

07 – Injections XSS

Thibaut HENIN
www.arsouyes.org

Web technologies

HTML, Javascript, ...

World Wide Web...

Were the magic happens

The screenshot shows the homepage of the Université Bretagne Sud website. The browser address bar displays the URL <https://www.univ-ubs.fr/fr/index.html>. The website header includes the logo for Université Bretagne Sud (ubs) and a search bar with the text "L'université en pratique". A navigation menu lists various sections: L'UNIVERSITÉ, FORMATION, RECHERCHE, INTERNATIONAL, VIE DES CAMPUS, ENTREPRISES & INSTITUTIONS, and ACTUALITÉS.

The main banner features the text "sport" and "Science" in a stylized font, with the headline "Fête de la Science - Nuit de la Scien..." and the sub-headline "L'Université ouvre ses portes pour une découvert...". A "Lire la suite" button is visible below the banner.

Below the banner, there are two featured articles under the heading "Chroniques UBS":

- jeux : la fin d'une mode ou le début d'une nouvelle ère ?**
LEGO - Recherche & Innovation
: publié sur The Convers...
- Rentrée 2**
Université E
Si les inscript

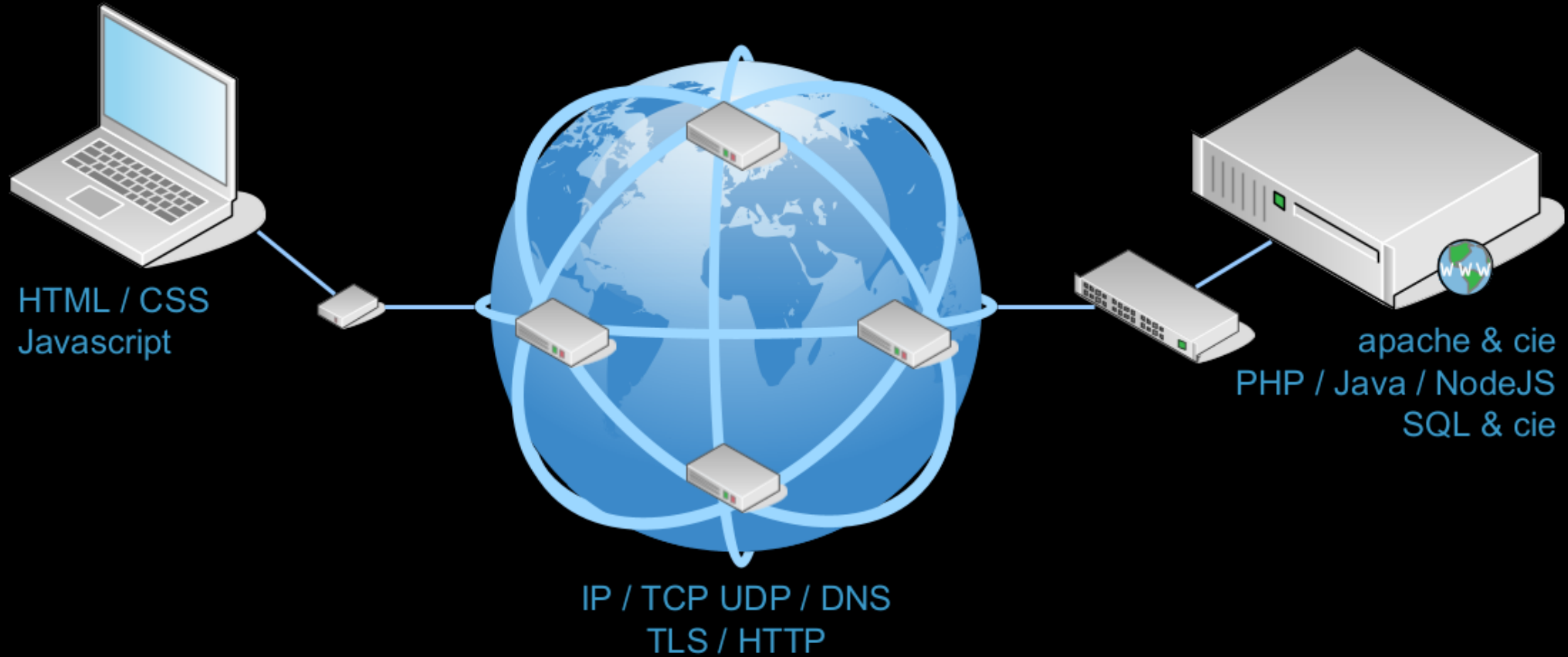
At the bottom, there is an "Agenda" section with the following events:

- 10/10**
Journée d'étude Médiévalismes et Orientali...
Faculté lettres & sciences humaines - Laboratoire HCTI
Rendez-vous le 10 octobre à 10h à l'Université
- 10/10**
Fête de la Science - Planète conférences - Vannes - Co...
Culture - Recherche & Innovation
Conférence de Léo Gerville-Réache, le 10 octobre à Vannes.

A "Rechercher des formations" button is located in the bottom right corner.

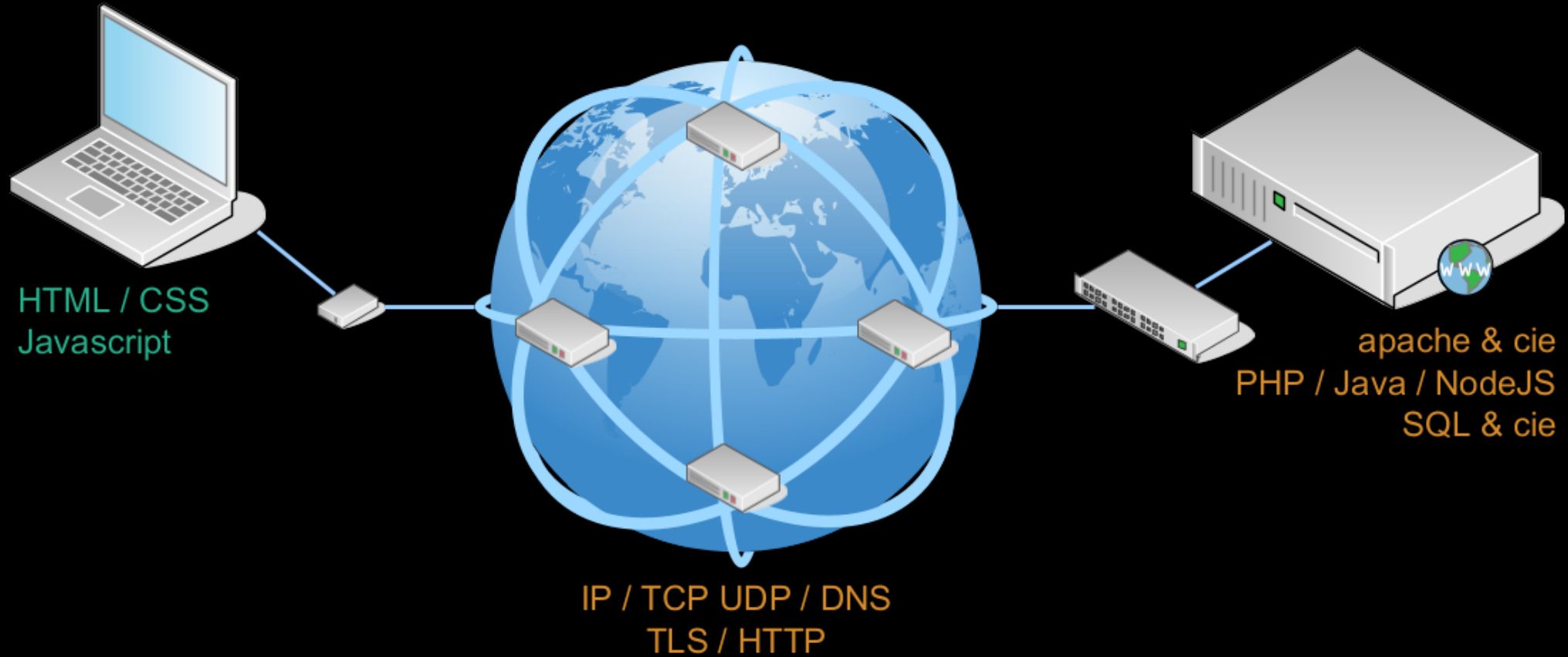
Network protocols

How the magic happens



Network protocols

How the magic happens



HTML / CSS / Javascript

Fastest introduction ever

HTML Basis

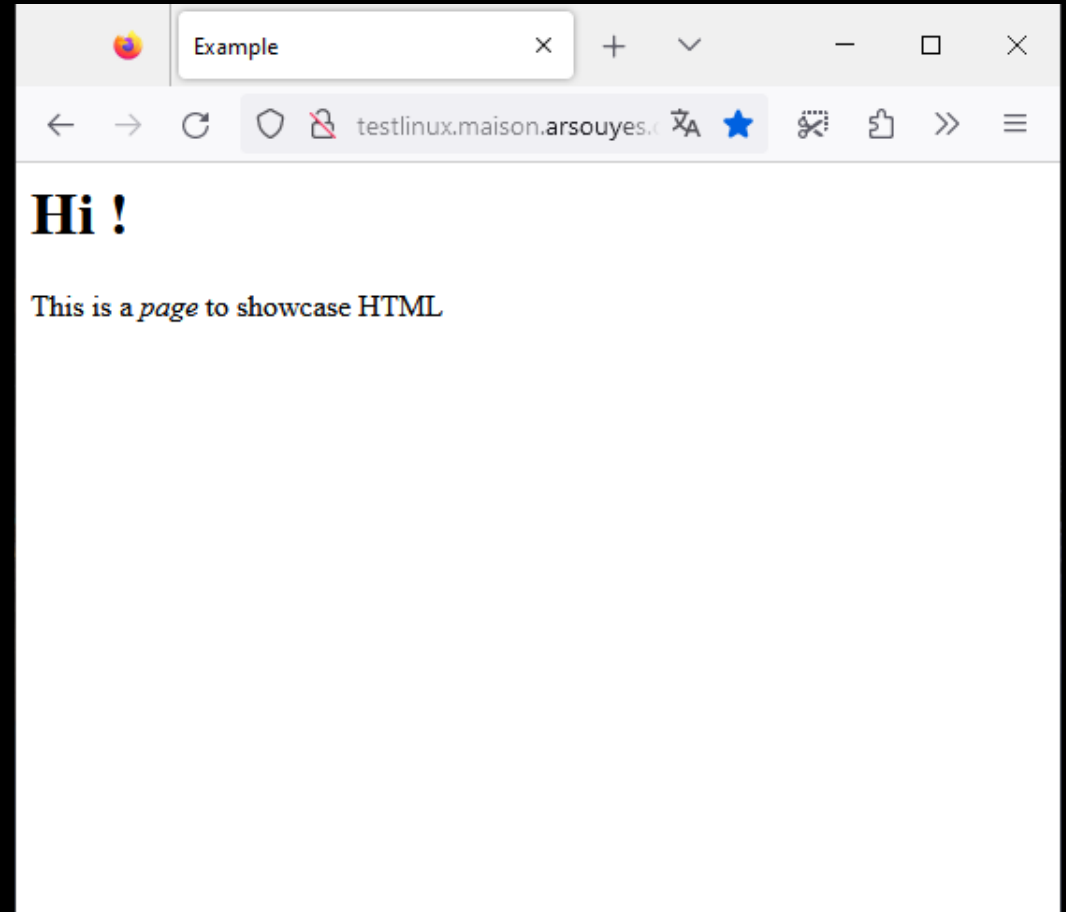
```
<html lang="en">
  <head>
    <title>Example</title>
  </head>

  <body>

    <h1>Hi !</h1>

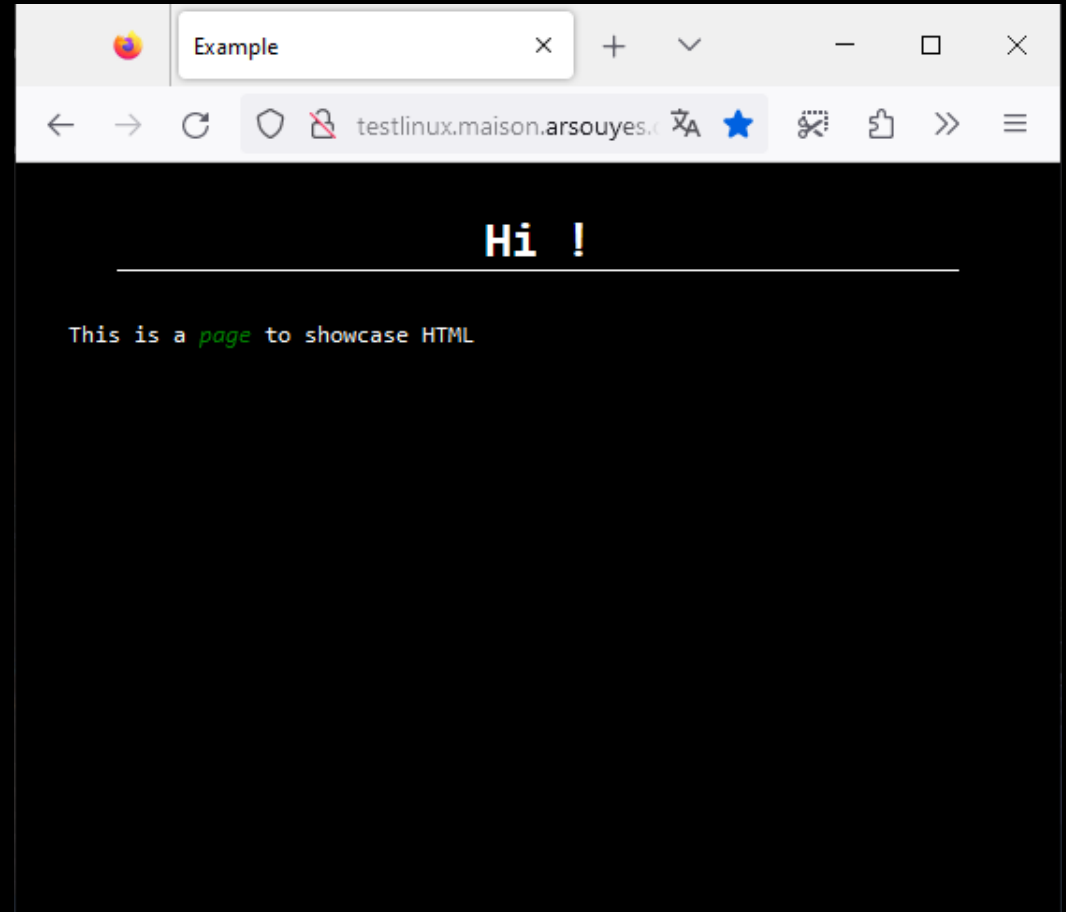
    <p>This is a
      <em>page</em>
      to showcase HTML</p>

  </body>
</html>
```



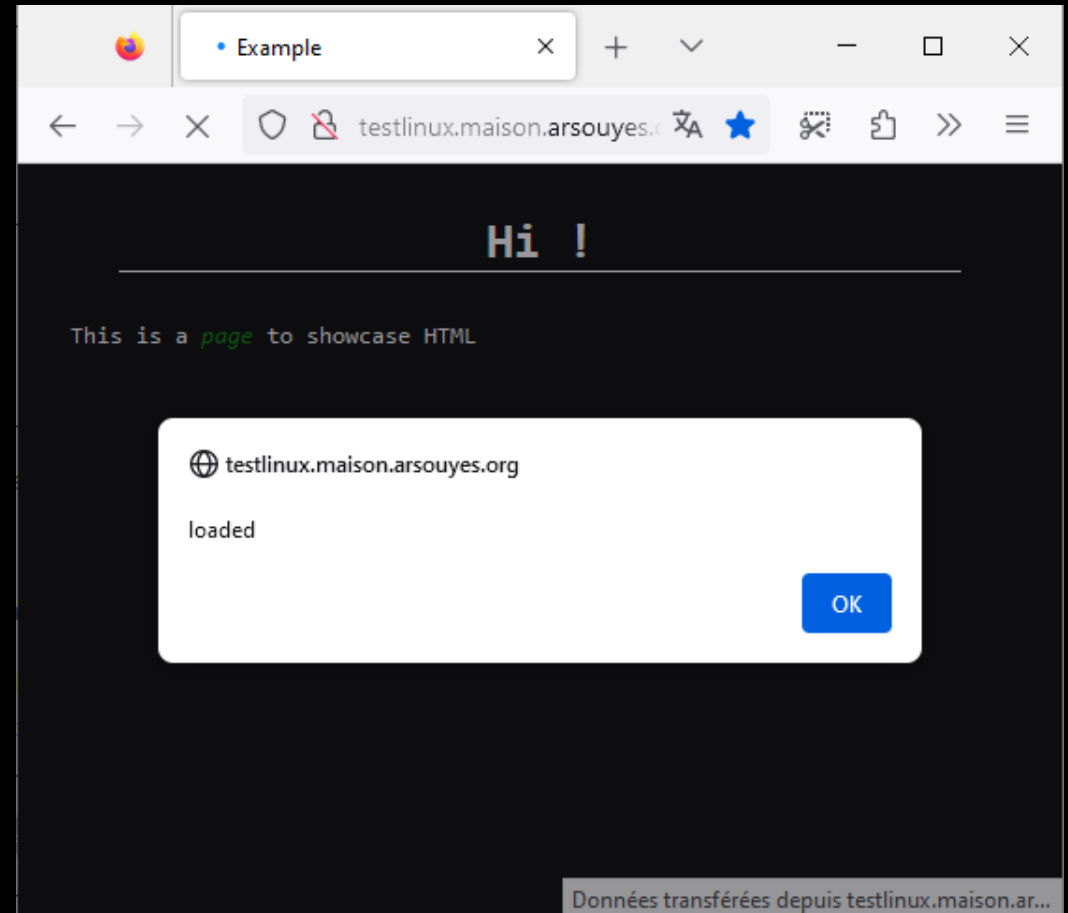
CSS Basis

```
body {  
  background-color: black ;  
  color: white ;  
  font-family: monospace ;  
  margin: 0 auto 0 auto ;  
  width: 90% ;  
}  
h1 {  
  margin: 1em ;  
  text-align: center ;  
  border-bottom: solid 1px ;  
}  
em {  
  color: green ;  
}
```



JS Basis

```
window.onload =  
  function() {  
    alert("loaded") ;  
  } ;
```



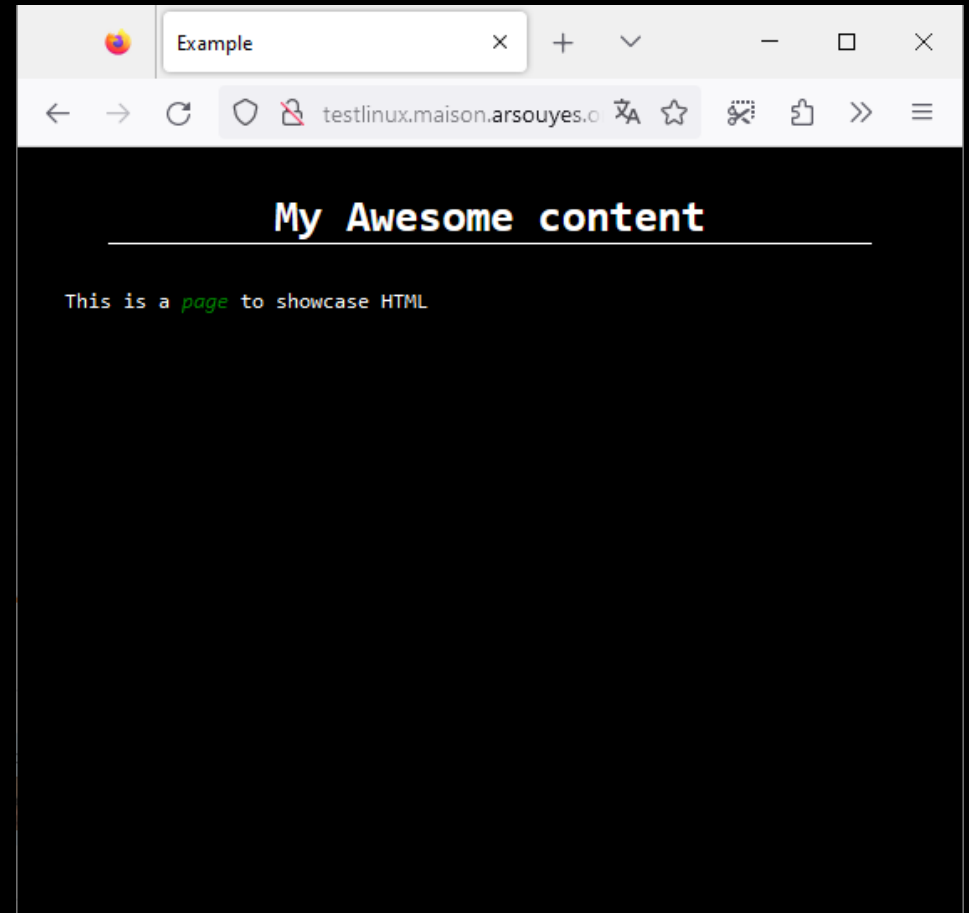
Ajax Basis

Main.js

```
window.onload = function() {  
  var xhttp = new XMLHttpRequest();  
  xhttp.onreadystatechange = function() {  
    if (this.readyState == 4  
        && this.status == 200) {  
      document.getElementsByTagName("h1")[0].innerHTML  
        = this.responseText;  
    }  
  };  
  xhttp.open("GET", "Test.txt", true);  
  xhttp.send();  
}
```

Test.txt

My Awesome content



XSS Reflected

Cross Site Scripting

Example of Vulnerable Application

```
<html lang="en">
  <head>
    <title>Example</title>
    <link href="style.css" rel="stylesheet" />
    <script type="text/javascript" src="main.js"></script>
  </head>
  <body>
    <h1>Hi !</h1>
    <p>Hello <em>
      <?php echo $_GET["user"] ?? "you" ; ?>
    </em>!</p>
  </body>
</html>
```

Example of Vulnerable Application

```
<html lang="en">
  <head>
    <title>Example</title>
    <link href="style.css" rel="stylesheet" />
    <script type="text/javascript" src="main.js"></script>
  </head>
  <body>
    <h1>Hi !</h1>
    <p>Hello <em>
      <?php echo $_GET["user"] ?? "you" ; ?>
    </em>!</p>
  </body>
</html>
```

http://example.com/

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

http://example.com/

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo null ?? "you" ; ?>  
</em>!</p>
```

http://example.com/

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo null ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "you" ; ?>  
</em>!</p>
```


http://example.com/

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo null ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  You  
</em>!</p>
```

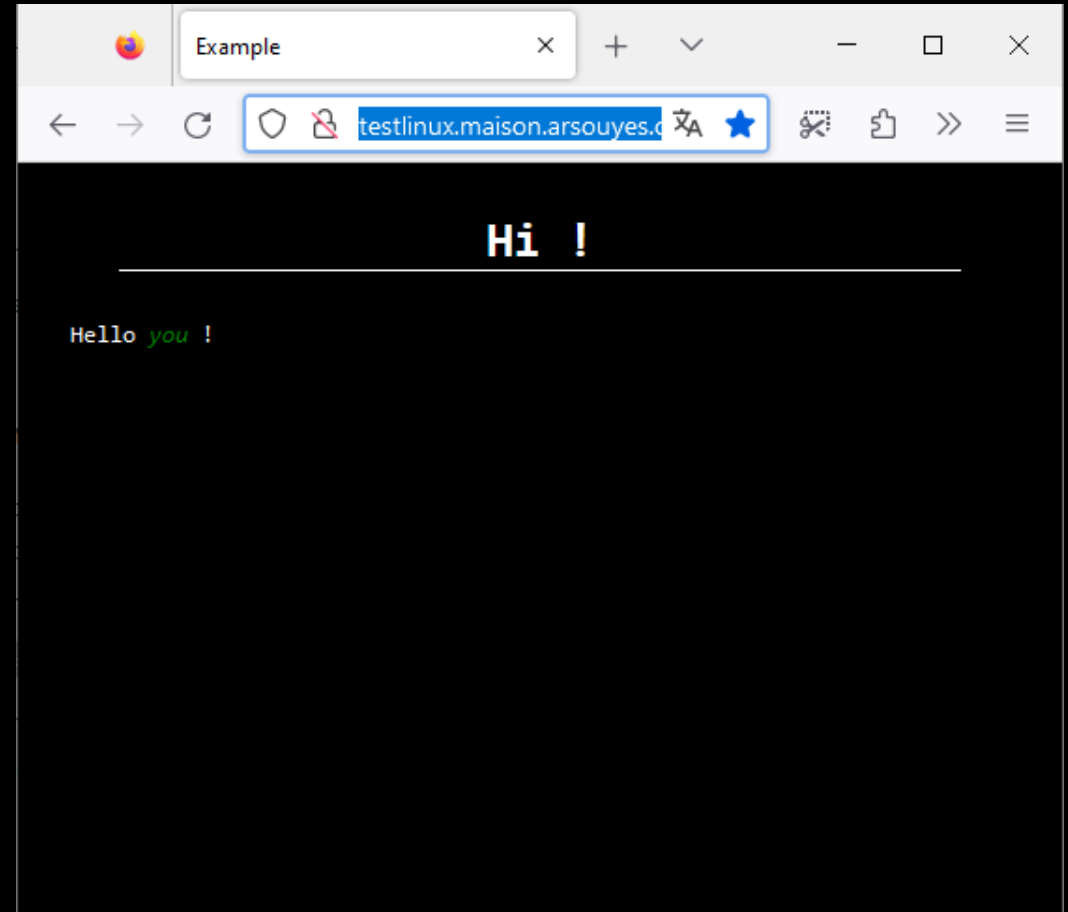
http://example.com/

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo null ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  You  
</em>!</p>
```



<http://example.com/?user=tbowan>

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

http://example.com/?user=tbowan

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "tbowan" ?? "you" ; ?>  
</em>!</p>
```

http://example.com/?user=tbowan

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "tbowan" ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "tbowan" ; ?>  
</em>!</p>
```

http://example.com/?user=tbowan

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "tbowan" ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "tbowan" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  tbowan  
</em>!</p>
```

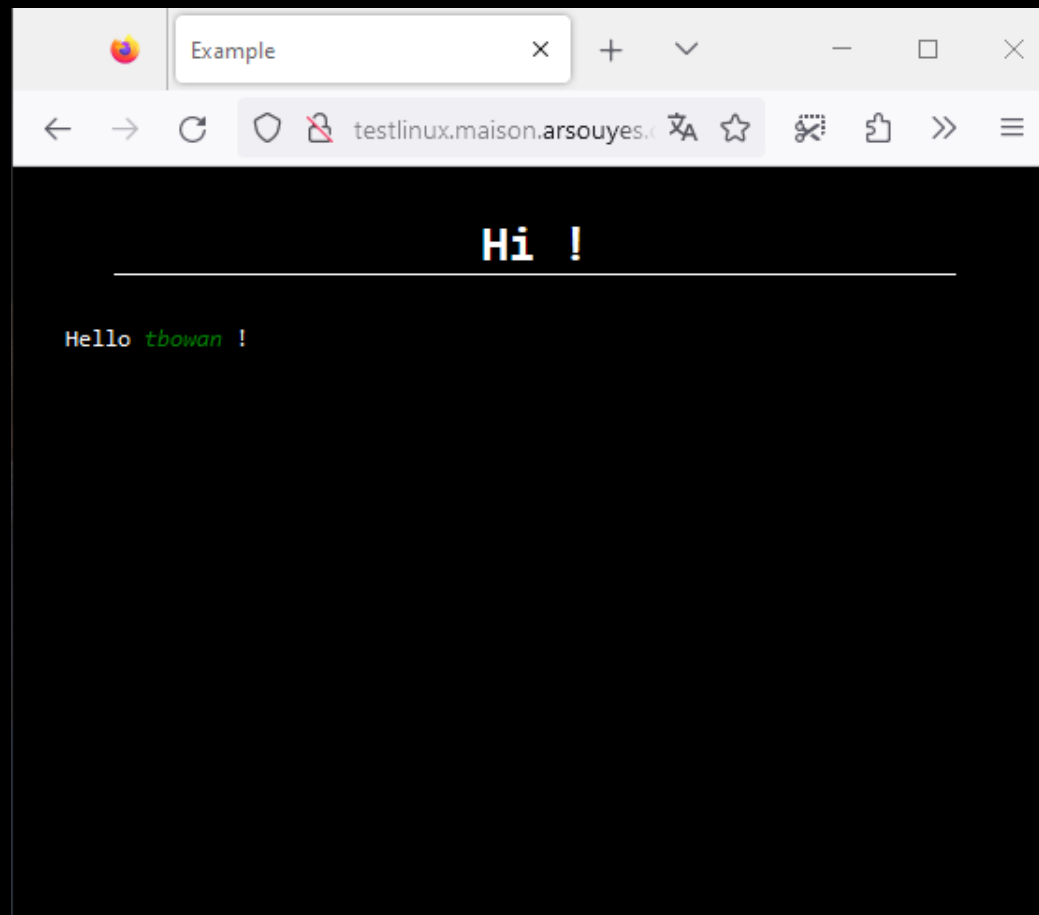
http://example.com/?user=tbowan

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "tbowan" ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "tbowan" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  tbowan  
</em>!</p>
```



`http://example.com/?user=<h1>tbowan</h1>`

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```


`http://example.com/?user=<h1>tbowan</h1>`

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "<h1>tbowan</h1>" ?? "you";?>  
</em>!</p>
```

`http://example.com/?user=<h1>tbowan</h1>`

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "<h1>tbowan</h1>" ?? "you";?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "<h1>tbowan</h1>" ; ?>  
</em>!</p>
```

`http://example.com/?user=<h1>tbowan</h1>`

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "<h1>tbowan</h1>" ?? "you";?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "<h1>tbowan</h1>" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <h1>tbowan</h1>  
</em>!</p>
```

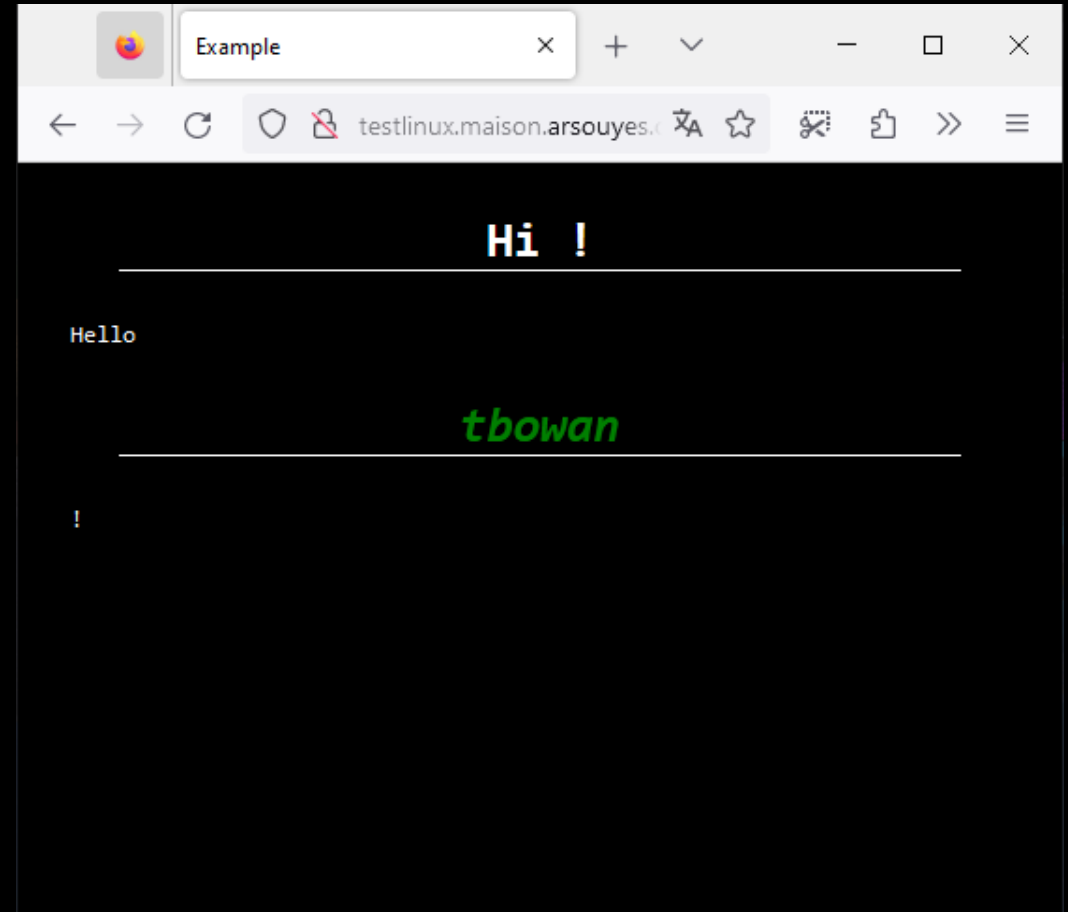
`http://example.com/?user=<h1>tbowan</h1>`

```
<p>Hello <em>  
  <?php echo $_GET["user"] ?? "you" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "<h1>tbowan</h1>" ?? "you";?>  
</em>!</p>
```

```
<p>Hello <em>  
  <?php echo "<h1>tbowan</h1>" ; ?>  
</em>!</p>
```

```
<p>Hello <em>  
  <h1>tbowan</h1>  
</em>!</p>
```



Better injection :

```
http://example.com/?user=
  tbowan</em>!</p>
  <h1>Congratulation</h1>
  <p>Click
    <a href="evil.com">HERE</a>
    to win your price
  <em>
```

Better injection :

```
http://example.com/?user=
tbowan</em>!</p>
<h1>Congratulation</h1>
<p>Click
    <a href="evil.com">HERE</a>
    to win your price
</em>
```

```
<p>Hello <em>
  <?php echo "tbowan</em>..pricece<em>" ;?>
</em>!</p>
```

Better injection :

```
http://example.com/?user=
tbowan</em>!</p>
<h1>Congratulation</h1>
<p>Click
    <a href="evil.com">HERE</a>
    to win your price
<em>
```

```
<p>Hello <em>
  <?php echo "tbowan</em>...pricece<em>" ;?>
</em>!</p>
```

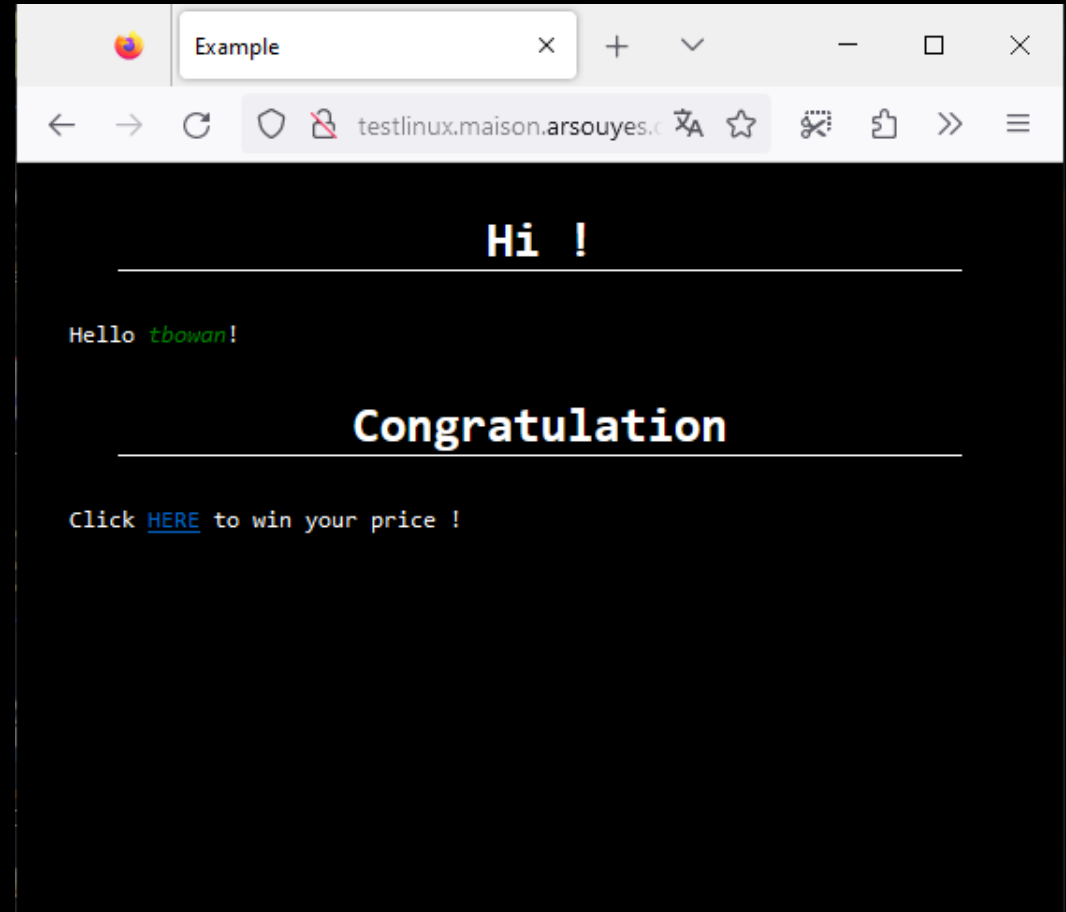
```
<p>Hello <em>
  tbowan</em>!</p>
<h1>Congratulation</h1>
<p>Click
    <a href="evil.com">HERE</a>
    to win your price
<em>
</em>!</p>
```

Better injection :

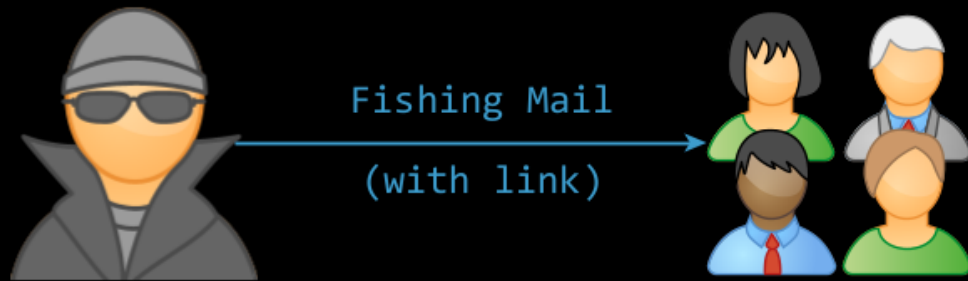
```
http://example.com/?user=
tbowan</em>!</p>
<h1>Congratulation</h1>
<p>Click
  <a href="evil.com">HERE</a>
  to win your price
<em>
```

```
<p>Hello <em>
  <?php echo "tbowan</em>...pricece<em>" ;?>
</em>!</p>
```

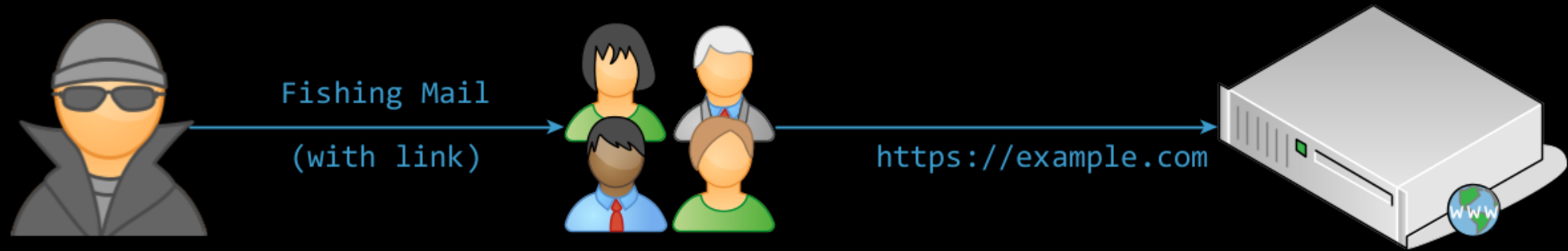
```
<p>Hello <em>
  tbowan</em>!</p>
<h1>Congratulation</h1>
<p>Click
  <a href="evil.com">HERE</a>
  to win your price
<em>
</em>!</p>
```



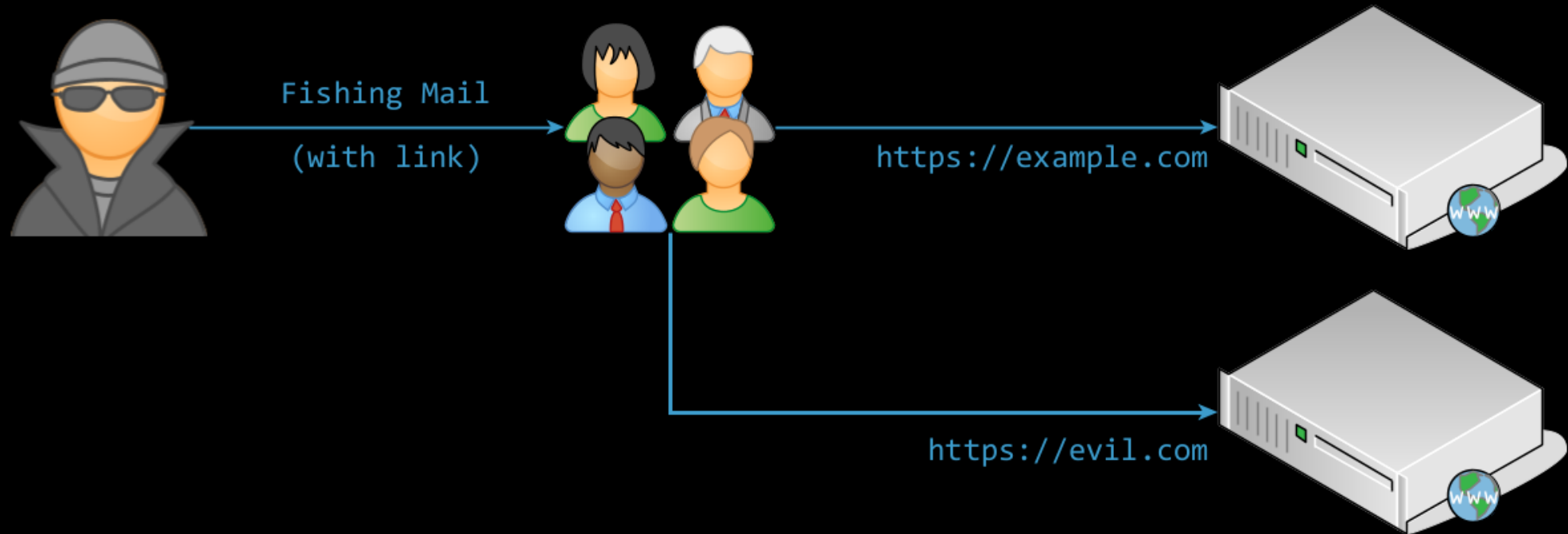
Principle



Principle



Principle



JS Injection

a.k.a. XSS – Cross Site Scripting

```
http://example.com/?user=  
  <script>  
    alert("Virus Detected");  
  </script>
```

JS Injection

a.k.a. XSS – Cross Site Scripting

```
http://example.com/?user=  
  <script>  
    alert("Virus Detected");  
  </script>
```

```
<p>Hello <em>  
  <?php echo "<script>...</script>" ;?>  
</em>!</p>
```

JS Injection

a.k.a. XSS – Cross Site Scripting

```
http://example.com/?user=  
  <script>  
    alert("Virus Detected");  
  </script>
```

```
<p>Hello <em>  
  <?php echo "<script>...</script>" ;?>  
</em>!</p>
```

```
<p>Hello <em>  
  <script>  
    alert("Virus Detected");  
  </script>  
</em>!</p>
```

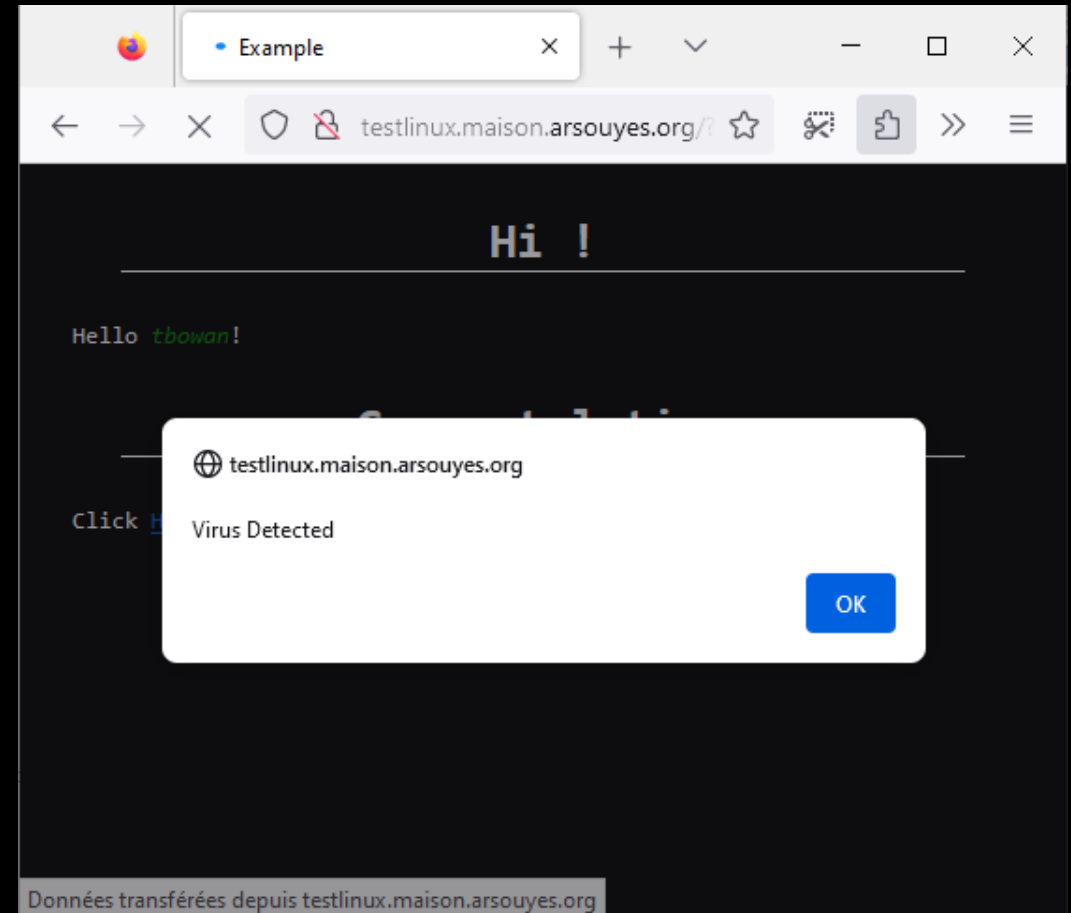
JS Injection

a.k.a. XSS – Cross Site Scripting

```
http://example.com/?user=  
  <script>  
    alert("Virus Detected");  
  </script>
```

```
<p>Hello <em>  
  <?php echo "<script>...</script>" ;?>  
</em>!</p>
```

```
<p>Hello <em>  
  <script>  
    alert("Virus Detected");  
  </script>  
</em>!</p>
```



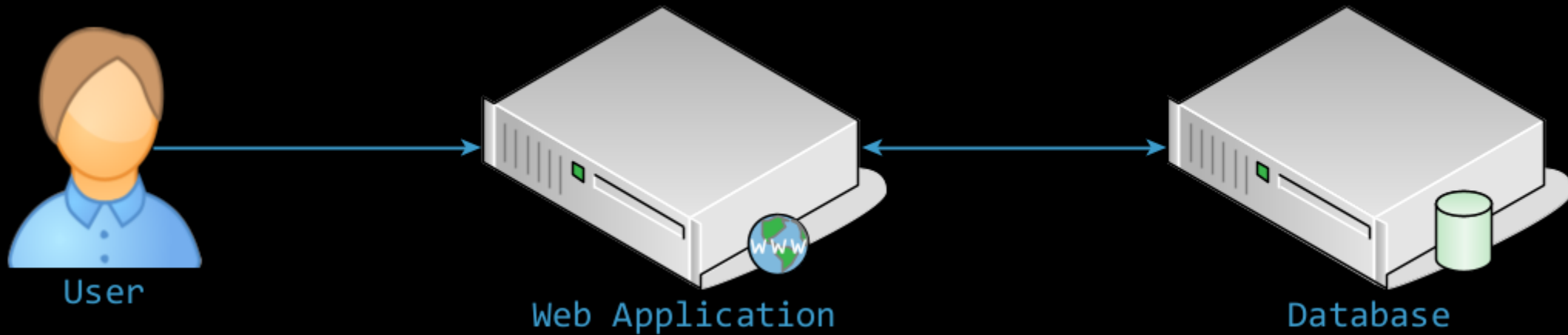
Problem :

Need to send a link to every victim

XSS - Stored

Cross Site Scripting

Persistent applications



Example of data creation

i.e. adding a blog comment

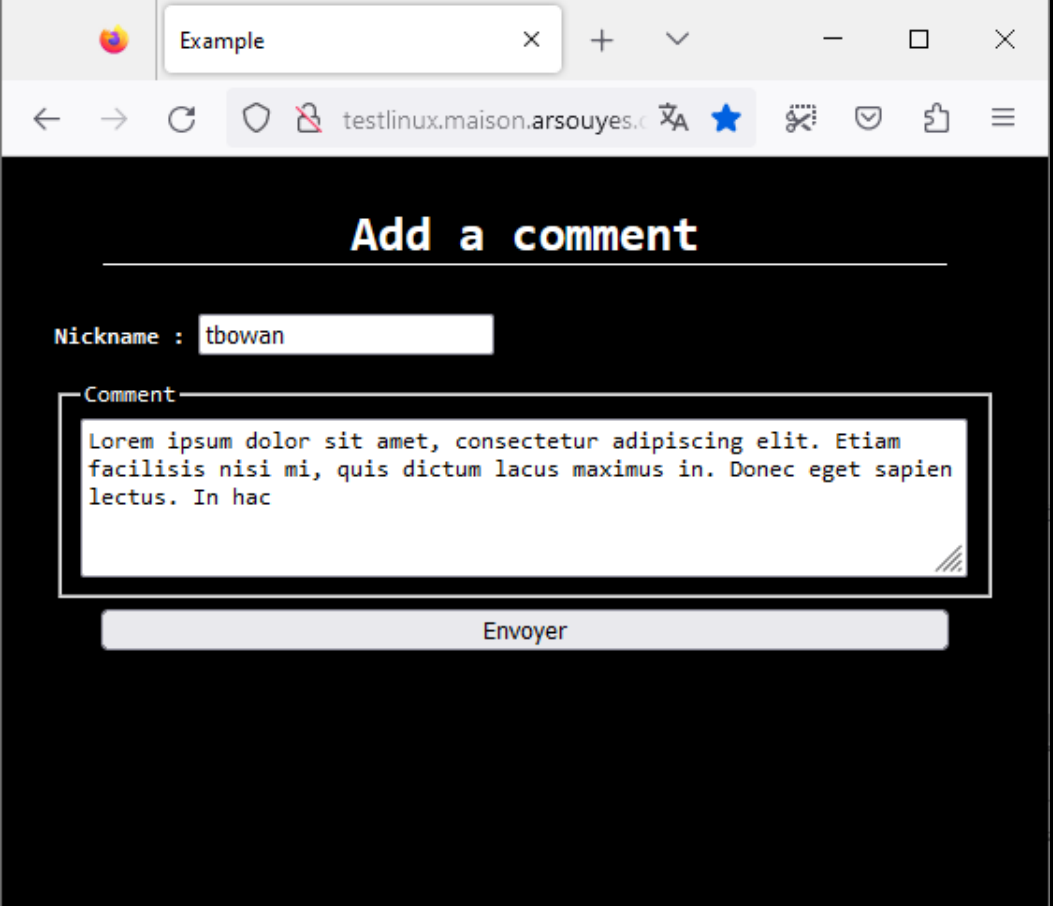
```
<h1>Add a comment</h1>
<form method="post"
      action="addComment.php">

<input type="hidden"
      name="article"
      value="123" />

<p><strong>Nickname :</strong>
  <input type="text" name="author" />
</p>

<fieldset><legend>Comment</legend>
  <textarea name="content"></textarea>
</fieldset>

<input type="submit" />
</form>
```



The screenshot shows a web browser window with the title 'Example'. The address bar contains 'testlinux.maison.arsouyes.c'. The page content is a form titled 'Add a comment'. The form has a 'Nickname' field with the value 'tbowan'. Below it is a 'Comment' field containing the text 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam facilisis nisi mi, quis dictum lacus maximus in. Donec eget sapien lectus. In hac'. At the bottom of the form is a button labeled 'Envoyer'.

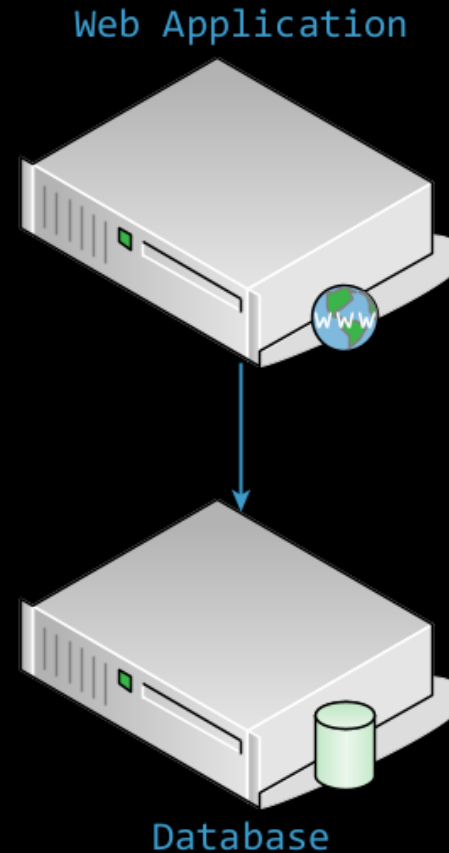
Example of data creation

i.e. adding a blog comment

```
<?php // Add Comment

$cmd = $pdo->prepare("
    . "insert into comment"
    . " (article, author, content)"
    . " values"
    . " (:article, :author, :content)"
);

$cmd->exec([
    "article" => $_POST["article"],
    "author"  => $_POST["author"],
    "content" => $_POST["content"]
]);
```



Example of database

Article	Author	content
...
123	Tbowan	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam facilisis nisi mi, quis dictum lacus maximus in. Donec eget sapien lectus. In hac
...

Example of data retrieval

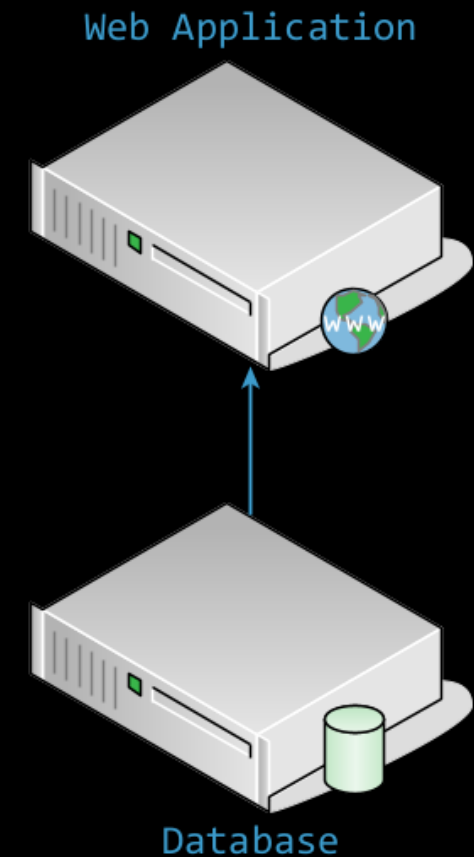
i.e. displaying a blog comment

```
<?php // show post

$cmd = $pdo->prepare("
    . "select * from comment"
    . " where article = :article"
    );

$stmt = $cmd->exec(["article" => $_GET["id"] ]);

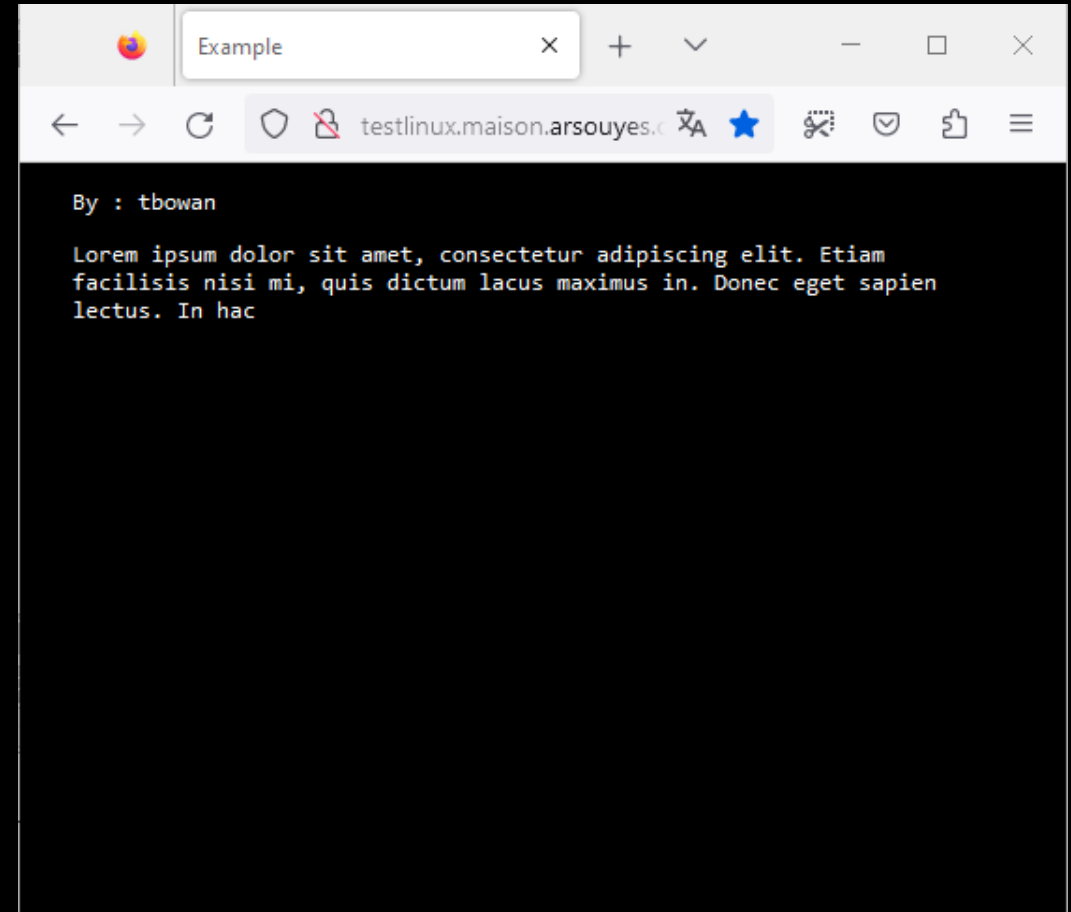
foreach ($stmt as $row) {
    echo '<div class="comment">' ;
    echo '<p>By : ' . $row["author"] . '</p>' ;
    echo $row["content"] ;
    echo '</div>' ;
}
}
```



Example of data retrieval

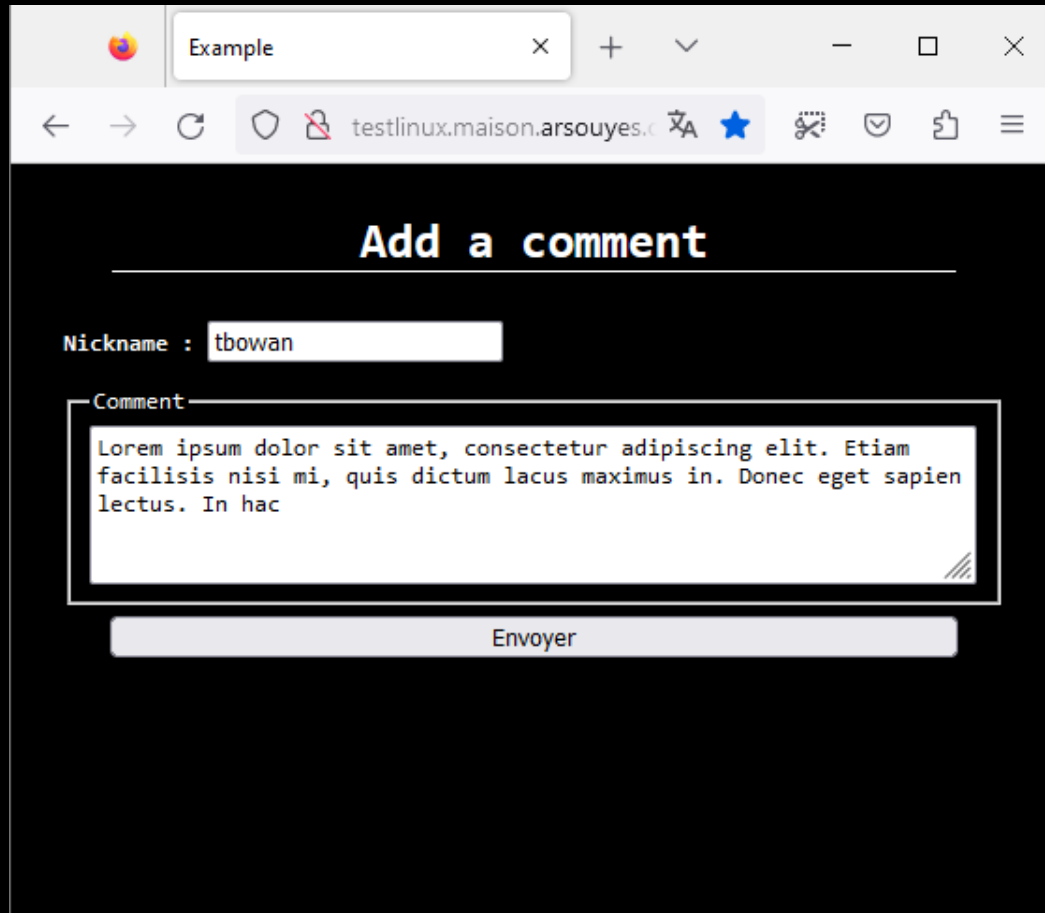
i.e. displaying a blog comment

```
<div class="comment">  
<p>By : tbowan</p>  
  Lorem ipsum dolor sit amet,  
consectetur adipiscing elit. Etiam  
facilisis nisi mi, quis dictum lacus  
maximus in. Donec eget sapien lectus.  
In hac  
</div>
```

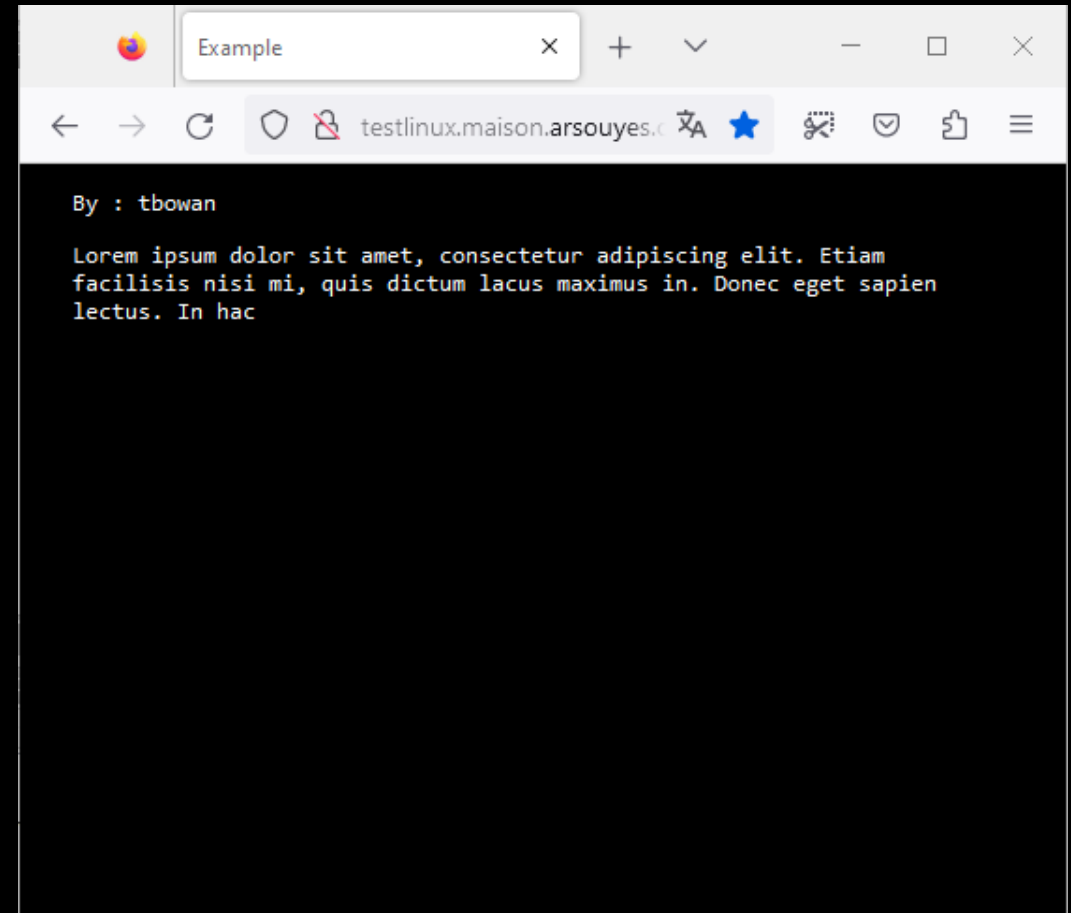


Example of interaction

i.e. adding a blog comment



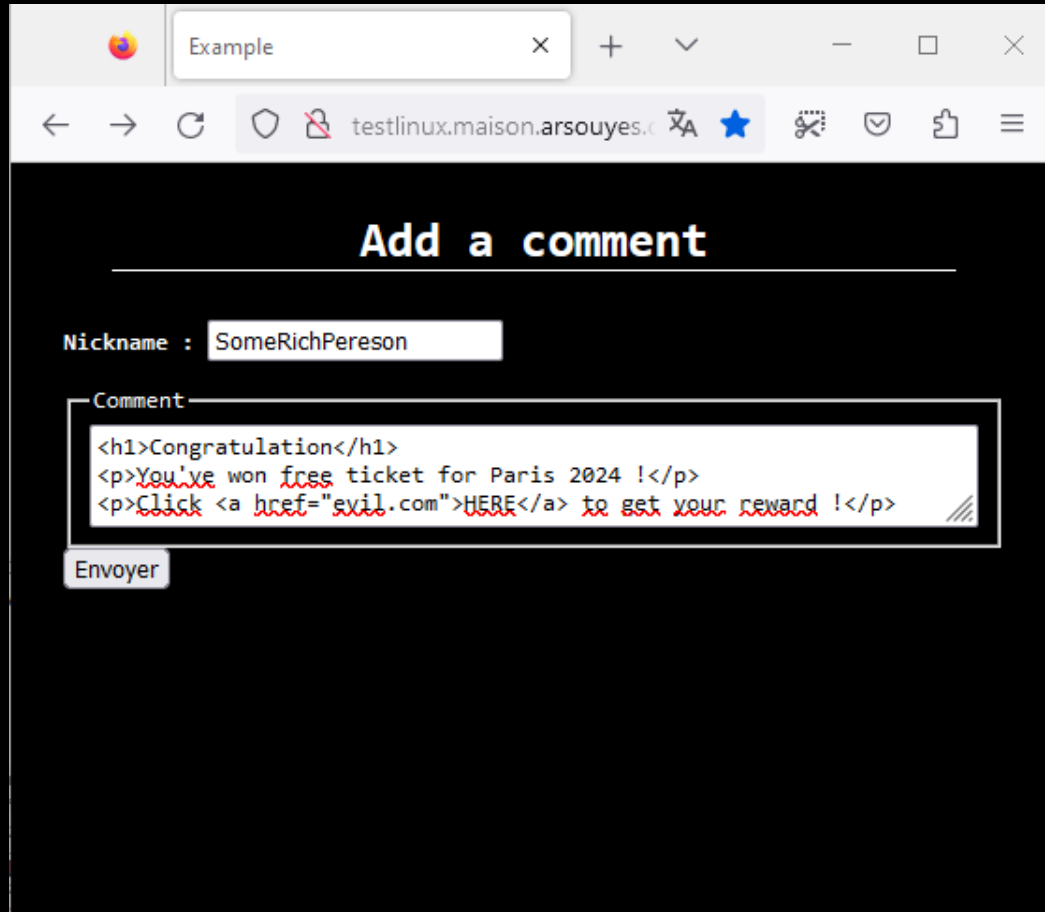
The screenshot shows a web browser window with the title 'Example'. The address bar contains 'testlinux.maison.arsouyes.c'. The main content area has a heading 'Add a comment' with a horizontal line underneath. Below the heading, there is a 'Nickname' label followed by a text input field containing 'tbowan'. Underneath that is a 'Comment' label followed by a large text area containing the text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam facilisis nisi mi, quis dictum lacus maximus in. Donec eget sapien lectus. In hac'. At the bottom of the form is a button labeled 'Envoyer'.



The screenshot shows the same web browser window after the comment has been submitted. The main content area now displays the text: 'By : tbowan' followed by the comment text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam facilisis nisi mi, quis dictum lacus maximus in. Donec eget sapien lectus. In hac'.

Example of persistent injection

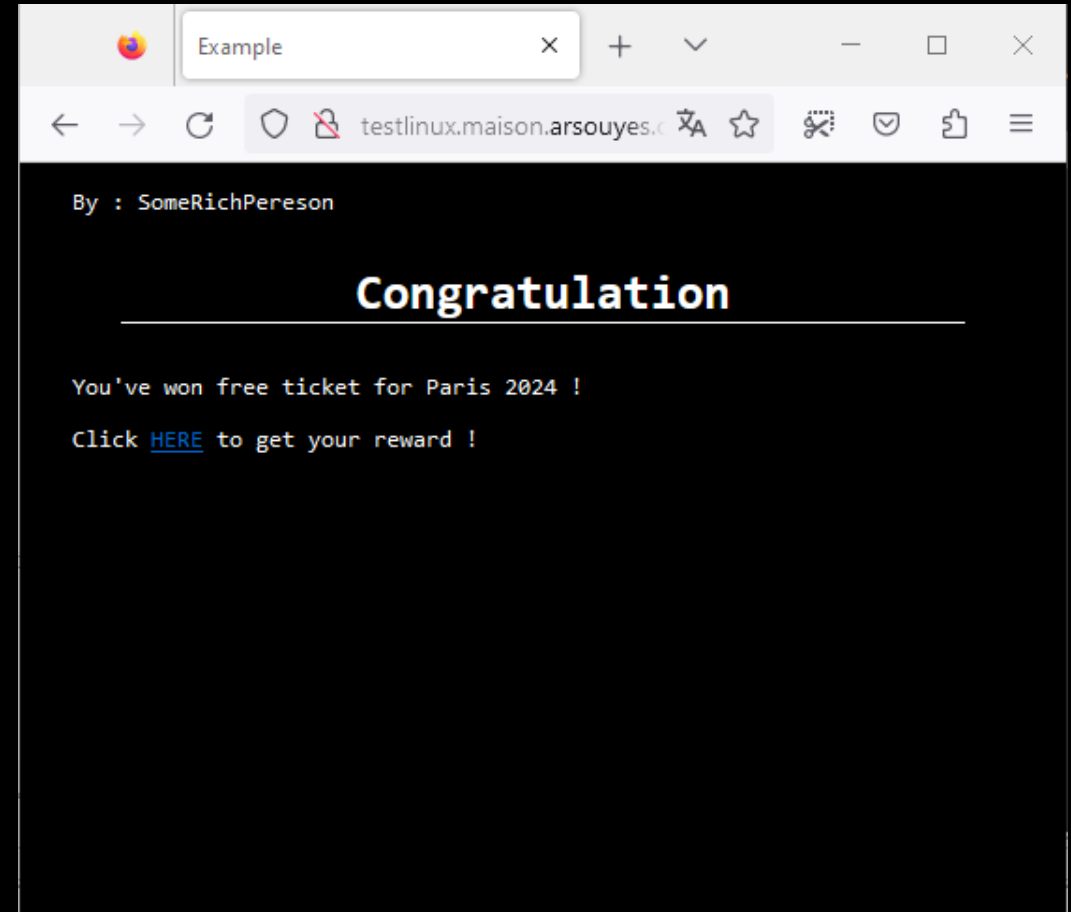
i.e. adding some fishing HTML



The screenshot shows a web browser window with the address bar containing "testlinux.maison.arsouyes.c". The page title is "Example". The main content area has a heading "Add a comment" underlined. Below the heading, there is a form with a "Nickname" field containing "SomeRichPereson" and a "Comment" text area. The text area contains the following HTML code:

```
<h1>Congratulation</h1>
<p>You've won free ticket for Paris 2024 !</p>
<p>Click <a href="evil.com">HERE</a> to get your reward !</p>
```

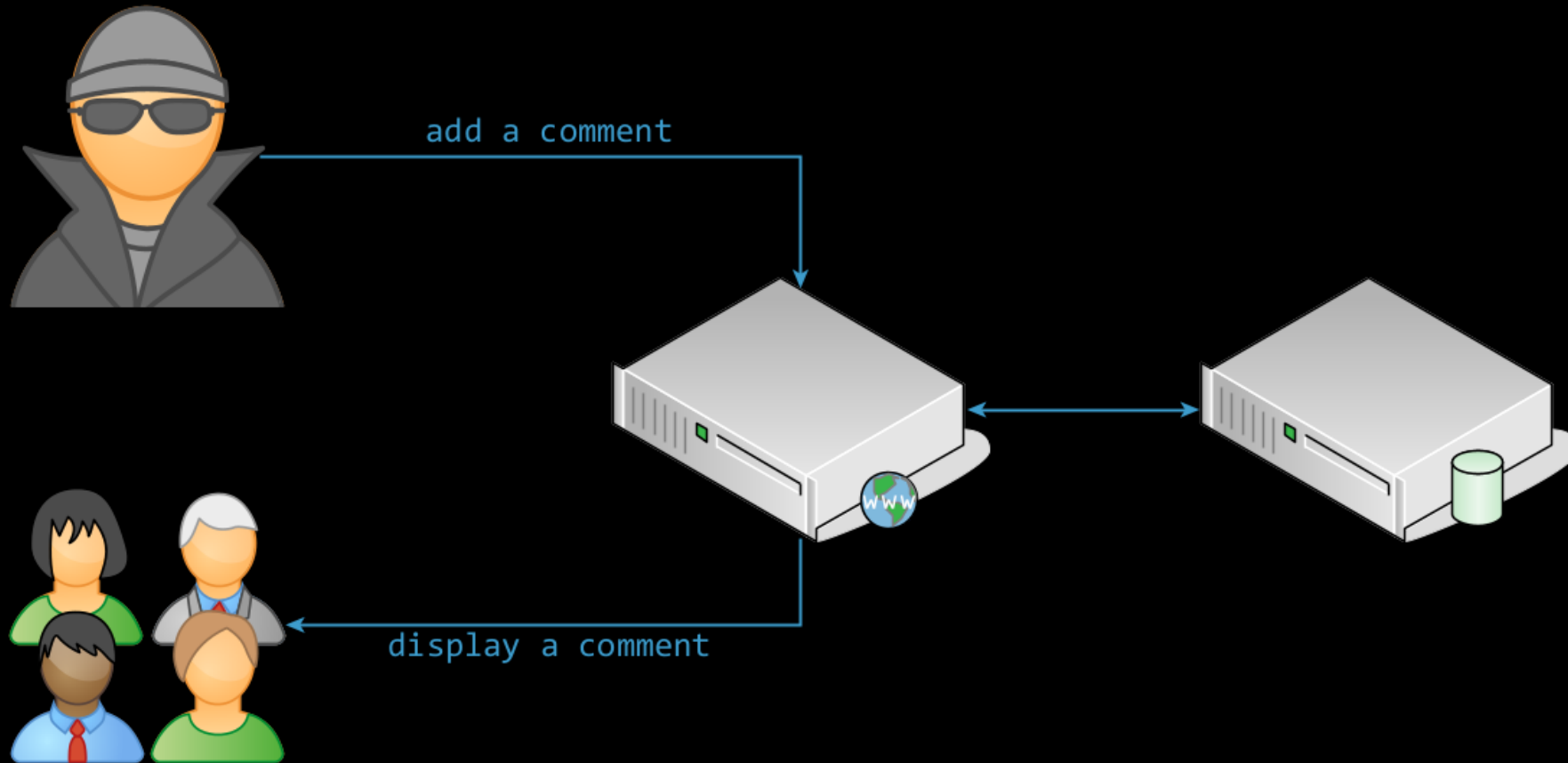
 Below the text area is an "Envoyer" button.



The screenshot shows the same web browser window after the comment is submitted. The page content now displays the rendered HTML: "By : SomeRichPereson" followed by a heading "Congratulation" underlined. Below the heading, the text reads: "You've won free ticket for Paris 2024 !" and "Click [HERE](#) to get your reward !".

Principle of stored XSS

Use vulnerable website to host your evil content



Risks

Information theft

(cookies, form data, ...)

Botnet

(relay for other attacks, crypto mining, ...)

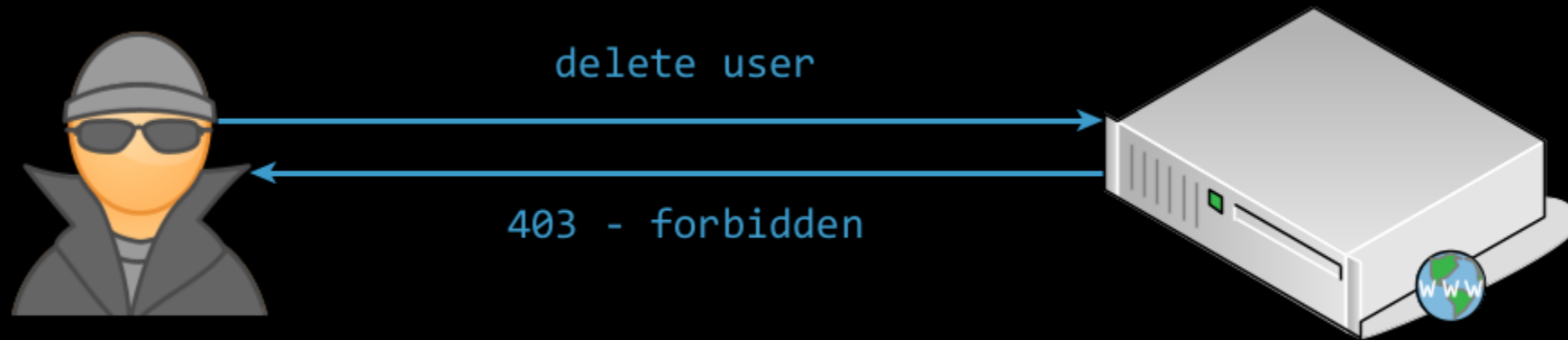
Request execution

(XSRF)

XSRF / CSRF

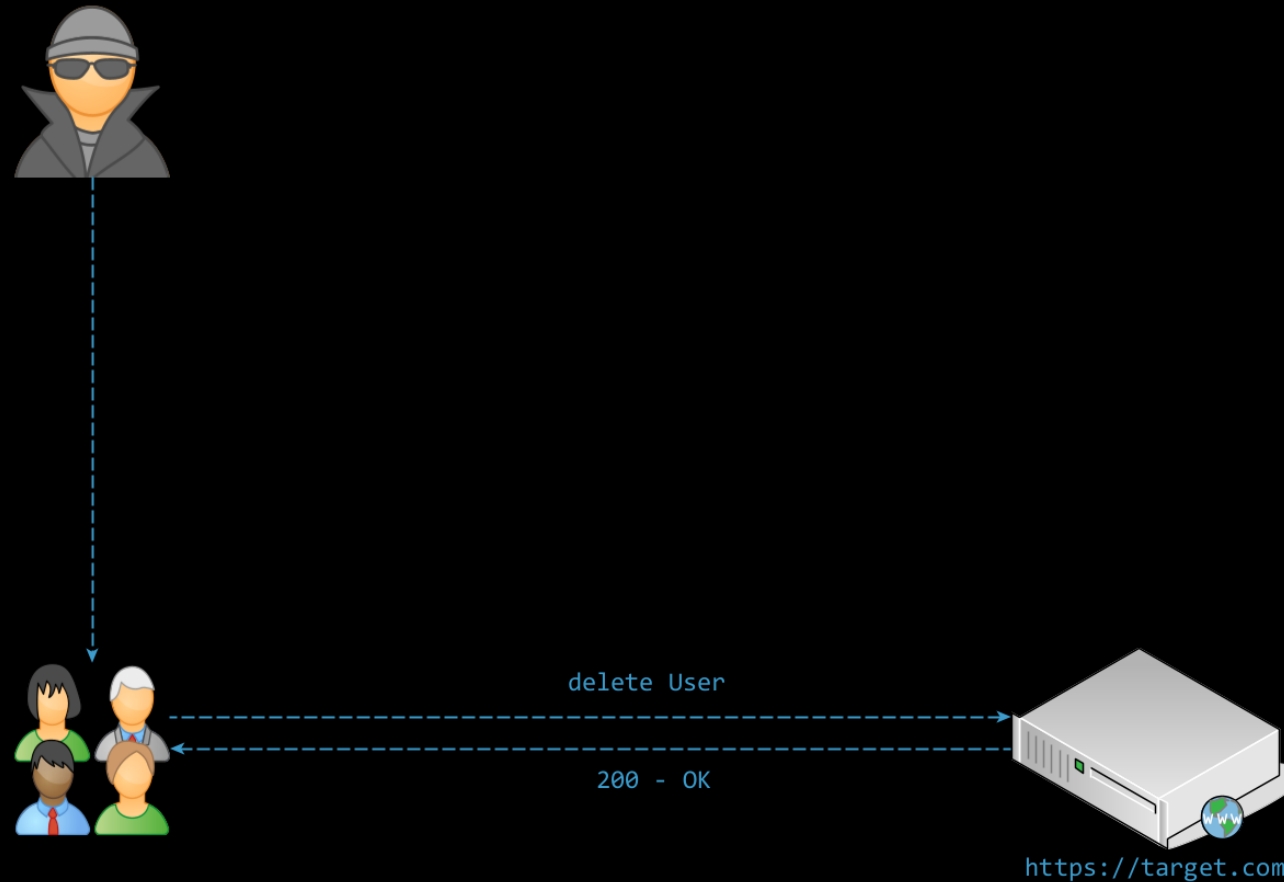
Cross Site Request Forgery

Principle protected feature



Principle

Need some admin to do the call



Principle

Exploit a third (vulnerable) party



<https://vulnerable.com>



delete User

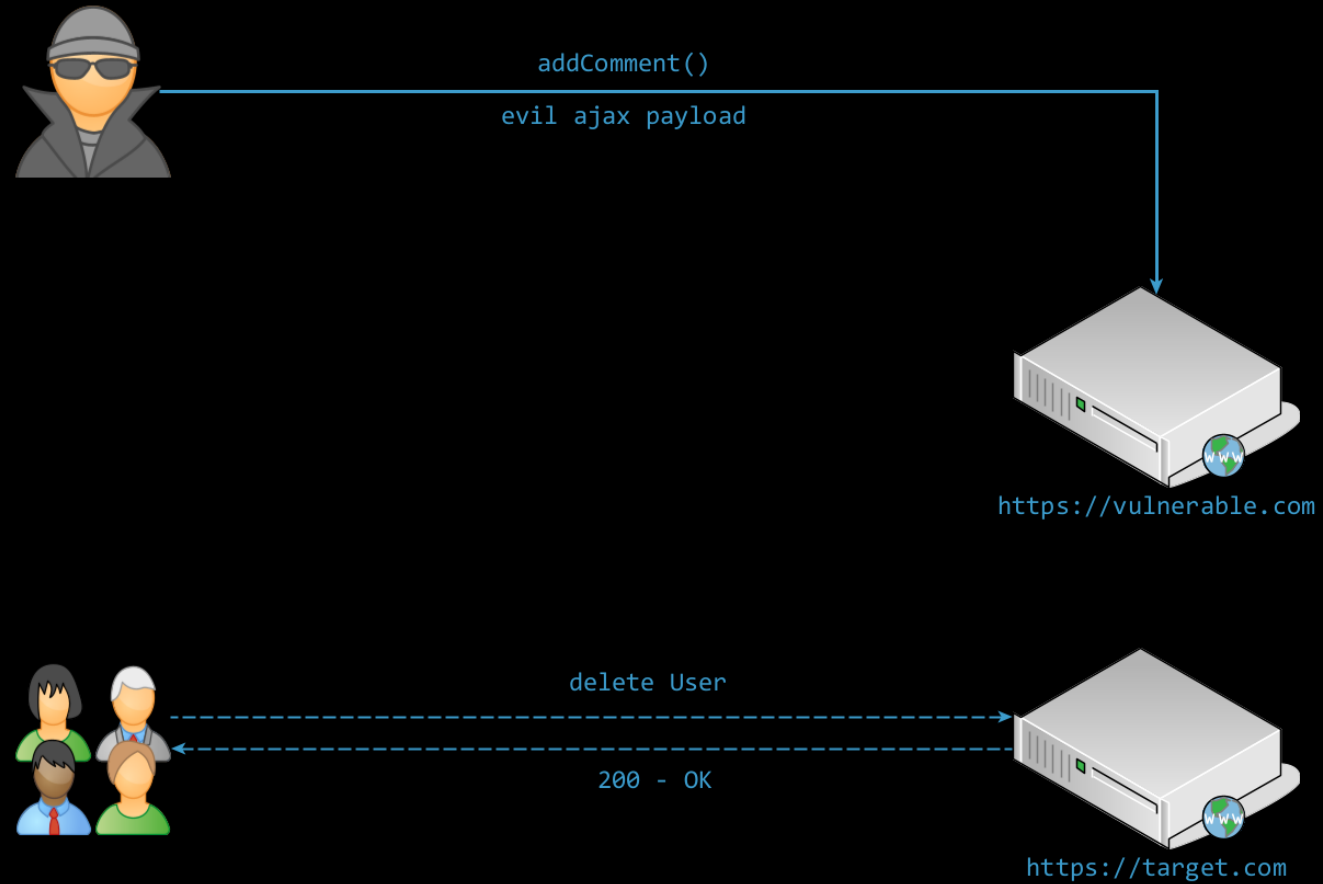
200 - OK



<https://target.com>

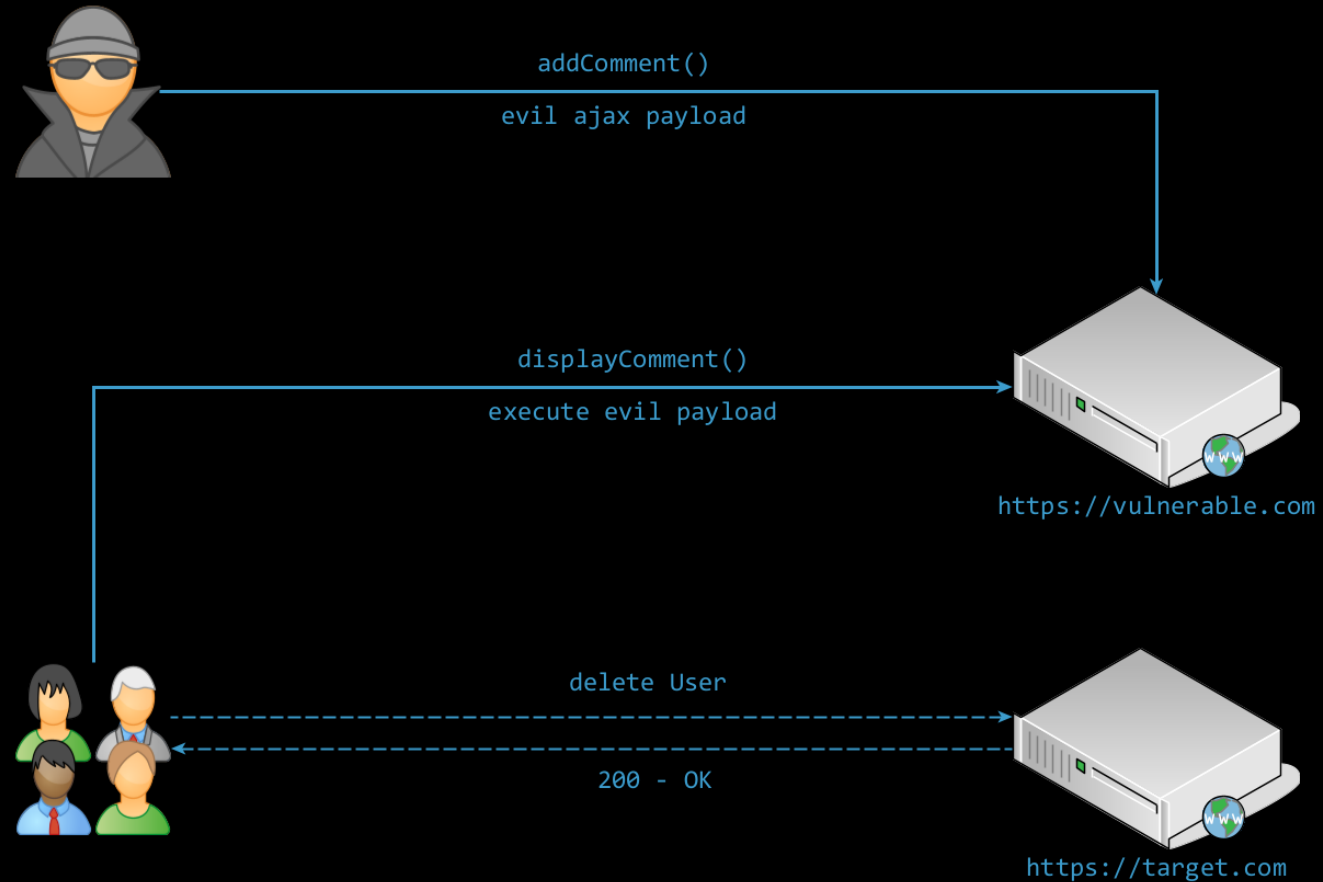
Principle

Inject Ajax that do the call

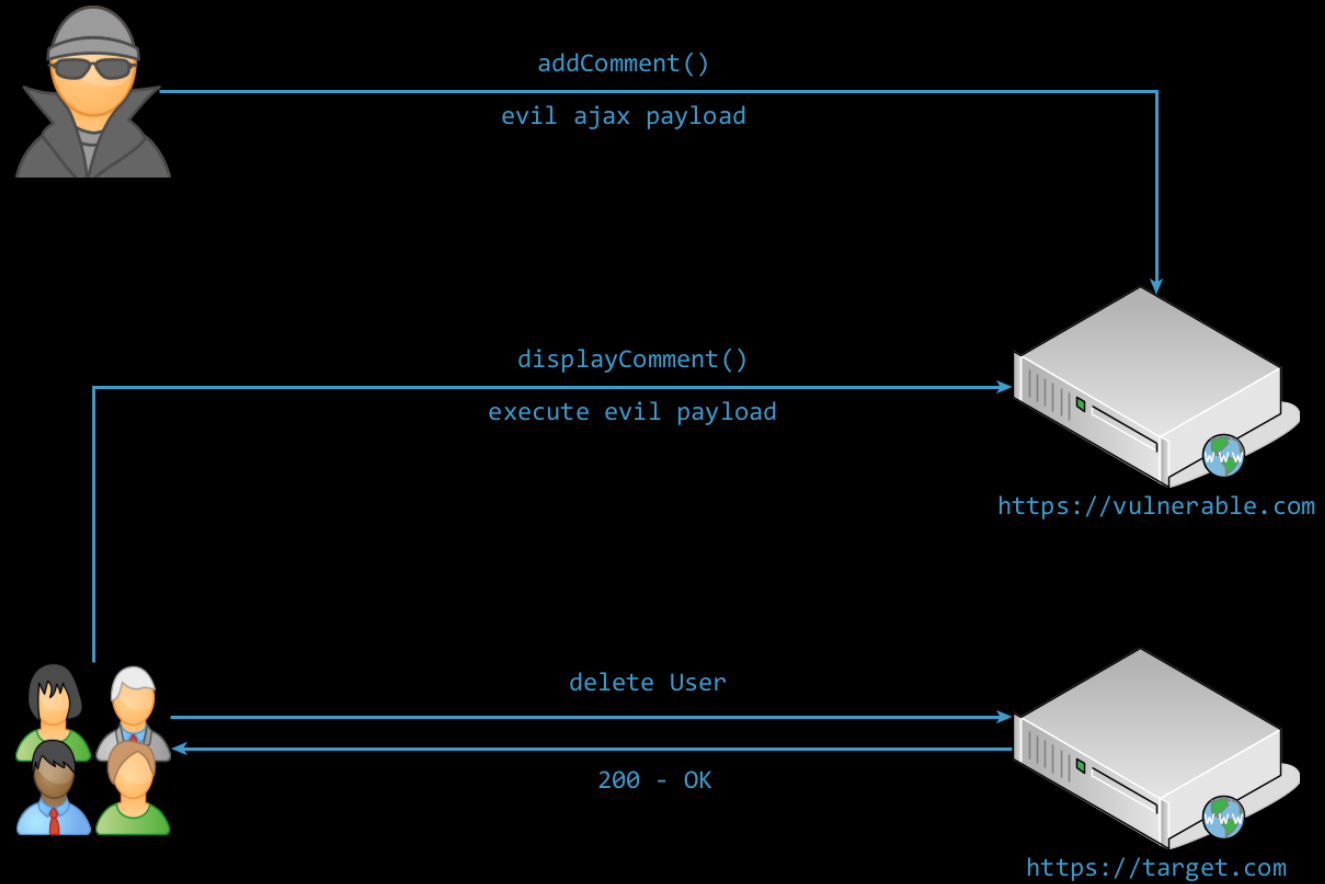


Principle

Wait some admin get the payload



Principle Enjoy



Protections

Where to fix it ?

Complementary layers

Client Side

- Javascript

Server Side

- PHP, Java, ...

Protocols

- Cookies, CORS, ...

Browsers

- Chrome, firefox,...

Third parties

- Libs, ...

Server side

PHP

Escape / Delete tags

`htmlspecialchars`, `htmlspecialchars`, `strip_tags`

Encode attributes

`urlencode`

Client side

Javascript

Escape / Delete tags

Depends on frameworks

Use html5 <template>

`textContent` vs. `innerHTML`

Cookies

Protocol (app + browsers)

Expires

(validity duration)

Secure

(transmit only if TLS)

Domain

(validity on domain name)

HttpOnly

(only send to server)

Path

(path of resources)

SameSite

(transmit only to same site)

SOP

Same-Origin-Policy

Same Origin

Two resources share same origin if...

Same protocol

(http, https, ftp, ...)

Same domain name

Same port

(80, 443, 8080, 8443, ...)

Politic for other origins

Mainly for XMLHttpRequest()

Cannot access other content

But can be embedded in html

(scripts, img, video, forms, ...)

Can do requests

(GET et POST)

CORS

Cross Origin Ressource Sharing

Principle

Finer grained request to outside

New HTTP Header

Browser ask for rights

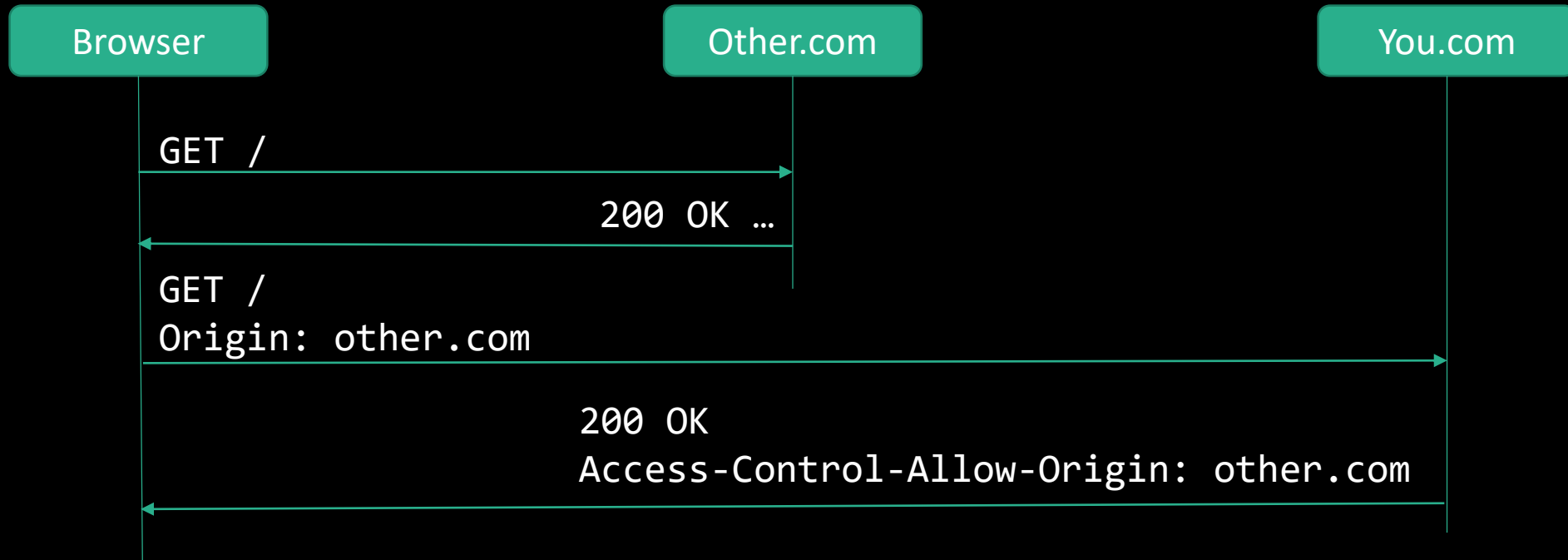
(Origin, Access-Control-Request-Method)

Server check/setup rights

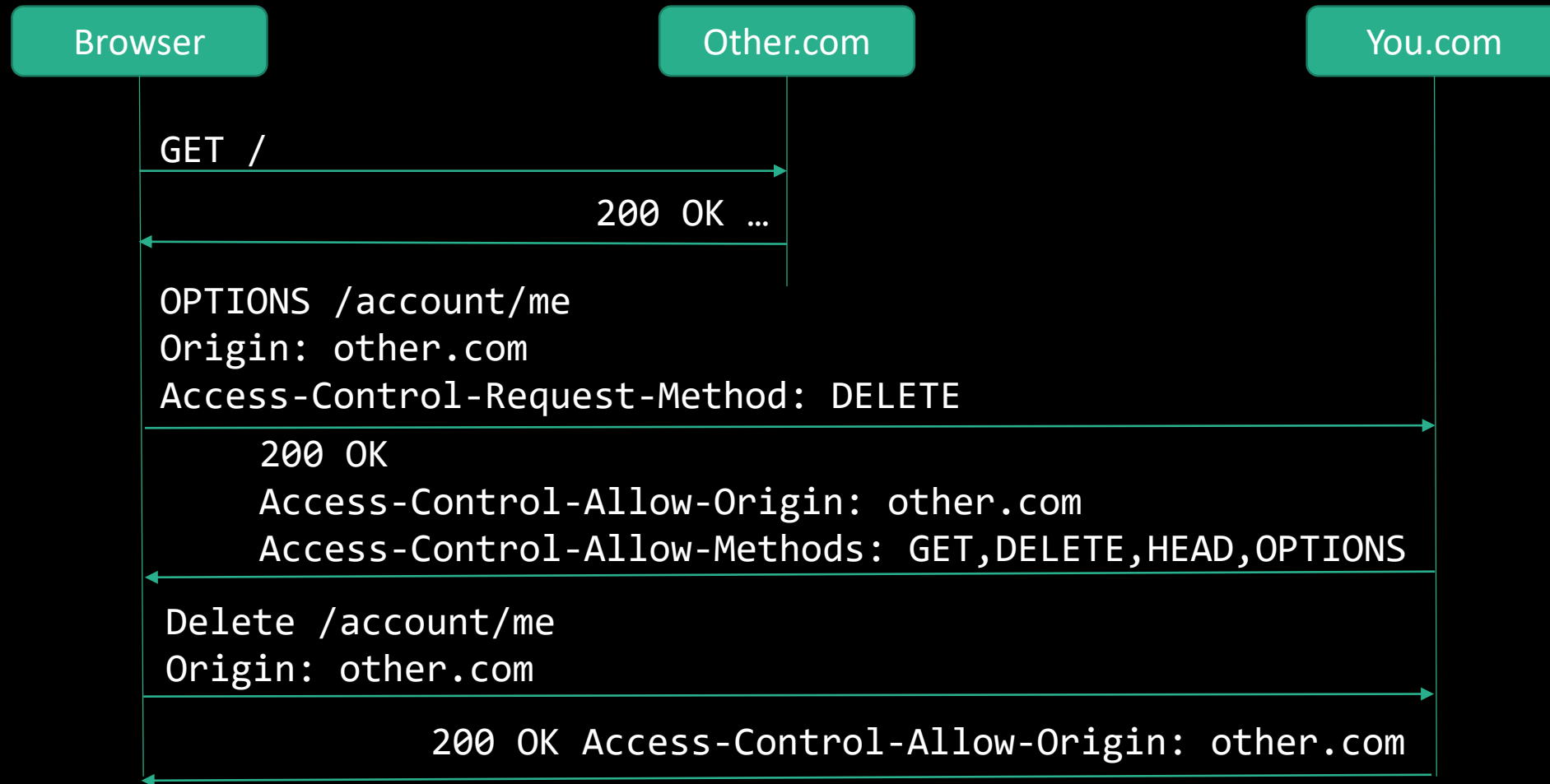
(Access-Control-Allow-Origin, Access-Control-Allow-Methods)

Simple request

(GET, POST, HEAD + content type)



« preflight » Request (everything else)



CSP

Content Security Policy

Principle : Headers

Server set the policy

HTTP header

(Content-Security-Policy)

HTML header

(meta, Content-Security-Policy)

Principle : rules

Restriction on usable origins

Type of contents

(default-src, script-src, style-src, ...)

Allowed Origin

('self', domaine, protocole+domaine)

Principle : reports

Error notification to an endpoint

A URL

(to get JSON report from browsers)

A mode « report only »

(To check policy before going to production)

Anti CSRF

Available techniques

CSRF Token

Server generate random value

(unique for each session)

Put on a form

```
<input type=hidden>
```

Checked on submit

Double submit

Idem but...

Cookie instead of session

Variants

(ciphered / signed cookie)

Re-authentication

Re-ask for password

(for important requests only)


Captcha

Turing test

(painfull for humans)

Veuillez cocher la case ci-dessous pour continuer.

Je ne suis pas un robot

 reCAPTCHA
Confidentialité - Conditions