Introduzione agli argomenti di Fondamenti di Reti e Sicurezza Informatica

Lezione 1: Introduzione e Stack di Protocolli

1. Internet e i Protocolli di Comunicazione

Cos'è Internet

Internet è una vasta rete globale di reti di computer che utilizzano il protocollo IP (Internet Protocol) per comunicare tra loro. È costituito da milioni di reti private, pubbliche, aziendali e governative, tutte collegate tramite una vasta gamma di tecnologie di trasmissione come cavi in fibra ottica, wireless e altre infrastrutture.

Protocollo di Comunicazione

Un protocollo di comunicazione è un insieme di regole che definiscono come i dati vengono trasmessi e ricevuti su una rete. Esempi di protocolli includono TCP (Transmission Control Protocol), IP (Internet Protocol), HTTP (Hypertext Transfer Protocol) e FTP (File Transfer Protocol).

2. Stack di Protocolli

Livelli dello Stack

Lo stack di protocolli è suddiviso in cinque livelli principali:

- **Livello Applicazione**: Fornisce servizi di rete direttamente agli utenti finali, come email, file transfer, web browsing. Protocollo esempio: HTTP.
- **Livello Trasporto**: Assicura la trasmissione dati end-to-end tra host, fornendo servizi come l'affidabilità (TCP) o la consegna veloce (UDP).
- **Livello Rete**: Gestisce l'instradamento dei pacchetti di dati tra dispositivi su reti diverse. Protocollo esempio: IP.
- **Livello Collegamento**: Controlla la trasmissione dei dati tra due dispositivi su una singola rete fisica. Protocollo esempio: Ethernet.
- **Livello Fisico**: Si occupa della trasmissione effettiva dei bit su un mezzo fisico, come cavi o segnali wireless.

3. Storia e Sviluppi

La storia di Internet inizia negli anni '60 con lo sviluppo di ARPANET, finanziato dal Dipartimento della Difesa degli Stati Uniti. Negli anni '80, il protocollo TCP/IP divenne lo standard per la trasmissione di dati su Internet. Con la creazione del World Wide Web negli anni '90, Internet è diventato accessibile a un pubblico più ampio.

4. Modelli di Comunicazione

Client/Server

Il modello client/server è un'architettura di rete in cui un server fornisce risorse o servizi a uno o più client. Ad esempio, un web server fornisce pagine web a un browser web.

Peer-to-Peer (P2P)

Il modello peer-to-peer (P2P) è un'architettura di rete in cui i nodi possono agire sia come client che come server, condividendo risorse direttamente tra loro. Esempi di P2P includono le reti di file sharing come BitTorrent.

Lezione 2: Livello Applicazione, SMTP

1. Protocollo SMTP (Simple Mail Transfer Protocol)

SMTP è un protocollo utilizzato per inviare email da un client a un server di posta elettronica e tra server di posta elettronica. Utilizza la porta 25 per la comunicazione.

Interazione con POP e IMAP

Mentre SMTP gestisce l'invio delle email, POP (Post Office Protocol) e IMAP (Internet Message Access Protocol) gestiscono la ricezione delle email sui client.

2. Struttura dei Messaggi Email

Un messaggio email è composto da due parti principali: header e body.

Header

Il header di un messaggio email contiene informazioni di controllo come mittente, destinatario, data e oggetto dell'email.

Body

Il corpo del messaggio contiene il contenuto effettivo dell'email, che può essere in formato testo o HTML.

Lezione 3: Livello Applicazione/Trasporto, DNS

1. DNS (Domain Name System)

Il sistema di nomi di dominio (DNS) è un servizio che traduce i nomi di dominio in indirizzi IP e viceversa. Il DNS traduce i nomi di dominio leggibili dall'uomo (come www.example.com) in indirizzi IP utilizzabili dai computer (come 192.0.2.1).

Tipi di Record

- **A**: Mappa un nome di dominio a un indirizzo IPv4.
- **AAAA**: Mappa un nome di dominio a un indirizzo IPv6.
- **CNAME**: Alias di un altro nome di dominio.
- **MX**: Specifica i server di posta elettronica responsabili per un dominio.

2. Processo di Risoluzione DNS

Richiesta DNS

Un client invia una richiesta a un resolver DNS.

Risposta DNS

Il resolver interroga vari server DNS (root, TLD, autoritativi) fino a ottenere la risposta corretta.

Lezione 4: Livello Applicazione, HTTP

1. Protocollo HTTP (Hypertext Transfer Protocol)

HTTP è il protocollo utilizzato per la trasmissione di documenti ipertestuali sul Web.

Metodi HTTP

- **GET**: Richiede una rappresentazione di una risorsa specificata.
- **POST**: Invia dati al server per elaborazione.
- **PUT**: Carica una rappresentazione di una risorsa specificata.
- **DELETE**: Elimina una risorsa specificata.

2. Struttura delle Richieste e Risposte HTTP

Richiesta HTTP

Una richiesta HTTP comprende una linea di richiesta (metodo, URL, versione HTTP), header di richiesta e un corpo opzionale. Esempio:

```
GET /index.html HTTP/1.1
Host: www.example
```

Risposta HTTP

Comprende una linea di stato (versione HTTP, codice di stato, frase di ragione), header di risposta e un corpo opzionale. Esempio:

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1274
```

Lezione 5: Livello Trasporto (UDP, TCP)

1. Protocollo TCP (Transmission Control Protocol)

Il protocollo TCP è un protocollo di trasporto affidabile che garantisce la consegna dei dati nell'ordine corretto. Utilizza un handshake a tre vie per stabilire una connessione.

Caratteristiche di TCP

- **Affidabilità**: Garantisce che i dati persi siano ritrasmessi.
- **Controllo di Flusso**: Regola la velocità di invio dei dati per prevenire il sovraccarico del ricevitore.
- **Controllo di Congestione**: Regola la quantità di dati inviati per evitare la congestione della rete.

2. Protocollo UDP (User Datagram Protocol)

UDP è un protocollo di trasporto leggero e non affidabile. Non garantisce la consegna dei dati né il loro ordine.

Caratteristiche di UDP

- **Assenza di Connessione**: Non stabilisce una connessione prima di inviare dati.
- **Velocità**: È più veloce di TCP poiché non ha overhead per garantire affidabilità.
- **Casi d'uso**: Applicazioni che richiedono velocità e possono tollerare la perdita di alcuni pacchetti, come streaming video e giochi online.

Lezione 6a: Livello Rete (IP, NAT)

1. Protocollo IP (Internet Protocol)

Indirizzamento IP

- **IPv4**: Utilizza indirizzi a 32 bit. Esempio: 192.0.2.1
- **IPv6**: Utilizza indirizzi a 128 bit. Esempio: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

Struttura dei Pacchetti IP

Comprende un header con informazioni di controllo (indirizzo sorgente, indirizzo destinazione, TTL, ecc.) e un payload con i dati effettivi.

2. NAT (Network Address Translation)

Il NAT traduce gli indirizzi IP privati in indirizzi IP pubblici, permettendo a più dispositivi su una rete privata di condividere un singolo indirizzo IP pubblico.

Caratteristiche del NAT

- **Conservazione degli Indirizzi IP**: Riduce la necessità di indirizzi IP pubblici.
- **Sicurezza**: Nasconde gli indirizzi IP privati, rendendo più difficile l'attacco diretto ai dispositivi interni.

Lezione 6b: Livello Rete (Subnetting)

1. Subnetting

Il subnetting è la suddivisione di una rete IP in sottoreti più piccole per migliorare la gestione e la sicurezza della rete.

Vantaggi del Subnetting

- **Migliore Gestione**: Permette di organizzare la rete in segmenti più piccoli.
- **Sicurezza**: Limita la diffusione di traffico e attacchi alla rete.

Calcolo degli Indirizzi di Rete e Broadcast

Per calcolare gli indirizzi di rete e broadcast di una sottorete, si utilizzano l'indirizzo IP e la subnet mask.

2. CIDR (Classless Inter-Domain Routing)

La notazione CIDR rappresenta i blocchi di indirizzi IP in forma compatta, ad esempio 192.0.2.0/24.

Lezione 6c: Livello Rete (Firewalls)

1. Firewall

Un firewall è un dispositivo di sicurezza di rete che monitora e controlla il traffico di rete in base a regole di sicurezza predefinite.

Tipi di Firewall

- **Packet-Filtering**: Controlla i pacchetti in base a regole definite su indirizzi IP, porte e protocolli.
- **Stateful**: Tiene traccia dello stato delle connessioni e permette solo i pacchetti che fanno parte di una connessione stabilita.
- **Application-Level**: Controlla il traffico a livello di applicazione, esaminando il contenuto dei pacchetti.

2. Esempi di Regole Firewall

- **Permettere il Traffico**: Consente il traffico in base a criteri specifici, come indirizzi IP sorgente/destinazione e numeri di porta.
- **Bloccare il Traffico**: Blocca il traffico in base a criteri specifici, come indirizzi IP sorgente/destinazione e tipi di protocollo.

Lezione 7: Livello Data Link (Ethernet, WLAN)

1. Ethernet

Ethernet è un protocollo di rete per reti locali (LAN) che utilizza cavi di rame o fibra ottica per la trasmissione dei dati.

Topologie di Reti Ethernet

Le configurazioni fisiche e logiche di una rete Ethernet, come bus, stella e anello. La topologia più comune è la topologia a stella.

Collision Domain

Un'area di rete in cui i pacchetti possono collidere tra loro, con conseguente necessità di meccanismi di controllo delle collisioni.

2. WLAN (Wireless Local Area Network)

Standard Wi-Fi

Le tecnologie wireless come IEEE 802.11a/b/g/n/ac/ax che consentono la connessione senza fili tra dispositivi. I dispositivi Wi-Fi devono supportare lo stesso standard per comunicare tra loro.

Sicurezza WLAN

Metodi per proteggere le reti wireless da accessi non autorizzati, come WEP, WPA, WPA2 e WPA3. La crittografia e l'autenticazione sono importanti per garantire la sicurezza delle reti WLAN.

Lezione 8: Livello Fisico

1. Livello Fisico

Il livello fisico del modello OSI si occupa della trasmissione dei bit su un mezzo fisico. Include componenti come cavi, connettori e trasmettitori/ricevitori.

Tipi di cavi

- **Cavo Coassiale**: Utilizzato per trasmissioni a banda larga e CATV.
- **Fibra Ottica**: Utilizza luce per trasmettere dati ad alte velocità e lunghe distanze.
- **Coppia Ritorta**: Utilizzata nelle reti Ethernet, es. cavo UTP (Unshielded Twisted Pair).

2. Tecnologie di Trasmissione

Modulazione

Le tecniche di modulazione, come AM (Amplitude Modulation) e FM (Frequency Modulation), vengono utilizzate per codificare i dati sui segnali di trasmissione.

Codifica

I processi di codifica convertono i dati digitali in segnali fisici per la trasmissione, come NRZ (Non-Return-to-Zero) e Manchester Encoding.

Lezione 9: Introduzione alla Cybersecurity

1. Fondamenti di Sicurezza Informatica

- **Confidenzialità**: Protezione delle informazioni da accessi non autorizzati.
- **Integrità**: Garanzia che i dati non siano stati alterati in modo non autorizzato.
- **Disponibilità**: Assicurarsi che le informazioni e i servizi siano disponibili quando necessario.

2. Minacce e Vulnerabilità

- **Malware**: Software dannoso progettato per danneggiare o compromettere i sistemi, come virus, worm e trojan.
- **Phishing**: Tecniche di inganno per ottenere informazioni sensibili, come password e dati personali.
- **Attacchi DDoS**: Attacchi di Denial of Service distribuiti che mirano a rendere indisponibili i servizi di rete sovraccaricando i server.

3. Metodi di Protezione

- **Crittografia**: Tecniche per proteggere i dati mediante la codifica in modo che solo i destinatari autorizzati possano leggerli.
- **Autenticazione**: Metodi per verificare l'identità degli utenti, come password, token di sicurezza e biometria.
- **Controllo degli Accessi**: Meccanismi per controllare chi può accedere a risorse e informazioni specifiche.