Relazione Approfondita di Cybersecurity

Analisi dei Protocolli di Rete

Basata sulle lezioni del Prof. Franco Arcieri

Report a cura di: Chiappini Mario e Salvucci Franco

Anno Accademico 2025-2026

Contents

1 Introduzione ai Modelli di Rete

Le lezioni introducono i due principali modelli di rete a strati, che servono a standardizzare e organizzare le complesse funzioni della comunicazione di rete.

- **Modello OSI (Open Systems Interconnection)**: Presentato come un modello concettuale e di riferimento. È strutturato in 7 livelli (Physical, Data Link, Network, Transport, Session, Presentation, Application). La sua utilità è primariamente didattica e per definire standard in modo rigoroso.
- **Modello TCP/IP**: Definito come il modello pratico e operativo su cui si basa Internet. È composto da 4 livelli (Network Access, Network, Transport, Application) che raggruppano le funzioni del modello OSI in modo più pragmatico.

2 Livello 2: Network Access e Data Link

Questo livello è fondamentale perché gestisce la comunicazione diretta tra dispositivi connessi allo stesso segmento di rete locale (LAN). È responsabile dell'indirizzamento fisico (MAC) e del controllo degli errori sul mezzo trasmissivo.

2.1 Struttura Pacchetto Ethernet

Il frame Ethernet è l'unità dati del livello 2. La sua struttura è cruciale per il corretto instradamento locale:

- **Destination/Source Ethernet Address** (6 bytes ciascuno): Sono gli indirizzi MAC (Media Access Control). Ogni scheda di rete ha un indirizzo MAC unico. L'indirizzo di destinazione può essere *broadcast* (tutti 1) per inviare un pacchetto a tutti i dispositivi sulla rete.
- **Length or Type Field** (2 bytes): Questo campo è "intelligente" e serve a due scopi:
  - **Length (IEEE 802.3)**: Se il valore è ≤ 1500 ($0x05DC$), indica la lunghezza in byte del campo dati che segue.
  - **Type (Ethernet II)**: Se il valore è > 1500 ($0x05DC$), indica l'EtherType, ovvero quale protocollo di livello superiore (es. IP, ARP) è incapsulato nel payload. Valori noti sono $0x0800$ per IP e $0x0806$ per ARP.
- **Dati (Payload)** (da 46 a 1500 bytes): Contiene i dati del protocollo di livello superiore. Il limite minimo di 46 bytes assicura che il frame sia sufficientemente lungo per essere rilevato correttamente sulla rete. Se il payload è più corto, viene aggiunto del *padding* (riempimento) per raggiungere la dimensione minima.
- **Frame Check Sequence (FCS)** (4 bytes): Un checksum calcolato sull'intero frame. Serve al ricevente per verificare se il pacchetto è stato corrotto durante la trasmissione.

2.2 Address Resolution Protocol (ARP)

L'ARP è un protocollo di servizio essenziale che colma il divario tra il Livello 3 (logico) e il Livello 2 (fisico).

- **Problema**: Per inviare un pacchetto IP (Livello 3) a un host sulla stessa rete locale, un dispositivo conosce l'indirizzo IP di destinazione, ma ha bisogno di conoscere l'indirizzo MAC (Livello 2) corrispondente per costruire il frame Ethernet.
- **Soluzione**: L'ARP utilizza un semplice formato di richiesta e risposta. I pacchetti ARP sono trasportati direttamente all'interno dei frame Ethernet, identificati dall'EtherType $0x0806$.
- **Processo (IPv4 su Ethernet)**:
  1. L'host A (mittente) invia un **ARP Request** (Operazione 1) in broadcast (MAC destinazione $FF:FF:FF:FF:FF:FF$) chiedendo "Chi ha l'indirizzo IP X?". L'host A include il proprio MAC (SHA) e IP (SPA).
  2. Tutti gli host sulla LAN ricevono la richiesta, ma solo l'host B (destinatario) che possiede l'IP X risponde.
  3. L'host B invia un **ARP Reply** (Operazione 2) in unicast (direttamente al MAC di A) dicendo "Io ho l'indirizzo IP X, e il mio MAC è Y".
- Questa informazione (associazione IP-MAC) viene poi memorizzata in una cache ARP su ogni host per ridurre il traffico futuro.

2.3 Network Sniffing

Lo sniffing è la tecnica di intercettazione e analisi del traffico di rete.

- **Obiettivo**: "Ascoltare" i pacchetti che transitano su un segmento di rete, anche se non sono destinati all'host che esegue lo sniffing. Questo è possibile ponendo la scheda di rete in *modalità promiscua*.
- **Tool**: I due strumenti principali menzionati sono `tcpdump` (un potente tool da riga di comando) e `Wireshark` (un analizzatore grafico più intuitivo e semplice da usare).
- **Tecniche**: Nelle reti moderne basate su switch, lo sniffing non è banale come nelle vecchie reti basate su hub. Per intercettare il traffico tra altri host è necessario posizionarsi in un punto strategico, ad esempio utilizzando un **TAP (Test Access Point)**, un dispositivo hardware che duplica il traffico che lo attraversa verso una porta di monitoraggio.

3 Livello 3: Network Layer

Questo livello è responsabile dell'indirizzamento logico (IP) e dell'instradamento dei pacchetti (*datagrammi*) attraverso reti diverse (Internet).

3.1 Protocollo IP (DOD IP)

Il protocollo IP (Internet Protocol), descritto in RFC791, è il pilastro del Livello 3. È un protocollo *connectionless* e *best-effort* (non garantisce la consegna, l'ordine o l'assenza di errori).

- **Version** e **IHL**: I primi 4 bit indicano la versione (es. 4 per IPv4), i successivi 4 bit (IHL) la lunghezza dell'header in parole da 32 bit. Un IHL minimo di 5 corrisponde a 20 byte (5×4 byte), l'header IP minimo senza opzioni.
- **Total Length**: Indica la dimensione totale dell'intero datagramma (header + dati) in byte. La dimensione massima è 65.535 byte, anche se raramente utilizzata.
- **Frammentazione (Identification, Flags, Fragment Offset)**: Campi usati per dividere i datagrammi in pezzi più piccoli (frammenti) per attraversare reti con una MTU (Maximum Transmission Unit) inferiore.
  - **Identification**: Un valore unico per tutti i frammenti di un singolo datagramma, per permetterne il riassemblaggio.
  - **Flags**: 3 bit. Il bit 'Don't Fragment' (DF) impedisce la frammentazione. Il bit 'More Fragments' (MF) è attivo per tutti i frammenti tranne l'ultimo.
  - **Fragment Offset**: Indica la posizione di questo frammento all'interno del datagramma originale, misurata in unità di 8 byte.
- **Time to Live (TTL)**: Un contatore a 8 bit che previene i loop di instradamento. È impostato dal mittente e decrementato di almeno uno da ogni router (gateway) che processa il pacchetto. Quando il TTL raggiunge lo zero, il pacchetto viene scartato e (generalmente) viene inviato un messaggio ICMP "Time Exceeded" al mittente.
- **Protocol**: Un campo a 8 bit che identifica il protocollo di Livello 4 incapsulato nel payload. I valori più comuni sono 1 (ICMP), 6 (TCP) e 17 (UDP).
- **Header Checksum**: Un checksum a 16 bit calcolato solo sull'header IP. Poiché il TTL (e potenzialmente le opzioni) cambia ad ogni hop, questo checksum deve essere ricalcolato da ogni router.
- **Source/Destination Address**: Gli indirizzi IP a 32 bit (per IPv4) del mittente e del destinatario finale.

3.2 Internet Control Message Protocol (ICMP)

Definito in RFC792, ICMP è un protocollo di supporto per IP. Non trasporta dati utente, ma messaggi di controllo, diagnostica e errore. I pacchetti ICMP sono incapsulati direttamente in IP (Protocollo IP 1).

- **Header**: Composto da **Type** (tipo di messaggio), **Code** (sottotipo/specifica del motivo) e **Checksum**.
- **Tipi di Messaggi Comuni**:
  - **Type 8 (Echo Request)** e **Type 0 (Echo Reply)**: Usati dal comando `ping` per verificare la raggiungibilità e misurare la latenza.

  - **Type 3 (Destination Unreachable)**: Inviato quando un router o l'host di destinazione non può consegnare il pacchetto. I codici specificano il motivo, es: *Code 1 (Host Unreachable)*, *Code 3 (Port Unreachable)* (molto comune, indica che il servizio/porta non è in ascolto).
  - **Type 11 (Time Exceeded)**: Inviato da un router quando scarta un pacchetto perché il suo TTL è arrivato a zero. Questo è il meccanismo sfruttato da `traceroute` per mappare la rotta verso una destinazione.
  - **Type 5 (Redirect)**: Usato dai router per informare un host su una rotta migliore per una specifica destinazione.

4 Livello 4: Transport Layer

Questo livello fornisce la comunicazione logica tra processi applicativi su host diversi. Introduce il concetto di **porta** per il multiplexing.

4.1 User Datagram Protocol (UDP)

Definito in RFC768, UDP è un protocollo *semplice*, *connectionless* e *inaffidabile* (unreliable).

- **Caratteristiche**: Non stabilisce una connessione prima di inviare dati. Non garantisce la consegna, l'ordine di arrivo o l'assenza di duplicati.
- **Header**: Estremamente leggero (8 byte):
  - **Source Port** e **Destination Port** (16 bit ciascuno): Permettono di indirizzare i dati all'applicazione corretta (servizio) sull'host.
  - **Length** (16 bit): Lunghezza totale del datagramma UDP (header + dati).
  - **Checksum** (16 bit): Un checksum opzionale (sebbene fortemente raccomandato) che copre header UDP, dati e uno "pseudo-header" IP.
- **Casi d'uso**: Ideale per applicazioni "transaction-oriented" dove la velocità è prioritaria sulla affidabilità, come DNS, DHCP, streaming video/audio e giochi online.

4.2 Transmission Control Protocol (TCP)

Definito in RFC793, TCP è il protocollo *affidabile* e *orientato alla connessione* di Internet.

- **Caratteristiche**: Fornisce un flusso di byte affidabile e ordinato. Gestisce il controllo di flusso (per non sovraccaricare il ricevente) e il controllo di congestione (per non sovraccaricare la rete).
- **Header TCP** (minimo 20 byte):
  - **Source/Destination Port** (16 bit ciascuno).
  - **Sequence Number (SN)** (32 bit): Usato per ordinare i segmenti e rilevare duplicati. Indica il numero di sequenza del primo byte di dati nel segmento.

- **Acknowledgment Number (ACK)** (32 bit): Se il bit ACK è attivo, questo campo indica il prossimo SN che il mittente si aspetta di ricevere. È il meccanismo base per la conferma di ricezione.
- **Data Offset** (4 bit): Indica la lunghezza dell'header TCP in parole da 32 bit.
- **Control Bits** (6 bit): Flag fondamentali per la gestione della connessione:
  - **SYN**: Usato per iniziare una connessione ("Synchronize").
  - **ACK**: Indica che il campo Acknowledgment Number è valido.
  - **FIN**: Usato per terminare una connessione ("Finish").
  - **RST**: Resetta la connessione in modo anomalo.
- **Window** (16 bit): Usato per il *controllo di flusso*. Indica quanti byte di dati il mittente di questo segmento è disposto ad accettare.
- **Checksum** (16 bit): Calcolato su header, dati e uno "pseudo-header" IP per garantire l'integrità.
- **Options**: Campo opzionale per funzionalità avanzate (es. MSS, Window Scale).

4.3 TCP State Machine e Handshake

Una connessione TCP è un'entità complessa che progredisce attraverso vari stati.

- **Stati Principali**: Includono **LISTEN** (server in attesa), **SYN-SENT** (client in attesa di risposta), **SYN-RECEIVED** (server in attesa di conferma), ed **ESTABLISHED** (connessione attiva, trasferimento dati).
- **Three-Way Handshake (Apertura)**: Il processo per stabilire una connessione:
  1. Il client (active open) invia un segmento **SYN** e passa allo stato SYN-SENT.
  2. Il server (passive open) riceve il SYN, invia un segmento **SYN+ACK** e passa a SYN-RECEIVED.
  3. Il client riceve il SYN+ACK, invia un segmento **ACK** e passa a ESTABLISHED. L'invio di dati può iniziare.
  4. Il server riceve l'ACK e passa anch'esso a ESTABLISHED.
- **Chiusura della Connessione (FIN)**: La chiusura è gestita indipendentemente per ciascuna direzione.
  - Un lato invia un **FIN** (passando a FIN-WAIT-1).
  - L'altro lato riceve il FIN, invia un **ACK** (passando a CLOSE-WAIT).
  - Quando anche il secondo lato ha finito di inviare dati, invia il suo **FIN** (passando a LAST-ACK).
  - Il primo lato riceve il FIN, invia un **ACK** e passa a TIME-WAIT (uno stato di attesa per garantire che l'ultimo ACK sia ricevuto).

5 Protocolli e Servizi Aggiuntivi

5.1 Dynamic Host Configuration Protocol (DHCP)

Definito in RFC1531, DHCP è un protocollo fondamentale per l'amministrazione di rete che automatizza l'assegnazione dei parametri di configurazione IP.

- **Scopo**: Fornisce agli host (client) i parametri necessari per operare su una rete IP, come indirizzo IP, subnet mask, default gateway e server DNS.
- **Funzionamento**: È un'estensione del protocollo BOOTP. Utilizza messaggi **BOOTREQUEST** (dal client) e **BOOTREPLY** (dal server), incapsulati in UDP.
- **Header DHCP**: Include campi cruciali per il processo di allocazione:
  - **Xid** (Transaction ID): Usato per associare richieste e risposte.
  - **Chaddr** (Client Hardware Address): Il MAC address del client, usato dal server per identificare il dispositivo.
  - **Ciaddr** (Client IP Address): Usato dal client quando sta rinnovando un lease esistente.
  - **Yiaddr** ("Your" IP Address): Il campo dove il server inserisce l'indirizzo IP che sta offrendo/assegnando al client.
  - **Siaddr** (Server IP Address): L'indirizzo IP del server DHCP.
- Il processo (non esplicitato ma implicito) segue i passi DORA (Discover, Offer, Request, Acknowledge), che utilizzano questi campi per negoziare e confermare un lease di un indirizzo IP.