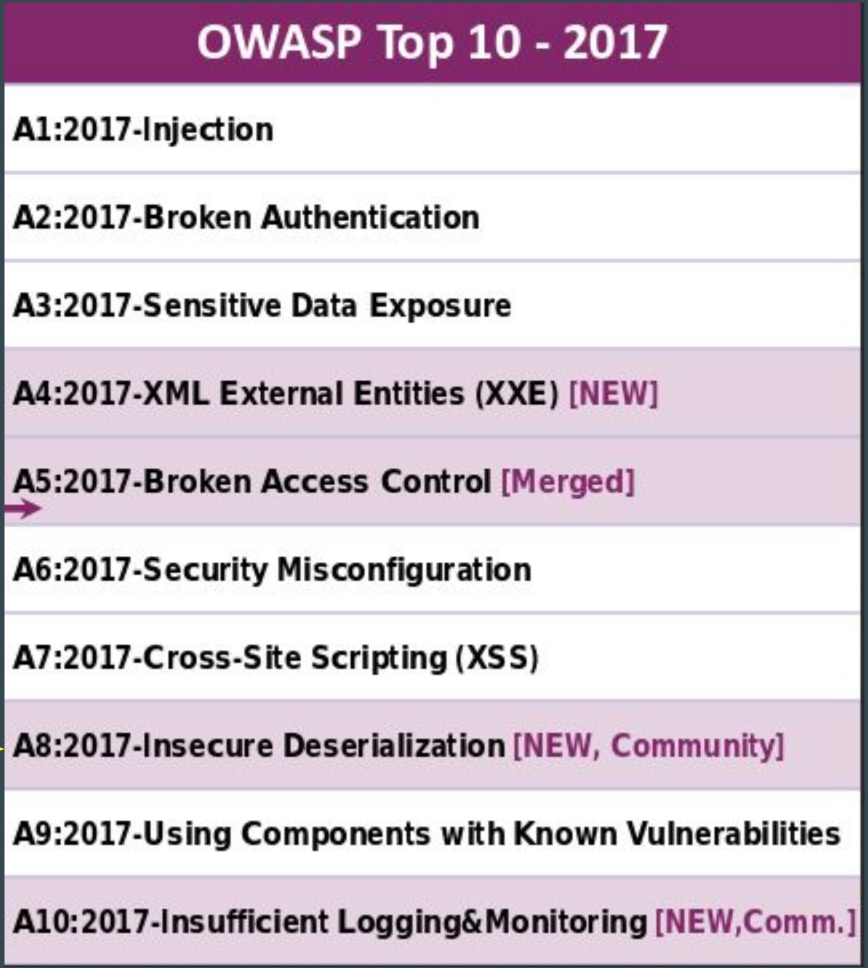
Insecure Deserialization

@raadfhaddad

Contents

Contents

OWASP Top 10 - 2017

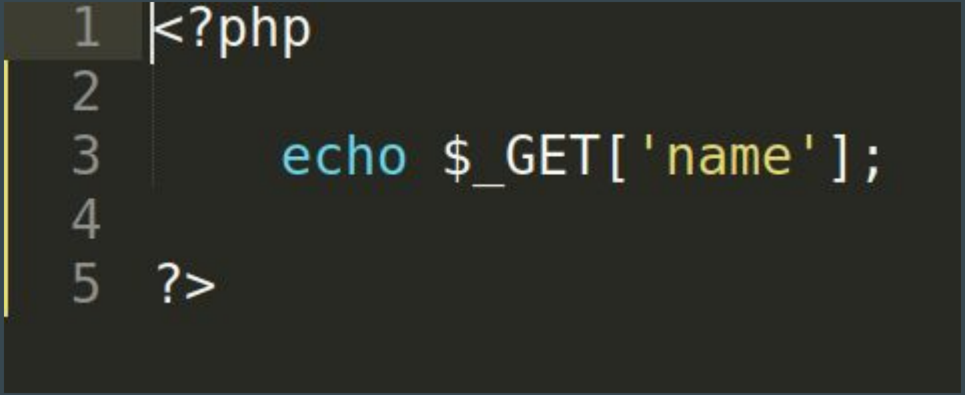


OWASP Top 10 - 2017
A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE) [NEW]
A5:2017-Broken Access Control [Merged]
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization [NEW, Community]
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Power of PHP Language

- Server side scripting: create dynamic web-pages

https://localhost/mypage.php?name=Raad Haddad



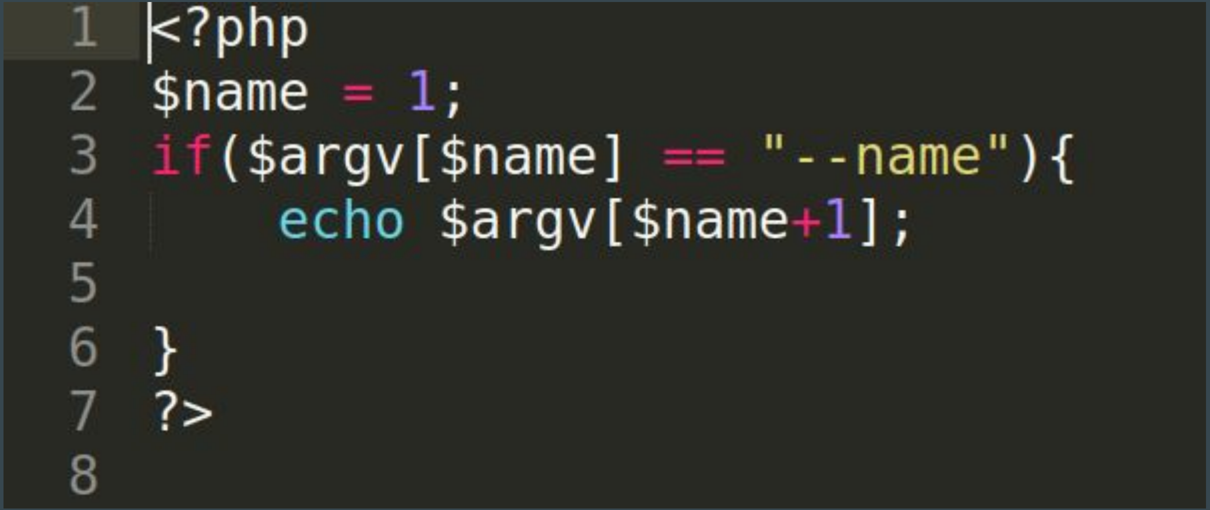
```php
<?php

    echo $_GET['name'];

?>
```

Power of PHP Language - 2

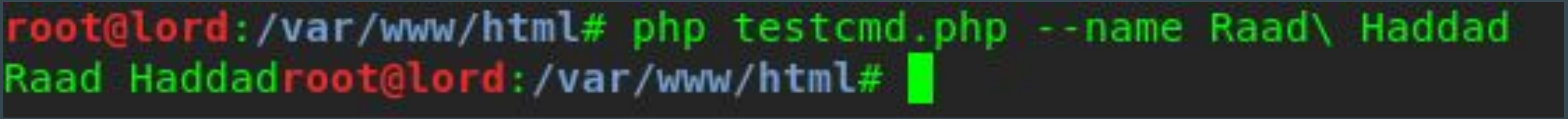
- Command line scripting: create PHP scripts without browser/server.



```php
<?php
$name = 1;
if($argv[$name] == "--name"){
	echo $argv[$name+1];

}
?>

```

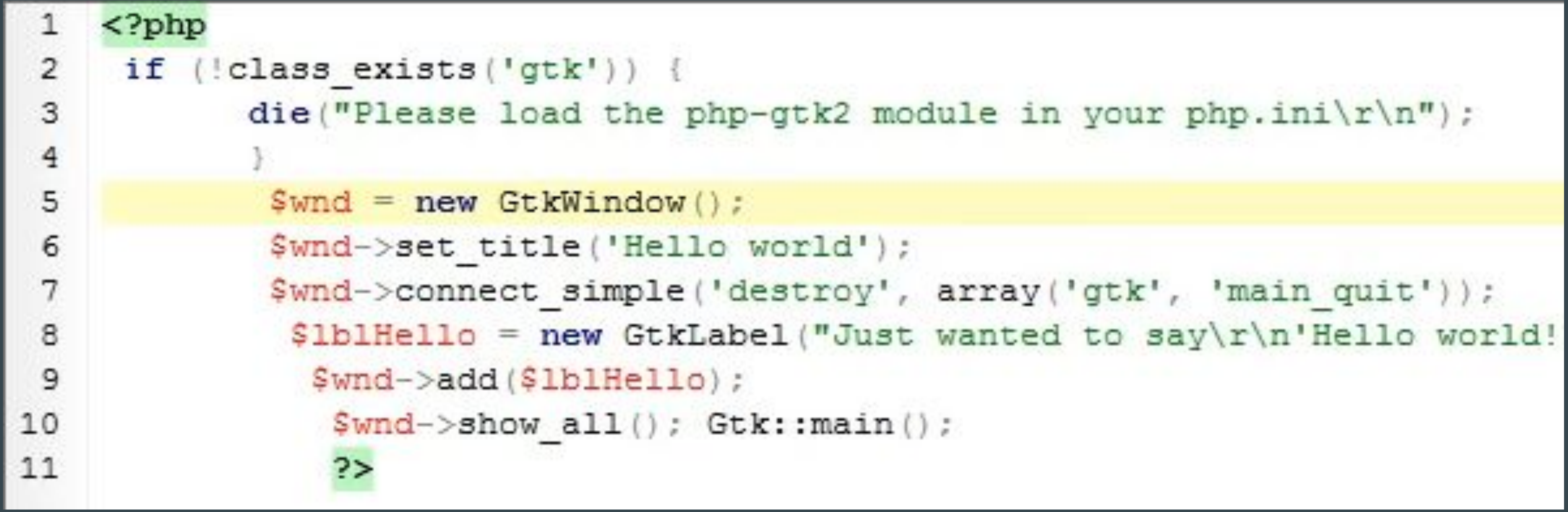


```
root@lord:/var/www/html# php testcmd.php --name Raad\ Haddad
Raad Haddadroot@lord:/var/www/html# 
```

Power of PHP Language - 3

- Desktop application: write and develop Desktop Application

  (not the best way to do this)



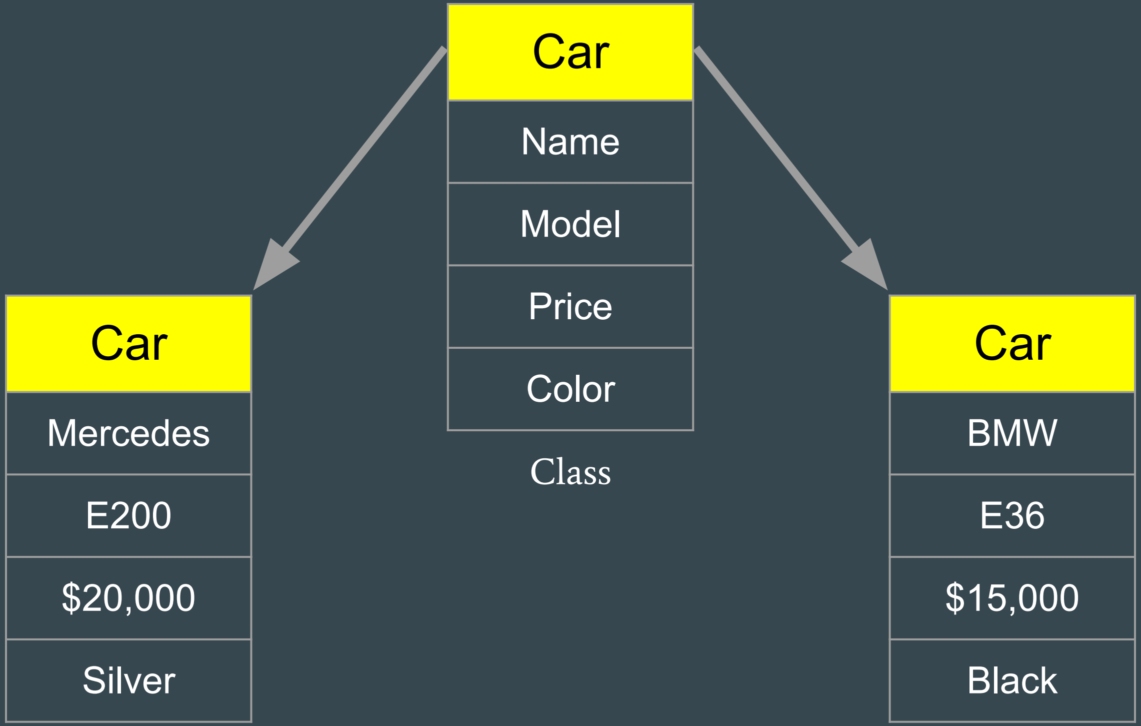
```php
<?php
 if (!class_exists('gtk')) {
       die("Please load the php-gtk2 module in your php.ini\r\n");
       }
        $wnd = new GtkWindow();
        $wnd->set_title('Hello world');
        $wnd->connect_simple('destroy', array('gtk', 'main_quit'));
         $lblHello = new GtkLabel("Just wanted to say\r\n'Hello world!
          $wnd->add($lblHello);
           $wnd->show_all(); Gtk::main();
           ?>
```

Source: https://youtu.be/363MbLx0eRw

Classes and Objects

What is your favourite car brand?

Classes and Objects

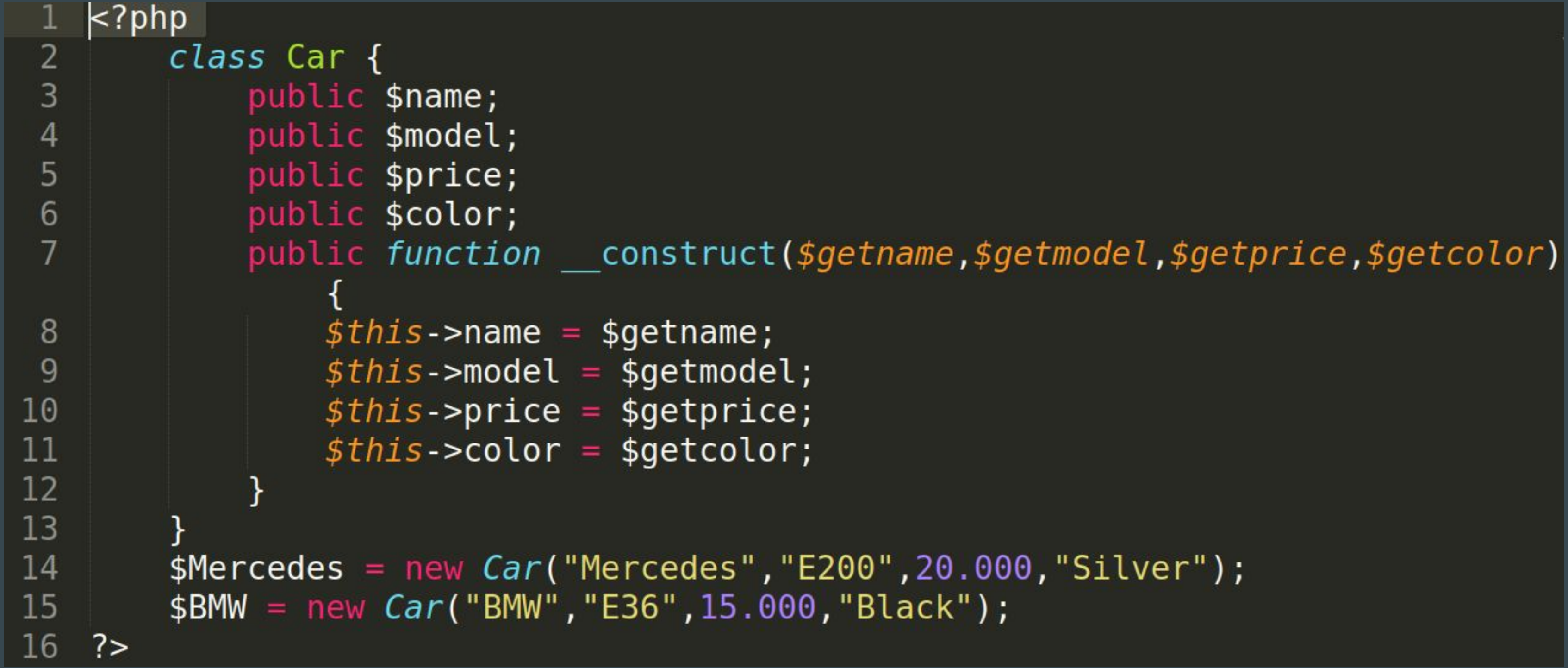


Class

Object

Object

Classes and Objects in PHP



```php
<?php
    class Car {
        public $name;
        public $model;
        public $price;
        public $color;
        public function __construct($getname,$getmodel,$getprice,$getcolor)
            {
            $this->name = $getname;
            $this->model = $getmodel;
            $this->price = $getprice;
            $this->color = $getcolor;
        }
    }
    $Mercedes = new Car("Mercedes","E200",20.000,"Silver");
    $BMW = new Car("BMW","E36",15.000,"Black");
?>
```

PHP Magic Methods

Do some magic with your class:

- *__construct()*
- *__destruct()*
- *__wakeup()*
- *__sleep()*
- *__clone()*

….

PHP Magic Methods

__*destruct()* => allows you to perform some operations before destroying an object

__*wakeup()* => it is called when we unserialize()

__*sleep()* => it is called when we serialize()

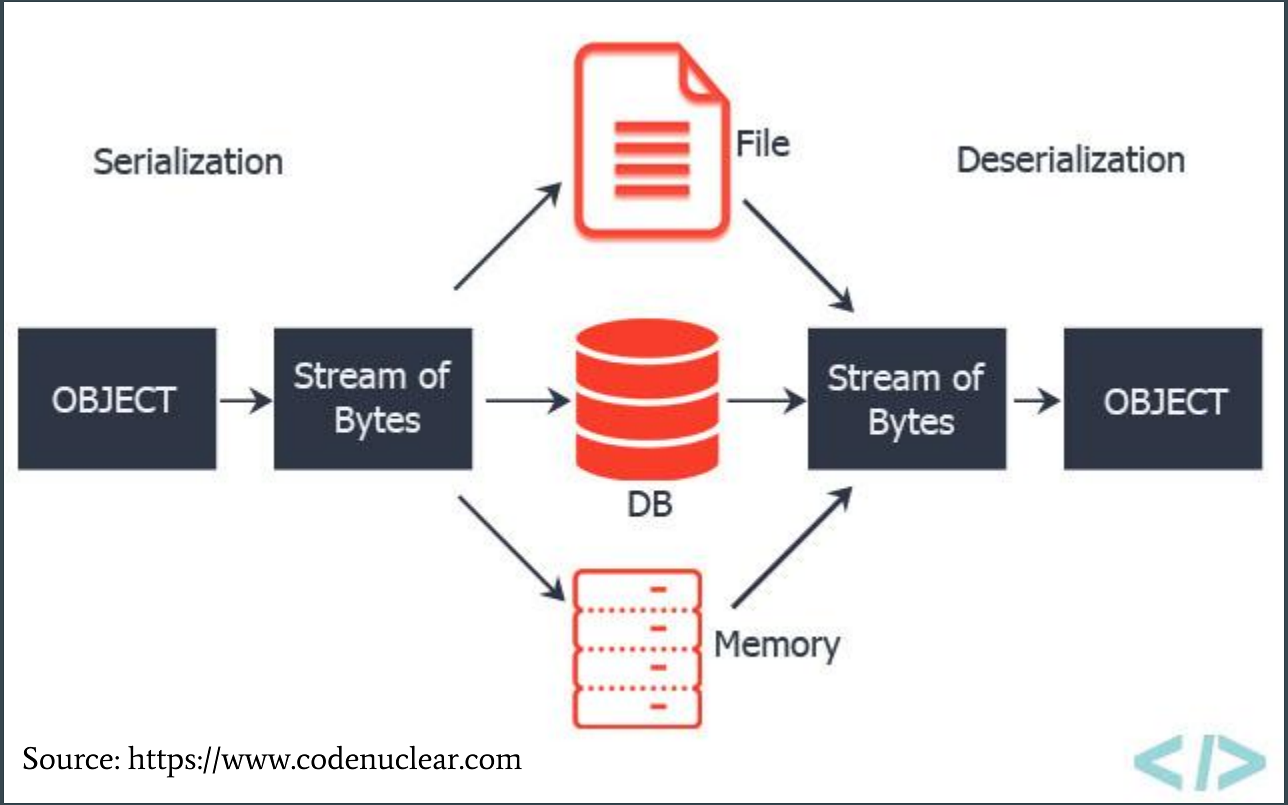
POP Chains in PHP

- Class has Interesting Operations
  - File access
  - Command execution
  - Mail access
  - Session start

    .....

- Class has Magic Methods
- Class transfer workflow

Object Serialization and Deserialization



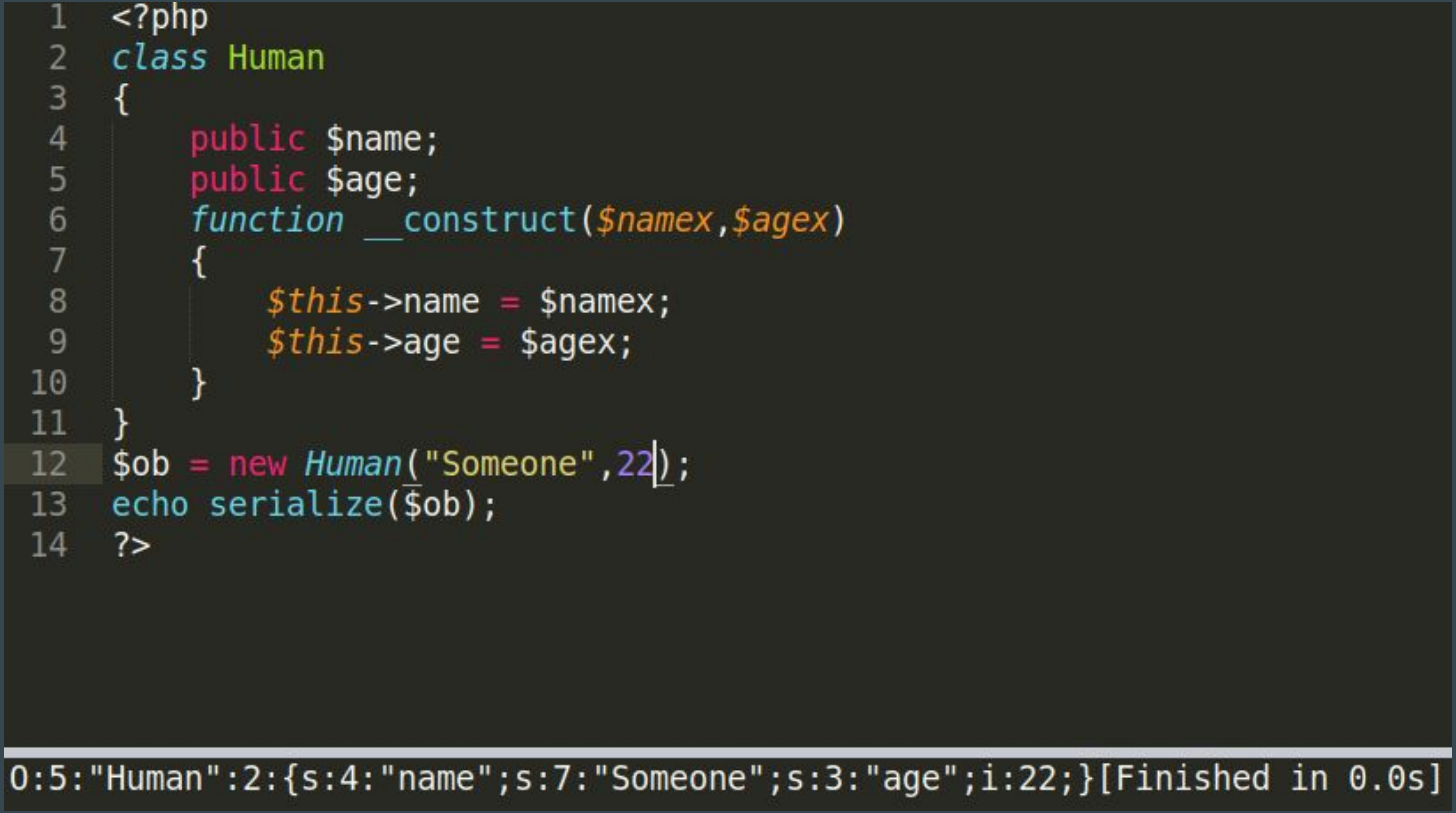
Source: https://www.codenuclear.com

Object Serialization

We serialize objects to:

- Store data in a persistence storage
- Transfer Data
- Store data to files

PHP - serialize() built-in Function



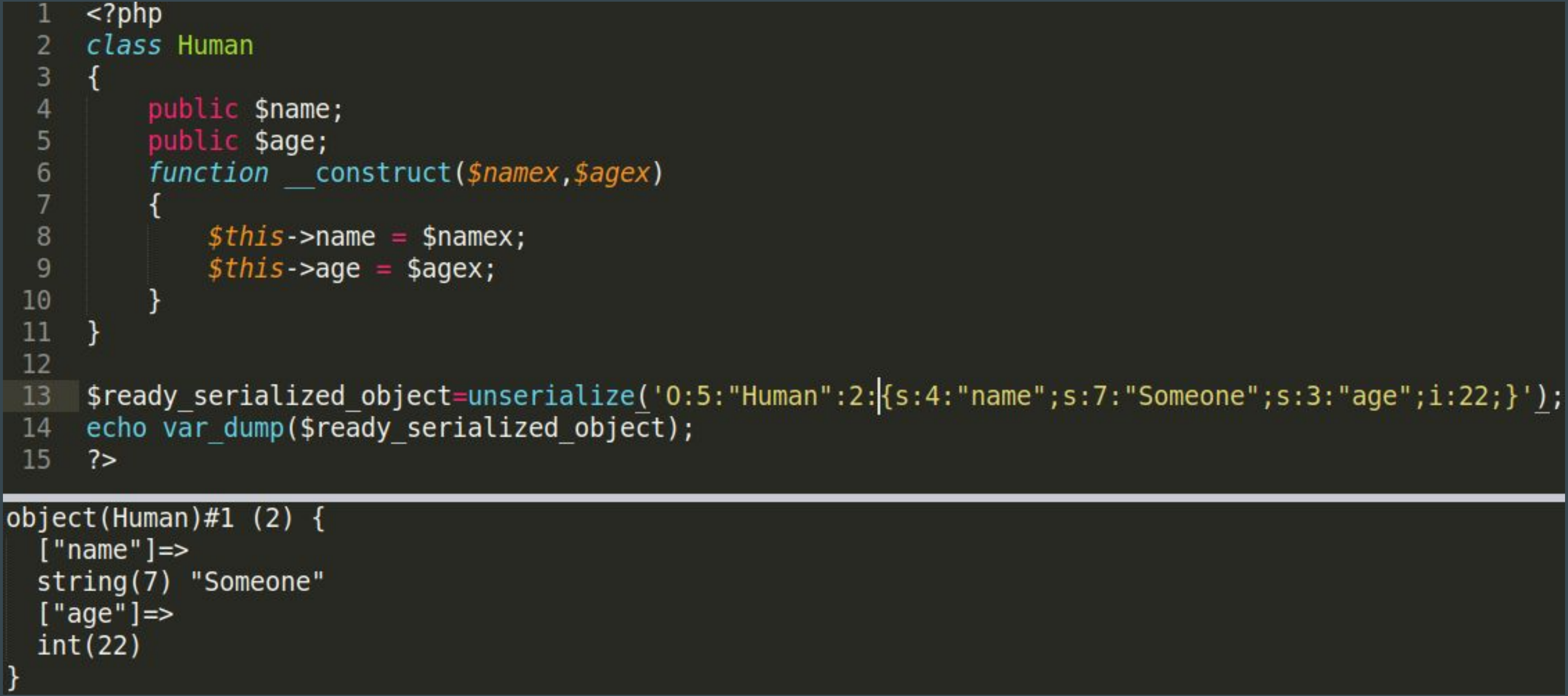
```php
<?php
class Human
{
	public $name;
	public $age;
	function __construct($namex,$agex)
	{
		$this->name = $namex;
		$this->age = $agex;
	}
}
$ob = new Human("Someone",22);
echo serialize($ob);
?>
```

```
O:5:"Human":2:{s:4:"name";s:7:"Someone";s:3:"age";i:22;}[Finished in 0.0s]
```

Serialized Object

```
O:3:"Log":3:{                              #class name
    s:4:"type";                            #first property
        s:3:"XSS";                         #first property’s value
    s:7:"logfile";                         #second property
        S:7:"log.txt";                     #second property’s value
    s:7:"logdata";                         #third property
        s:19:"Attack was detected";        #third property’s value
}
```

PHP - unserialize() built-in Function



```php
<?php
class Human
{
	public $name;
	public $age;
	function __construct($namex,$agex)
	{
		$this->name = $namex;
		$this->age = $agex;
	}
}

$ready_serialized_object=unserialize('O:5:"Human":2:{s:4:"name";s:7:"Someone";s:3:"age";i:22;}');
echo var_dump($ready_serialized_object);
?>
```

```
object(Human)#1 (2) {
  ["name"]=>
  string(7) "Someone"
  ["age"]=>
  int(22)
}
```

Security issues with unserialize()

- Never pass user’s inputs to *unserialize*( ) function
- Deserialization of user’s input is the major security problem

Local File Inclusion

The File Inclusion vulnerability allows an attacker to include a file, usually exploiting a "dynamic file inclusion" mechanisms implemented in the target application. The vulnerability occurs due to the use of user-supplied input without proper validation.

- OWASP

Possible RCE?

Let’s check it out....

Questions ?

@raadfhaddad