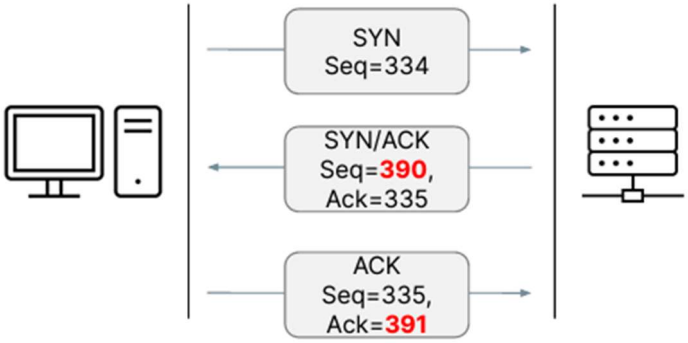
I protocolli di rete sono un insieme di regole e convenzioni che permettono a dispositivi elettronici di comunicare tra loro su una rete. Essi definiscono come i dati devono essere formattati, trasmessi e ricevuti per garantire una comunicazione corretta ed efficiente. I protocolli possono essere visti come linguaggi standardizzati che i dispositivi utilizzano per comunicare.
I protocolli fanno in modo che:
- I dispositivi di diversi produttori possano comunicare senza problemi (**interoperabilità**)
- I dati trasmessi siano consegnati in modo corretto e completo (**Affidabilità**)
- La rete utilizzi le risorse disponibili al meglio (**Efficienza**)
- Le informazioni trasmesse siano protette da accessi non autorizzati (**Sicurezza**)

I protocolli di rete possono essere classificati in vari modi, a seconda del livello del modello OSI in cui operano e della loro funzionalità:

1. Protocolli di Livello Fisico e Data Link:
   - Ethernet: Protocollo del livello Data Link utilizzato nelle LAN (Local Area Network). Fornisce accesso al mezzo fisico e controllo degli errori di trasmissione.
   - Wi-Fi: Standard di comunicazione wireless per reti locali. Permette la trasmissione dati senza l'uso di cavi, utilizzando onde radio.
2. Protocolli di Livello di Rete:
   - IP (Internet Protocol): Responsabile dell'instradamento e dell'indirizzamento dei pacchetti attraverso la rete.
   - ICMP (Internet Control Message Protocol): Utilizzato per diagnosticare errori e gestire il traffico di rete (es. comando ping).
3. Protocolli di Livello di Trasporto:
   - TCP (Transmission Control Protocol): Garantisce una consegna affidabile e ordinata dei dati grazie al controllo di flusso e ritrasmissione.
   - UDP (User Datagram Protocol): Fornisce un servizio di consegna veloce ma senza connessione, senza garanzia di ricezione.
4. Protocolli di Livello di Applicazione:
   - HTTP (HyperText Transfer Protocol): Utilizzato per la trasmissione di contenuti web tra browser e server.
   - FTP (File Transfer Protocol): Utilizzato per il trasferimento di file tra client e server. Supporta autenticazione e gestione remota dei file.
   - SMTP (Simple Mail Transfer Protocol): Utilizzato per l'invio delle e-mail da client a server o tra server di posta.

TCP
Il protocollo TCP è un protocollo orientato alla connessione e garantisce la trasmissione affidabile e ordinata dato che prima di effettuare uno scambio di pacchetti viene effettuata una comunicazione tra i due dispositivi (three-way-handshake)
SYN: il mittente invia un pacchetto (SYN) al destinatario
SYN-ACK: il destinatario risponde con un pacchetto SYN-ACK
ACK: il mittente risponde con un pacchetto ACK stabilendo così la connessione
Questo scambio avviene tramite dei flag dove:
il mittente usa un numero generato casualmente e lo usa come flag SYN
successivamente, il destinatario prende quel numero, lo incrementa di uno e lo usa come flag di ACK e genera un altro SYN casuale.
Infine, i mittente incrementerà a sua volta il SYN flag ricevuto e invertirà i codici SYN e ACK. Da ora in poi i due dispositivi comunicheranno usando questi due codici



UDP (User Datagram Protocol)
È un protocollo di trasporto della suite TCP/IP, progettato per essere veloce e leggero, ma non affidabile. Funziona in modo connectionless, cioè non stabilisce una connessione prima di inviare i dati e non verifica se i dati arrivano a destinazione.
Il suo funzionamento inizia con un applicazione che genera i dati da inviare e li incapsula in un **datagramma** contenente la porta sorgente e la porta di destinazione, la lunghezza del datagramma e il **checksum** (anche se opzionale). Il datagramma viene inviato direttamente al livello IP e successivamente inoltrato verso il destinatario che riceverà il datagramma senza alcuna garanzia che sia completo, ordinato o anche solo arrivato.

Il datagramma è un pacchetto autonomo specifico per i protocolli connectionless, come UDP e IP, contenente sia i dati da inviare che le informazioni per poterlo instradare nella rete.

Il checksum Il checksum è un valore di controllo calcolato sommando i bit dei dati contenuti all'interno del pacchetto e serve per verificare che i dati siano arrivati tutti correttamente.
Se il valore combacia con quello inviato, si presume che i dati siano integri, altrimenti, il pacchetto viene considerato corrotto.

DNS (Domain Name System)
Opera al livello 7 della pila (Application Layer) e la sua funzione principale è tradurre l'indirizzo IP di un server, come ad esempio 142.250.184.36, in un dominio leggibile dagli esseri umani, come www.store.google.com.

Passaggi:

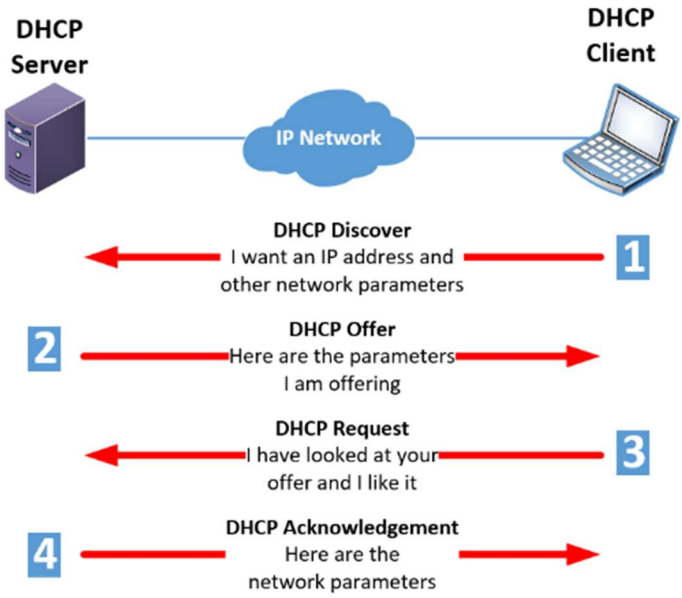
1. **Local DNS**
   quando il client tenta di accedere a "www.store.google.com" il sistema operativo verifica se l'indirizzo IP è già presente nella cache del DNS locale. Se non è presente, invia una richiesta di risoluzione al Resolver DNS
2. **Resolver DNS**
   Riceve la richiesta "www.store.google.com" da risolvere e se non ha le informazioni richieste nella sua cache, inizia il processo di risoluzione interrogando il root DNS per ricevere informazioni riguardanti il TLD che nel nostro caso è ".com".
3. **Root DNS**
   riceve la richiesta di risoluzione del TLD ".com" da parte del Resolver e se conosce direttamente il dominio "www.store.google.com" risponde con il suo indirizzo IP pubblico, altrimenti risponde con l'indirizzo IP del server Autoritativo per il dominio di "google.com".
4. **Auth. Server (google.com)**
   il Resolver DNS, riceve l'indirizzo IP del server autoritativo per "google.com" e quindi invia una richiesta per poter risolvere il sottodominio di "store.google.com"
5. **Auth. Server (store.google.com)**
   il server autoritativo per "store.google.com" riceve la richiesta da parte del Resolver e se ha già l'indirizzo IP di "www.store.google.com" risponde fornendolo direttamente, altrimenti risponde con l' indirizzo IP del server autoritativo per risolvere l'host name di "www.store.google.com"
6. **Auth. Server (www.store.google.com)**
   il Resolver riceve l'indirizzo IP del server autoritativo per risolvere l'host name di "www.store.google.com" e gli invia una richiesta per ottenere l'indirizzo IP che a sua volta invierà direttamente l'indirizzo IP finale per quel dominio.
7. **Local DNS**
8. Il Resolved ottiene cosí l'indirizzo IP finale che invierà al Local DNS del client e metterà le informazioni ricevute nella sua cache per non dover rifare il procedimento da capo.

FTP (porta 20 e 21)
Il File Transfer Protocol (FTP) è un protocollo di rete standard utilizzato per trasferire file da un host a un altro su una rete basata su TCP, come Internet. FTP consente il caricamento (upload) e lo scaricamento (download) di file tra client e server, fornendo un mezzo affidabile per il trasferimento di dati.
L'FTP viene spesso utilizzato dagli sviluppatori per caricare file, dalle aziende per poter distribuire software e viene utilizzato anche per trasferire file di backup verso server remoti.
Alcuni metodi (comandi) usati sono:
STOR (Store):
Serve per caricare (upload) un file dal client al server.
Il client invia [STOR nomefile], il server risponde con un codice di stato e aspetta il file.
RETR (Retrieve):
Serve per scaricare (download) un file dal server al client.
Il client invia [RETR nomefile], il server risponde e inizia il trasferimento del file.
LIST:
Serve per ottenere la lista dei file e directory presenti in una directory sul server.
Il client invia [LIST], e il server risponde con un elenco dettagliato.
Il suo funzionamento avviene usando la porta 21 per il controllo, dove vengono inviati i comandi e ricevute le risposte, e la porta 20 per il trasferimento dei dati.
Nella modalità attiva, il client apre una porta e attende una connessione dal server, mentre nella modalità passiva, il server apre una porta e attende una connessione da un client. Quest'ultima viene comunemente usata per superare eventuali problemi con firewalls e NAT.
FTP funziona anche senza il bisogno di fornire credenziali e tutti i dati vengono trasmessi in chiaro.

Telnet (porta 23)
Telnet è un protocollo di rete utilizzato per fornire una comunicazione bidirezionale interattiva tra due macchine su una rete permettendo a un utente di accedere a un computer remoto e di eseguire comandi come se fosse collegato localmente.
Fornisce un'interfaccia a riga di comando per interagire con sistemi remoti, ma è considerato non sicuro poiché trasmette i dati in chiaro, incluse credenziali sensibili. È uno strumento flessibile, compatibile con diversi dispositivi come router, switch e server, e supporta comandi per la gestione delle sessioni (es. open, close, quit).
Il protocollo Telnet viene impiegato per accedere e gestire server remoti, configurare dispositivi di rete come router e switch, automatizzare attività amministrative tramite scripting, e testare la connettività di rete o servizi su porte specifiche per scopi di debugging.
A causa delle sue vulnerabilità di sicurezza, Telnet è stato in gran parte sostituito da SSH (**Secure Shell**), che fornisce una connessione cifrata e sicura. SSH offre le stesse funzionalità di Telnet ma con un livello di sicurezza molto più elevato.

FTPS (porta 21, 989 e 990)
FTPS è la versione sicura del protocollo FTP che utilizza SSL/TLS per la cifratura dei dati che sono dei protocolli crittografici progettati per fornire sicurezza nelle comunicazioni di rete.
TLS (Transport Layer Security) è il successore di SSL (Secure Sockets Layer) offrendo migliori caratteristiche di sicurezza. Questi protocolli utilizzando una cifratura asimmetrica per la sessione di dati e una cifratura asimmetrica per lo scambio delle chiavi che consiste nel cifrare i dati usando una chiave pubblica e decifrarli con la corrispondente chiave privata
FTPS, inoltre, utilizza dei **certificati digitali** per autenticare il server, e opzionalmente il client, garantendo un'affidabilità nella comunicazione. Questi certificati possono essere auto-autenticati per uso interno oppure possono essere emessi da una CA (Certification Authority) riconosciuta per uso pubblico.
Il protocollo FTPS utilizza la porta 21 per effettuare una connessione esplicita dove il client si collega direttamente e invia il comando "AUTH TLS" oppure "AUTH SSL" per iniziare una sessione cifrata e se il server accetta, si avvia la cifratura del canale. Esiste anche una connessione implicita tramite la porta 990 che è cifrata già dall'inizio dove il server pretende che il client inizi subito con una connessione di tipo TLS/SSL. In questa modalità si usa la porta 989 per lo scambio dati (anche questa porta è cifrata)

DHCP (porta 67 e 68)
Il Dynamic Host Configuration Protocol (DHCP) è un protocollo di rete utilizzato per assegnare dinamicamente indirizzi IP e altre informazioni di configurazione di rete ai dispositivi client in una rete automatizzando il processo di configurazione degli indirizzi IP e a volte dell'indirizzo di gateway.
Il suo funzionamento consiste nell'invio di un pacchetto, da parte del client che vorrebbe un indirizzo IP, di tipo ***DHCPDISCOVER*** contenente il proprio indirizzo MAC e altre informazioni di identificazione che, inviato in broadcast, verrà ricevuto da parte di un server DHCP che risponderà con un pacchetto di tipo ***DHCPOFFER*** contenente l'indirizzo IP offerto, la durata del **lease** e altre informazioni di configurazione.
Il **lease** è un periodo di tempo durante il quale un indirizzo IP assegnato a un dispositivo client è valido. Alla scadenza del lease, il client deve rinnovare il lease per continuare a utilizzare l'indirizzo IP.
Quando un client DHCP non ha più bisogno dell'indirizzo IP, può inviare un messaggio ***DHCPRELEASE*** al server DHCP per liberare l'indirizzo IP, rendendolo disponibile per altri dispositivi.
Dopo che il client ha ricevuto uno o più pacchetti di tipo DHCPOFFER da parte del server, invierà in broadcast un pacchetto di tipo DHCPREQUEST contenente l'indirizzo IP scelto e l'identificativo del server che ha fatto l'offerta per potergli notificare di aver accettato una determinata offerta e infine il server a sua volta risponderà con un DHCPACK per poter confermare al client di poter utilizzare quel determinato indirizzo IP.
Il server DHCP utilizza la porta 67 per ricevere richieste dai client e la porta 68 per inviare le risposte



HTTP (porta 80)
L'HyperText Transfer Protocol (HTTP) è un protocollo di rete utilizzato per la trasmissione di informazioni ipertestuali, principalmente pagine web, su internet. HTTP funziona come un protocollo di richiesta-risposta tra client e server, dove il client invia richieste per risorse e il server risponde con i dati richiesti.
HTTP è alla base delle applicazioni web che richiedono lo scambio di dati tra client e server utilizzando dei metodi:

GET
Viene utilizzato dal client per richiedere una risorsa specifica presente sul server che risponderà con i dati richiesti se disponibili. I dati richiesti sono inclusi nell'URL della richiesta.

```
GET /index.html HTTP/1.1
```

POST
Viene utilizzato per inviare dati al server per l'elaborazione. I dati inviati sono inclusi nel corpo della richiesta.

```
POST /submit-form HTTP/1.1
```

PUT
Viene utilizzato per aggiornare o creare una determinata risorsa sul server. I dati inviati sono inclusi nel corpo della richiesta.

```
PUT /users/123 HTTP/1.1
```

DELETE
Viene utilizzato per eliminare una risorsa specifica sul server.

```
DELETE /users/123 HTTP/1.1
```

HEAD
È simile al metodo GET ma richiede solo i metadati di una risorsa (header http) senza il corpo del messaggio, utile per controllare se una risorsa è disponibile o per verificare i metadati.

```
HEAD /index.html HTTP/1.1
```

OPTIONS
Viene utilizzato per richiedere le opzioni di comunicazione supportate dal server per una risorsa specifica. Il server risponderà con i metodi HTTP supportati e altre informazioni di configurazione. Utile per verificare le capacità del server.

```
OPTIONS /index.html HTTP/1.1
```

Ogni volta che viene inviato un metodo, o richiesta http, viene ricevuto dal client un codice di stato che sta ad indicare lo stato della richiesta:
1xx Informational:
- 100 Continue: Il server ha ricevuto la richiesta iniziale e il client può continuare con la richiesta.
- 101 Switching Protocols: Il server accetta di cambiare il protocollo come richiesto dal client.

2xx Success:
- 200 OK: La richiesta è stata completata con successo.
- 201 Created: La richiesta ha portato alla creazione di una nuova risorsa.
- 204 No Content: La richiesta è stata elaborata con successo, ma non ci sono contenuti da restituire.

3xx Redirection:
- 301 Moved Permanently: La risorsa richiesta è stata spostata in modo permanente a un nuovoURL.
- 302 Found: La risorsa richiesta è stata trovata, ma temporaneamente si trova in un URL diverso.
- 304 Not Modified: La risorsa non è stata modificata dall'ultima richiesta.

4xx Client Error:
- 400 Bad Request: La richiesta è malformata o contiene errori.
- 401 Unauthorized: L'autenticazione è richiesta e non è stata fornita o è fallita.
- 403 Forbidden: Il server ha capito la richiesta, ma rifiuta di autorizzarla.
- 404 Not Found: La risorsa richiesta non è stata trovata sul server.

5xx Server Error:
- 500 Internal Server Error: Il server ha incontrato un errore interno e non può completare la richiesta.
- 502 Bad Gateway: Il server, agendo come gateway o proxy, ha ricevuto una risposta invalida dal

server a monte.
- 503 Service Unavailable: Il server non è disponibile per gestire la richiesta, spesso a causa di manutenzione o sovraccarico.

HTTPS (porta 443)
L'HyperText Transfer Protocol Secure (HTTPS) è la versione sicura del protocollo HTTP l'aggiunta della "S" indica l'uso della crittografia SSL/TLS per proteggere i dati in transito.
Uno degli aspetti più importanti di HTTPS è la **confidenzialità**. Quando un utente accede a un sito web tramite HTTPS, tutti i dati che invia e riceve vengono cifrati, ovvero trasformati in un formato illeggibile a chiunque tenti di intercettarli.
Oltre a garantire la riservatezza delle informazioni, HTTPS assicura anche **l'integrità** dei dati. Questo significa che i messaggi inviati dal client al server (e viceversa) non possono essere modificati o corrotti durante il transito senza che tale alterazione venga rilevata.
Infine, una caratteristica essenziale di HTTPS è **l'autenticazione**, ovvero la capacità del client di verificare che il server con cui sta comunicando sia effettivamente chi dichiara di essere.
le comunicazioni con HTTPS si basano su TLS (Transport Layer Security):

1- **Client Hello**
   Il client invia un messaggio al server indicando le suite di cifratura supportate e un numero casuale (nonce).
2- **Server Hello**
   il server risponde con la cifratura scelta , un proprio numero casuale e il certificato digitale nel quale è contenuta la chiave pubblica del server
3- **Verifica del certificato**
   il client verifica che il certificato sia firmato da una CA (Certification Autority) affidabile
4- **Generazione della chiave**
   il client genera una chiave di sessione simmetrica e la cifra con la chiave pubblica del server, inviandola poi al server.
5- **Sessione Sicura**
   il server decifra la chiave di sessione simmetrica e da questo punto in avanti, tutte le comunicazioni sono cifrate utilizzando questa chiave di sessione

HTTP è il protocollo fondamentale per il funzionamento del World Wide Web, ma le sue comunicazioni in chiaro lo rendono vulnerabile a varie minacce di sicurezza. L'adozione di HTTPS e l'implementazione di pratiche di sicurezza adeguate, inclusa la gestione sicura dei cookies, sono essenziali per proteggere la trasmissione dei dati e garantire un'esperienza sicura per gli utenti.

NetBIOS (p.56)
Il Network Basic Input/Output System (Net-BIOS) non è un protocollo vero e proprio ma è un'API che permette alle applicazioni di comunicare tra computer su una rete locale.
NetBIOS fornisce tre servizi principali:

-Name Service (porta UDP 137) che consente la registrazione e la risoluzione dei nomi NetBIOS in indirizzi IP come un DNS ma in una rete LAN. Questa funzionalità è cruciale perché consente ai dispositivi Windows di trovare altri host usando nomi brevi (es. PC-UFFICIO) invece di indirizzi IP, anche se questa funzione che nei sistemi moderni è stata sostituita dai DNS.

-**Datagram Service** (porta UDP 138) che è responsabile della trasmissione di pacchetti di dati senza connessione, sfruttando principalmente la comunicazione di tipo broadcast. Questo servizio viene usato per notificare la presenza di un dispositivo in rete. Poiché non stabilisce una connessione tra mittente e destinatario, è un meccanismo veloce ma privo di garanzie: non assicura che il messaggio arrivi, né che venga ricevuto nell'ordine corretto. Viene quindi utilizzato per traffico di rete secondario, come annunci e notifiche, piuttosto che per trasferimenti critici di dati.

-**Session service** (porta TCP 139) che permette la creazione di sessioni orientate alla connessione che vengono utilizzate per la condivisione di file e di stampanti

Nonostante la sua importanza storica, NetBIOS oggi rappresenta un elemento di rischio per la sicurezza delle reti moderne. Le sue operazioni si basano spesso su meccanismi non cifrati e vulnerabili a tecniche di spoofing o man-in-the-middle. Inoltre, la risoluzione dei nomi NetBIOS è suscettibile a manipolazioni malevole in ambienti misti, ad esempio con strumenti come Responder che intercettano e rispondono falsamente alle richieste di nomi in rete locale, ottenendo così credenziali o sessioni non autorizzate.

Le API (Application Programming Interface) sono un insieme di definizioni, regole e protocolli che consentono a diversi software di accedere alle funzionalità o ai dati di un altro software, sistema operativo, libreria o servizio, senza dover conoscere i dettagli del suo funzionamento interno.
A livello tecnico, un'API agisce come un intermediario. Ad esempio, quando un'applicazione richiede l'accesso a un servizio esterno (come una mappa, un sistema di pagamento, o una base di dati), lo fa tramite un'API che espone determinate funzioni o endpoint che possono essere utilizzati in maniera standardizzata. Questo consente a sviluppatori diversi di interagire con lo stesso sistema senza doverlo riscrivere o modificare.

SMB (porta 445)
Il Server Message Block (SMB) è un protocollo implementato in una suite di software livera chiamata SAMBA. Questo protocollo è utilizzato principalmente per la condivisione di file e stampanti tra diversi sistemi operativi, tra cui Windows, Linux e MacOS permettendo ad un client di accedere a file remoti come se fossero locali.
Grazie a SMB, un client può accedere a file remoti come se fossero locali, semplificando l'uso delle risorse condivise in reti miste. Le versioni iniziali si basavano su NetBIOS (porta TCP 139), ma con l'evoluzione del protocollo è stato introdotto SMB over TCP (porta 445), che ha migliorato efficienza e sicurezza. Dal punto di vista della cybersecurity, SMB rappresenta un punto debole, in particolare con le versioni più datate come SMB1, che sono vulnerabili ad attacchi gravi come EternalBlue.

IMAP (porta 143)
L'Internet Message Access Protocol è un protocollo utilizzato dai client di posta elettronica per accedere ai messaggi su un server di posta elettronica senza il bisogno di scaricare i messaggi in locale. Poiché i messaggi rimangono sul server, è essenziale implementare misure di sicurezza adeguate sul server di posta per prevenire accessi non autorizzati.

POP (porta 110)
Il Post Office Protocol è un protocollo utilizzato dai client di posta elettronica per recuperare i messaggi da un server di posta elettronica. POP è progettato per scaricare i messaggi sul dispositivo locale e, per impostazione predefinita, rimuoverli dal server. Questo consente agli utenti di accedere ai propri messaggi anche senza una connessione Internet attiva. Una volta scaricati, i messaggi risiedono localmente, quindi la sicurezza dipende anche dalla protezione del dispositivo locale.

SMTP (porta 25)
Il Simple Mail Transfer Protocol è un protocollo standard per l'invio di e-mail attraverso le reti IP.
SMTP è responsabile della trasmissione delle e-mail dai client di posta ai server di posta e tra server di posta. Funziona principalmente sulla porta TCP 25, ma può utilizzare anche la porta 587 per connessioni sicure.
Il funzionamento di SMTP si basa su un meccanismo di comunicazione tra client di posta (ad esempio, un'applicazione di posta come Outlook o Gmail) e un server SMTP.
Quindi:
-SMTP è usato solo per invio di e-mail, mentre IMAP e POP3 sono utilizzati per ricevere e-mail.

-IMAP mantiene le e-mail sul server, facilitando l'accesso da più dispositivi, mentre POP3 scarica le e-mail e le rimuove dal server (a meno che non sia configurato diversamente).

-IMAP è ideale per chi vuole accedere alle e-mail da più dispositivi, mentre POP3 è più adatto per un uso su un singolo dispositivo.

SSH (porta 22)
Il protocollo Secure SHell è un protocollo di rete utilizzato per connettersi in modo sicuro a un sistema remoto, come un server o una macchina virtuale, attraverso una rete non sicura (ad esempio, Internet).
È ampiamente utilizzato per l'amministrazione remota di server, la gestione di dispositivi di rete e per eseguire comandi e trasferire file in modo sicuro. SSH è diventato lo standard per la gestione di sistemi Unix-like (Linux, MacOS, ecc.), ma può essere utilizzato anche su Windows.
Quando un client si connette a un server SSH, avviene una negoziazione per stabilire una connessione sicura. Questo processo include l'uso di chiavi pubbliche e private per l'autenticazione e la crittografia del traffico per proteggere i dati durante il trasferimento.

Autenticazione a chiave pubblica: Il client genera una coppia di chiavi (una privata e una pubblica). La chiave pubblica viene copiata sul server mentre la chiave privata resta sul client. Quando si tenta di connettersi, il server invia una sfida che può essere decifrata solo dal client che possiede la chiave privata corretta. Una sfida (challenge) è un messaggio criptato con una determinata chiave che, se il client ha, e quindi riesce a decifrare, avviene la connessione, altrimenti no.

Autenticazione tramite password: Sebbene l'autenticazione tramite chiave sia più sicura, SSH supporta anche l'autenticazione tramite una password. Tuttavia, questa modalità è considerata meno sicura rispetto all'uso delle chiavi.

SSH supporta anche il **port forwarding**, che consente di instradare il traffico di rete attraverso la connessione sicura, rendendo possibile l'accesso a risorse interne di una rete privata in modo protetto. Inoltre, con **SFTP** (Secure File Transfer Protocol), SSH permette il trasferimento sicuro di file tra client e server, proteggendo i dati durante il trasferimento.
Un'altra funzionalità potente di SSH è il **tunneling**, che consente di creare canali sicuri per accedere a servizi remoti o risorse interne, come se fossero sulla rete locale. Infine, SSH supporta **l'X11 Forwarding**, che permette di eseguire applicazioni grafiche su un server remoto, ma di visualizzarle sul client, utile per l'accesso a desktop remoti in modo sicuro.
Queste funzionalità rendono SSH un protocollo versatile, essenziale per la gestione sicura e remota di sistemi e risorse in una rete.
SNMP (porte 161 e 162)

Il Simple Network Management Protocol è un protocollo standard utilizzato per la gestione e il monitoraggio dei dispositivi di rete. SNMP consente agli amministratori di rete di raccogliere informazioni e configurare dispositivi di rete come router, switch, server, stampanti raccogliendo dati su parametri come utilizzo della CPU, memoria, larghezza di banda e stato delle interfacce di rete per rilevare guasti e anomalie.
Grazie a SNMP è possibile modificare le impostazioni dei dispositivi interessati da remoto utilizzando un modello Client-Server, dove i dispositivi di rete (agenti SNMP) sono i server e il sistema di gestione di rete (NMS, Network Management System) è il client. Per il trasporto, viene utilizzato UDP (User Datagram Protocol) come protocollo, con le porte 161 per le richieste e le porte 162 per le notifiche (trap).
L'NMS, per ottenere informazioni specifiche sull'agente SNMP, utilizza il metodo GET, per modificare un valore di configurazione, utilizza il metodo SET e l'agente SNMP invia notifiche non richieste (trap) all'NMS per segnalare eventi significativi e anomalie.
La sicurezza di SNMP deve essere una priorità, utilizzando SNMPv3 con autenticazione e crittografia per proteggere le comunicazioni di rete e prevenire accessi non autorizzati. Implementare politiche di sicurezza adeguate è fondamentale per garantire l'integrità e la disponibilità delle reti gestite tramite SNMP.