Programmazione di Rete

- Orientata al **Datagram**
- Orientata alla **Connessione**

Client-Server

- Client sempre prima mossa
- Sono 2 o + processi
- ① Si crea socket
- ② Trasmissione & Ricezione
- Bisogna sapere porta del server

Web-Socket

- Protocollo lvl. App. fornisce un **Canale Bidirezionale**
- Utile per **App. REAL-TIME**, srv fa il PUSH
- HTTP **non** è BIDIREZIONALE (client fa REQUEST)
- **Upgrade di Protocollo** può essere fatto solo dal Client e srv accetta o meno la proposta

Approccio Asincrono

- Event-Driven
  - Sincroni
  - Asincroni
- Javascript per programmare Browser (FRONTEND)

Comandi di Rete

- Ping → Raggiungibilità e RTT
- TraceRoute → Percorso del Pacchetto
- NSLookUp → interrogazione al DNS (hostname → IP)
- IFConfig → Interfacce
- Route → Tab. Routing
- WhoIs → a chi è intestato l'IP/dominio DNS

HTTP/HTTPS

- Si Basa su TCP → Affidabile ma NON Sicuro
- HTTPS se c'è TLS/SSL → Connessione CRITTOGRAFATA
- RICHIESTA — Linea Richiesta, intestazione, corpo
- RISPOSTA — Linea Stato, intestazione, corpo
- DOM → rappresentazione Gerarchica della STRUTTURA
- URL → identifica una RISORSA HTTP

CERTIFICATO DIGITALE

- file che contiene INFO di chi Vuole Essere certificato
- contiene Chiave pubblica di chi Vuole Essere certificato
- Si può Usare sia per CLIENT che Server
- Chiave privata La Tengo io e NON c'è dentro

CGI

- Standard che consente ai WEB-SERVER di Generare Contenuti dinamici ed Interattivi ⇒ NASCITA DEL WEB DINAMICO
- Web è una porta di Servizi Remoti

SOA

- Suddivisione programma in SERVIZI Autonomi e Riusabili
- Potenza & Calcolo delegati al SERVER
- Protezione Proprietà Intellettuale
- Modello Economico PAY-PER-USE
- No Pirateria
- No distribuzione Aggiornamenti
- Parametri in RETE vengono CODIFICATI/SERIALIZZATI
  - RPC
  - Java RMI
  - CORBA
  - WEB-SERVICES

Sfruttano HTTP come Protocollo di trasporto

Per Elem. della Funzione

WEB-SERVICES

HTTP ↑ ESPONGO DEI SERVIZI

- fz Mappate su URL
- Passo Parametri
  - Sull'URL → get o delete
  - Nell'HEADER → post o put
- Ret nel BODY
  - formato JSON
- Utilizzato Perchè
  - Infrastruttura predisposta
  - Trapassa NAT e FIREWALL

CLOUD-COMPUTING

- ON-SITE ⇒ Proprietario di TUTTO
- IAAS ⇒ Ho una Macchina Virtuale
- PAAS ⇒ Gestisco L'Ambiente (DOCKER,...)
- SAAS ⇒ Solo uso Applicativo
- FAAS ⇒ Chiamata a Funzione Specifica

PUB/SUB

- Molto SCALABILE (utile Per IoT) e DATA CENTRIC
- Architettura disaccoppiata → componenti NON si Conoscono
- Ogni dato ha un etichetta (TOPIC) che Esprime La SEMANTICA
- BROKER ha tutte le responsabilità "DA SERVER"
- No PEER-TO-PEER ma molti a molti

MQTT

- QoS 0 ⇒ at Most One ⇒ Posso Perderlo
- QoS 1 ⇒ at Least One ⇒ Può Arrivare duplicato
- QoS 2 ⇒ Exactly One ⇒ Ancora più ACK
- TOPIC Gerarchici:
  - lvl1/lvl2/#
  - +/lvl2
- BROKER può avere HW e S.O ≠ da PUB/SUB e può Esserlo anche il linguaggio di Programmazione

Aggiungo Complessità e Pacchetti in RETE ↓

PROPRIETÀ SICUREZZA

- CONFIDENZIALITÀ ⇒ riservatezza
- INTEGRITÀ ⇒ Impedice Alterazione delle INFO
- DISPONIBILITÀ ⇒ Prestazioni & Robustezza Garantite
- AUTENTICITÀ ⇒ INFO è Stata MANIPOLATA?
- TRACCIABILITÀ ⇒ loggare ciò che Succede

CRITTOGRAFIA

- Rendere l'INFO incomprensibile a chi NON è AUTORIZZATO
- Algoritmo per CIFRARE e DECIFRARE (simmetrico o asimmetrico)
- Computazionalmente Sicuro se NON-ECONOMICO

CRITTOGRAF. SIMMETRICA

- Chiave UNICA scambiata attraverso un canale Sicuro
- CIFRARIO DI CESARE ⇒ shift di K a ogni SIMBOLO
- CIFRATURA MONOALFABETICA ⇒ Permutazione (analisi freq. è RISCHIO)
- CIFRATURA A BLOCCO ⇒ divido in gruppi di K-BIT e cifro il Blocco
  - DES → Data Encryption Standard
  - TRIPLO-DES → 3×DES con CHIAVI diverse
  - AES → è SOTA, si Allunga Sempre di più KEY

CRITTOGRAFIA ASIMMETRICA

- CHIAVE PUBBLICA ⇒ Nota a TUTTI
- CHIAVE PRIVATA ⇒ devo Tenerla Segreta
- dato CIFRATO con PUBLIC-KEY del destinatario che poi la DE-CRYPT con la sua PRIVATE-KEY
- Non c'è NESSUNA ENTITÀ TERZA e NON devo avere chiavi ≠ per ogni COMUNICAZIONE ma sono ONEROSI COMPUTAZ.

FIRMA DIGITALE
- Combino CIFRATURA + fz. HASH
- Invio msg. in chiaro + versione Hashata & Cifrata con La mia PRIVATE-KEY
- fz. hash NON deve essere INVERTIBILE e minimizzare il #Collisioni

CERTIFICATION AUTORITY — Convalida IDENTITÀ ed Emette CERTIFICATI

AUTENTICAZIONE
- LOCALE ⇒ accede il Loco al Servizio che effettua autenticazione
- REMOTA DIRETTA ⇒ accedo REMOTE a chi mi autentifica
- REMOTA INDIRETTA ⇒ accedo a ≠ servizi con servizio di AUTENT. SEPARATO
- OFF-LINE ⇒ Servizi autonomi senza dover contattare C.A.
- OTP ⇒ come SMS, codici

AUTORIZZAZIONE — Garantisce accesso Alle Risorse SOLO agli UTENTI che ne hanno il DIRITTO

FIREWALL & INTRUSION DETECTION SYSTEM — Rilevano & Bloccano Minacce