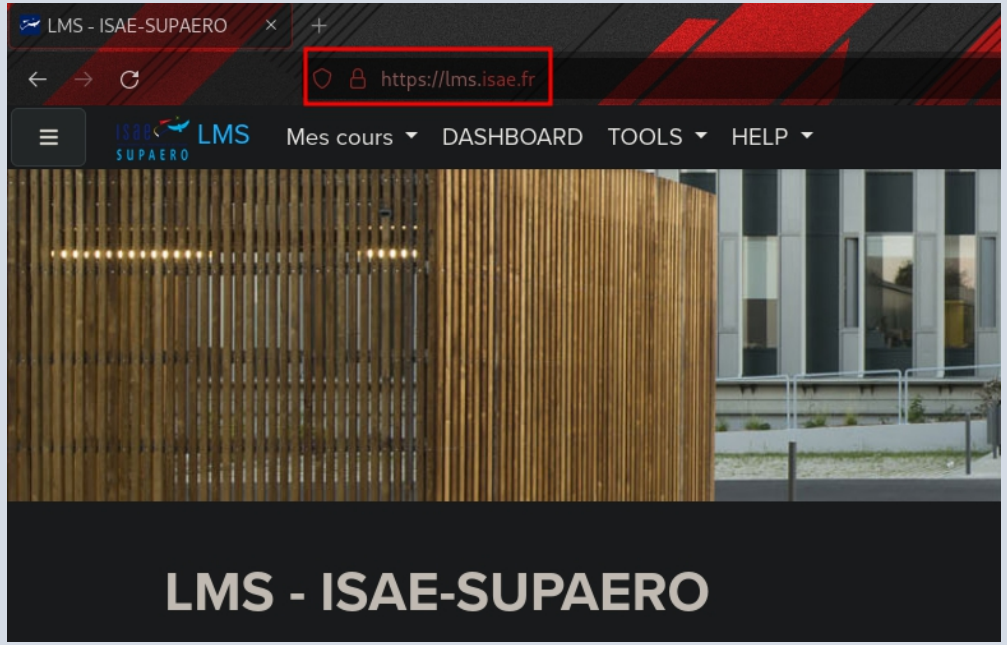
Web Introduction

01 - SCSC - Cybersecurity

Client/Seveur

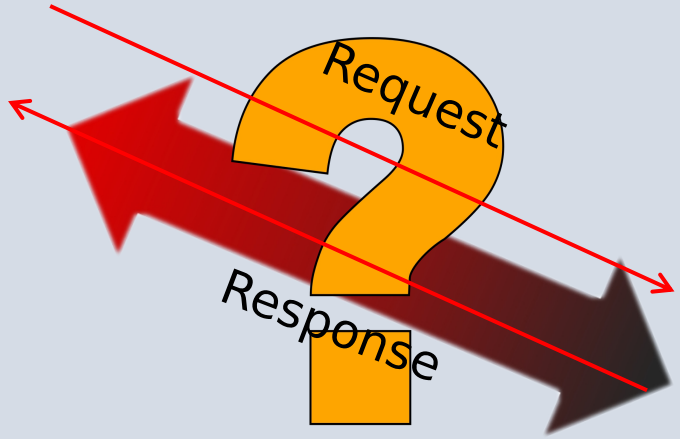


Client

- HTML
- CSS
- Javascript

Serveur

- MySQL
- PHP



Web Representation

ISAE
Institut Supérieur de l'Aéronautique et de l'Espace
SUPAERO

GET/POST Methods

```
1  GET / HTTP/1.1
2  Host: lms.isae.fr
3  Cookie: MoodleSessionLmsIsae=tpcja8h4dlh1mqpon85on1hhg3
4  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0
5  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Upgrade-Insecure-Requests: 1
9  Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: none
12 Sec-Fetch-User: ?1
13 Cache-Control: max-age=0
14 Te: trailers
15 Connection: close
16
17 user=alex&page=index
```

Example of a GET Header

HTTP Protocole

- GET: parameters in the url, ask data to the server

http://lms.isae.fr?user=alex&page=index

- POST: parameters only in the header, submit data to the server

Often used when sending forms

HTML & CSS

Hypertext Markup Language

```html
<!DOCTYPE html>
<html>
<body>

  <h1>My super Website</h1>
  <h2>Some random stuff</h2>
  <p>This is a paragraph.</p>
  <p>Click <a href='www.dangerous-wesite.com'>here</a>.</p>


  <h2>My cat</h2>
  <p>Here is my cat.</p>
  <img src='img/cat.png' />

  <h2>My shopping list</h2>
  <ul>
    <li>Another cat</li>
    <li>More cat food</li>
    <li>A new laptop</li>
  </ul>

</body>
</html>
```

Cascading Style Sheets

```css
body {
  background-color: lightblue;
}

head {
  border-style: solid;
  border-bottom-width: 15px;
}

p {
  color: red;
  text-align: center;
}

p.center {
  color: blue;
}
```

Javascript

Javascript

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript</h2>
<p id="demo">Hello, User!</p>

<input id="user">
<input id="pass">
<button type="button" onclick="myFunction()">Log In</button>

<script>
var x, y, z;  // Declare 3 variables
x = 5;    // Assign the value 5 to x
y = 6;    // Assign the value 6 to y
z = x + y;  // Assign the sum of x and y to z


function myFunction() {
  user = document.getElementById("user").value;
  pass = document.getElementById("pass").value;
  if (user == "admin" && pass == "password") {
    document.getElementById("demo").innerHTML = "You are now Admin!";
  }
}
</script>

</body>
</html>
```

Real programing language:

- Variables
- Logic statment (if, else ...)
- Loops
- Functions
- Classes
- ...

Goal:

- Dynamic web pages

PHP - MySQL

PHP & SQL

```php
...
<?php
    echo "Hi, I'm a PHP script!";
    phpinfo();
    if (strpos($_SERVER['HTTP_USER_AGENT'], 'MSIE') !== FALSE) {
        echo 'You are using Internet Explorer.<br />';
    }
    echo "Hi $_POST['name']; ?>. You are $_POST['age'] years old."

    $query = sprintf("SELECT message FROM Messages WHERE author='%s'",
    $_POST['name']);

    $result = mysql_query($query);
    echo "Result: $result"
?>
...
```

This code is stored and executed on the server.

Cookies

__Secure-3PAPISID: "NagwDTgZJrDGc8Up/ATSDEmhGVL9GkFSsx"
Created: "Sat, 25 Sep 2021 13:47:49 GMT"
Domain: ".youtube.com"
Expires / Max-Age: "Mon, 25 Sep 2023 13:47:49 GMT"
HostOnly: false
HttpOnly: false
Last Accessed: "Tue, 28 Sep 2021 19:18:01 GMT"
Path: "/"
SameSite: "None"
Secure: true
Size: 51

Example of a cookie

Faille Injection SQL

```php
...
<?php
  $sql = "SELECT * FROM Users
    WHERE user=$_POST['user'] AND pass=$_POST['pass']";
?>
...
```

Users

user	pass
admin	???
alex	password

POST:

user=**alex**&pass=**password**

SELECT * FROM Users WHERE user="alex" AND pass="password";

OK

Faille Injection SQL

```php
...
<?php
  $sql = "SELECT * FROM Users
    WHERE user=$_POST['user'] AND pass=$_POST['pass']";
?>
...
```

Users

user	pass
admin	???
alex	password

POST:

user=”&pass=**password**

SELECT * FROM Users WHERE user=”””” AND pass=”password”;

ERROR

Faille Injection SQL

```
...
<?php
  $sql = "SELECT * FROM Users
    WHERE user=$_POST['user'] AND pass=$_POST['pass']";
?>
...
```

Users

user	pass
admin	???
alex	password

POST:

user=**"; --**&pass=**password**

SELECT * FROM Users WHERE user=""; --" AND pass="password";

Return nothing

Faille Injection SQL

```
...
<?php
  $sql = "SELECT * FROM Users
    WHERE user=$_POST['user'] AND pass=$_POST['pass']";
?>
...
```

Users

user	pass
admin	???
alex	password

POST:

user=**admin”; --**&pass=**password**

SELECT * FROM Users WHERE user=”admin”; --” AND pass=”password”;

SUCCES

Faille XSS

```
...
<p id="feedback">Please, feel free to leave a commentary</p>
<input id="input" type="text">
<button id="btn" onclick="postCommentary()">Send</button>

<script>
  function postCommentary() {
    commentary = document.getElementById('input').value;
    document.getElementById("feedback").innerHTML =  commentary;
  }
</script>
...
```

Vulnerable XSS script

Payload:

- <script>...</script>
-
- ...

Goal:

- Grab the admin cookie, by redirecting to a malicious website
- ...