

WIRESHARK:

Diagnistica / Analizzo Reti Analizzando PDU (Protocol Datagram Unit)

Imbustamento Multipla mi permette di Simulare un dialogo tra 2 Entità Allo Stesso livello in ORIZZONTALE, anche se in REALTÀ comunicano tutti a livello fisico

PROTOCOLLO \Rightarrow Convenzione / Significato che si dà ai BIT, Quindi la SEMANTICA

Ogni PDU ha:

■ HEADER \Rightarrow Come minimo Indirizzo Mittente & Destinazione

■ PAYLOAD \Rightarrow

Un dispositivo Solitamente ha K-Interfacce, ma Come Minimo c'è la LOOPBACK che è un Interfaccia di Rete Virtuale con me stesso per Comunicare Come Se fossi in RETE ma in Realtà sono Sullo Stesso PC

ETHERNET :

PREAMBULO \Rightarrow Per SINCRONIZZARE il Ricevitore, e Subito dopo trasmette L'INDIRIZZO di destinazione, così Se Qualcuno è in Ascolto e msg non è per lui Allora smette di Ascoltare

ARP \Rightarrow Address Resolution Protocol, da Indirizzo MAC ad IP

CONNESSIONE HTTP:

Si Basa Sul **TCP**, che mi Assicura **AFFIDABILITÀ** e **NON SICUREZZA**

DNS (protocollo Lvl Applicazione) è Basato su UDP

FTP \Rightarrow File Transfer Protocol mette si ** Quando digito la PSW ma **NON** è un protocollo Cifrato e Quindi **NON SICURO**

PROTOCOLLO HTTP:

- ② Apertura Connessione Tramite **3-Way-Handshake**
- ③ Se con HTTPS allora Autenticazione del Server & Negoziazione di una Chiave di Cifratura.
- ③ Richieste (DEL CLIENT) e Risposte (DEL SERVER)
- ④ Chiusura della Connessione TCP.

HTTPS Lavora Sulla Porta **443** invece che 80 di HTTP

TCP vs UDP:

■ Quando msg **NON** Sta tutto in un Solo PACCHETTO a LVL IP

■ Per Affidabilità

■ Mi Servono TUTTI i SINGOLI MESSAGGI

INVIO FILE VIDEO



NON tollera Nessuna Perdita

anche Qui Vo lig l'Affidabilità
ma Posso Tollercare di accorgermi della Perdita
e rieffettuare la RICHIESTA

■ Uso Real-Time

STREAMING VIDEO



Se perdo Qualcosa Continuo

DA WEB a WEBSERVICES:

HTTP Nato per soddisfare le richieste che un CLIENT effettua ad un SERVER di una DETERMINATA Risorsa

Ci possono Essere problemi di privacy in:

REQUEST \Rightarrow Cookie Nei Browser

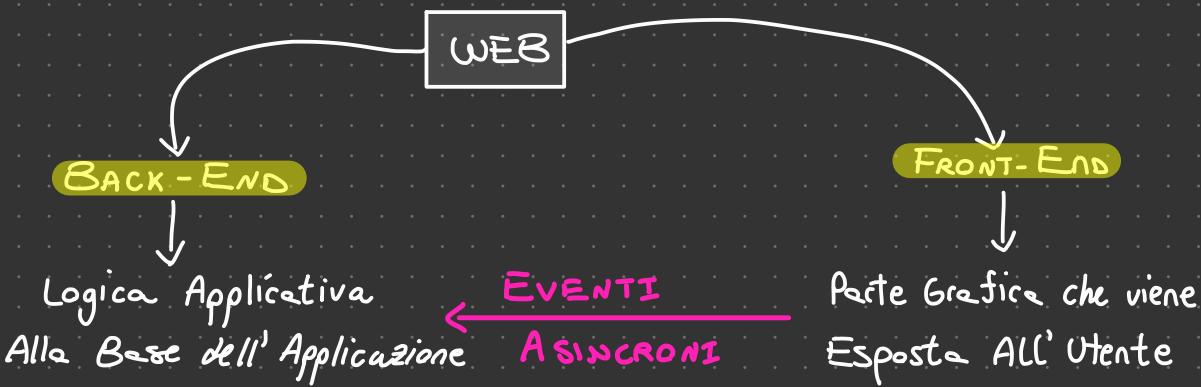
RESPONSE \Rightarrow Quando Consulto i Voti

Devo Riuscire ad AUTENTICARE il Server, il Problema è che io Passo attraverso DNS per RISOLVERE correlazione Nome \leftrightarrow IP

WEB è un Applicazione su INTERNET.

JAVASCRIPT:

Linguaggio Interpretato per Programmare il BROWSER (lato FRONT-END)



Non Sono Prevedibili / Non So Quando Arrivano ed infatti Si Gestiscono Attraverso una Programmazione di TIPO EVENT-DRIVEN e NON BATCH Classica.

JAVA SCRIPT mi Permette proprio di: } Essendo che Gira Sul CLIENT
 ■ Gestire Eventi:
 ■ Modificare La Página } non devo Sovraccaricarlo

INTERFACCIA SOCKET:

Assembly della Programmazione di Rete

Per un Applicazione di Rete Solitamente Sono almeno 2 i processi che dialogano (secondo diversi PARADIGMI)

PARADIGMA CLIENT-SERVER:

ha a che fare con la GUI ed il Front-End (fa Richieste)

processo che Risponde Alle Richieste Ricevute

UDP: Invio di INFO Semplici dove ogni pacchetto è Indipendente e le perdite non Sono Tenute in Considerazione

TCP: Dati più Pesanti, Tra pacchetti c'è una Relazione e fanno Parte di un Messaggio più Grande.

Se ho una perdita procedo al RINVIO

PRO

■ Stessa Naturalezza Scrittura/Lettura come se La Rete non ci fosse

CONTRO

■ Più Ritardo
■ Trasmettitore e Ricevitore lavorano con il S.O.

Posso farlo con Entrambi... Questo Paradigma mi descrive Solo come Avviene l'Interazione Tra i 2 Soggetti:

- ① Client fa Sempre il Primo Passo con una Richiesta
- ② Server Risponde alla Richiesta

Client può RICHIEDERE il DATO ma Anche fornirlo

Host identifica con IP mentre il Processo con IP+PORTA

EVENTO SINCRONO: È prevedibile, ad Esempio SET-INTERVAL (n) per l'Allarme perché lo Imposto io.

Questo è un Esempio di POLLING, da EVITARE il più possibile

HTTP NON permette che il SERVER possa Mandare in Maniera asincrona INFO al Client ed è per questo che è Nato il Protocollo di livello Applicativo Web-Socket

URL

Identificare una Risorsa a livello di Rete, quindi Ad Ogni livello del Modello ISO/OSI ha un Concetto di Indirizzo:

- URL - Lvl 7 - Livello Applicazione
- Porte - Lvl 4 - Livello di Trasporto
- IP - Lvl 3 - Livello Rete
- MAC - Lvl 2 - Livello Collegamento Dati

Non Necessariamente INDIRIZZO/PERCORSO LOGICO è Quello Fisico Sul SERVER (anche se rimane un filesystem Gerarchico)

INTRODUZIONE CYBERSECURITY:

Cosa Voglio Proteggere? Che **Natura** ha?

AFFIDABILITÀ

Eventi Naturali
oppure **ERRORE UMANI**

TCP

fault/Failure

Fulmini/Alluvioni

SICUREZZA

Minaccia

Malizia, Dolo
umano

Dos/Trojan/...

TLS/SSL

PROTEGGERE:

Proprietà:

CONFIDENZIALITÀ → Segretezza / Riservatezza

INTEGRITÀ → No Alterazioni (TCP NON Basta)

DISPONIBILITÀ → Interruzione del Servizio

AUTENTICITÀ → Riuscire ad identificare correttamente chi fa cosa

TRACCIABILITÀ → Loggare TUTTO Quello che viene fatto

PRIVACY ≠ NON DIRE LE COSE (Segretezza)

MINACCIE & ATTACCHI:

Possibile violazione della Sicurezza

Violazione Effettiva

- **INTERNO:** iniziati da entità interna

- **ATTIVO:** alterare risorse / funzionamento

- **ESTERNO:** iniziati da entità esterna

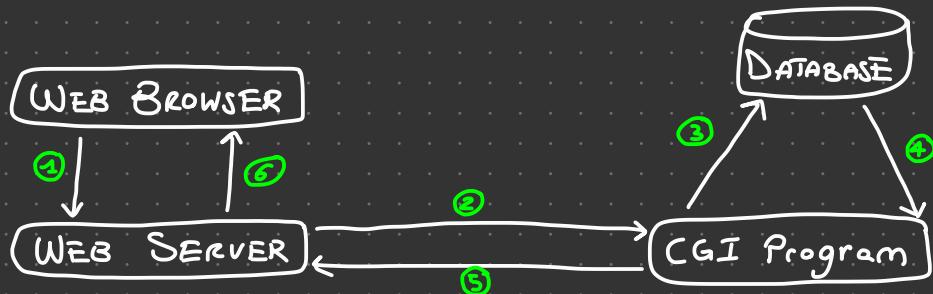
- **PASSIVO:** carpire INFORMAZIONE

Bisogna Sempre utilizzare un Appraccio MODULARE (che mi riduce la complessità)

La Sicurezza dipende anche degli UTENTI, da dove spesso nascono le Vulnerabilità (SMART WORKING - USB - DISTRIBUZIONE INFO)

Sicurezza NON ha NULLA a che vedere con la SEGRETEZZA DEI METODI (algoritmo OPEN è "più Sicuro")

CGI: (Common Gateway Interface)



- ① HTTP Request
- ② Chiamata All'Applicazione in BACK-END
- ③ Interroga il DATABASE
- ④ Risultato della Query
- ⑤ Standard Output (HTML che è stato Generato)
 - ↳ NON Presente Sulla macchina SRV Salvato come FILE
- ⑥ HTTP Response

PERCHÈ DEVE ESSERE STATICA? Posso Generarla al Volo, Nascita del WEB Dinamico ⇒ Elaborazioni Su SERVER per Generare la PAGINA HTML

Così facendo non ha Problemi di firewall perché Sfrutta il fatto che il Traffico HTTP è Sempre Lecito.

Tramite CGI Server mi Crea al Volo La pagina Senza dover averla Salvata (SAREBBE IMPOSSIBILE SALVARNE PER TUTTI)

WEB SERVICES:

Non centra nulla con il WEB BROWSER

Invoco un SERVIZIO

Ad Esempio APP per Smartphone che hanno funzionalità che Vengono Invocate e Risiedono Sul SERVER.

Non più Programmazione Monolitica, ma:

SERVICE ORIENTED ARCHITECTURE (SOA):



Ho un processo che gira vicino all' Utente e poi le funzioni più complesse sono su un SERVER Remoto (introduco un RITARDO)

Ci sono vantaggi come:

- Memoria, calcolo delegato al SERVER
- Protezione della PROPRIETÀ INTELLETTUALE (Solo io ho l'Algoritmo)
- Non devo distribuire AGGIORNAMENTI SOFTWARE \Rightarrow Solo GUI
- Modello Economico PAY PER USE
- Eliminazione Pirateria \Rightarrow Per usare Bisogna Pagare SEMPRE

REQUISITI: Infrastruttura di RETE è Strettamente Necessaria altrimenti ho solo l'INTERFACCIA GUI.

Devo prestare Attenzione All' INTERFACCIA

Sia STUB che CALLED OBJECT hanno stessa Interfaccia
Sulle Macchine Locali Sul SERVER

IL Caller è convinto di fare una Chiamata LOCALE ma in realtà codifica Parametri e Chiama una Procedure REMOTA

■ RPC → in C su TCP

■ Java RMI → su TCP

■ CORBA → Standard INDEPENDENTE dal Linguaggio

■ WEB SERVICE

HTTP/HTTPS come l'Applicativo

Sfutta HTTP

Chiamata a Servizi su SERVER

WEBSERVICE BASATI SU REST:

url mi Codifica le Procedure REMOTA che chiama.

Parametri dipende dal METODO DELLA RICHIESTA

Nell' URL <= GET <

Post => Nel Body della Richiesta

Grazie ai WEB SERVICES il CLIENT può Essere anche un WEB BROWSER.

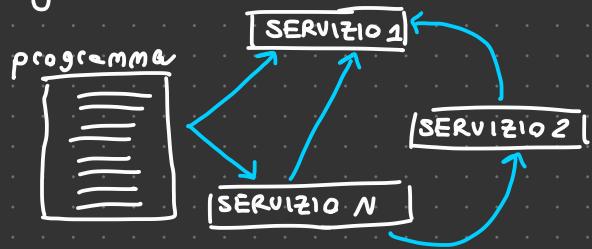
→ È un modo di Programmare Applicazioni Distribuite.

SERIALIZZAZIONE → Quando Passo parametri Sulla RETE, fatto Semplicemente Tramite JSON.

Sul BACK-END ho una 'LIBRERIA', ovvero Molte Funzioni che vengono Esposte

ARCHITETTURA SOA a MICROSERVIZI:

Non più Applicazioni Monolitiche con Operazioni Sequentiali ma Applicazioni Strutturate come Segue:



Tanti SERVIZI più PICCOLI /LEGGERI possibile :

- Chiamando Stessa Funzione con Parametri diversi
- Riscrivere la funzione

Così facendo :

- Sono più Scalabile
- facilita il LOAD BALANCING

La MODULARIZZAZIONE può degenerare se i VARI Moduli si parlano TROPPO allora Andrebbe messo NELLO Stesso Modulo

ARCHITETTURA ORIENTATO AI SERVIZI mi permette di Mescolare i Linguaggi perché ho la Rete in mezzo che mi Serializza il tutto.

Bisogna stare attenti sui TIPI BASE da un Lingueggio all'altro

LOAD BALANCING:

In fase di progettazione del BACK-END andrebbero fatti dei MICRO-SERVIZI

Posso farlo attraverso:

■ MACCHINA VIRTUALE \Rightarrow S.O./architetture diverse, che è MOLTO Pesante e non utile nei DATACENTER

■ CONTAINERIZZAZIONE \Rightarrow Virtualizzazione più leggera, mantiengo lo stesso S.O. Senza duplicazione dell'hardware.

È come se avessi un AMBIENTE DUPLICATO, una scatola a parte.

■ UN UNICO IP che distribuisce il Carico ad Altre macchine (ESPONGO IP DAL LOAD BALANCER)

■ tramite DNS posso fornire IP \neq allo stesso URL...ma DNS ha la CACHE Quindi NON funziona Troppo Bene

I micro-servizi devono essere STATELESS, ovvero non devono avere dipendenze l'un l'altro di DATI altrimenti si devono aspettare a vicenda.

CLOUD - COMPUTING \Rightarrow Novità Economica dove chi Possiede le macchine non è chi Possiede il Software che gira sopra

CLOUD:

- ON-SITE \Rightarrow Gestisco tutto io (da HW a SW)
- IAAS \Rightarrow Infrastructure as A Service, affitto una virtual machine su cui Posso fare Quello che Voglio.
- PaaS \Rightarrow Platform As A Service, affitto un Ambiente con già un S.O. e ci Metto i miei Container (ex. DOCKER)
- SaaS \Rightarrow Software As a Service, affittano l'uso di Applicazioni già Installate

Quale costa di più? chiaramente IAAS perchè ti do più libertà di manovra su Quello che puoi fare

- FaaS \Rightarrow function As A Service, espongo dei microservizi che faccio pagare per ogni volta che viene Invocata
Anche conosciuto come SERVERLESS COMPUTING (anche se il server c'è e come)

WEB SOCKET:

RECAP PRIMA:

UNICO EVENTO SINCRONO è l'ALARM perché imposto io un valore

EVENTI ASINCRONI spesso Sono d'Interfaccia Utente Perché NON so Quando Capita

Javascript per Lete CLIENT, ma Per Eseguire Codice Backend entro in gioco le CGI e WEB SERVICES

Per WEB Classico

Mandarmi una Página

Web Costruite dinamicamente

Utilizzo del Web come Protocollo di Lvl Trasporto per chiamata a funzione

Costruire Applicazioni distribuite

È un Protocollo di Lvl Applicativo che fornisce un canale di Comunicazione Bidirezionale Simmetrico attraverso una singola Connessione TCP che Da un certo punto in poi si Smette di parlare HTTP e si Passa a WEB-SOCKET.

La Connessione TCP Rimane (non cade), ma Cambie Solo il modo in cui la uso visto che passo da un MODELLO CLIENT-SERVER (http) a un MODELLO PEER-TO-PEER (web Socket).

Quindi non devo più fare il POLLING ma sarà il BACK-END che mi fa il PUSH (ad esempio una NUOVA NOTIZIA)

HTTP non è SIMMETRICO, c'è un client e un Server ed il Server non dà Risposte se non c'è un Client che lo INTERROGA

Web-Socket è in Composizione ad HTTP. Nel caso in cui usiamo il BROWSER, altrimenti se voglia un APP sfrutta solo la CONNESSIONE TCP.

LIMITAZIONI:

NAT non ha CACHE ∞ e Quindi non sa Quale CONNESSIONE è Aperta e Quale è chiusa e per Questo devo continuare a trasmettere dei PACCHETTI di KEEP-ALIVE per tenere la CONNESSIONE viva Nella CACHE del NAT.

WEB-SOCKET, se prima Avevo HTTPS, Quindi  diventa Web-Socket sicuro perchè Sotto rimane tutto invertito

PUB-SUB:

È un **PARADIGMA** diverso Rispetto Al **CLIENT-SERVER**.

→ A Lvl Applicativo

Sono io che per ordine metto delle
Regole d'Interazione

È importante l'ordine d'Interazione ⇒ **Client sempre per primo**

C: Sono 3 entità:

- Un SOLO **Broker** ⇒ diviso per **TOPIC**
- Più **Publisher** ⇒ **Pubblica** un dato sul Broker
- Più **Subscriber** ⇒ **Colui che si Abbona** (prima interazione)
e Quando ci sarà un Nuovo dato con Quel TOPIC
Allora Broker farà una **NOTIFY** ai SUBSCRIBER

IL DATO è Sempre Con Etichette **TOPIC** che de le
Semantica del DATO

PUBLISHER & SUBSCRIBER Non si Conoscono

Da CLIENT-SERVER posso Sempre Ricondursi a PUB-SUB con le Seguenti Osservazioni:

- Server deve Essere Sempre Accesso e Raggiungibile
- Server deve avere IP Pubblico e Quindi è una **vulnerabilità** essendo che è in Ascolto.

Tutti gli Oneri del SERVER Vanno sul BROKER ma sia i PUBLISHER che SUBSCRIBER Sono dei CLIENT e Quindi ho Spostato la COMPLESSITÀ Solo sul BROKER

→ che può Essere UNICO con molti TOPIC al Suo Interno.

Client-Server è molto più difficile Modificare & Aggiungere un dispositivo, mentre con PUB-SUB Basta Aggiungere un SUBSCRIBER

PUB-SUB è MOLTI a MOLTI

PUB-SUB è DATA-CENTRIC visto che devo Conoscere Solo il BROKER mentre Client-Server è HOST-CENTRIC visto che devo conoscere tutti gli Attori in gioco

PROTOCOLLI:

- MQTT
- AMQP
- Kafka

IMPLEMENTAZIONI:

- Mosquitto / Paho
- Rabbit MQ
- Apache Kafka

CERTIFICATO DIGITALE Serve ad Autenticare il BROKER, perché il BROKER riconosce subscriber da USERNAME e PSW.

→ in MQTT gestisce da Solo la CONCORRENZA ma i TOPIC sono Senza diritti

AFFIDABILITÀ DI MQTT:

→ Qualità Del Servizio = Affidabilità → Aggiungo Complessità perchè ho 3 Entità

■ QoS Lvl 0 ⇒ At Most One, Quindi il

dato arriva al Massimo una Volta... non viene duplicato e può non arrivare

■ QoS Lvl 1 ⇒ At Least One, Quindi

arriva Sicuramente una Volta ma può arrivare anche un duplicato e devo Tenerne Conto (AD ESEMPIO CREANDO UN TIME STAMP NEL DATO).

Ancora più Transazioni di Messaggi.

■ QoS Lvl 2 ⇒ Exactly One, arriva ESATTAMENTE una Volta ed aggiunge molte Nuovi Scambi di Messaggi (il più costoso)

Viaggiano Quindi nuovi MESSAGGI nel TCP che sono Stati introdotti da chi ha Sviluppato MQTT per Garantire Affidabilità

END TO END

→ Visto che Per ogni Singola Transazione c'è già TCP

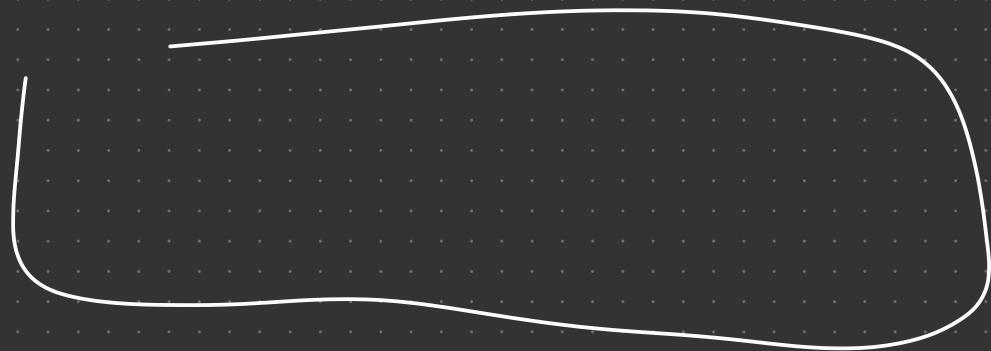
MQTT con Sotto TSL/SSL diventa Sicuro su una Singola Tratta ma non end To End.

Ciò che garantisce Tra 2 Entità non è portabile su 3 ed una possibile Soluzione è cifrare il dato così Se Qualcuno

Attacco il BROKER trova un dato Incomprensibile

STRUTTURA DEI TOPIC:

Posso Costruire un Albero di TOPIC per fare delle SUBSCRIBE su dei Sotto-Topic



Esempio della chat a differenza dei Web Socket devo Implementarmi tutta la Grafica (con WEB SOCKET faceva tutto il BROWSER) e far Scaricare l'Applicativo a chi lo Vuole Usare

CONCETTI BASE CRIPTOGRAFIA:

Ricordarsi i 5 Aspetti che Riguardano la SICUREZZA.

Scienza che si Occupa di Proteggere l'Informazione Rendendola Sicura per non renderla comprensibile a chi non è autorizzato
CrittoAnalisi è il Contrario, provare a decifrare l'informazione.

Brutta Pratica è Quella di Nascondere l'Algoritmo, perché Quello che deve Rimanere Segreto è la Chiave e non l'Algoritmo.

Tramite CIFRATURA rendo incomprensibile il **'TESTO'**

Che può Essere qualsiasi cosa

Da un testo ad un Immagine

Chiave è l'Insieme dei Parametri che Specializza l'Algoritmo

Con chiavi di DECIFRATURA e CIFRATURA uguali ho Algoritmo che è Simmetrico e le chiave deve essere SEGRETA.

Posso Avere 2 Chiavi, legate Matematicamente Alla Nascita, una pubblica ed una Private se ho Algoritmo ASIMMETRICO
Quindi con chiavi DIVERSE → deve RIMANERE SEGRETA

Si dice che è COMPUTAZIONALMENTE SICURA se:

- Costo Violazione è Superiore al Valore dell'Informazione
- Tempo Necessario a Violarlo è Superiore al Tempo di Vita INFO

Cambio Quindi le chiavi di Sessione in Tempi Inferiori a Quelli Necessari alla Computazione di DECIFRATURA

BRUTE FORCE \Rightarrow Costo $2^{\# \text{numero_di_Bit}}$ e Per Questo si Sta Allungando sempre di più la chiave

CIFRATURA SIMMETRICA è preferibile a Lvl computazionale con il problema del Come mi Scambio la Chiave (VISTO CHE È UGUALE)

CIFRATURA SIMMETRICA :

CIFRARIO DI CESARE: fare una SHIFT di K Posizioni ogni Singolo Carattere, dove K è la Chiave.

Cifratura di Ogni Singolo Blocco (BYTE), dove Posso Avere 256 Possibili Simboli (E NON CARATTERI)

Posso Cifrare anche delle Immagini.

Chiavi Possibili Sono 255 ovvero #Simboli - 1 $\Rightarrow 2^8 - 1$

Con 255 Chiavi al giorno d'oggi è molto Semplice fare 255 prove diverse.

Alfabeto sono tutti i Possibili Simboli che Posso Generare con un'unità Base di BYTE.

CIFRATURA MONOALFABETICA: Ogni Carattere viene Sostituito da un altro (PERMUTAZIONE) e Quindi diventa (#Chiave)!. Così facendo ho aumentato di Molto i CASI POSSIBILI

BRUTE-FORCE è l'UPPER BOUND \Rightarrow Termina Sempre dopo Aver provato tutti i casi, ma esistono anche modi più furbi d'Attacco.

CIFRATURA A BLOCCHI: Cambia il Mio Blocco, Quindi Invece di Lavorare su 1 Byte **Lavoro Su k-Byte** e così facendo vado ad aumentare le Possibili Permutazioni.

Dati K-Bit; Possibili 2^K ingressi Vengono Permutati

Se non Sono Multipli di BYTE allora devo Completare il multiplo con Numeri Casuali (**CHE POI DEVO TOGLIERE QUANDO DECIFRO**)

AES è SOTA

CHIAVE PUBBLICA e PRIVATA:

Sono formalmente uguali (sceglio io quale condividere)

Quello che cifro con una posso decifrarla solo con l'altra

Se qualcuno vuole inviarmi un messaggio CIFRATO allora dovrà Cifrarlo con la mia Chiave Pubblica

In questo caso non mi serve Entità Terza visto che c'è uno scambio diretto delle chiavi Pubbliche.

L'unico dubbio che sale è... Siamo sicuri che Bob è Bob? Come Verifico l'Identità? Ci si baserà su un Entità Terza che è Chiamata CERTIFICATE AUTHORITY.

Catena della fiducia è un Esempio di Possibile Soluzione che viene Utilizzata per i Passaporti/Spid/...

Se più persone parlano con Bob verrà usata Sempre e solo la CHIAVE PUBBLICA di Bob.

CIFRATURA ASIMMETRICA: Integrità e Autenticazione

INTEGRITÀ:

firmare un **Messaggio** per renderlo non ripudiabile.

Si Sfrutta RSA in Modo Inverso rispetto a quanto fatto per cifrare:

-
-

firmare è ONEROso (per via della CIFRATURA ASIMMETRICA)

FUNZIONE DI HASH:

Sono f.z.:

- **COERENTI** \Rightarrow Input uguali Output uguali
- **STRAVOLGERE STATISTICHE DEI SIMBOLI DI INPUT** \Rightarrow Per impedire l'interpretazione accidentale del Messaggio Originale
- **UNIVOCHE** \Rightarrow Bisogna minimizzare la possibilità che 2 msg diversi generino lo stesso Hash
- **NON INVERTIBILI** \Rightarrow Risalire al Messaggio dall'Hash deve essere COMPUTAZIONALMENTE IMPOSSIBILE

Utilizzata anche per la Gestione delle Password e non solo per la firma

Nel Mondo Virtuale la firma \neq firma di Qualcuno

AUTORITÀ DI CERTIFICAZIONE:

Voglio la CERTEZZA che la chiave pubblica sia di chi mi Aspetto e mi fido di un Entità Terza.

Tramite un file detto CERTIFICATO DIGITALE con i Campi Necessari per l'Identificazione.

L'Equivalento della foto della CARTA D'IDENTITÀ Nel Certificato è la CHIAVE PUBBLICA di chi Vuole Essere Certificato

Certificato, Essendo che lo do a Tutti NON può Contenere la chiave PRIVATA.

Non Basta un Certificato ma ho Bisogno Anche di un ENTE CERTIFICATORE che Verifica che sia IO.

IL Certificato ha una SCADENZA

AUTENTICAZIONE:

• LOCALE \Rightarrow Ad un dispositivo (RAPPORTO DIRETTO, NON C'È RETE IN MEZZO)
solo persone

DIRETTA \Rightarrow da Remoto (ex. HomeBanking), dove c'è una RETE tra i 2 Attori
REMOTA anche Macchine
INDIRETTA \Rightarrow

• OFF-LINE \Rightarrow

Basate su Qualcosa che l'utente:

- CONOSCE \Rightarrow Segreti come PIN
- POSSIENEDE \Rightarrow cose fisiche o Elettroniche
- È \Rightarrow Aspetti BIOMEDICI come impronte digitali

Autenticazione a N-fattori per MIXARLI e Aggiungere Sicurezza

USERNAME e PSW è il Più Semplice ed è esposta al Pericolo di BRUTEFORCING (come la Chiave).

Si può fare anche un dizionario (con Parole di Senso Compito)

ONE TIME PASSWORD:

Viene Generata ad Ogni Accesso ad Esempio:

- SMS
- TOKEN (google Auth)

REMOVA DIRETTA: Non Basta Cifrare il Canale, Potrei Eseguire un Atacco REPLAY dopo Essermi Accorto di Alcune Ripetizioni durante la comunicazione ed Allora lo Rieseguo

Come Autentico le 2 parti? Protocollo di Sfida

M SERVER \Rightarrow Associazione nome \leftrightarrow IP

M UTENTE \Rightarrow Manda un msg al Server (SEMPRE DIVERSO) e poi Srv risponde con il msg cifrato con la sua Chiave privata e quindi poi Posso Verificare con la Chiave Pubblica del SRV se Effettivamente è lui o No.

Protocollo di Sfida si Basa su TCP, visto che Potrei perdere qualche Pacchetto (ESISTE ANCHE SU UDP ma PIÙ COMPLICATO)

UTENTE Lo Autentico Tramite Username e Password

AUTORIZZAZIONE:

Servizio di Controllo degli Accessi; soggetti ≠ possono avere diritti a Modalità di Interazione con le Risorse diverse.

PROTOCOLLI e APPLICATIVI DI RETE:

- ① ESTENDO IP (aggiungo Sicurezza ad IP) e diventa **IPSec**.
- ② Aggiungo Sicurezza a livello di Trasporto \Rightarrow **SSL o TLS** sopra al TCP.
- ③ Rendo Sicura l'Applicazione \Rightarrow Devo però Assicurarmi di **Proteggere** in tutte le fasi (caso non scontata)

Quando Metto sicurezza ad un livello tutti Quelli Sopra diventano sicuri

TLS non protegge END-TO-END e Bisogna Inventarsi Qualcosa come fatto in MQTT.

IPSec ha il Vantaggio di Proteggere sia TCP che UDP ma anche lui non protegge il Messaggio Quando è nel Broker o nel Server di posta

PGP (Pretty Good Privacy)

RENDERE SICURA CONNESSIONE TCP :

SSL ⇒ Secure Socket Layer

FASE 1 di HANSHAKE:

- Per il SERVER
- Mutua Autenticazione

Genera Chiavi di Sessione

Serve un Certificato, Altrimenti Inutilizzabile

Viene Aggiunto NAC che è la firma digitale di Quel Messaggio
Message Authentication Code



Contiene TIMESTAMP Anche e Quindi ci Protegge da
Atteggi di TIPO REPLAY.

WEP (Wired Equivalent Privacy)

Sicurezza del WiFi, Più Vulnerabile di ETHERNET perché è difficilmente confinabile ed Ascoltabile Passivamente (senza rilevare Presenza)

WEP è durato poco

WPA - WPA2 - WPA3

SOTÀ, La corsa è Quella di creare CHIAVI PIÙ LUNGHE possibili perché con l'EVOLUZIONE dell' Hardware gli Attaccanti hanno a disposizione risorse Sempre più Potenti.

FIREWALL:

ROUTER DI FRONTERA \Rightarrow Router da dove passano TUTTI i pacchetti del mondo esterno Verso la mia Sotto-Rete e Viceversa.

FIREWALL deve Bloccare dei Pacchetti sì, MA con Intelligenza

Può lavorare solo a LVL 3 ma anche a Superiori, come per accettare SYN-ACK dei pacchetti SYN che Sono usciti e Quindi deve Entrare nel HEADER TCP.

Ma Allora devo avere un FIREWALL Statefull, quindi con MEMORIA perché devo Ricordarmi che cosa è uscito

FIREWALL è la PORTA BLINDATA del Mondo Virtuale/Informatica

NON può fare Nulla se la RETE Viene Infettata da dentro, ad Esempio con USB Infette.

FIREWALL può Essere Anche Installato sul PC (porta Blindata della Stanza)

DMZ \Rightarrow Demilitarized Zone

INTRUSION DETECTION SYSTEM:

Complementare al FIREWALL, è Equivalente all' Allarme che Segnala Anomalie Nella Rete.

Come Se fosse un Antivirus che sfrutta Wireshark per Analizzare La RETE.

Ecco perché Si Sfrutta il WEB e non si Inventano Sempre dei Protocolli Specifici... altrimenti un INTRUSION DETECTION SYSTEM potrebbe Bloccare un COMPORTAMENTO LEGITIMO

COME SALVARE PSW SUL SERVER:

Server lo Autentico Tramite Certificato, ma il **CLIENT**?

Psw mai Salvate in chiaro su SERVER BACK-END, ma vengono sottoposte alla funzione di HASH, che essendo Invertibile NON de Nessuna Informazione

FRONTEND NON fa HASH! Lo fa il BACKEND

2 Utenti Potrebbero avere Psw uguali e Quindi anche Stessa HASH ed Allora si Aggiunge alla Password il TIME-STAMP e solo dopo si fa WASHING e così ha un Risultato Univoco

Utente Scrive Psw Per Esteso, in Chiaro (perchè Canale è cifrato) e poi il SERVER Se il Controllo.

Colonna "SALE", dove c'è il TIMESTAMP (deve essere ≠ ogni riga.)

user	Hash-PSW-SALE	SALE	...
utente	hash	stamp	

Mi aiuta Anche a
Gestire Scadenza Password

Sono in Chiaro Senza Problemi