Web Security 2
Back-end e database

Riccardo BONAFEDE

Università degli Studi di Padova

https://cybersecnatlab.it

License & Disclaimer

License Information

This presentation is licensed under the Creative Commons BY-NC License

To view a copy of the license, visit:

http://creativecommons.org/licenses/by-nc/3.0/legalcode

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Obiettivi

- Comprendere a grandi linee il concetto di back-end
- Comprendere il concetto di database ed il loro funzionamento

© CINI – 2021

Prerequisiti

- Modulo NS_1.1 - Fondamenti di reti di calcolatori
- Modulo WS_1.1 - Il browser web e HTTP

Argomenti

- Back-end
  - Variabili GET/POST
  - Url-encoding
- Database

© CINI – 2021

Argomenti

- Back-end
  - Variabili GET/POST
  - Url-encoding
- Database

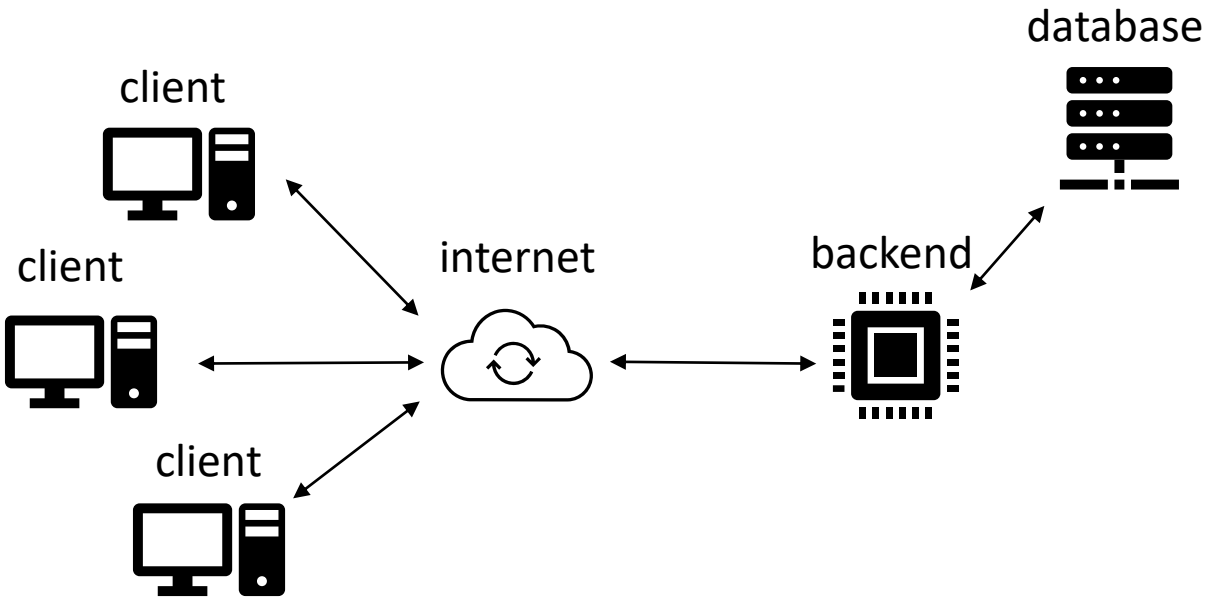
Back-end

- Con il passare del tempo i siti web si sono evoluti, e hanno acquisito nuovi utilizzi:
  - Social network
  - Home banking
  - E-Commerce

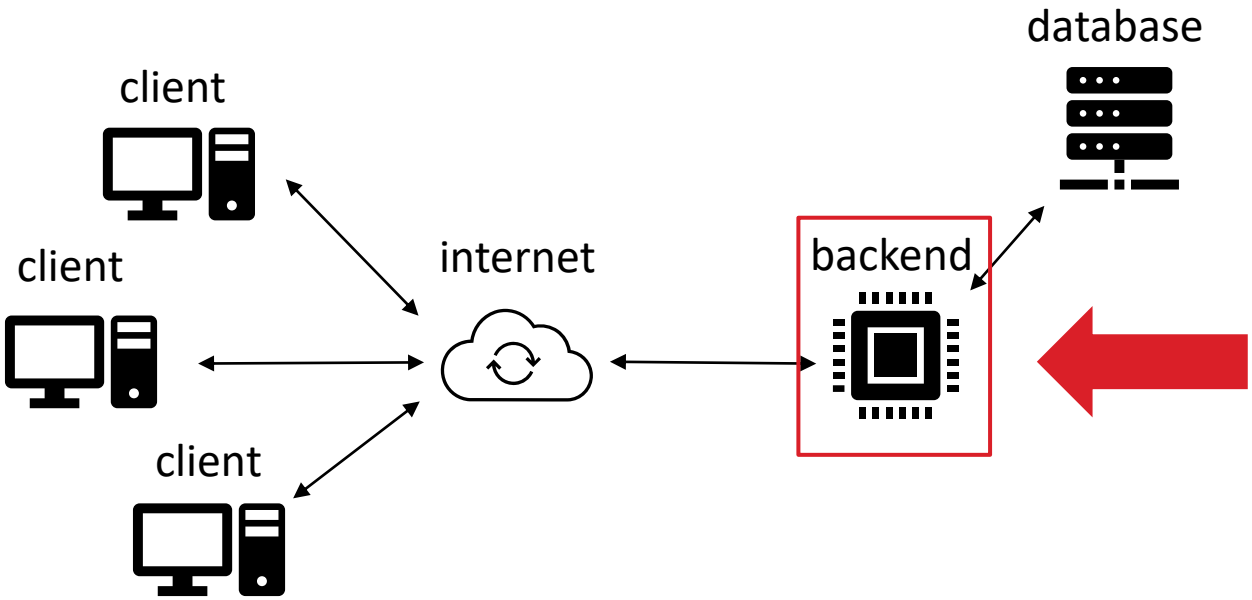
Back-end

- I siti web sono diventati dei veri e propri programmi, che necessitano lo scambio di informazione tra l'utente e il web server

Back-end



Back-end



Back-end

- Il back-end è la parte di un sito web che agisce lato server
- È responsabile dell'elaborazione dei dati dell'utente

© CINI – 2021

Back-end

- I back-end possono essere scritti in un qualsiasi linguaggio
- Tipicamente si utilizzano dei linguaggi o dei framework dedicati, ad esempio:
  - PHP
  - Flask
  - Ruby on Rails
  - …

Back-end

- La comunicazione tra front-end e back-end avviene generalmente all'interno delle richieste HTTP
- I dati del front-end vengono passati in diversi modi e formati:
  - Variabili GET, POST
  - All'interno del body di una richiesta

Argomenti

- Back-end
  - Variabili GET/POST
  - Url-encoding
- Database

Back-end

- Le variabili GET sono il metodo più comune per passare informazioni al back-end
- Sono inserite all'interno della path nell'URL
- Il nome deriva dal fatto che possono essere passate all'interno di una richiesta GET
  - Ma possono essere mandate con un qualsiasi metodo

Variabili GET

- Le variabili GET sono inviate nell'url, in una parte detta query
- Da specifica la query si trova dopo un ?

```
<schema>://host/<directory>/<file>?<query>#fragment
```

Variabili GET

- Le variabili GET sono poi serializzate come una serie di coppie chiave-valore
- Ogni coppia è separata da un &
- E fra ogni chiave-valore è presente un =

```
http://foo.bar/?nome=mario&cognome=rossi
```

Variabili POST

- Le variabili *POST* vengono inviate nel body di una richiesta
- Vengono supportate molte sintassi diverse:
  - `application/x-www-form-urlencoded`
  - `multipart/form-data`
  - … Anche custom
- La sintassi utilizzata va specificat nell'header `Content-Type`

Variabili POST

- Esempio di richiesta POST

```
POST / HTTP/1.1
Host: www.example.com
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 7

foo=bar
```

Variabili POST


```
POST / HTTP/1.1
Host: www.example.com
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 7

foo=bar
```

- Esempio di richiesta POST
- Content-Type definisce il formato dei dati inviato

Variabili POST


```
POST / HTTP/1.1
Host: www.example.com
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 7

foo=bar
```

- Esempio di richiesta POST
- Content-Type definisce il formato dei dati inviato
- Nel Body sono presenti i dati, in questo caso usando la stessa sintassi delle variabili GET

Variabili GET/POST e PHP


```php
<?php
    echo 'Ciao ' . $_GET['nome'];
?>
```

- Saluta chi visita la pagina, prendendo il nome da una variabile GET

© CINI – 2021

Variabili GET/POST e PHP

- PHP utilizza delle variabili globali per leggere i dati forniti dagli utenti:
  - $_GET ← Variabili GET
  - $_POST ← Variabili POST
  - $_COOKIE ← Cookie

© CINI – 2021

Testing

- Per fare delle prove con PHP può essere utile usare il server di sviluppo integrato
- Per lanciarlo basta dare il comando:
  - $ php –S 127.0.0.1:5000

Testing

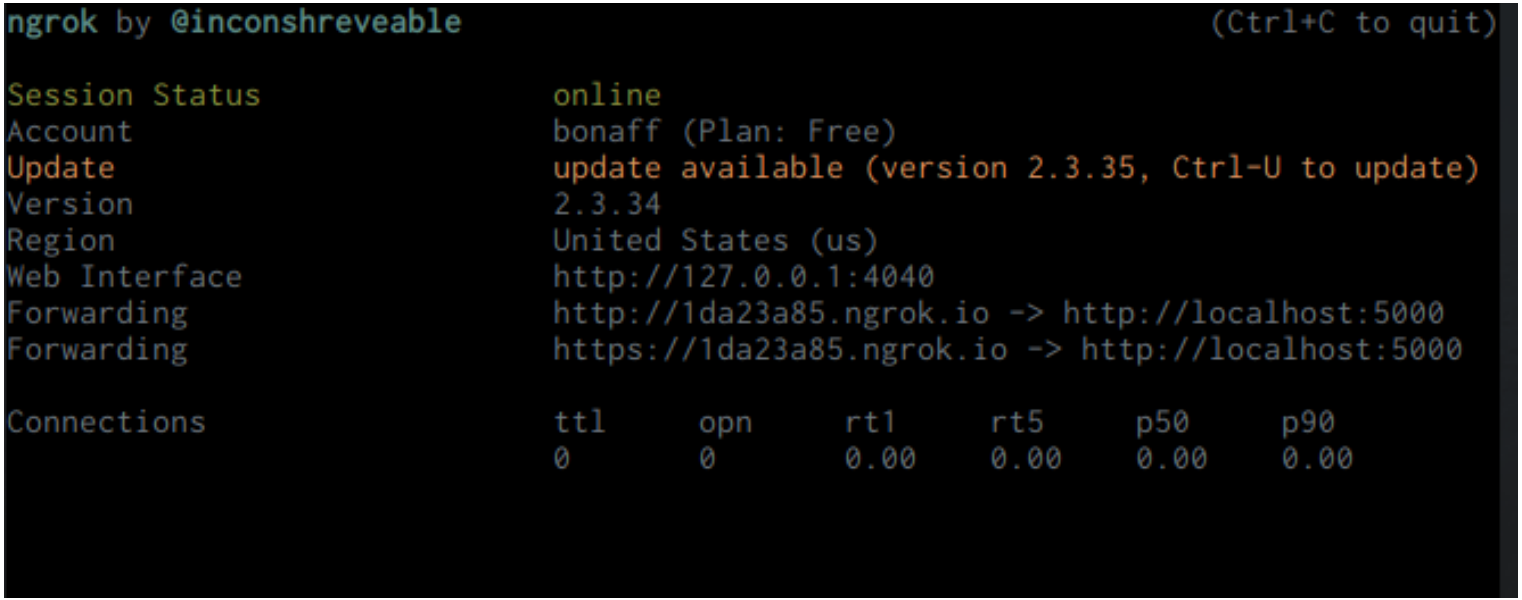
- Tutti i file nella cartella da cui viene lanciato il comando saranno serviti sulla porta locale 5000
- I file con estensione .php veranno eseguiti

© CINI – 2021

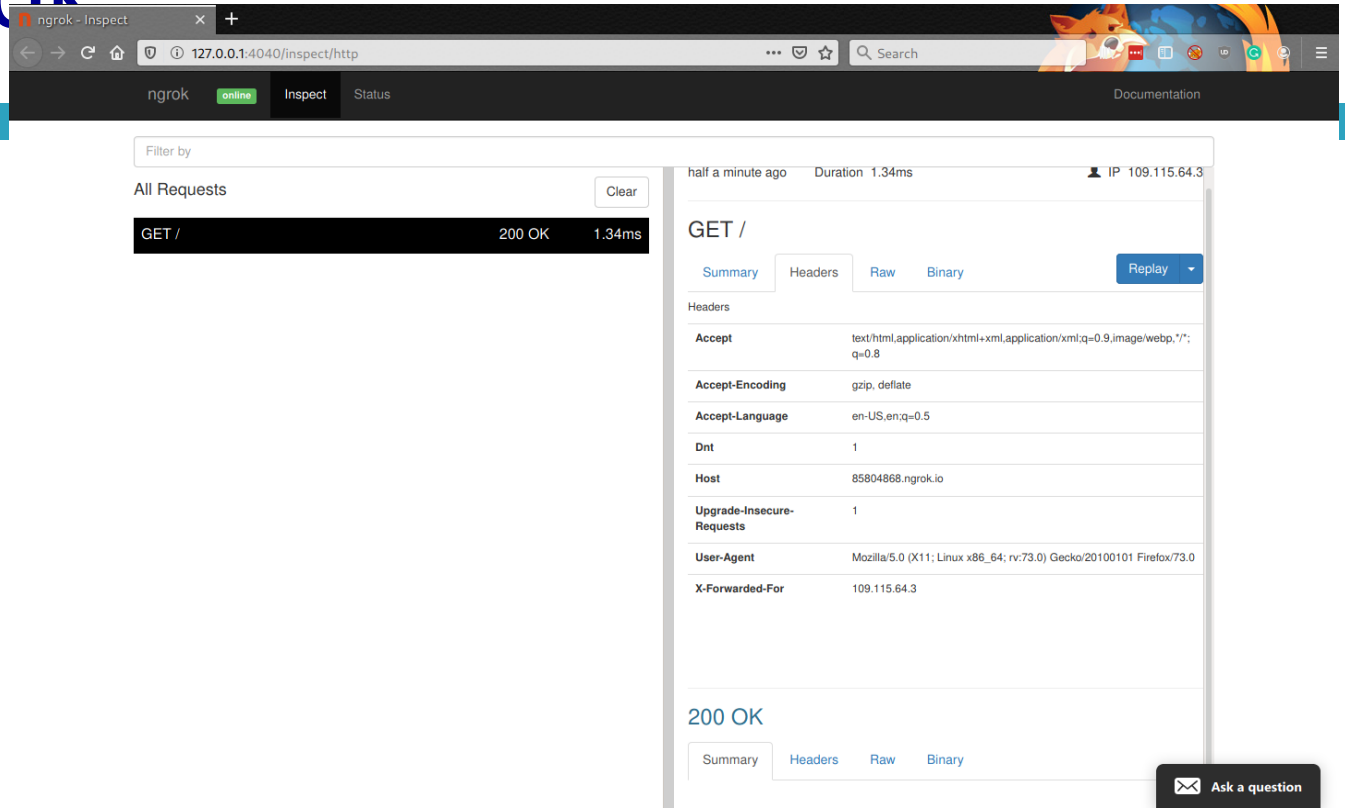
ngrok

- Per esporre il sito in rete è possibile usare ngrok
- Una volta installato, si può lanciare con il commando
- `$ ngrok http 5000`
  - Espone la porta 5000 in rete

ngrok


ngrok by @inconshreveable (Ctrl+C to quit)
Session Status online
Account bonaff (Plan: Free)
Update update available (version 2.3.35, Ctrl-U to update)
Version 2.3.34
Region United States (us)
Web Interface http://127.0.0.1:4040
Forwarding http://1da23a85.ngrok.io -> http://localhost:5000
Forwarding https://1da23a85.ngrok.io -> http://localhost:5000
Connections ttl opn rt1 rt5 p50 p90
0 0 0.00 0.00 0.00 0.00

ngrok


ngrok - Inspect
127.0.0.1:4040/inspect/http
Search
ngrok
online
Inspect
Status
Documentation
Filter by
All Requests
Clear
GET /
200 OK
1.34ms
half a minute ago
Duration 1.34ms
IP 109.115.64.3
GET /
Summary
Headers
Raw
Binary
Replay
Headers
Accept
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding
gzip, deflate
Accept-Language
en-US,en;q=0.5
Dnt
1
Host
85804868.ngrok.io
Upgrade-Insecure-Requests
1
User-Agent
Mozilla/5.0 (X11; Linux x86_64; rv:73.0) Gecko/20100101 Firefox/73.0
X-Forwarded-For
109.115.64.3
200 OK
Summary
Headers
Raw
Binary
Ask a question

Argomenti

- Back-end
  - Variabili GET/POST
  - Url-encoding
- Database

Url encoding

- Problema: come inserisco i caratteri & e ? In una variabile GET/POST ?
- Questi caratteri sono speciali, per esempio il & è utilizzato come separatore delle variabili

```
nome=&&cognome=rossi
```

URL encoding

- Questo problema si risolve trasformando I dati usando una codifica *safe*
- Questa codifica è chiamata “URL encoding" o anche “Percent Encoding”

URL encoding

- L'URL encoding funziona trasformando ogni carattere nel suo valore HEX e aggiungendoci un % davanti

  urlencode(#) == %23

- Tutti i caratteri riservati devono essere URL-encodati
- Anche tutti i caratteri non stampabili
- Gli spazi hanno una forma abbreviata: +

URL encoding

- Ricapitolando, l'URL seguente:

```
http://foobar.com/?var=hello &# world
```

- Viene riscritto come:

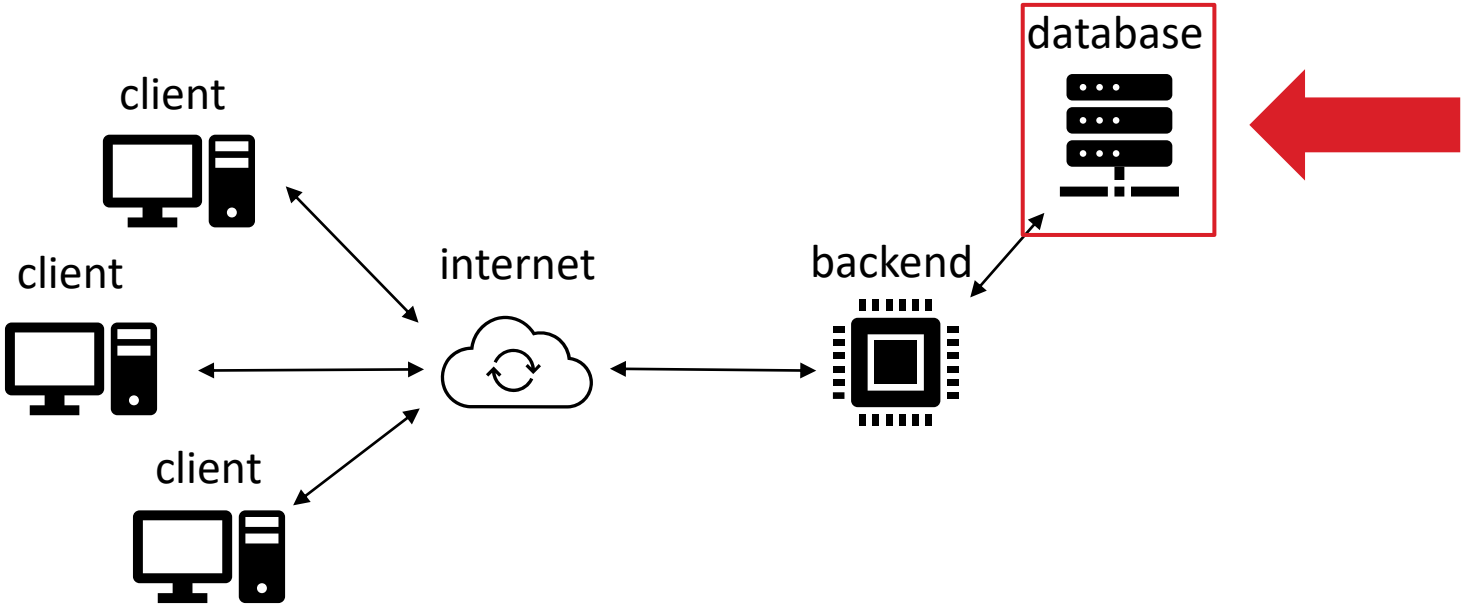
```
http://foobar.com/?var=hello+%26%23+world
```


© CINI – 2021

Argomenti

- Back-end
  - Variabili GET/POST
  - Url-encoding
- Database

Database



Database

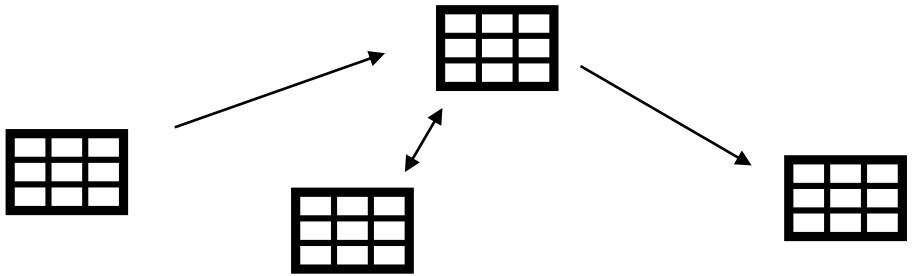
- Spesso i server hanno bisogno di salvarsi dati:
  - Informazioni dell'utente
  - Messaggi scambiati
  - I post di un blog
- I database sono programmi che offrono questa funzionalità

Database

- I vantaggi di usare database rispetto a salvare tutto su dei file sono molteplici:
  - Permettono il salvataggio di dati strutturati
  - Sono ottimizzati per le operazioni di lettura e scrittura
  - Sono standardizzati

Database relazionali

- Una tipologia di database è quella dei database relazionali, come ad esempio i database SQL
- Permettono la creazione di relazioni fra tipi di dato



Database SQL

- I database SQL salvano informazioni su delle tabelle

Username	admin	salvatore
Email	admin@..	aranzulla@..
Password	12345	mario2019

Id	1	2	3
Titolo	Come ..	Installare...	Le basi di ...
Testo	Per fare	Oggi ...	I databse. .
User	admin	salvatore	admin

Database SQL

- Le interazioni con i database SQL sono fatte mediante *query*
- Le query sono delle istruzioni inviate al database che fanno uso di uno specifico linguaggio
  - Nel caso dei database SQL-like questo linguaggio si chiama SQL (Structured Query Language)

Query SQL

- Recupera le password dell'utente che ha come username «admin»

Username	admin	salvatore
Email	admin@..	aranzulla@..
Password	12345	mario2019

```
SELECT password FROM utenti WHERE username = 'admin'
```

Database SQL

- I back-end interagiscono con i database utilizzando specifiche funzioni

```
$userQuery = mysqli_query("SELECT * FROM users
    WHERE email = '"    . $_POST['email']       . "'
    AND password = '"   . $_POST['password']    . "'"
);
```

Web Security 2
Back-end e database

Riccardo BONAFEDE

Università degli Studi di Padova

https://cybersecnatlab.it