arriva dall'esterno
- Il firewall stateless lo accetta, anche se nessuna connessione HTTP è stata avviata da un client interno.
- Un attaccante potrebbe sfruttare questa debolezza per inviare pacchetti falsi alla rete interna

Per porre rimedio a questi tipi di problemi si veda la tipoligia di firewall sucessiva

2.2.2 Stateful packet filtering

Definition 2.2.4: Stateful packet filtering

Lo **Stateful packet filtering** è una tecnica di filtraggio di pacchetti tenendo traccia delle connessioni attive e delle loro fasi

Un firewall stateful tiene traccia di ogni connessione TCP attiva e controlla le tre fasi fondamentali della connessione TCP (assicurandosi che ogni pacchetto in entrata o uscita abbia senso):

- **Setup**: Il firewall rileva il pacchetto SYN iniziale, segnalando l'inizio di una connessione
- **Trasferimento dati**: I pacchetti con ACK vengono accettati solo se appartengono a una connessione già avviata
- **Chiusura**: Quando un pacchetto FIN o un timeout segna la fine di una connessione, il firewall non accetta più pacchetti fino a che non se apre un altro

Chapter 3

Livello di rete

3.1 Overview del libeello di rete

Definition 3.1.1: Livello di rete

Il **livello di rete** (Network Layer) è responsabile del trasporto dei pacchetti da un host mittente a un host destinatario attraverso una o più reti intermedie, collandosi sopra il Data Link Layer e sotto il livello di trasporto

GLi obbiettivi principiali del Network Layer sono:

- **Modello di servizio**: definisce come il livello di rete fornisce il servizio ai livelli superiori
- **Instradamento e forwarding**:
  - **Forwarding**: trasmissione di un pacchetto dall'ingresso di un router alla sua uscita appropriata.
  - **Routing**: determinazione del percorso che i pacchetti devono seguire attraverso la rete.

3.1.1 Fuznioni del network layer

Il Network Layer può essere suddiviso in due funzioni principali:

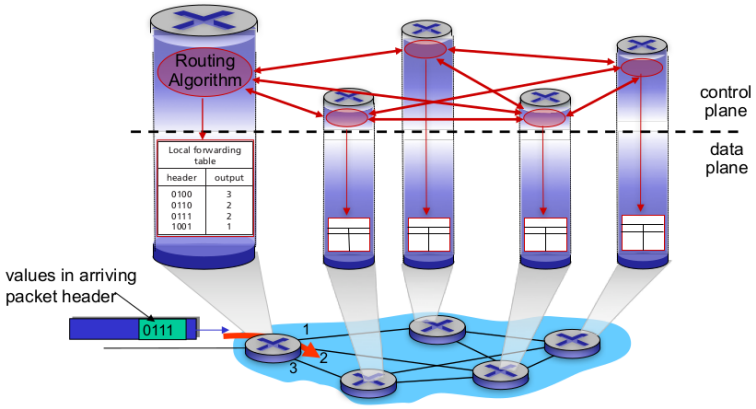
- **Data Plane**: gestione del traffico dei pacchetti in tempo reale (forwarding)
- **Control Plane** (piano di controllo): determinazione del percorso ottimale per i pacchetti (routing)

Adesso vediamo entrambi nel dettaglio:

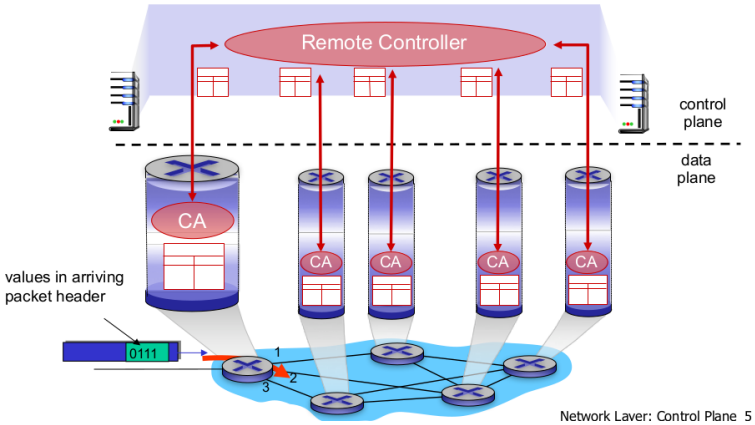
Data Plane

Il Data Plane è la parte del Network Layer responsabile della gestione effettiva del traffico dei pacchetti in tempo reale. A differenza del Control Plane, che decide il percorso dei pacchetti, il Data Plane si occupa di spostarli da un router all'altro fino alla destinazione.
Esistono due principali approcci alla gestione del Control Plane:

- **Per-Router Control Plane**: ogni ruoter ha il proprio algoritmo di routing e prende decisioni autonomamente e comunicano tra di loro per scambiarsi informazioni di instradamento



- **Logically Centralized Control Plane**: Un controller centralizzato (tipicamente remoto) prende le decisioni di instradamento e le comunica ai router, i router eseguono solo il forwarding senza prendere decisioni di routing



In quasi tutte le infrastrutture di rete viene utilizzata il Logically Centralized Control Plane, mentre il Per-Router Control Plane viene utilizzata sopratutto nei sistemi di rete dinamici (ad esempio bluetooth col cellulare e altri dispositivi)
Fare le altre robe

3.2 Router

3.3 IP datagram format

3.3.1 Frammentazione IP

L'offset, le flag e l'identificatore permette la ricostruzione dell'intero pacchetto che era stato frammentato.

Example 3.3.1

Supponiamo di avere:

- datagramma di 4000 byte
- MTU di 1500 byte

Ci servono 3 frammenti, che hanno lo stesso ID di pacchetto, il flag che indica i frammenti a 1 per i primi due e 0 per l'ultimo (proprio per indicare che e' l'ultimo) e gli offset (in byte/8 per compattezza).

3.3.2 ATM

Solo 5 byte, che sono un'etichetta di flusso, che serve per prenotare le risorse sul cammino determinato dal routing. I router che ricevono lungo il percorso leggono l'etichetta e sanno dove instradarli. E' l'emulazione di una commutazione di circuito fatto a pacchetto.
C'e' quindi un contratto a priori che garantisce una certa quantita' di flusso a disposizione.
La dimensione dei pacchetti ATM deve essere abbastanza piccolo da rendere veloce e gestibile la trasmissione, ma non troppo piccole da avere una porzione importante presa dall'header (overhead)

3.4 IP addressing

Example 3.4.1

Reti di classe C Alto sx:

- rete di classe C
- Default Gateway 223.1.1.4, seguendo le best practices dovrebbe essere 223.1.1.254 (non 255 perche' sarebbe broadcast)

Per identificare la subnet in realta non si cambia la maschera perche boh

Example 3.4.2 (Subnets)

Ci sono 3 subnets, router separati per ogni sottorete connessi in modo completo (anello, ridontante). ma ci sono anche le reti degeneri che collegano i tre router, quindi in realta' ce ne sono 6 di reti.

Example 3.4.3 (DHCP)

Il DHCP server vede che arriva una richiesta in broadcast alla porta DHCP chiede al router se esiste un indirizzo e lo da al client.
Facendo spoofing del mac address, e' possibile apparire come una scheda di rete nuova ed e possibile spammare i DHCP in broadcast fino a esaurire gli IP.
Tale attacco puo' essere fermato solo tramite autenticazione1

3.5 IPv6

Non c'e' frammentazione, c'e' priorita' del flusso. Etichetta di flusso ci dice gia' la porta di destinazione, dato che per ogni flusso il router v6 salva la porta in una tabella separata.

3.6 Forwarding Generalizzato

Nelle SDN, abbiamo la possibilita' di definire su ogni router una politica locale nella quale abbiamo tre entita' di interesse:

- header
- counter: controllo prestazioni
- azioni: cosa fare se accade qualcosa

Le regole di gestione dei pacchetti sono semplici e generalizzati:

- azioni: