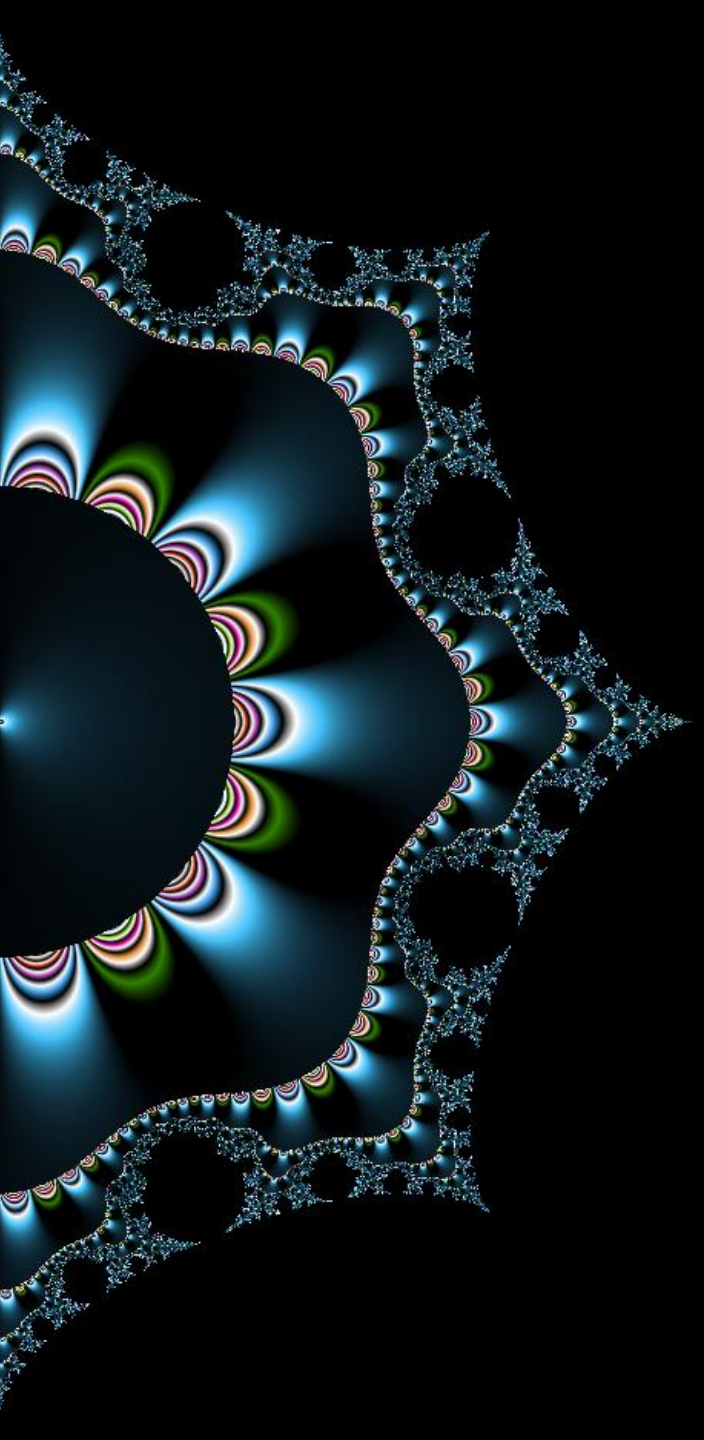


XSS
Injection HTML & JS

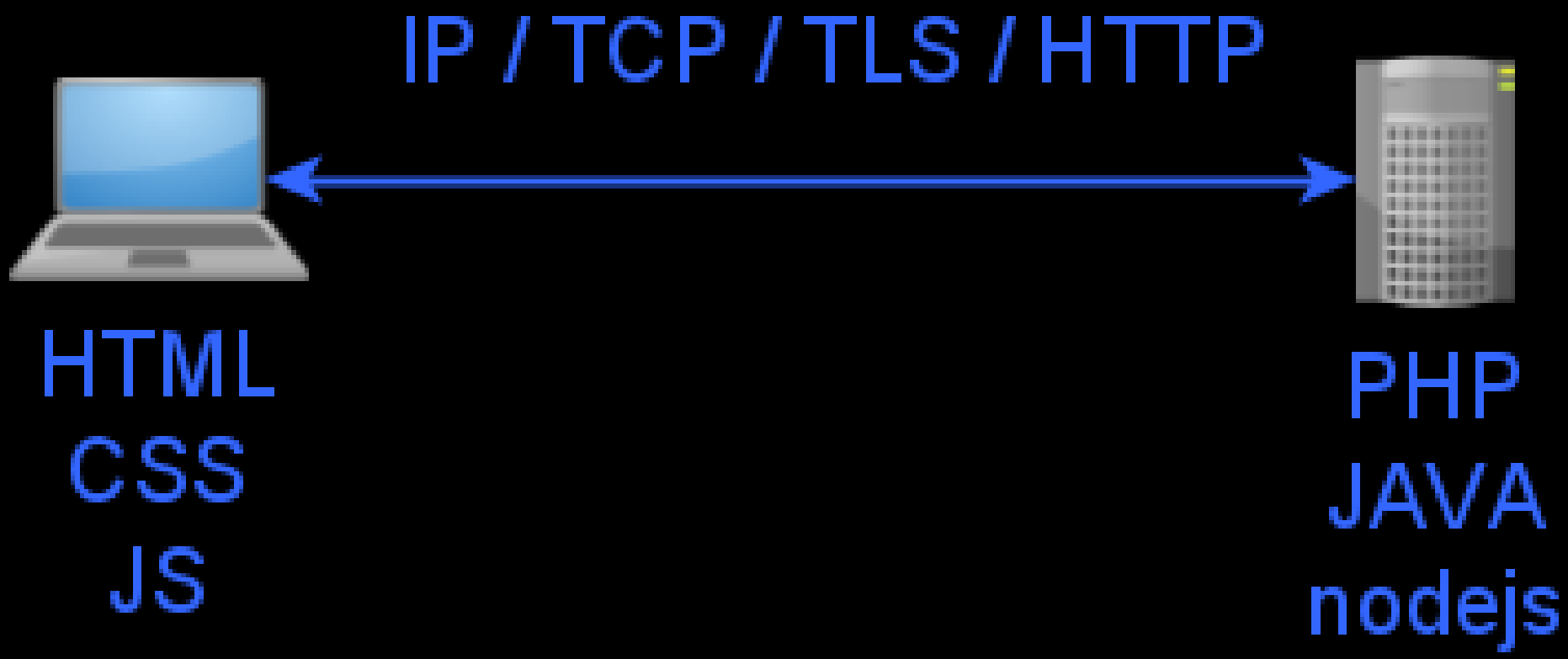
Thibaut HENIN

tbowan@arsouyes.org

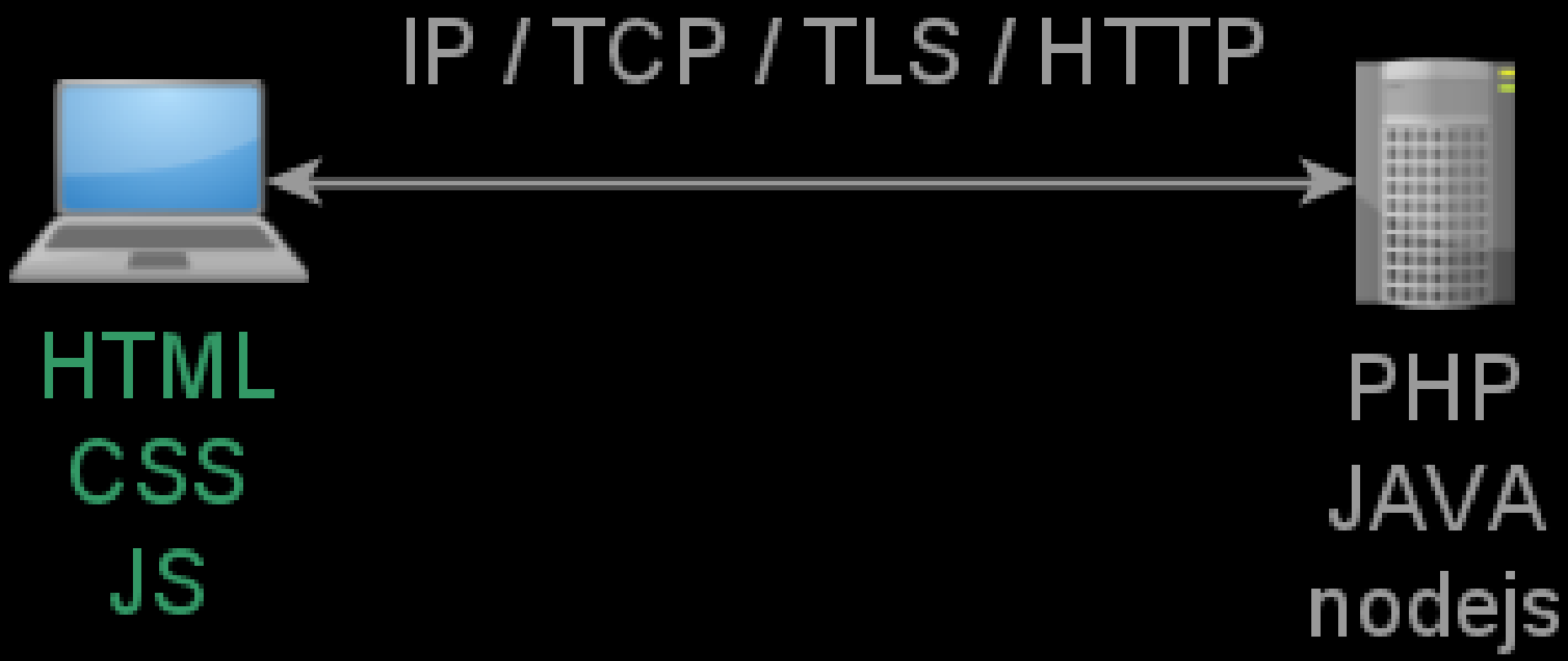
Technologies Web

HTML, Javascript, …

Technologies Web

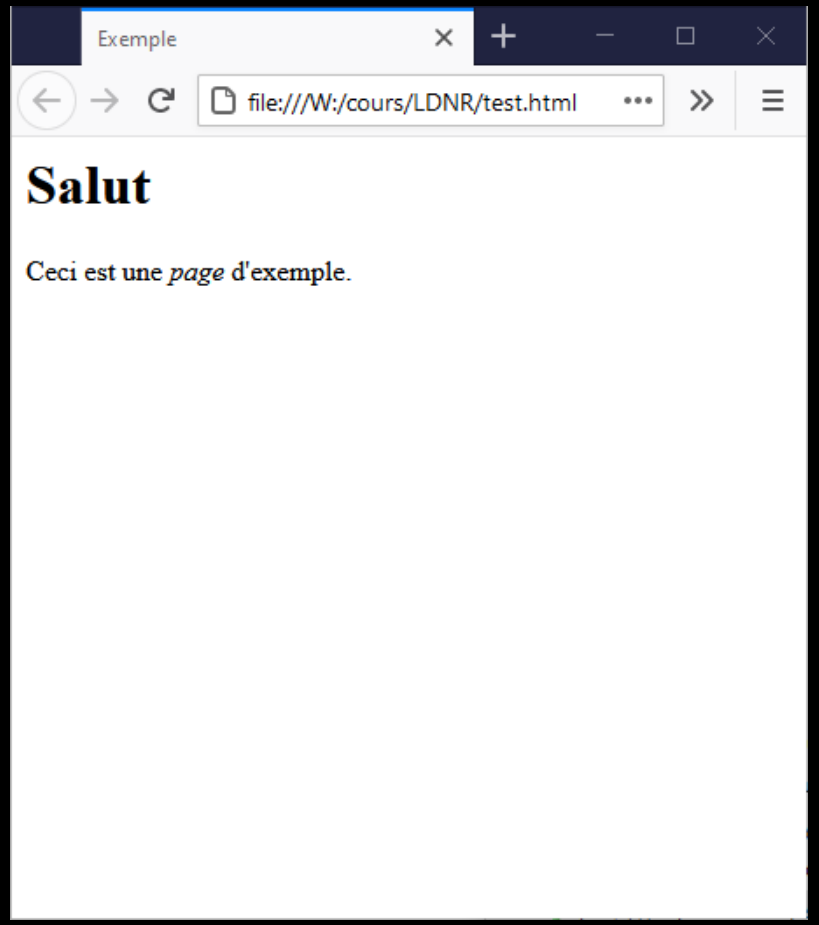


Technologies Web



Base du HTML

```
<html lang="fr">
  <head>
    <title>Exemple</title>
  </head>
  <body>
    <h1>Salut</h1>
    <p>Ceci est une
      <em>page</em>
      d'exemple.</p>
  </body>
</html>
```

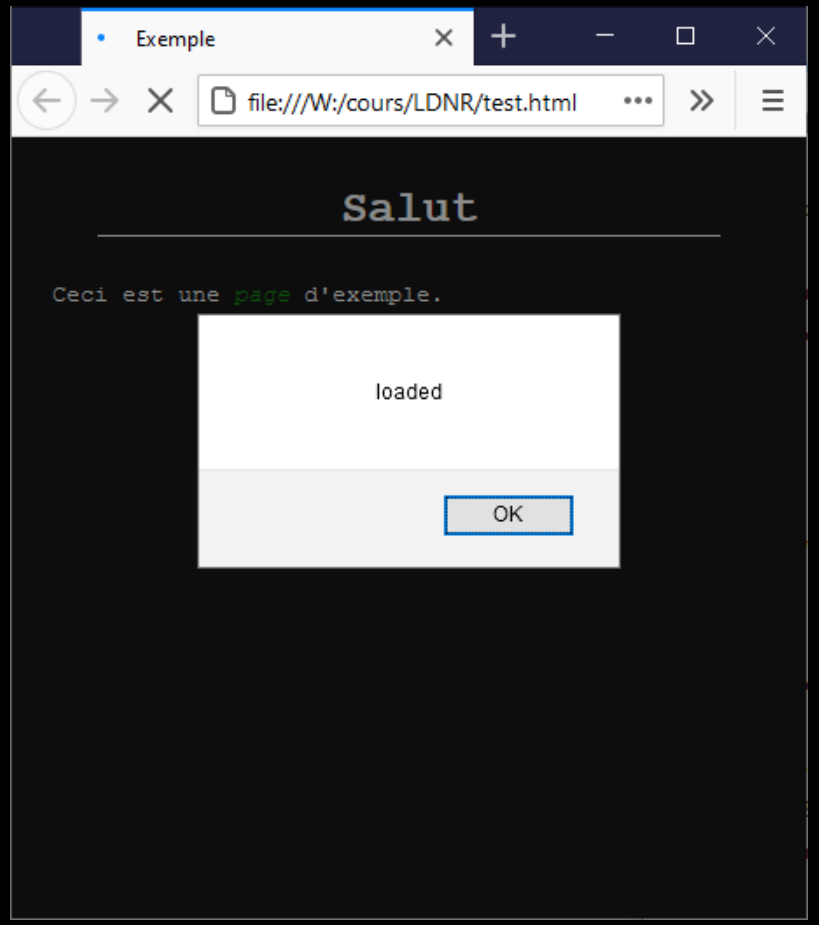


Base du CSS

```
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 ;
}
```

Base du JS

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



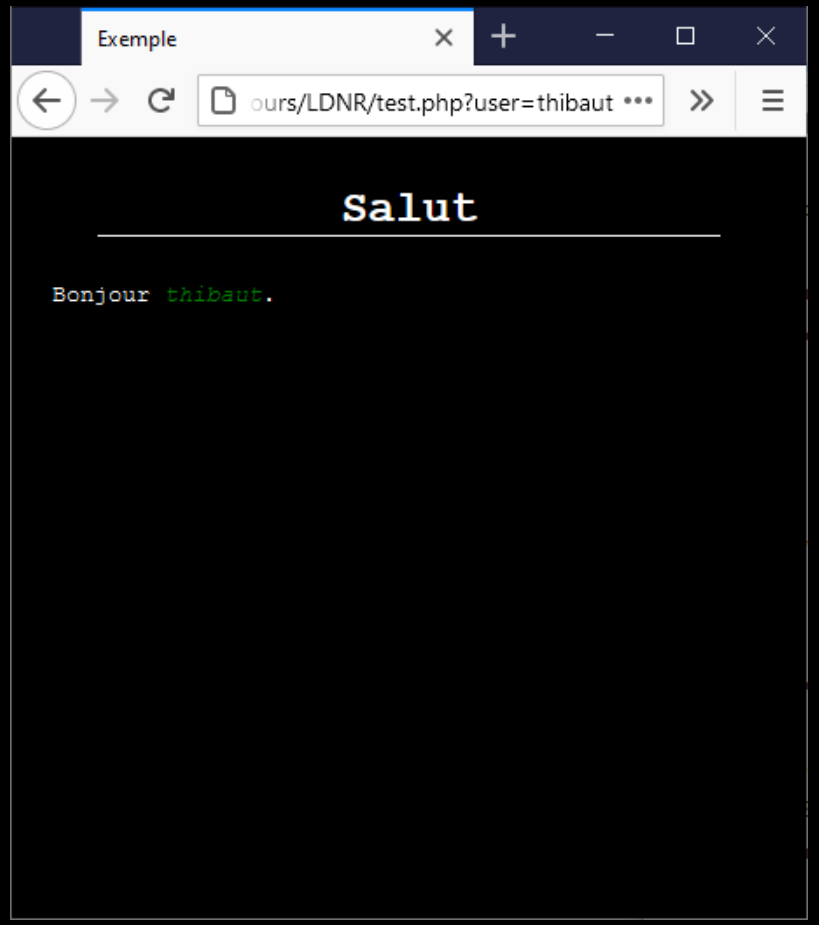
XSS - Reflected

Cross Site Scripting

Application

```
<html>
  <body>
    <h1>Salut</h1>
    <p>Bonjour <em>
<?php
  echo $_GET["user"] ;
?>php
    </em>.</p>
  </body>
</html>
```

https://localhost/test.php?user=thibaut



Injection HTML

```
User = <h1>thibaut</h1>
```

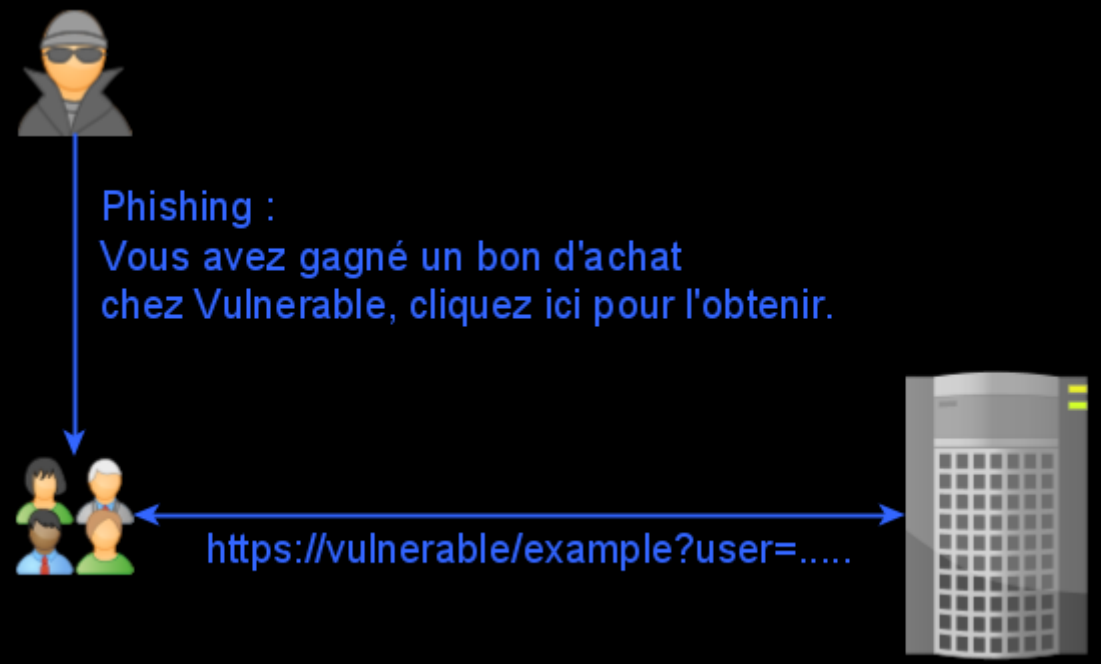
```
?user=%3Ch1%3Ethibaut%3C%2Fh1%3E
```

Injection HTML (bis)

```
User =

thibaut</em>.</p>
<h1>Vous avez gagné</h1>
<p>Cliquez
<a href="https://evil-website.org">
ici</a>
pour remporter votre prix<em>
```

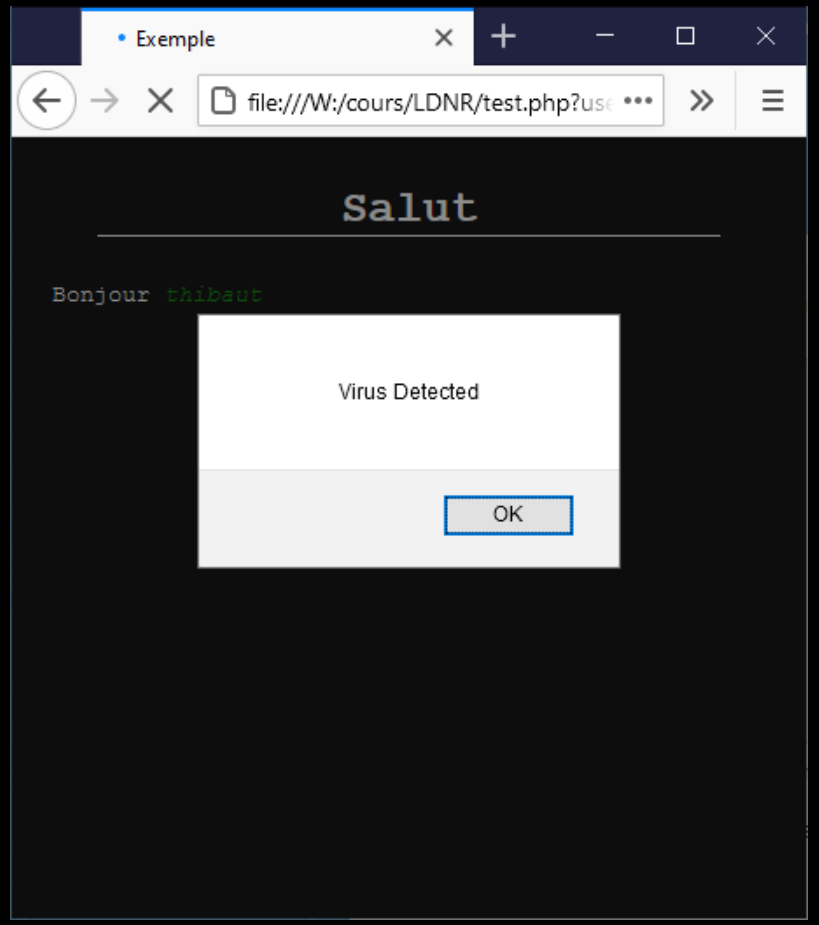
Principe – Réflexion



Injection JS - Exécution de code

```
User =

thibaut
<script>
alert("Virus Detected");
</script>
```



XSS - Stored

Cross Site Scripting

Principe – persistante

addComment.php

```
<?php

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

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

showPost.php

```
<?php

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

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

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

Principe – persistante

Risques des injections JS

Vol d’informations

(cookies, données de formulaire, …)

Enrôlement dans un botnet

(relai d’attaques, minage de cryptomonaie, …)

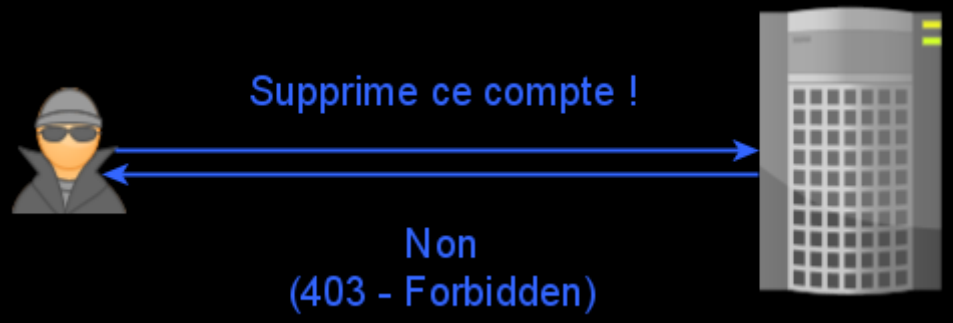
Exécutions de requêtes

(XSRF)

XSRF

Cross Site Request Forgery

Principe – fonctionnalités protégées



Principe –les victimes font les requêtes