

# Web Security: introduzione al Web e HTTP

Leonardo Taccari

`<s1069964@studenti.univpm.it>`

# Sommario

World Wide Web

Uniform Resource Locator (URL)

HyperText Transfer Protocol (HTTP)

Conclusioni

Riferimenti

# World Wide Web

# World Wide Web

- ▶ World Wide Web, AKA WWW, AKA Web <sup>1</sup>
- ▶ È una collezione globale di documenti e altre risorse, collegate da **hyperlinks** e **URL**.
- ▶ I documenti vengono scambiati attraverso il protocollo di comunicazione **HTTP**
- ▶ La maggior parte dei documenti sono degli **ipertesti** scritti nel linguaggio **HTML**

---

<sup>1</sup>Lo chiameremo semplicemente Web a seguire.

# World Wide Web

## Un po' di storia

- ▶ Proposto da Tim Berners-Lee al CERN nel 1989 e rilasciato pubblicamente nel 1993
- ▶ Nel 1994 viene fondato il World Wide Web Consortium (W3C) che si occupa degli standard che costituiscono il Web
- ▶ Inizialmente nato per scambiare **ipertesti**, poi **ipermedia**
- ▶ **Web 1.0**, **web statico**, l'utente può esclusivamente leggere i contenuti senza possibilità di modificarli
- ▶ **Web 2.0**, **web dinamico**, l'utente può anche modificare i contenuti

# World Wide Web

Home page del primo sito web della storia

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#) )

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help ?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#) , etc.

Home page del primo sito web della storia, pubblicato il 20 dicembre 1990 su <https://info.cern.ch/hypertext/WWW/TheProject.html> (catture da Wayback Machine dell'Internet Archive)

# Uniform Resource Locator (URL)

- ▶ Ogni risorsa nel Web è identificata da un indirizzo detto **Uniform Resource Locator (URL)**
- ▶ Ad esempio, `https://www.example.org/index.html` identifica l'**home page** di **www.example.org**
- ▶ Nel **browser web** l'URL viene mostrato in una barra

# Uniform Resource Locator (URL) I

## Anatomia di un URL

- ▶ Un **URL** è costituito da diverse parti
- ▶ La sintassi generale è  
`scheme://[userinfo@]host[:port][path][?query][#fragment]`

**scheme** indica il protocollo utilizzato, ad esempio `http` o `https`

**userinfo** (opzionale) solitamente nel formato `username:password` utilizzata per accedere a risorse che richiedono delle credenziali

**host** hostname o indirizzo IP

**port** (opzionale) port, di default 80 per HTTP e 443 per HTTPS <sup>2</sup>

**path** (opzionale) "percorso", ogni componente è prefissato da un `/`

**query** (opzionale) "query string", costituisce coppie attributi-valori separati da `&`, solitamente utilizzate per sottomettere dati



# Uniform Resource Locator (URL) II

## Anatomia di un URL

**fragment** (opzionale) identifica un determinato frammento all'interno della risorsa (ad esempio permette di "evidenziare" una parte di testo)

- ▶ Esempio: `https://www.example.org/index.html`

`https` è lo **schema** seguito da :

`www.example.org` è l'**host** prefissato da //

`/index.html` è il **path** costituito da un singolo componente

---

<sup>2</sup>Vedremo meglio ciò in Network Security!

# HyperText Transfer Protocol (HTTP)

- ▶ **HyperText Transfer Protocol (HTTP)** è il protocollo di comunicazione utilizzato per scambiarsi risorse nel Web
- ▶ Basato sul modello **client-server**: la comunicazione viene iniziata dal **client** che **richiede** una **risorsa** al **server** che a sua volta fornisce al client la risorsa in una **risposta**
- ▶ L'applicazione client HTTP viene detta **browser web**
- ▶ L'applicazione server HTTP viene detta **web server**

# HyperText Transfer Protocol (HTTP)

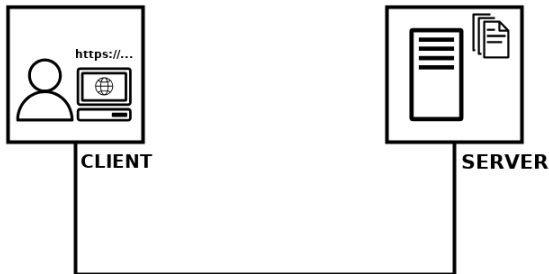
## Flusso di una richiesta e risposta HTTP

- ▶ Vediamo il flusso di richiesta e risposta HTTP
- ▶ Come esempio pratico ci chiediamo che cosa succede quando scriviamo nella barra del browser `https://www.kittenwar.com/c_images/2006/08/10/84180.1.jpg`

# Flusso di una richiesta e risposta HTTP

## Digitazione dell'URL nel browser

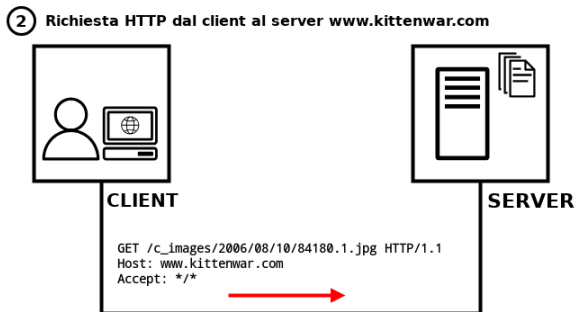
① **Digitazione URL** <[https://www.kittenwar.com/c\\_images/2006/08/10/84180.1.jpg](https://www.kittenwar.com/c_images/2006/08/10/84180.1.jpg)>



Digitiamo nel browser web l'URL all'immagine di Joe Dirt

# Flusso di una richiesta e risposta HTTP

## Richiesta HTTP

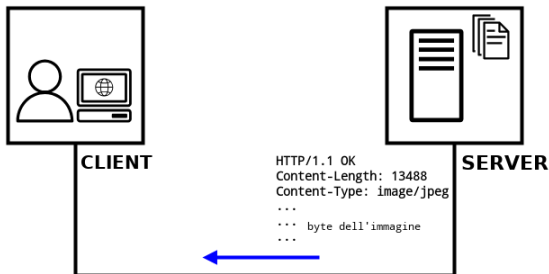


Il browser spedisce al server la richiesta HTTP per ottenere la risorsa

# Flusso di una richiesta e risposta HTTP

## Risposta HTTP

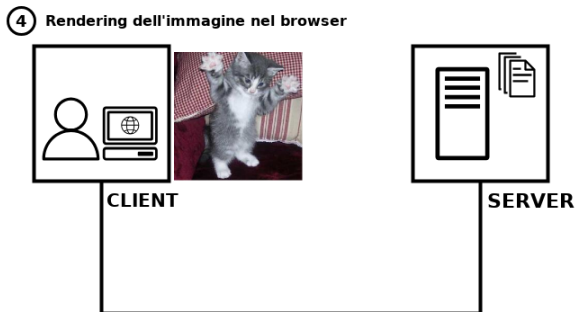
### ③ Risposta HTTP dal server al client



Il server web restituisce in risposta la risorsa richiesta

# Flusso di una richiesta e risposta HTTP

## Rendering dell'immagine nel browser



L'immagine viene visualizzata nel browser web

# HyperText Transfer Protocol (HTTP)

## Messaggio HTTP

- ▶ Le richieste e le risposte HTTP sono dei **messaggi HTTP**
- ▶ I messaggi HTTP hanno la seguente struttura:
  - riga iniziale (start line)** indica la versione HTTP e il metodo della richiesta (per richieste) o lo status della risposta (per risposte)
  - header** metadata che descrivono il messaggio
  - riga vuota (empty line)** riga vuota (CRLF, carriage return e line feed) che separa l'header dal body
  - body** (opzionale) contenuto del messaggio



# HyperText Transfer Protocol (HTTP) I

## Richiesta HTTP

- ▶ Spedita dal **client** al **server**
- ▶ È costituita da:
  - request-line** contiene la tripletta **metodo (method)**, **request-target** e **protocollo (protocol)**
    - metodo (method)** metodo (detto anche verbo), ad esempio: GET (per ricevere una risorsa) o POST (per spedire dati)
    - request-target** solitamente un URL relativo (senza dominio)
    - protocollo (protocol)** versione del protocollo HTTP utilizzata
  - request header** metadati, Host: è l'unico sempre richiesto, si suddividono in **request header** e **representation header**
  - body** (opzionale) contenuto della richiesta

# HyperText Transfer Protocol (HTTP) I

## Risposta HTTP

- ▶ Spedita dal **server** al **client**
- ▶ È costituita da:
  - status line** contiene la tripletta **protocollo (protocol)**, **status code** e **status text**
    - protocollo (protocol)** versione del protocollo HTTP utilizzata
    - status code** codice numerico che indica l'esito della richiesta, ad esempio 200, 403, 404, 503 <sup>3</sup>
    - status text** descrizione testuale dello status code
  - response header** metadati della risposta, si suddividono in **response header** e **representation header**
  - body** (opzionale) contenuto della risposta

---

<sup>3</sup>Diversi status code sono rappresentati in <https://http.cat>

# HyperText Transfer Protocol (HTTP)

Un po' di pratica!

- ▶ **curl** è un client **HTTP**
- ▶ Non è un vero e proprio browser: non permette di visualizzare gli ipermedia ma ci permette di effettuare richieste HTTP e visualizzare la risposta HTTP
- ▶ È molto potente ed utile sia per capire HTTP che per risolvere challenge
- ▶ Leggiamo velocemente la pagina di manuale `curl(1)` con `man curl`
- ▶ Vediamo e commentiamo insieme  
`curl -http1.1 -verbose http://www.example.org/index.html`

# Conclusioni

- ▶ Abbiamo visto che cosa è il **World Wide Web** e la sua storia
- ▶ Abbiamo visto che cosa sono gli **URL**, gli identificatori di risorse alla base del Web
- ▶ Abbiamo visto il protocollo di comunicazione **HTTP** e visto in dettaglio - ed anche in pratica! - le richieste e risposte HTTP

# Riferimenti

- ▶ Materiale Didattico del Portale di allenamento delle Olimpiadi Italiane di Cybersicurezza
- ▶ MDN Web Docs