

# Report Descrittivo sull'Esercizio di Configurazione Server Web con Analisi HTTP e HTTPS

W4D4 - Progetto finale di LUCIO BOSCHI classe CSPT 0524

---

## Indice del Report

### 1. Introduzione

- Obiettivi dell'esercizio
- Strumenti e ambiente utilizzati

### 2. Configurazione di Rete

- Dettagli della configurazione IP per Kali Linux e Windows
- Test di connessione tramite ping

### 3. Configurazione del Server Apache

- Installazione di Apache su Kali Linux
- Configurazione del protocollo HTTPS con certificato SSL autofirmato
- Avvio e test del server web

### 4. Test di Connessione

- Accesso al server tramite protocollo HTTP
- Accesso al server tramite protocollo HTTPS
- Gestione del certificato autofirmato

### 5. Analisi del Traffico con Wireshark

- Cattura e analisi del traffico HTTP
- Cattura e analisi del traffico HTTPS
- Differenze tra traffico HTTP e HTTPS

### 6. Confronto tra HTTP e HTTPS

- Sicurezza
- Visibilità dei dati
- Utilizzo e scenari applicativi

### 7. Conclusioni

- Riflessioni sulla configurazione e sull'analisi
- Importanza dell'HTTPS per la sicurezza

## Introduzione

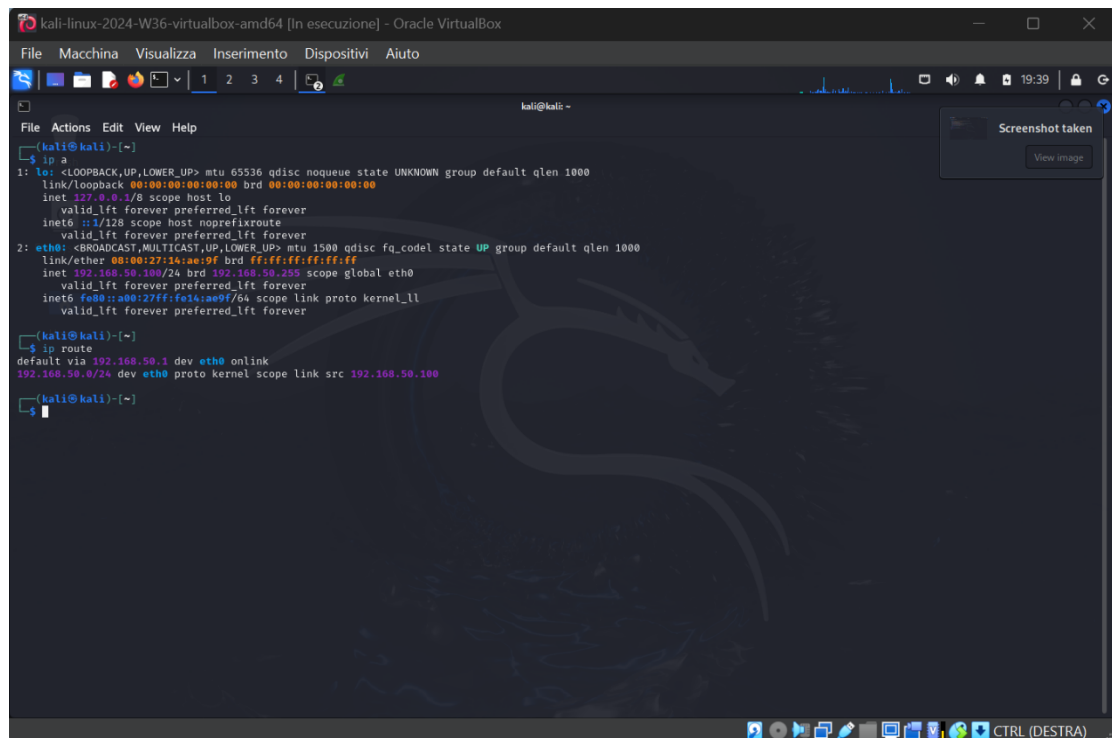
L'obiettivo dell'esercizio è stato configurare un server web su una macchina Kali Linux, consentendo l'accesso da un client Windows tramite i protocolli HTTP e HTTPS. Inoltre, è stata effettuata un'analisi del traffico generato usando Wireshark per confrontare il comportamento dei due protocolli.

Le immagini fornite documentano ogni fase del processo, dalla configurazione di rete alla cattura e analisi del traffico di rete.

## 1. Configurazione di Rete

Per configurare la rete interna tra le due macchine virtuali (Kali Linux e Windows 10), si è proceduto come segue:

- La macchina **Kali Linux** è stata configurata con:

A screenshot of a Kali Linux terminal window. The terminal shows the following commands and their outputs:

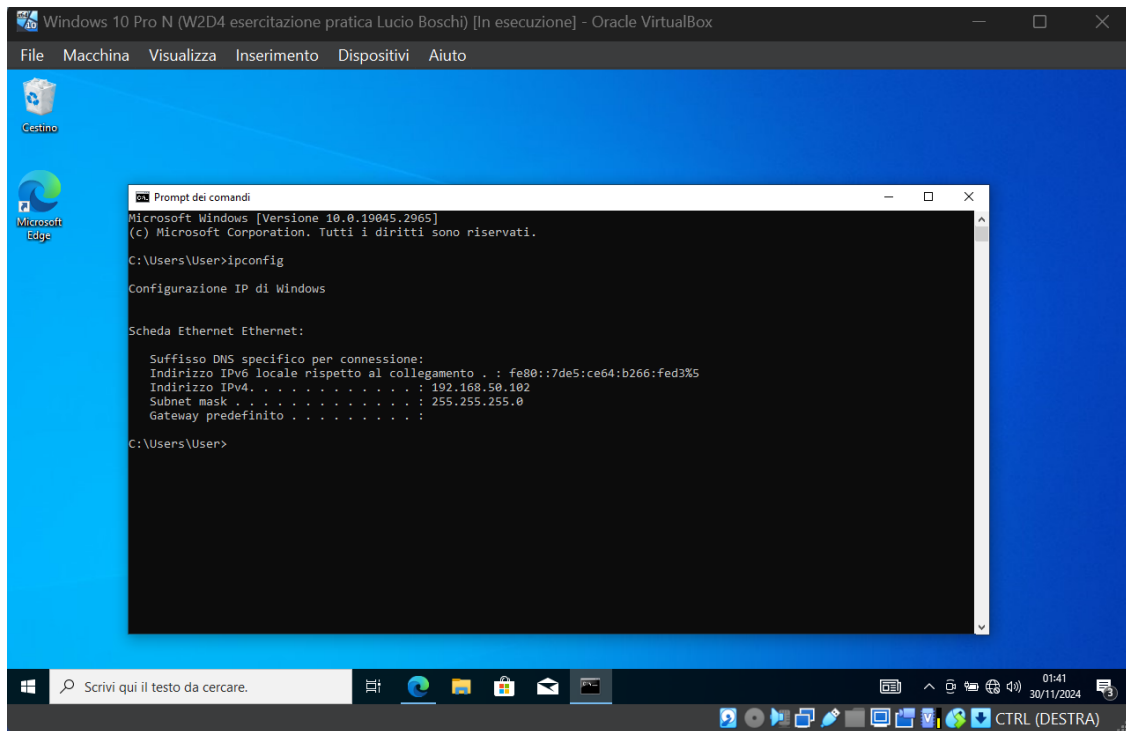
```
kali@kali:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:14:ae:9f brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.100/24 brd 192.168.50.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe24:ae9f/64 scope link proto kernel ll
        valid_lft forever preferred_lft forever

kali@kali:~$ ip route
default via 192.168.50.1 dev eth0 onlink
192.168.50.0/24 dev eth0 proto kernel scope link src 192.168.50.100

kali@kali:~$
```

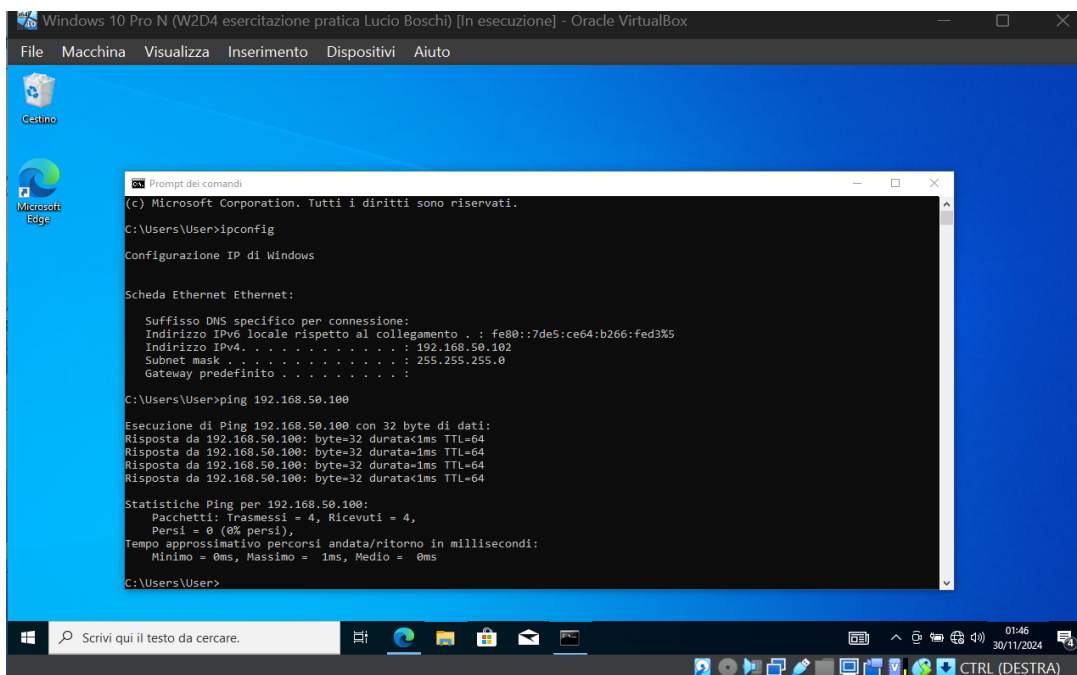
- IP: 192.168.50.100
- Gateway: 192.168.50.1
- Subnet mask: 255.255.255.0
- DNS: 8.8.8.8, 8.8.4.4

- La macchina **Windows 10** è stata configurata con:



- IP: 192.168.50.102
- Gateway: 192.168.50.1
- Subnet mask: 255.255.255.0

Le connessioni sono state testate tramite **ping**, con esito positivo.



## 2. Configurazione del Server Apache

Sulla macchina Kali Linux è stato installato il server Apache per servire contenuti tramite HTTP e HTTPS:

### 1. Installazione di Apache:

- Il server è stato installato usando `sudo apt install apache2`.

### 2. Configurazione HTTPS:

- È stato generato un certificato *SSL autofirmato* usando *OpenSSL*.
- Il modulo SSL è stato attivato con `sudo a2enmod ssl`.
- È stato modificato il file di configurazione di default di Apache (*default-ssl.conf*) per includere il certificato appena creato.

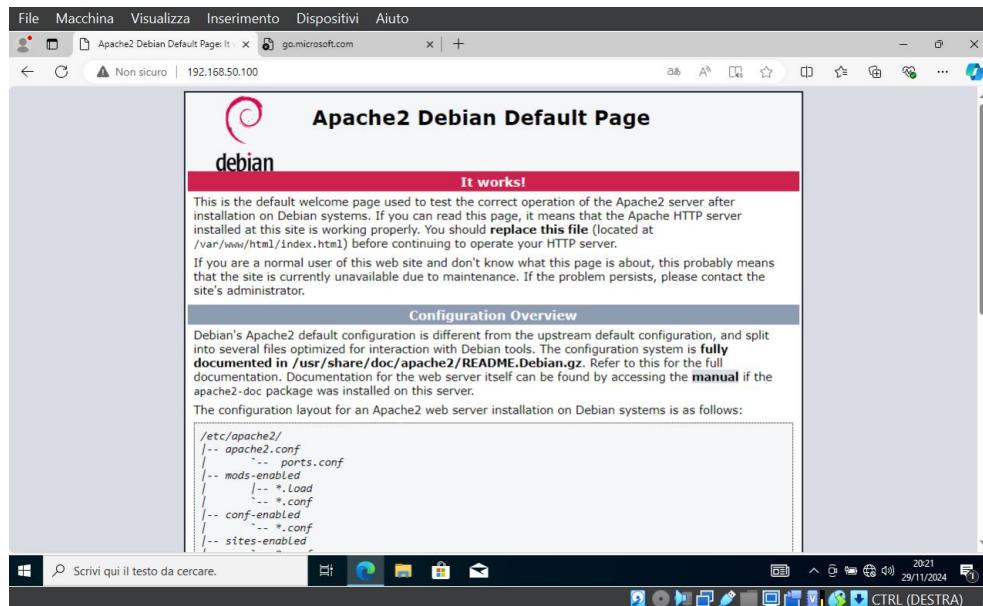
### 3. Avvio del server:

- Apache è stato avviato e il servizio è stato reso disponibile per il traffico sia HTTP che HTTPS.

### 3. Test della Connessione

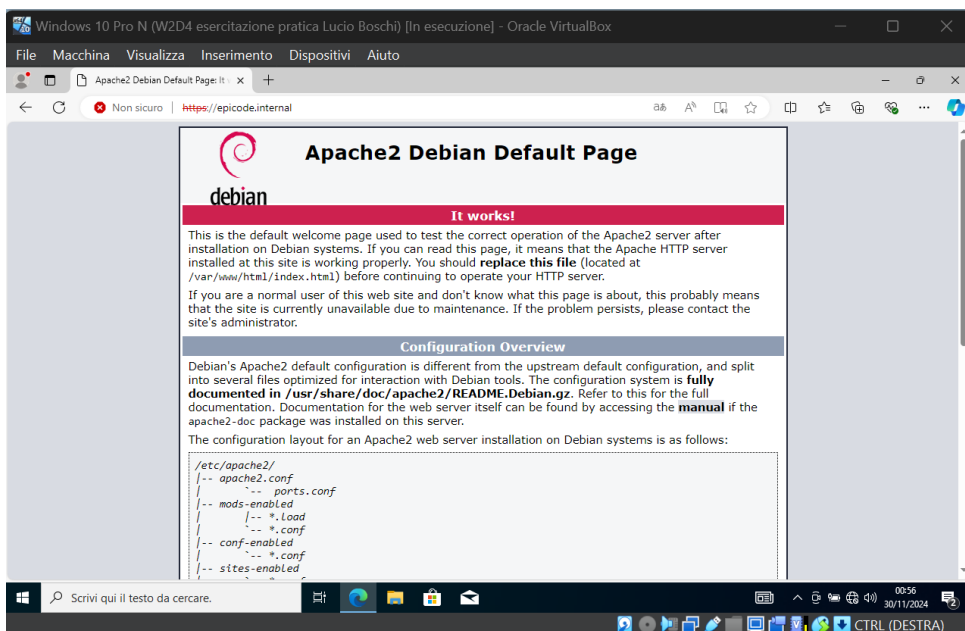
Dalla macchina Windows, sono stati effettuati i test di connessione:

#### 1. Accesso HTTP:



Utilizzando `http://192.168.50.100`, è stata visualizzata la pagina predefinita di Apache (Debian Default Page).

#### 1. Accesso HTTPS:



Utilizzando `https://epicode.internal`, è stata visualizzata la stessa pagina tramite connessione cifrata. Il browser ha segnalato un certificato autofirmato, ma è stato possibile proseguire accettando manualmente il certificato.

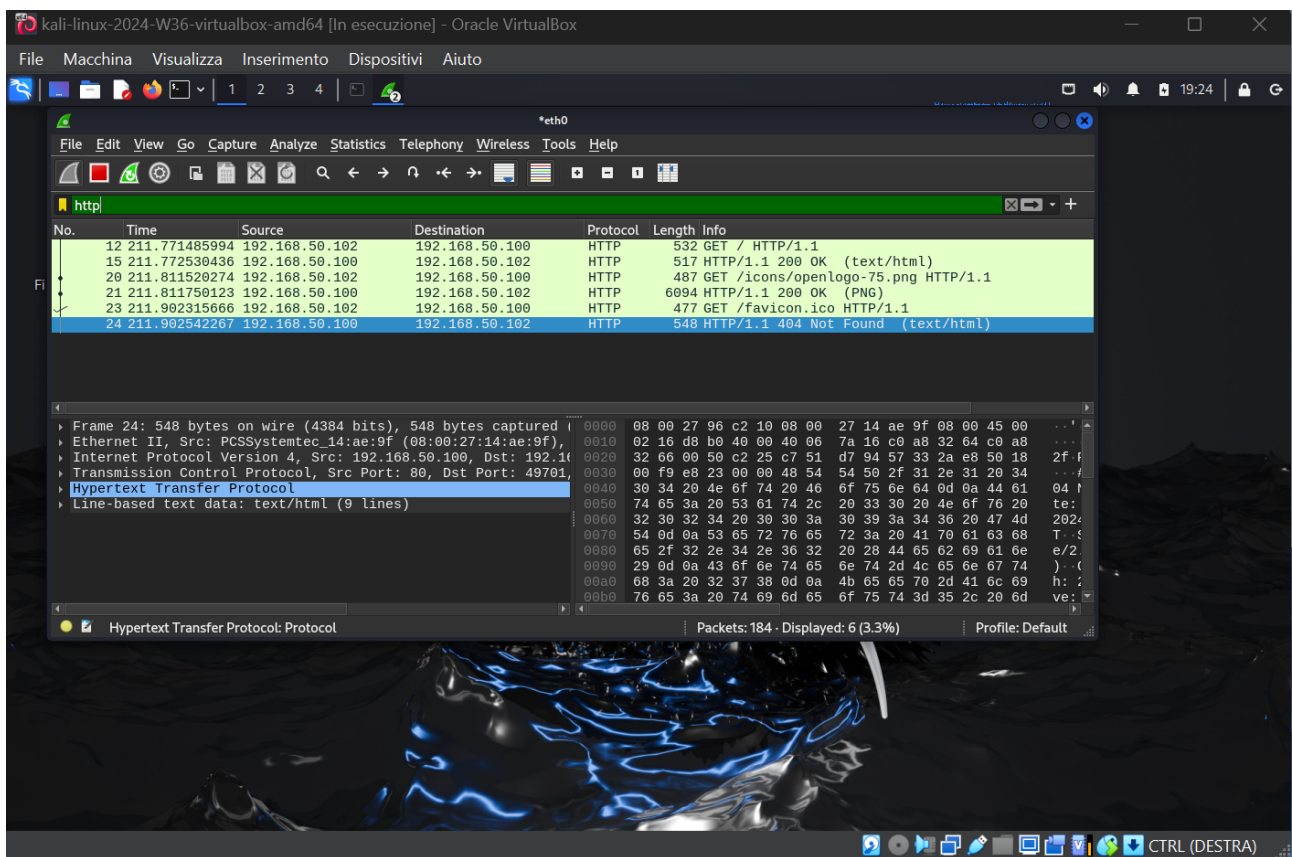
Le immagini mostrano chiaramente il risultato di entrambe le connessioni.

## 4. Cattura del Traffico con Wireshark

Utilizzando Wireshark sulla macchina Kali Linux, sono state catturate le comunicazioni tra il client Windows e il server Apache:

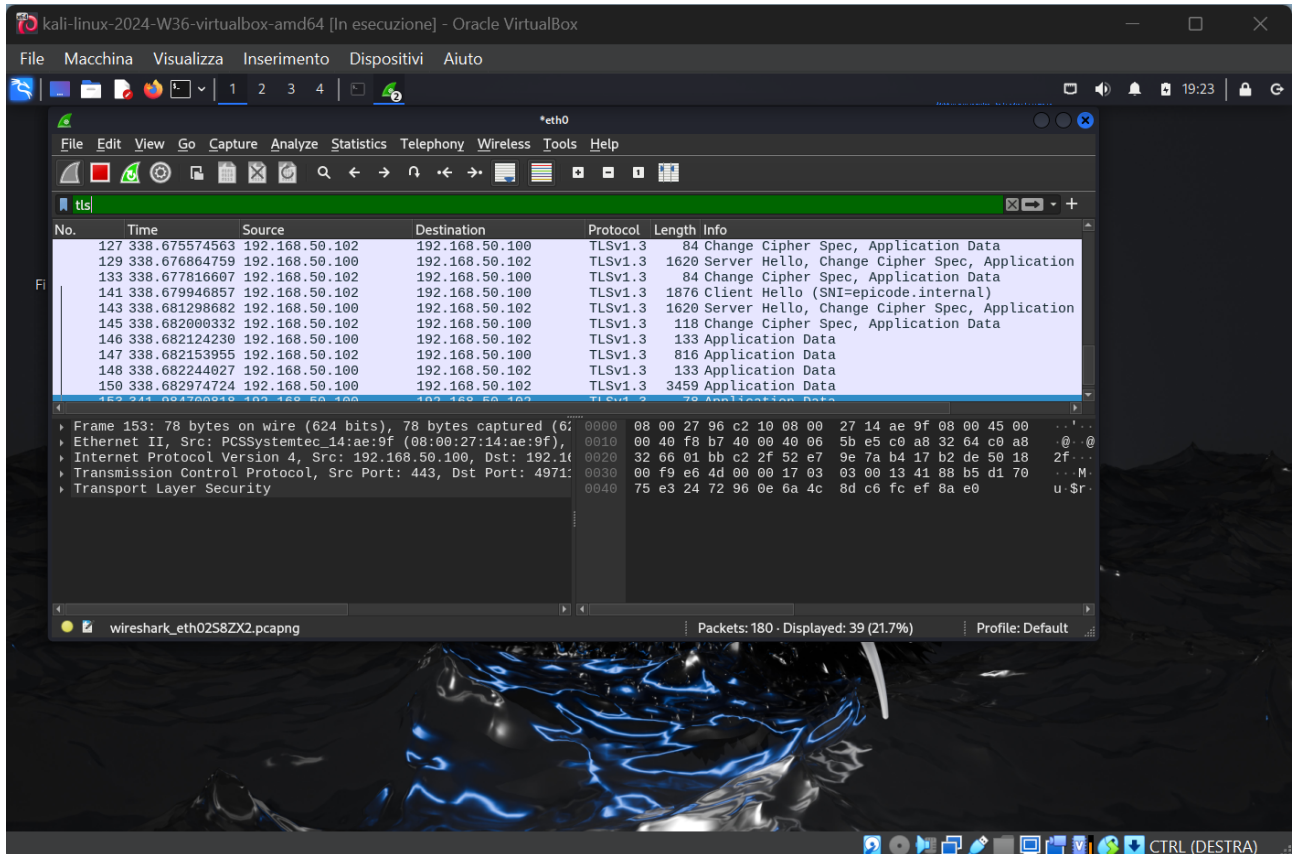
### 1. Traffico HTTP:

- Wireshark ha catturato il traffico in chiaro, dove sono visibili richieste HTTP come GET / HTTP/1.1 e dettagli dei contenuti richiesti.



## 2. Traffico HTTPS:

- Wireshark ha mostrato il traffico cifrato utilizzando il protocollo TLS. I pacchetti catturati evidenziano l'handshake TLS e i dati crittografati, impedendo la visualizzazione del contenuto.



Le immagini fornite mostrano chiaramente:

- I dettagli del traffico HTTP in chiaro.
- I pacchetti TLS cifrati e non leggibili.

## 5. Confronto tra HTTP e HTTPS

Caratteristica	HTTP	HTTPS
Sicurezza	Nessuna, traffico in chiaro	Cifratura con TLS
Visibilità su Wireshark	Contenuto leggibile, inclusi header e dati	Solo handshake e pacchetti cifrati
Utilizzo	Adatto a contenuti non sensibili	Essenziale per protezione dati

## Conclusioni

L'esercizio ha dimostrato:

- La configurazione di un server web con supporto per HTTP e HTTPS.
- L'importanza del protocollo HTTPS per proteggere i dati grazie alla crittografia.
- L'utilità di Wireshark per l'analisi del traffico di rete.

Grazie all'uso delle immagini, è stato possibile documentare ogni passaggio, evidenziando il processo di configurazione e le differenze tra i protocolli HTTP e HTTPS.