Università degli Studi di Verona
Corso di Laurea Magistrale in Informatica

Seminario sulla sicurezza delle reti

Andrea Zwirner


andrea@linkspirit.it


@AndreaZwirner

Chi sono

- Mi interesso di sicurezza informatica dallo scorso millennio
  - “Connettere” significava “intrecciare”
  - Hacker non aveva ancora alcun significato

- Ho fondato Linkspirit, azienda che si occupa di
  - Consulenza nella progettazione sicura di software e sistemi
  - Verifiche di sicurezza su software e sistemi
  - Formazione in materia di sicurezza informatica

Cosa faccio

- Partecipo ad alcuni progetti liberi legati la divulgazione della cultura sulla sicurezza informatica

ISECOM
INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES

www.isecom.org

Hacker Highschool
SECURITY AWARENESS FOR TEENS

www.hackerhighschool.org

OWASP
The Open Web Application Security Project

www.owasp.org

Virtualmente Sicuri – Progetto Scuole

www.progettoscuole.it

La sicurezza informatica

- Insieme di misure di carattere organizzativo, tecnologico e procedurale mirate a garantire
  - CONFIDENZIALITÀ
  - INTEGRITÀ
  - DISPONIBILITÀ

dell'informazione.

Come funziona

- Definizione di politiche di accesso a servizi e informazioni
  - autenticazione → chi è chi
  - autorizzazione → chi può fare cosa
- Difesa perimetrale
- Difesa interna

Sicurezza informatica offensiva

- Utilizzo di strumenti e metodologie usate dagli attaccanti reali
- Accesso a servizi ed informazioni senza possedere i permessi previsti dalle politiche di sicurezza
  - Lettura → Confidenzialità
  - Scrittura → Integrità / Disponibilità
- Ha il fine di evidenziare le vulnerabilità di sicurezza presenti nel sistema

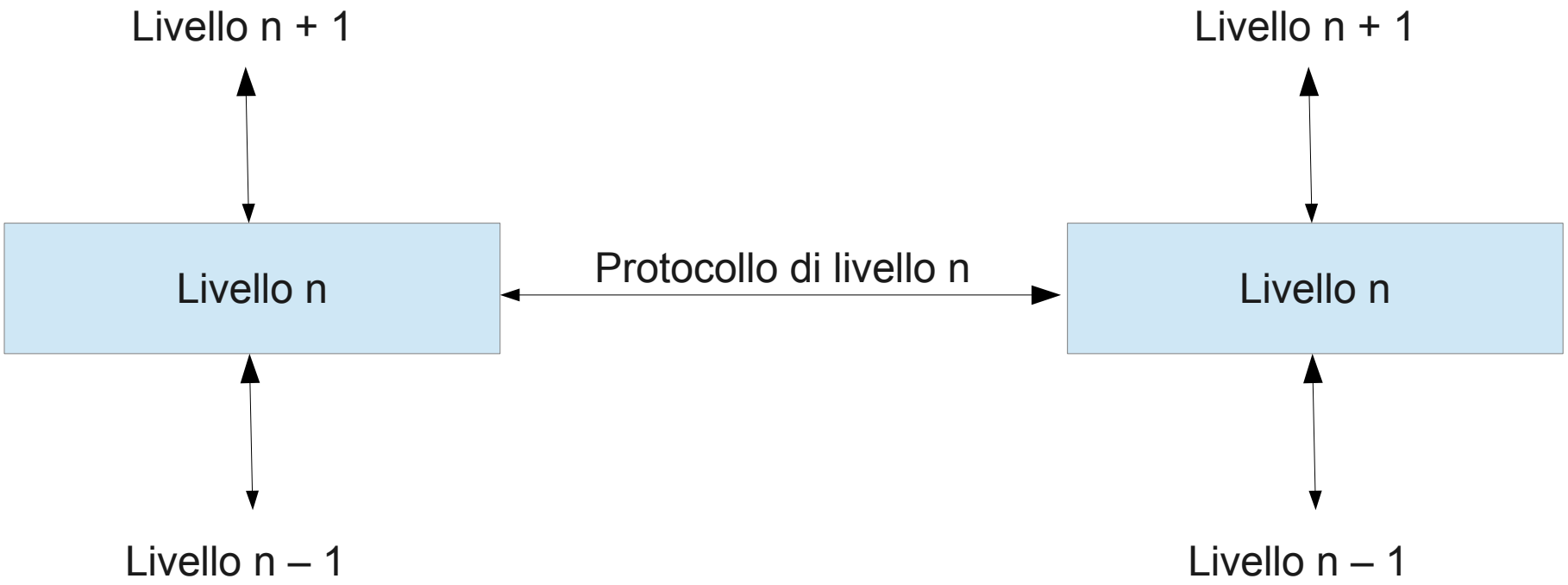
Di cosa parliamo oggi

- Sicurezza del livello di rete
- Utilizzo del livello di rete per la rilevazione e mitigazione ai livelli più alti

Lo standard Open Systems Interconnection

Architettura a livelli

- Il servizio implementato dal protocollo a livello n viene fornito al livello n + 1



Lo standard ISO / OSI

Il livello Fisico

- Controlla i dispositivi hardware che compongono la rete
- Tensioni, segnali, modulazioni, codifiche, trasmissioni simultanee, etc

Lo standard ISO / OSI

Il livello di Collegamento

- Garantisce l'affidabilità del livello fisico
- Incapsulamento e gestione degli header, controllo degli errori (CSMA/CD), controllo di flusso

Lo standard ISO / OSI

Il livello di Rete

- Generalizzazione dei livelli infriori rispetto ai superiori
- Routing, risoluzione indirizzi, gestione di frammentazioni, gestione di protocolli differenti nell'utilizzo di gateway

Lo standard ISO / OSI

Il livello di Trasporto

- Gestisce la trasparenza e l'affidabilità del trasporto end-to-end
- Gestione delle connessioni, gestione congestioni

Lo standard ISO / OSI

Il livello di Sessione

- Gestisce la comunicazione e sincronia tra applicazioni cooperanti
- Aggiunge al trasporto le logiche di cooprazione del livello applicativo

Lo standard ISO / OSI

Il livello di Presentazione

- Standardizza ed offre servizi di comunicazione comune al livello applicativo
- Gestisce la sintassi dell'informazione da trasferire (e.g. crittografia, compressione, etc)

Lo standard ISO / OSI

Il livello Applicativo

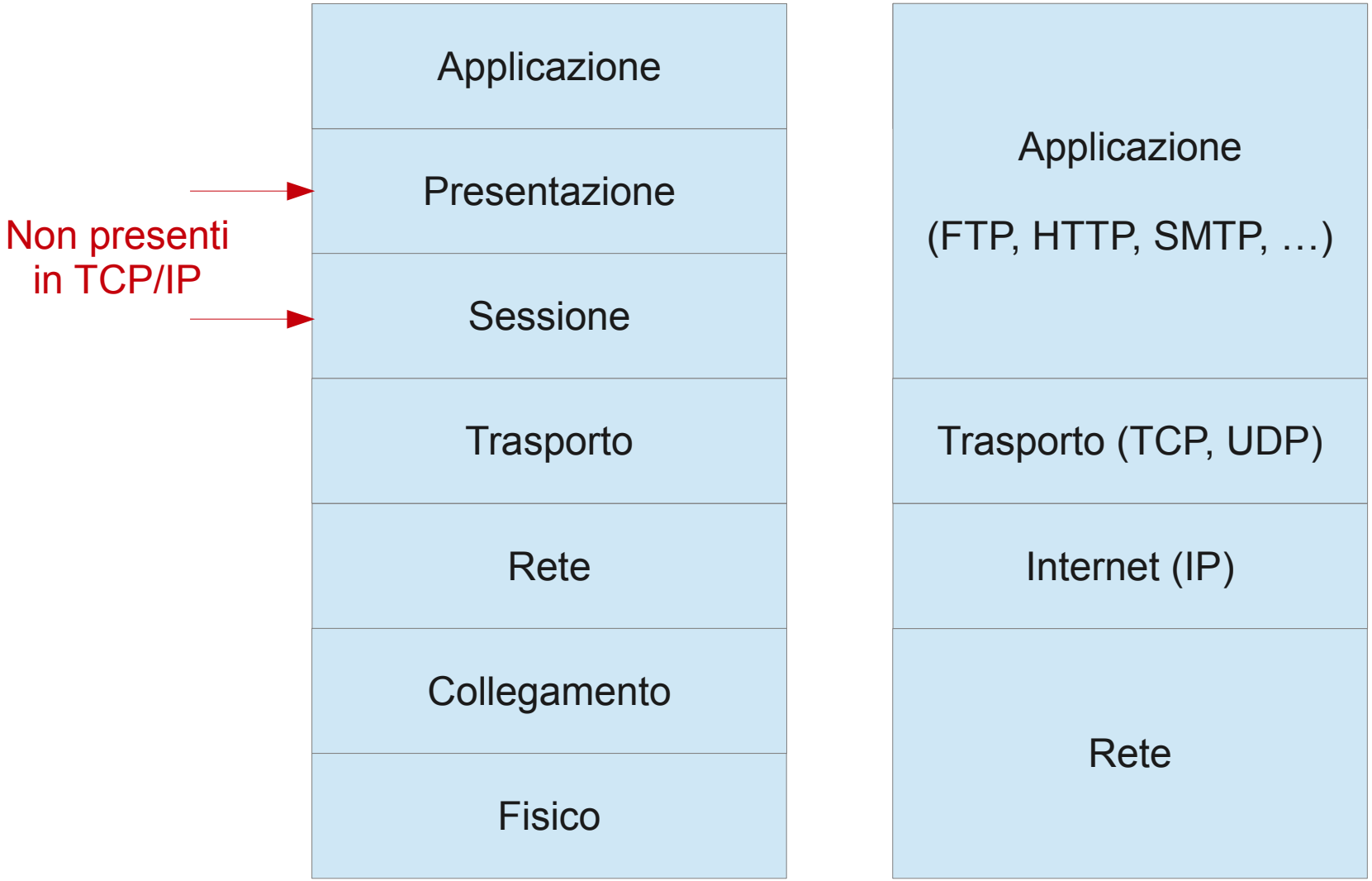
- Gestisce l'interfaccia fra utente e la macchina
- Fornisce i protocolli con cui operano le applicazioni

Over layer 7

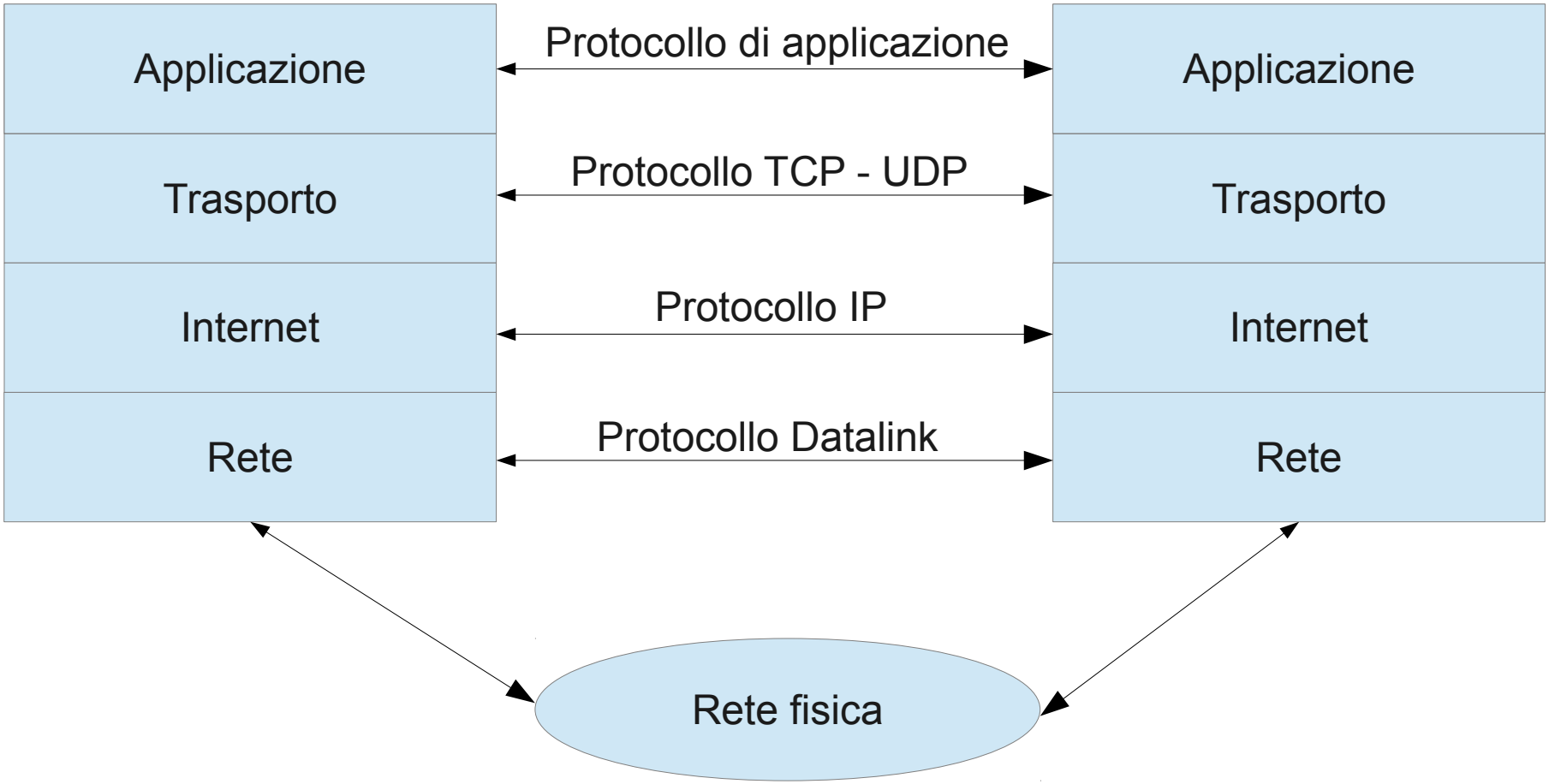
Il livello Utente

- Identifica l'utente (autenticazione) che esegue l'applicazione
- Permette di operare in base al suo livello di autorizzazione

I livelli ISO/OSI e TCP/IP



Protocolli oggetto di indagine



Sicurezza del livello di rete

- Concezione antiquata della sicurezza di rete
- Basata su socket e sull'associazione statica protocollo – porta
  - IP + porta del mittente
  - IP + porta del destinatario
- Ha limitazioni fortissime
- E' ancora uno strumento di network security ampiamente diffuso (e spesso usato in esclusiva)

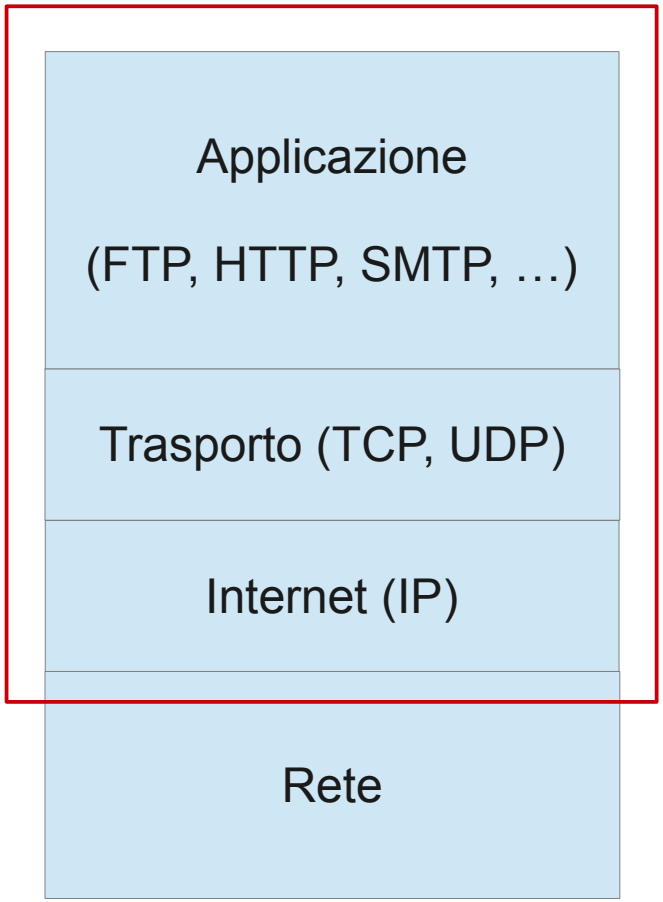
Limitazioni evidenti

- Filtrare protocolli di rete su porte non standard
- Filtrare in base all'applicazione utilizzata
- Filtrare in base all'utente che esegue l'applicazione

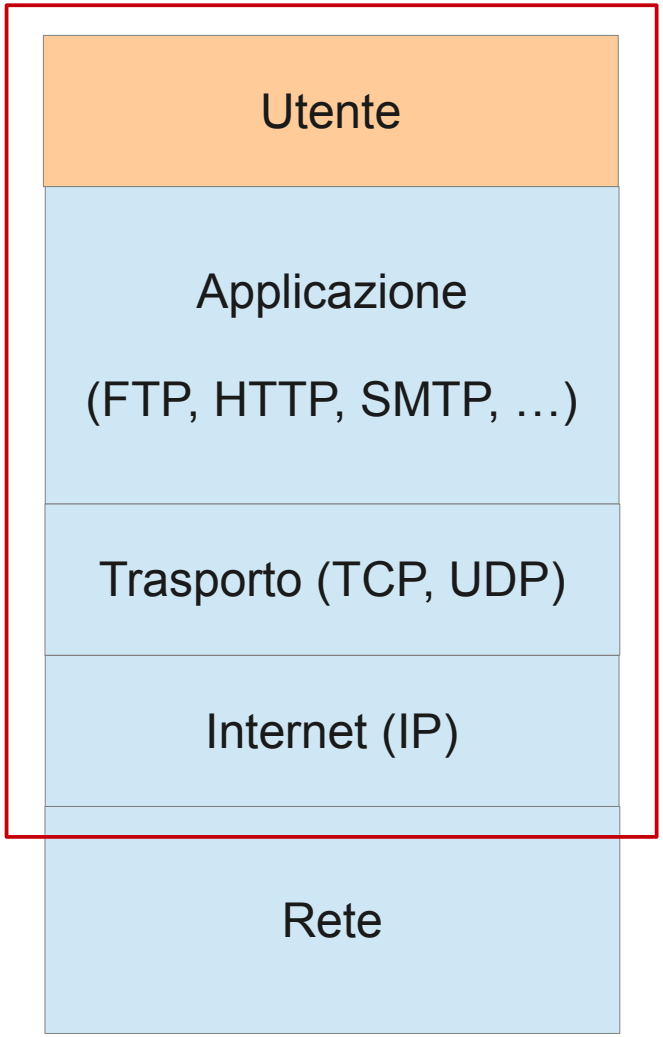
Sicurezza dal livello di rete

- L'ambito della sicurezza di rete non si limita pertanto al livello di rete
- A livello di rete transitano (incapsulate) tutte le informazioni destinate ai livelli superiori
- E' possibile mettere in campo controlli a livello di rete basati su tali informazioni
- Ciò permette di rilevare e filtrare minacce prima che si presentino al livello di destinazione

Ambito della sicurezza di rete



Ambito della sicurezza di rete



Transmission Control Protocol

- Orientato alla connessione
  - Necessario stabilire una connessione prima di avviare la trasmissione di dati
- Affidabile
  - Garantisce il recapito dei segmenti di comunicazione attraverso i meccanismi di *ack(nowledgements)* e ritrasmissione (timeout)
- Trasporta flussi di byte

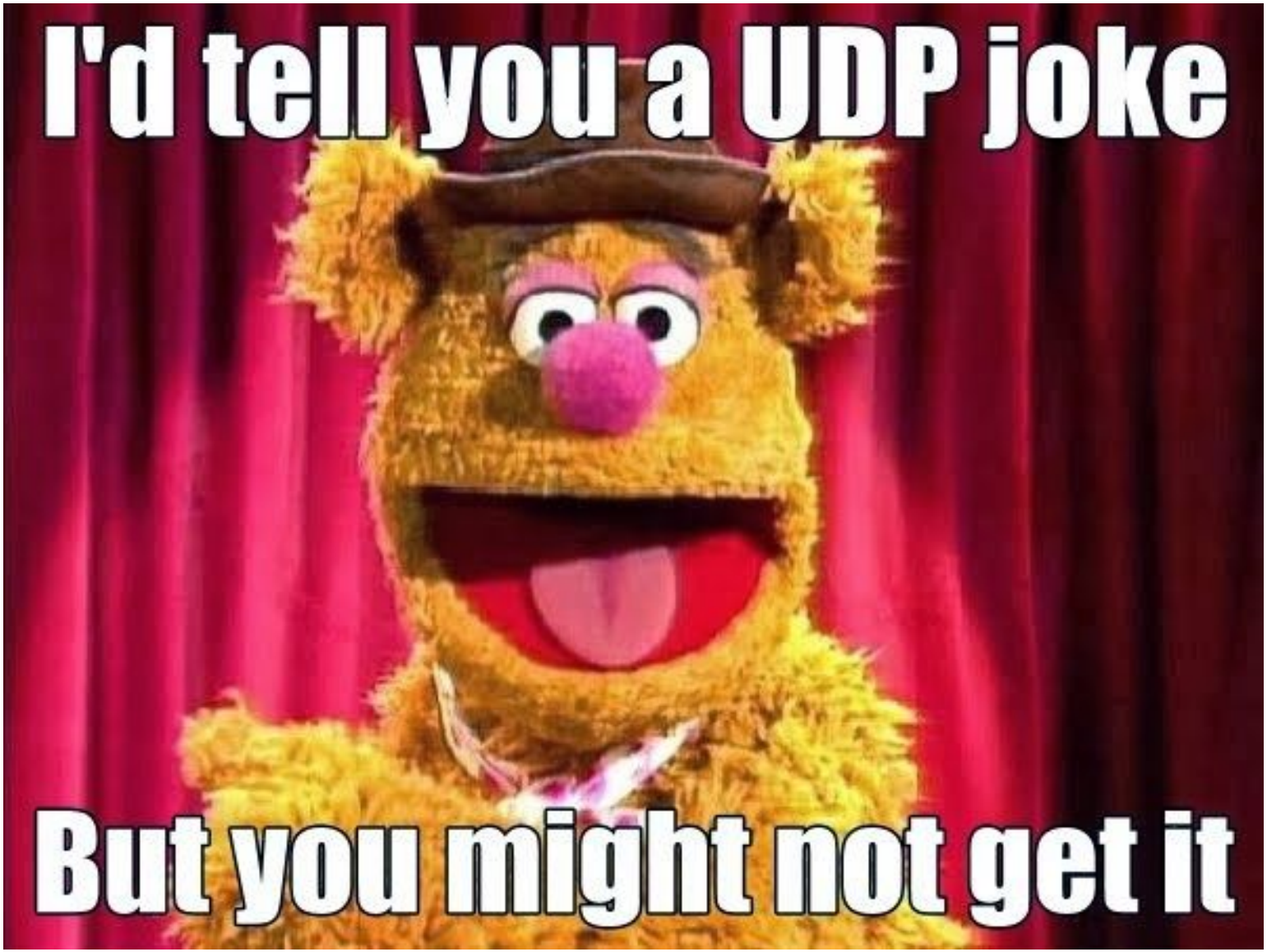
Transmission Control Protocol

- Garantisce l'arrivo a destinazione dei flussi nell'ordine di invio
- Implementa controllo degli errori di trasmissione mediante checksum
- Implementa controllo di flusso e di congestione mediante meccanismo della finestra scorrevole

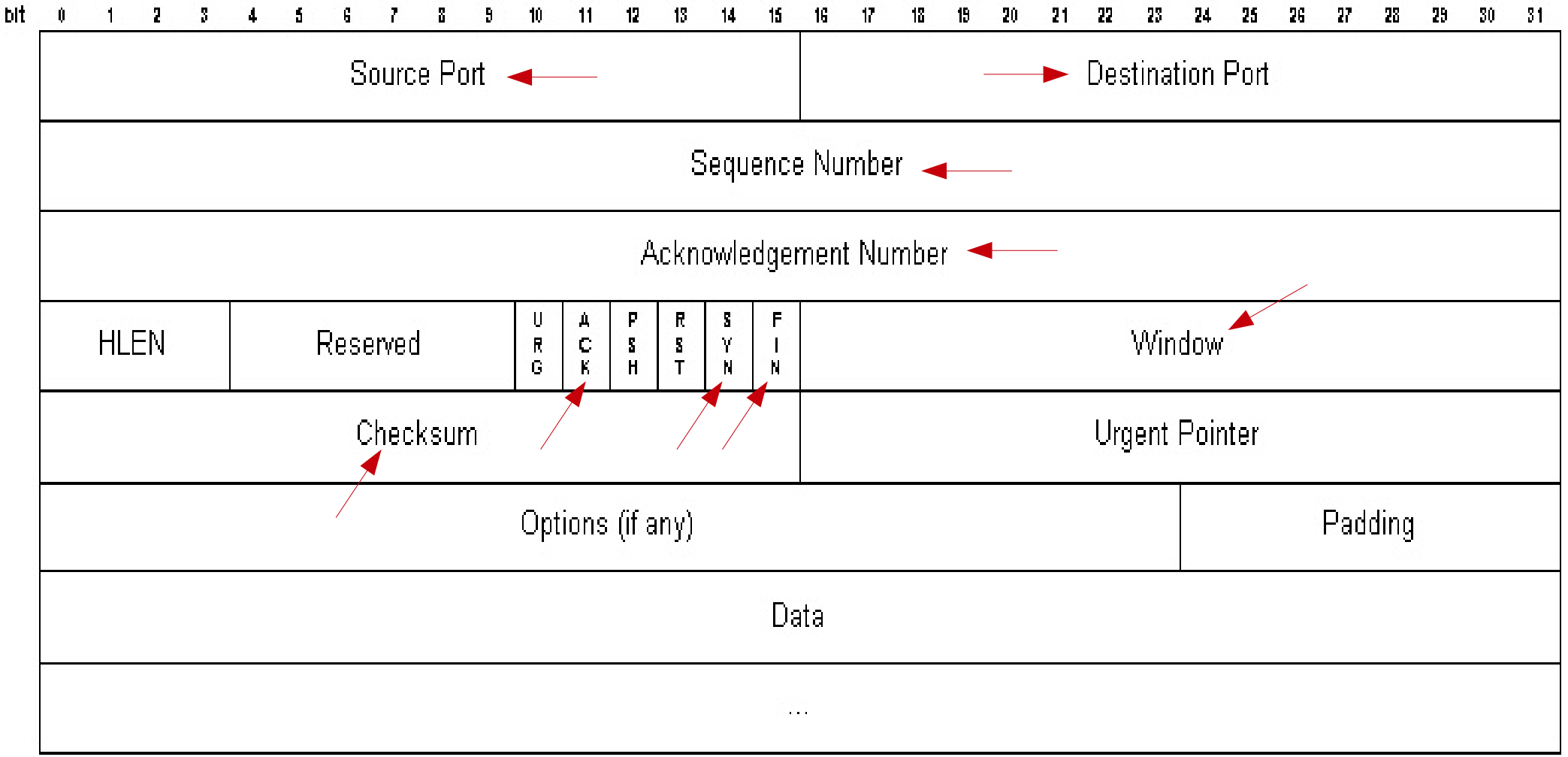
User Datagram Protocol

- Privo di connessione
- Non garantisce il recapito dei messaggi
- Invio dei soli datagrammi richiesti a livello applicativo, nessun pacchetto di servizio per la gestione di stati
  - Possono giungere a destinazione anche frammenti di messaggio

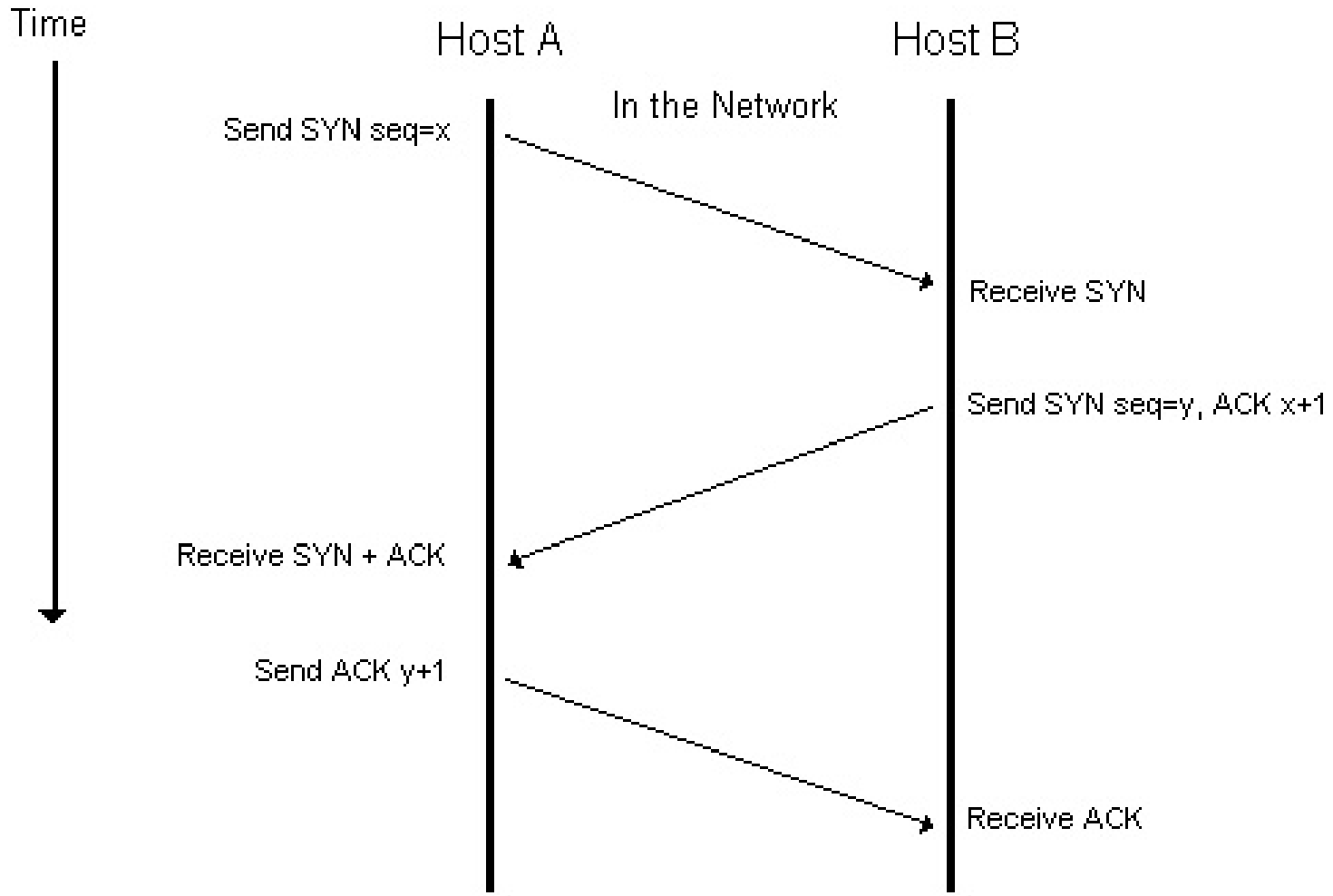
User Datagram Protocol



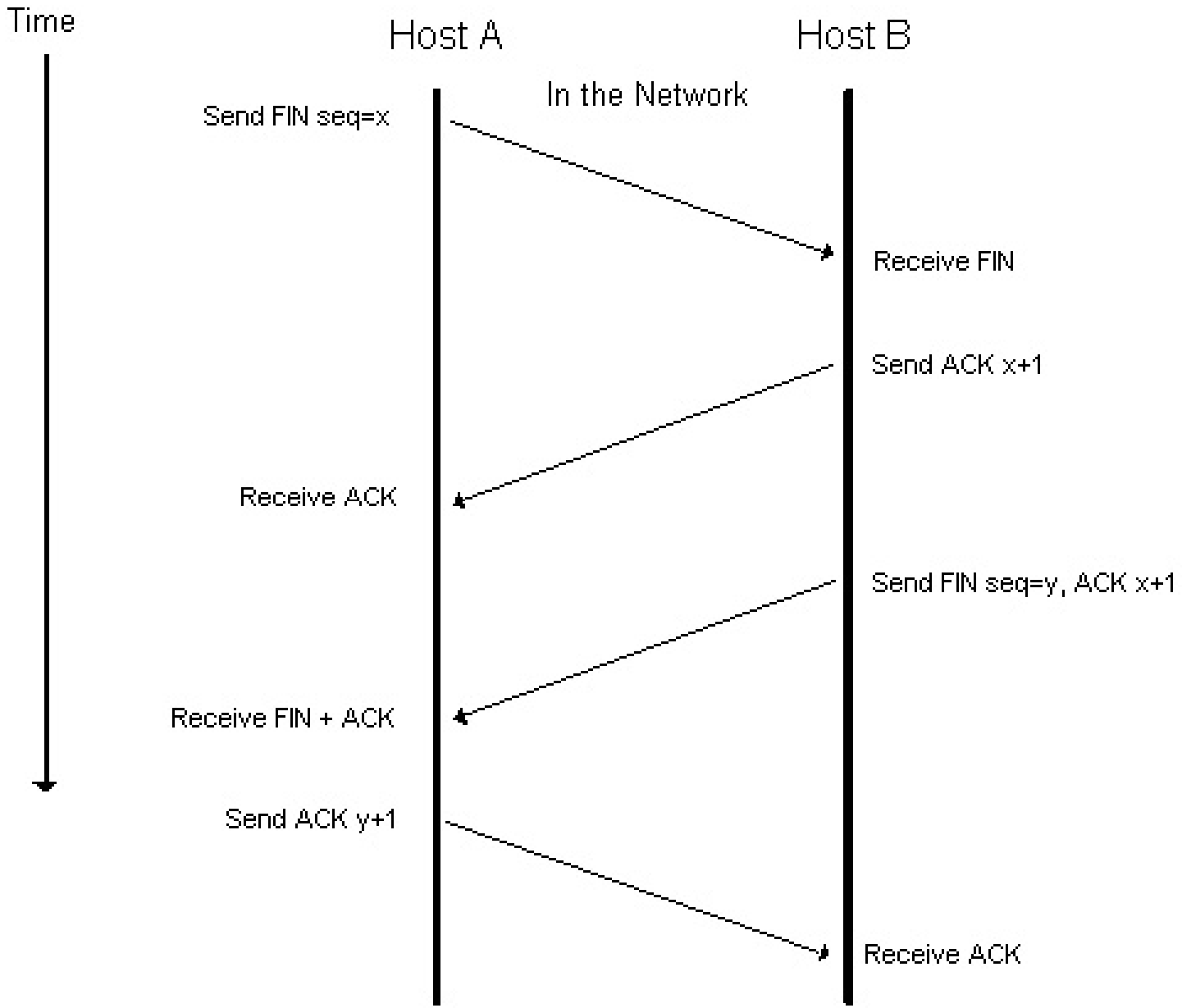
Header TCP


bit 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Source Port
Destination Port
Sequence Number
Acknowledgement Number
HLEN
Reserved
URG
ACK
PSH
RST
SYN
FIN
Window
Checksum
Urgent Pointer
Options (if any)
Padding
Data
...

Apertura connessione TCP – three-way handshake



Chiusura connessione TCP – four-way handshake



Chiusura connessione TCP

- Non del tutto standard
  - Dipende dell'implementazione dello stack TCP

- Alle volte è a tre vie (disconnessione immediata, senza attesa di ACK):
  - FIN / ACK
  - FIN / ACK
  - ACK

Esempio 1

Analisi di connessione e disconnessione di connessioni TCP ed UDP

Wireshark

Netcat

Esempio 2

Analisi del traffico di una connessione TCP (contenente dati)

Wireshark

Netcat

Python-echoer

Esercizio 1

Procedura Python per la scansione delle porte di un sistema

Evidenziare le porte TCP standard (1 – 1024) aperte
su un dato sistema (BackBox)

Suggerimento: vedere socket.connect_ex

Rilevazione e contromisure

- Monitoraggio del numero di richieste di connessione [SYN] in un intervallo temporale
  - Su porte diverse
  - Concluse con immediato [FIN,ACK] o [RST] di eventuali connessioni stabilite

- Filtraggio degli host che superano un limite su tale valore
  - a livello di host scansionato
  - a livello di apparati di rete (protezione altri host di rete)

Address Resolution Protocol

- Serve ad ottenere l'indirizzo fisico (MAC) di un host di cui si conosce l'indirizzo IP
- Il mittente invia una richiesta in broadcast (tutta la rete)
- Il destinatario la riceve e risponde con un messaggio diretto in cui include il proprio MAC address
- Il mittente salva il MAC nella propria tabella ARP e lo utilizza per l'invio della comunicazione in unicast

Address Resolution Protocol

- Dove sta il problema in tutto questo?

Address Resolution Protocol

- Dove sta il problema in tutto questo?

- Qualunque altro host in rete potrebbe rispondere alla richiesta fornendo il proprio MAC address

ARP spoofing

- Ci ha pensato per primo Alberto Ornaghi (ALoR), inventando la pratica dell'ARP spoofing o ARP poisoning

- Italians: spaghetti, pizza, mandolino, ARP spoofing

Funzionamento dell'ARP spoofing

- Alterazione malevola delle tabelle ARP delle postazioni sotto attacco
- Invio di risposte ARP (non richieste) che includono indirizzi IP non di proprietà dell'attaccante
  - Perché gli host accettano e mettono in cache risposte a richieste ARP mai effettuate?

Obiettivi dell'ARP spoofing

- Porsi nel mezzo di una comunicazione (Attacchi Man-in-the-middle o MITM)
  - Ricezione del traffico per l'host destinatario
  - Lettura ed eventuale alterazione dello stesso
  - Ritrasmissione

Esempio 3

ARP poisoning ed intercettazione del traffico di rete

Ettercap

Wireshark

Server FTP pubblici del Debian Project

Contromisure all'ARP poisoning

- Utilizzo di tabelle ARP statiche
- Filtrare le risposte ARP non richieste
- Attivare le opzioni di port security sugli switch
- Implementare il protocollo S-ARP
- Utilizzare protocolli che garantiscano l'identità dell'host remoto

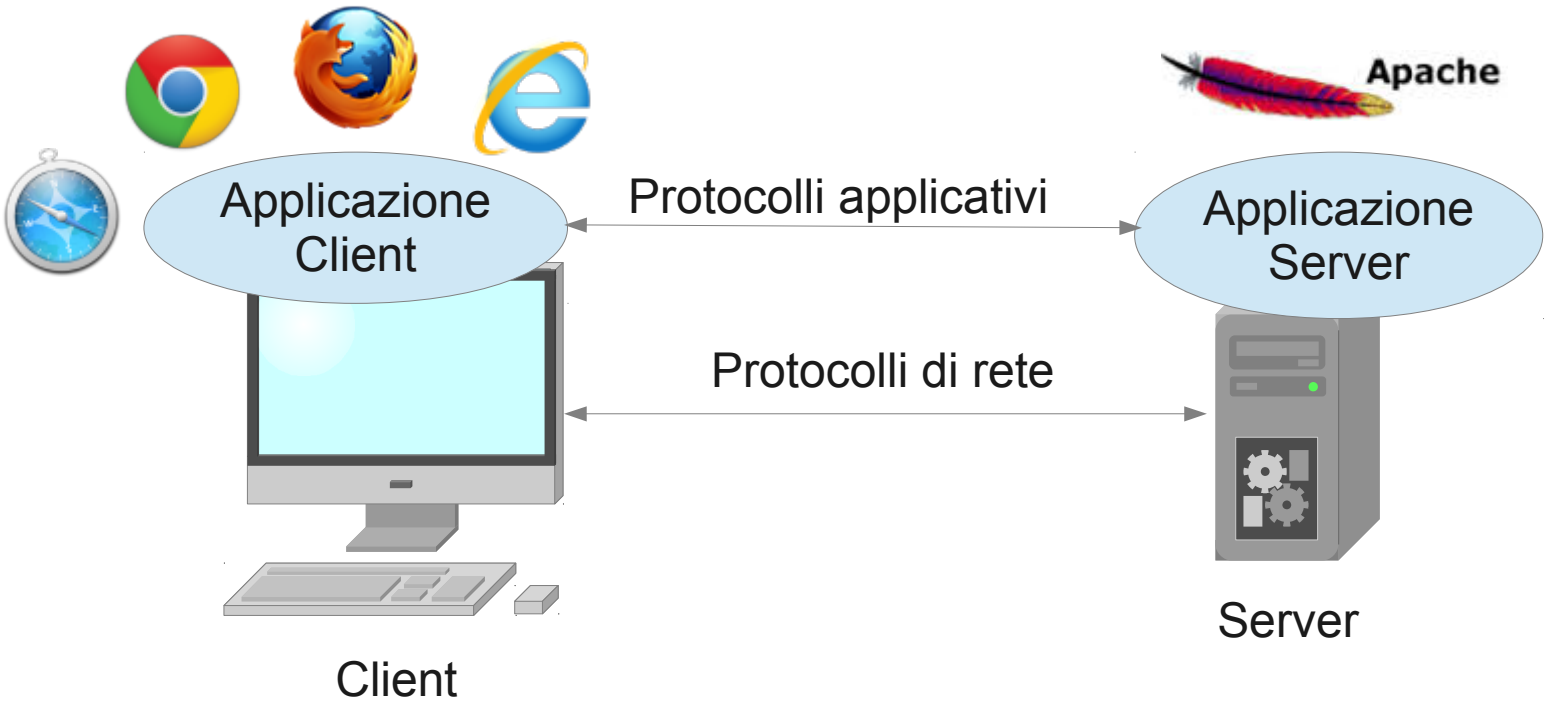
Rilevazione di pratiche di ARP poisoning

- Rilevare gli host che inviano risposte ARP non richieste
- Rilevare le modifiche al MAC address associato agli IP di rete
- Per le macchine critiche (gateway, server, etc), rilevare la presenza di MAC non coincidenti nelle risposte ARP in transito
- Rilevare la presenza di indirizzi MAC assegnati a più di un IP (*) nelle tabelle ARP degli host di rete

L'ascesa a livello applicativo

- Le applicazioni client–server interagiscono fra loro
- Possono appoggiarsi a servizi esterni
- Possono prevedere autenticazione
- Possono prevedere differenti livelli di autorizzazione basati sull'autenticazione

Architettura client-server



L'ascesa a livello applicativo

- I protocolli di comunicazione utilizzati in tali relazioni sono noti e le informazioni transitano a livello di rete
  - I dati in transito fra applicazioni e con eventuali servizi esterni possono essere ispezionati
  - Il sistema di autenticazione / autorizzazione può essere interrogato
  - La correlazione di tali informazioni può essere utilizzata per la definizione di regole di filtraggio

Vantaggi

- Aumento del livello di granularità del filtraggio
  - Filtraggio a livello IP delle connessioni che hanno effettuato più di un certo numero di tentativi di login
    - Inefficace se effettuato a livello applicativo: la gestione delle sessioni è effettuata “in collaborazione” col client
  - Filtraggio di protocolli o applicazioni in base all'utente che ne sta facendo uso

Vantaggi

- Operare in maniera indipendente dalla piattaforma
  - Posta elettronica
    - Server: Exchange, Postfix, Dovecot, etc
    - Client: Outlook, Thunderbird, Mail, etc
    - Protocolli standard: POP, IMAP, SMTP over TCP/IP
  - Web
    - Server: IIS, Apache, Nginx, Lighttpd, etc
    - Client: Internet Expolorer, Firefox, Chrome, Safari, etc
    - Protocollo: HTTP

Svantaggi

- Necessità di implementare sistemi di “chiarificazione” del traffico di rete
  - C'è la necessità di poter “porre nel mezzo” il firewall per ispezionare il traffico
  - Rischio di perdere “capra e cavoli”, nel caso di compromissione del sistema di chiarificazione
  - Questioni legali (posso applicare il filtraggio, ma non sapere cosa fa)

Sicurezza proattiva

- Può essere messo in atto filtraggio a livello di rete in base ad eventi rilevati e riportati dal livello applicativo
  - Attivazione di regole di filtraggio in base ad analisi dei log, eventualmente centralizzati
  - Push di regole di filtraggio dinamiche in base ad eventi rilevati da applicazioni e sonde di rete (IPS)
  - Blocco a livello IP

Vulnerabilità di rete e risvolti applicativi

- D'altra parte, vulnerabilità a livello di rete possono essere causa di compromissioni a livello applicativo
  - Insicurezza a livello di trasporto
  - Mancata segmentazione (routing / vlan)

Senza andare lontano... Top Ten 2013

- A1: Injection
- A2: Broken Authentication and Session Management
- A3: Cross-Site Scripting (XSS)
- A4: Insecure Direct Object References
- **A5: Security Misconfiguration**
- **A6: Sensitive Data Exposure**
- A7: Missing Function Level Access Control
- A8: Cross-Site Request Forgery (CSRF)
- **A9: Using Known Vulnerable Components**
- A10: Unvalidated Redirects and Forwards

Esempio: Sensitive Data Exposure

- Mancato utilizzo di crittografia sul canale per tutta la durata della sessione, ma solo per la fase di autenticazione.
- Trasporto in chiaro di informazioni riservate (token di sessione)
- Funzioni di completamento automatico attive per moduli contenenti dati confidenziali

Esempio 4

ARP poisoning e furto di sessione autenticata

Ettercap

Wireshark

Applicazione web vulnerabile ad A6

Esercizio 2

Procedura Python per attacco a forza bruta a login FTP

Effettuare un attacco a forza bruta al login di un dato server FTP, utilizzando un dizionario con la struttura

```
username:password
```

Suggerimento: vedere ftplib

Suggerimento: http://[redacted]/dizionario.txt

Esempio 5

Definizione delle modalità di rilevazione dell'attacco a forza bruta

Analisi del traffico di rete

Analisi dei log di sistema

Nella prossima puntata

- Rilevazione delle anomalie e gestione degli allarmi da parte di
  - Host sotto attacco
  - Apparati di rete
- Configurazione di IPS
- Definizione di politiche di filtraggio
- Configurazione e verifica di firewall

Riferimenti

- Internetworking con TCP/IP, Douglas Comer – Addison Wesley ISBN 88-7192-139-9
- The Transmission Control Protocol, John Kristoff – http://condor.depaul.edu/jkristof/technotes/tcp.html
- Internet Security, Maurizio Cinotti – Hoepli Informatica ISBN 88-203-3045-8
- S-ARP: a Secure Address Resolution Protocol, Bruschi, Ornaghi, Rosti – Università degli Studi di Milano

- **Software**
  - Ettercap Project, Alberto Ornaghi (ALoR) – https://ettercap.github.io/ettercap/
  - Wireshark – https://www.wireshark.org/

Università degli Studi di Verona
Corso di Laurea Magistrale in Informatica

Seminario sulla sicurezza delle reti

Andrea Zwirner

andrea@linkspirit.it

@AndreaZwirner