**Università degli studi di Modena e Reggio Emilia**

**Dipartimento di Scienze Fisiche, Informatiche e Matematiche**

---

*Corso di Laurea in informatica*

# Titolo: prima riga Seconda riga Terza riga Quarta riga

**Relatore:**
**Luca Ferretti**

**Candidato:**
**Fabio Zanichelli**

---

**Anno Accademico 2021/2022**

# Indice

# Capitolo 1

# Introduzione

## 1.1 OS Fingerprinting

L'OS fingerprinting consiste nel rilevare da remoto il sistema operativo di un dispositivo analizzandone i pacchetti inviati. Le differenze di implementazione dello stack TCP/IP, infatti, determinano comportamenti diversi che, analizzati, consentono di ottenere informazioni importanti.
Il fingerprinting può essere effettuato in due modalità: attiva e passiva.
Nella prima si analizzano le risposte ricevute in seguito ad alcuni pacchetti inviati; questi ultimi sono appositamente costruiti in modo da massimizzare le informazioni che si possono ottenere dalla risposta. Nella seconda, invece, viene ispezionato il normale traffico del dispositivo target; si tratta quindi di una tecnica meno invasiva e che si espone meno al rischio di essere scoperti.

## 1.2 Stack TCP/IP

Per effettuare comunicazioni tramite internet, vi è il bisogno che tutti i dispositivi connessi rispettino determinati meccanismi; questo si rende necessario a causa dell'elevata eterogeneità derivata da hardware e software differenti. Questi meccanismi, che prendono il nome di *protocolli*, sono strutturati secondo diversi layer (livelli) formando lo stack TCP/IP.

Sebbene l'idea originale prevedesse un modello composto da sette livelli, de facto lo schema attualmente in uso ne prevede solamente quattro. Nonostante ciò, nella terminologia informatica la numerazione è rimasta quella precedente

| | |
|---|---|
| Livello 7 | Applicativo |
| Livello 4 | Trasporto |
| Livello 3 | Rete |
| Livello 2 | Fisico |

Tabella 1.1: Livelli dello stack TCP/IP

### 1.2.1 Funzionamento dello stack TCP/IP

Si consideri l'esempio dell'invio di una lettera tramite il servizio di poste. La procedura da seguire è la seguente:

1. Il mittente scrive il contenuto del messaggio su un foglio e successivamente lo inserisce nella busta.
2. Il mittente scrive, nella busta, il nominativo del destinatario.
3. Il mittente scrive il CAP e l'indirizzo del ricevente.
4. Il mittente, dopo aver incollato il francobollo, consegna la busta al servizio di poste.

Si noti che la sequenza di eventi che si verifica per la ricezione delle lettere è nell'ordine opposto a quello precedente:

1. Viene controllata la presenza del francobollo.
2. Il postino legge l'indirizzo del destinatario e consegna la lettera.
3. Verrà letto il nominativo per individuare a quale inquilino è diretta.
4. Il destinatario apre la busta e legge il contenuto del messaggio.

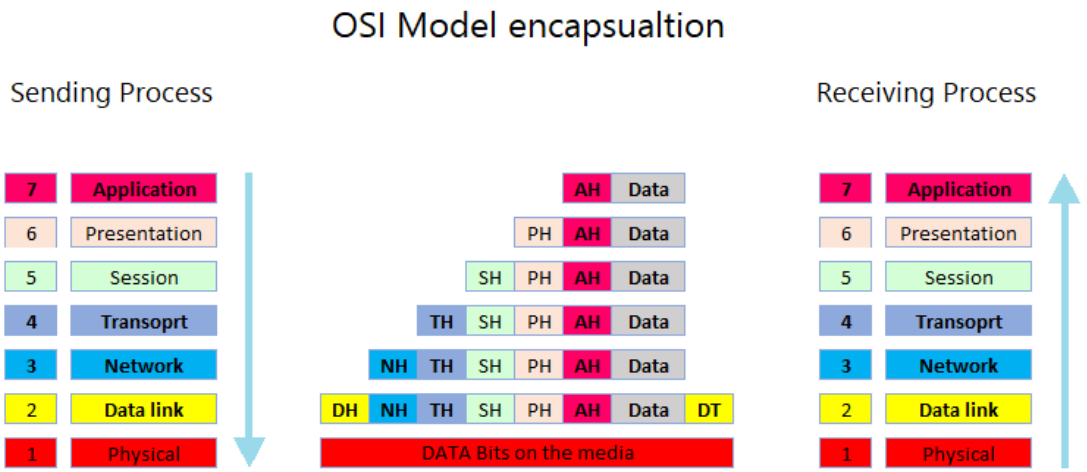


Figura 1.1: METTERE RIFERIMENTO IMMAGINE: http://infodoc.altervista.org/sistemi-e-reti/incapsulamento/

La sequenza di eventi descritta rappresenta ciò che avviene anche quando si comunica tramite internet. Ad ogni livello dello stack, infatti, vi è un protocollo che aggiunge una sua intestazione (*header*) al pacchetto del mittente, ad iniziare dal livello più alto disponibile per quel dispositivo e il ricevente valuterà queste in ordine inverso. Ad ogni layer il protocollo utilizzato può essere differente; fondamentale, ovviamente, che questo sia supportato da entrambi gli attori della conversazione.
Questa metodologia prende il nome di *incapsulamento*.

Per valutare il sistema operativo di un determinato dispositivo, si fa affidamento agli header di ogni livello.

## 1.3 Strumenti per il fingerprinting

Esistono numerosi tool per effettuare il fingerprinting, che eseguono anche attività correlate e fondamentali per quest'ultimo come ad esempio il port scanning. Nel corso di questo documento si farà principalmente riferimento a due strumenti:

- Nmap, per il fingerprinting attivo

- p0f, per il fingerprinting passivo

# Bibliografia

[1] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2):77–94, 1988.

[2] Google. Google scholar. `https://scholar.google.it/`, visited in Sep. 2016.