

Bypass Business Layer Access Control

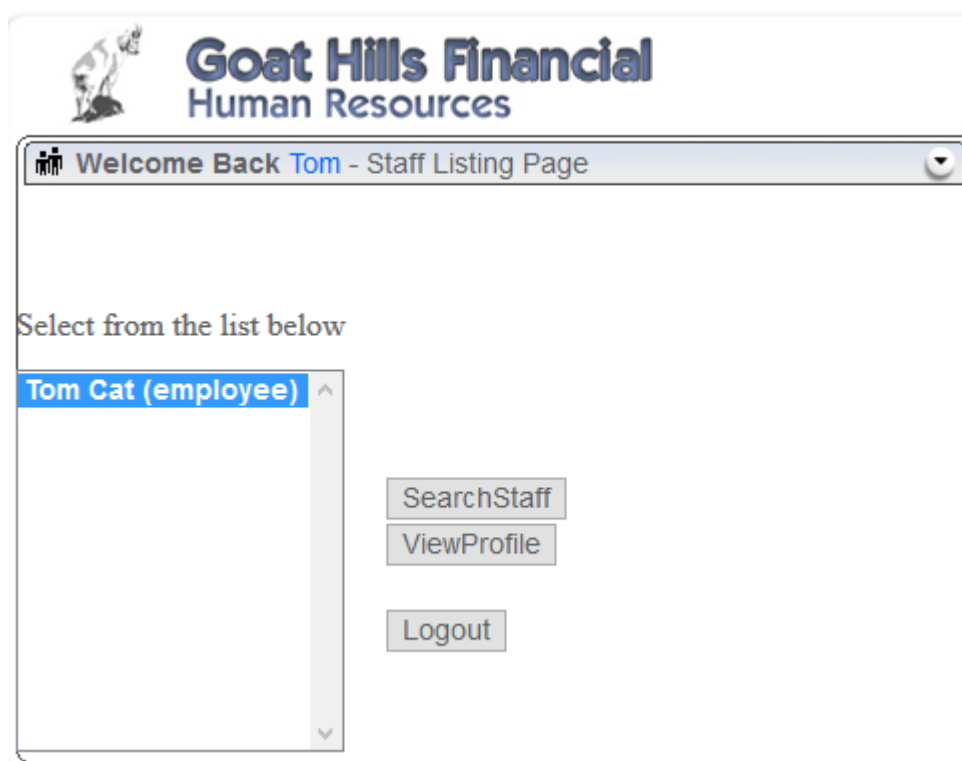
由於沒有對 Control 裡的 Delete 指令做權限管理，又透過 action 判斷 Control 指令，所以原本不應該有 Delete 權限的 Tom 執行了 Delete 操作。

使用密碼 john 進入 John Wayne 的帳號，有 ViewProfile 和 DeleteProfile 的操作



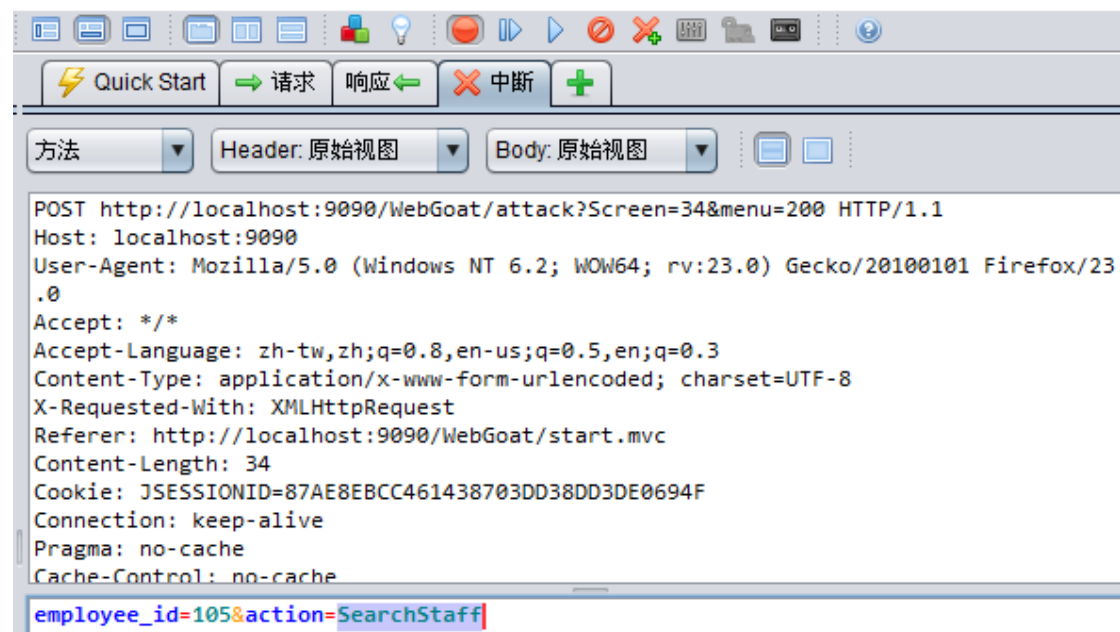
The screenshot shows the 'Goat Hills Financial Human Resources' web application. At the top, there is a logo of a goat and the text 'Goat Hills Financial Human Resources'. Below this is a navigation bar with a user icon and the text 'Welcome Back John - Staff Listing Page'. The main content area has the heading 'Select from the list below' followed by a scrollable list of staff members. The list includes: Larry Stooge (employee), Moe Stooge (manager), Curly Stooge (employee), Eric Walker (employee), Tom Cat (employee), Jerry Mouse (hr), David Giambi (manager), Bruce McGuire (employee), Sean Livingston (employee), Joanne McDougal (hr), and John Wayne (admin). To the right of the list are four buttons: SearchStaff, ViewProfile, CreateProfile, and DeleteProfile. At the bottom right is a Logout button.

使用密碼 tom 進入 Tom Cat 的帳號，只有 ViewProfile



The screenshot shows the 'Goat Hills Financial Human Resources' web application. At the top, there is a logo of a goat and the text 'Goat Hills Financial Human Resources'. Below this is a navigation bar with a user icon and the text 'Welcome Back Tom - Staff Listing Page'. The main content area has the heading 'Select from the list below' followed by a scrollable list of staff members. The list includes: Tom Cat (employee). To the right of the list are three buttons: SearchStaff, ViewProfile, and Logout.

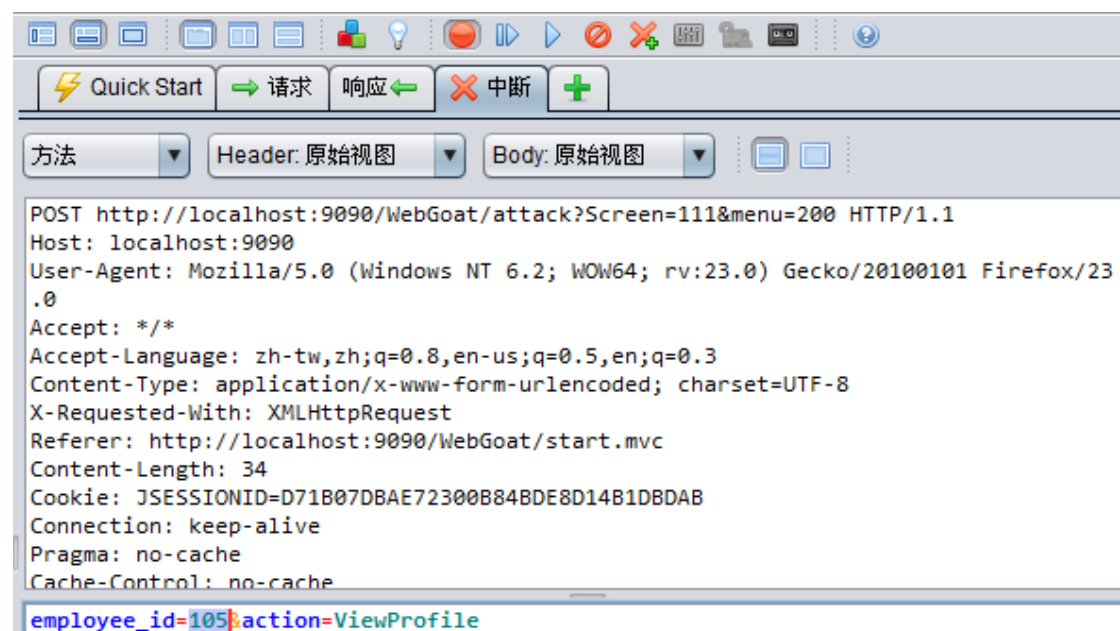
執行 ViewProfile 攔截請求，改 action 為 DeleteProfile



*** You have completed Stage 1: Bypass Business Layer Access Control.**

Bypass Data Layer Access Control

View 這個操作不能像 Delete 一樣對 Tom 進行權限上的控制，那麼與 Tom 出於同一層級的其它用戶也具有這個權限，所以說 Tom 可以通過攔截修改 employee_id 水平訪問其它人的資料。權限控管未做好，水平越權問題。





Goat Hills Financial Human Resources

Welcome Back Tom - View Profile Page

First Name:	Eric	Last Name:	Walker
Street:	1160 Prescott Rd	City/State:	New York, NY
Phone:	410-887-1193	Start Date:	12152005
SSN:	445-66-5565	Salary:	13000
Credit Card:	NA	Credit Card Limit:	0
Comments:	Late. Always needs help. Too intern-ish.		
Disciplinary Explanation:	Disc. Dates: 101013 Bothering Larry about webgoat problems		
Manager:	107		

ListStaffEditProfile




Logout

以 Tom 身份看到了 Eric Walker 的檔案

*** You have completed Stage 3: Bypass Data Layer Access Control.**

DOM-Based cross-site scripting

LAB: DOM-Based cross-site scripting



Java [Source] Solution Lesson Plan Hints Restart Lesson

STAGE 1: For this exercise, your mission is to deface this website using the image at the following location: [OWASP IMAGE](#)

Enter your name:

Submit Solution

依次輸入以下內容在名字欄位中

//XSS 插入圖片

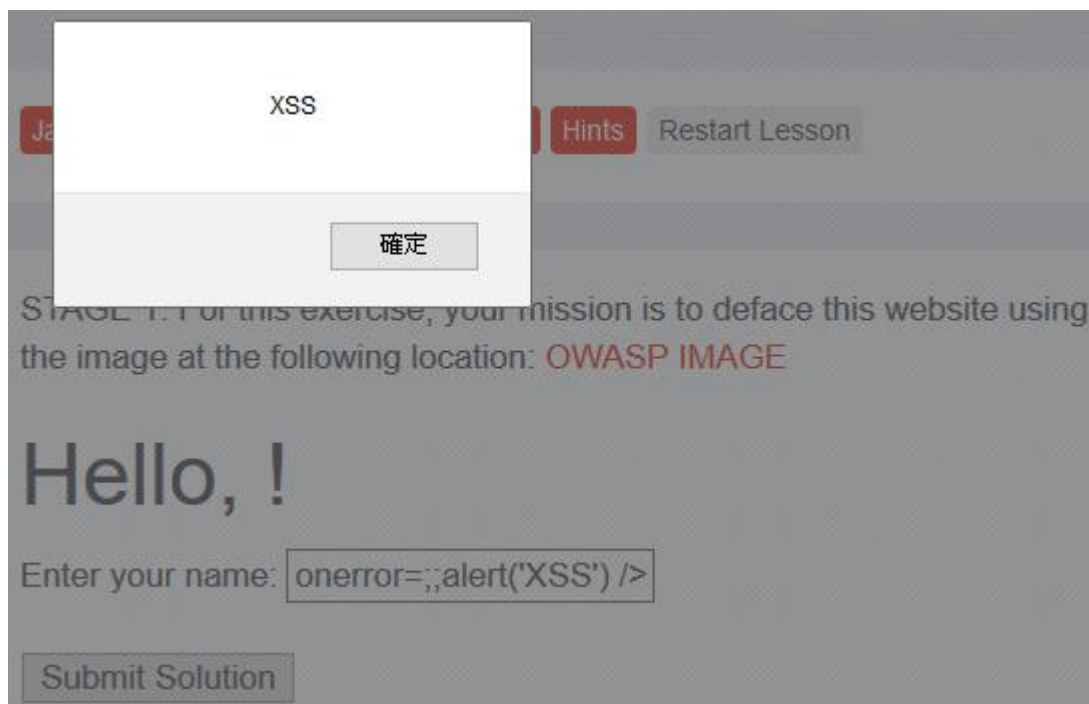
STAGE 1: For this exercise, your mission is to deface this website using the image at the following location: [OWASP IMAGE](#)

Hello,  **OWASP** !
The Open Web Application Security Project

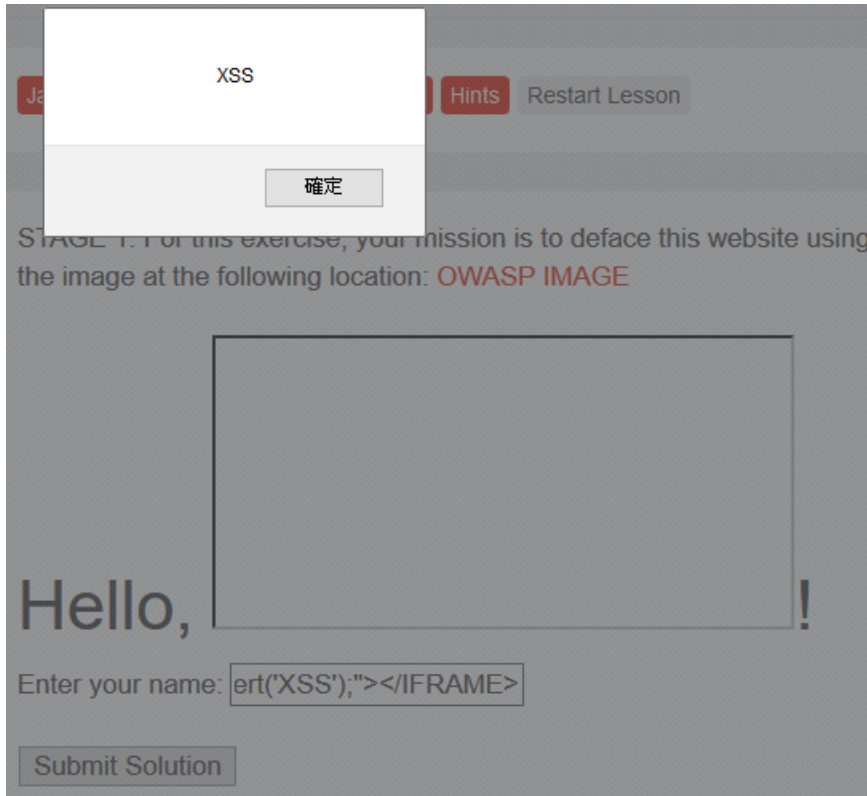
Enter your name:

[Submit Solution](#)

//XSS 插入 Alert



<IFRAME SRC="javascript:alert('XSS');"></IFRAME>//XSS 插入 iFrame



甚至可以直接偽造介面

[illegible]

STAGE 1: For this exercise, your mission is to deface this website using the image at the following location: [OWASP IMAGE](#)

Hello, Please enter your password:

Submit

```
/*AJAX Security-Dangerous Use of Eval
```

Eval 是 php 語言中執行一段 JS 代碼的意思，與剛剛基於 DOM 的不同，DOM 是

直接插入新節點，而這個是先關閉原本的 DOM，然後寫自己的 DOM，再組裝好剛剛被關閉 DOM 的後半部分。




通過 php 的 Eval，alert 被執行 123');alert(document.cookie);('

123 後的');使得原本的 DOM 不受影響，最後的('閉合掉了原本多出的')符號，插入代碼的樣子是('123');alert(document.cookie);(")*/

Buffer Overflows-Off-by-One Overflows

輸入姓名、房號

Off-by-One Overflows



Java [Source] Solution Lesson Plan Hints Restart Lesson

Welcome to the **OWASP Hotel!** Can you find out which room a VIP guest is staying in?

In order to access the Internet, you need to provide us the following information:

Step 1/2

Ensure that your first and last names are entered exactly as they appear in the hotel's registration system.

First Name:	<input type="text" value="f"/>	*
Last Name:	<input type="text" value="l"/>	*
Room Number:	<input type="text" value="1"/>	*

Submit

* The above fields are required for login.

會跳到選擇入住時間的頁面

Please select from the following available price plans:

Step 2/2

Ensure that your selection matches the hours of usage, as no refunds are given for this service.

Available Price Plans: 

Accept Terms

By Clicking on the above you accept the terms and conditions.

這時將表單詳細地顯示出來，且發現房號可進行 Buffer Overflow 攻擊

`<form enctype="" method="POST" name="form">`Please select from the following available price plans:

Step 2/2

Ensure that your selection matches the hours of usage, as no refunds are given for this service.

Available Price Plans:

`<select name="price_plan" value="$9.99 - 24 hours">` \$9.99 - 24 hours `</select>`

`<input name="SUBMIT" type="SUBMIT">` Accept Terms

`<input name="last_name">` l `<input name="first_name">` f `<input name="room_no">` 1

By Clicking on the above you accept the terms and conditions.

之後跳到這個頁面

*** To complete the lesson, restart lesson and enter VIP first/last name**

You have now completed the 2 step process and have access to the Internet

Process complete

Your connection will remain active for the time allocated for starting now.

We would like to thank you for your payment.

再次將表單詳細顯示出來，得到 VIP 資料

`<form enctype="" method="POST" name="form">` You have now completed the 2 step process and have access to the Internet

Process complete

Your connection will remain active for the time allocated for starting now.

<code><input name="a"></code>		<code><input name="b"></code>
f	<code><input name="c"></code>	6124962553630529028
<code><input name="d"></code>	Johnathan	<code><input name="e"></code>
Ravern	<code><input name="f"></code>	4321
<code><input name="g"></code>	John	<code><input name="h"></code>
Smith	<code><input name="i"></code>	56
<code><input name="j"></code>	Ana	<code><input name="k"></code>
Armeta	<code><input name="l"></code>	78
<code><input name="m"></code>	Lewis	<code><input name="n"></code>
Hamilton	<code><input name="o"></code>	9901

We would like to thank you for your payment.

後來看了此題的原始碼，發現其實並沒有真正還原 Buffer Overflow，而是透過以下程式碼


```

// And finally the check...
if(param3.length() > 4096)
{
    ec.addElement(new Input(Input.hidden, "d", "Johnathan"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "e", "Ravern"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "f", "4321"));
    ec.addElement("\r\n");

    ec.addElement(new Input(Input.hidden, "g", "John"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "h", "Smith"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "i", "56"));
    ec.addElement("\r\n");

    ec.addElement(new Input(Input.hidden, "j", "Ana"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "k", "Arneta"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "l", "78"));
    ec.addElement("\r\n");

    ec.addElement(new Input(Input.hidden, "m", "Lewis"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "n", "Hamilton"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "o", "9901"));
    ec.addElement("\r\n");

    s.setMessage("To complete the lesson, restart lesson and enter VIP first/last name");
}
if (("Johnathan".equalsIgnoreCase(param2) || "John".equalsIgnoreCase(param2)
    || "Ana".equalsIgnoreCase(param2) || "Lewis".equalsIgnoreCase(param2))
    && ("Ravern".equalsIgnoreCase(param1) || "Smith".equalsIgnoreCase(param1)
    || "Arneta".equalsIgnoreCase(param1) || "Hamilton".equalsIgnoreCase(param1)))
{
    // :)
    // Allows for mixed VIP names, but that's not really the point
    makeSuccess(s);
}

// Footer
ec.addElement(new br());
ec.addElement(new br());

```

只是檢查了第三個參數的長度，偽造了一個看似 Buffer Overflow 的漏洞，有點是為了出題而出題，並沒有真實還原 Buffer Overflow。

Concurrency-Thread Safety Problems

有些人寫程式喜歡用 static、const，但是往往忽略了多執行緒的問題，比如此題的原始碼

```
private static String currentUser;
```

這裡 currentUser 使用了 static，又沒有做保護，就會造成瀏覽器 A 訪問這個頁面時，B 同時訪問，資料就會被蓋掉

快速按下兩個頁面的 Submit(左邊先)

Thread Safety Problems

Java [Source] Solution Lesson Plan Hints Restart Lesson

The user should be able to exploit the concurrency error in this web application and view login information for another user that is attempting the same function at the same time. **This will require the use of two browsers.** Valid user names are 'jeff' and 'dave'. Please enter your username to access your account.

Enter user name:

jeff 的資料被 dave 蓋掉了

* Congratulations. You have successfully completed this lesson.

Enter user name:

Account information for user: dave

USERID	USER_NAME	PASSWORD	COOKIE
105	dave	dave	

Safety Problems

ution Lesson Plan Hints Restart Les

...e able to exploit the concurrency ei
...ew login information for another use
...at the same time. **This will requir**
...er names are 'jeff' and 'dave'.
...username to access your account.

dave

* Congratulations. You have successfully c

Enter user name:

Stored XSS

Tom 的檔案是可以編輯的，Jerry 可以查看 Tom 的檔案。Tom 對自己的檔案進行編輯，放入 XSS，被儲存到資料庫。Jerry 查看 Tom 檔案時，就中招了！

以 Tom 身份在 Street 欄位放入 XSS

Welcome Back Tom

First Name: Tom

Last Name: Cat

Street: <script>alert("XSS");</script>

City/State: New York, NY

Phone: 443-599-0762

Start Date: 1011999

SSN: 792-14-6364

Salary: 80000

Credit Card: 5481360857968521

Credit Card Limit: 30000

Comments: Co-Owner.

Manager: Tom Cat

Disciplinary Explanation: NA

Disciplinary Action: 0

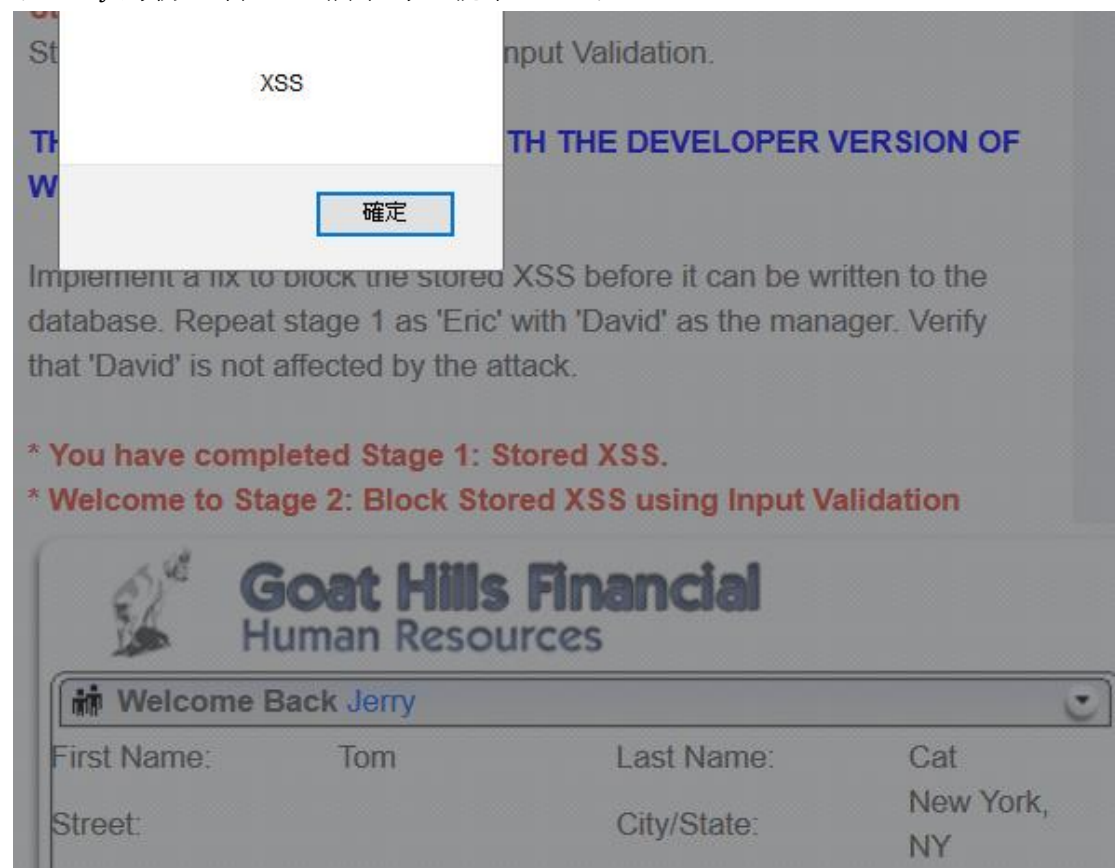
Dates:

ViewProfile

UpdateProfile

Logout

以 Jerry 身份查看 Tom 檔案時，就中 XSS 了

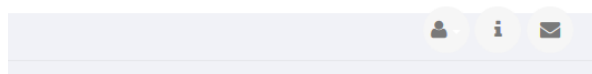


Improper Error Handling-Fail Open Authentication Scheme

要對錯誤有處理，處理不完整也可能造成漏洞，比如這題刪掉密碼這一個參數，也能成功登入，說明程式碼對獲取不到密碼這個參數時的錯誤處理不充分。

輸入 webgoat 帳號，然後輸入任意密碼

Fail Open Authentication Scheme



[Java \[Source\]](#) [Solution](#) [Lesson Plan](#) [Hints](#) [Restart Lesson](#)

Due to an error handling problem in the authentication mechanism, it is possible to authenticate as the 'webgoat' user without entering a password. Try to login as the webgoat user without specifying a password.

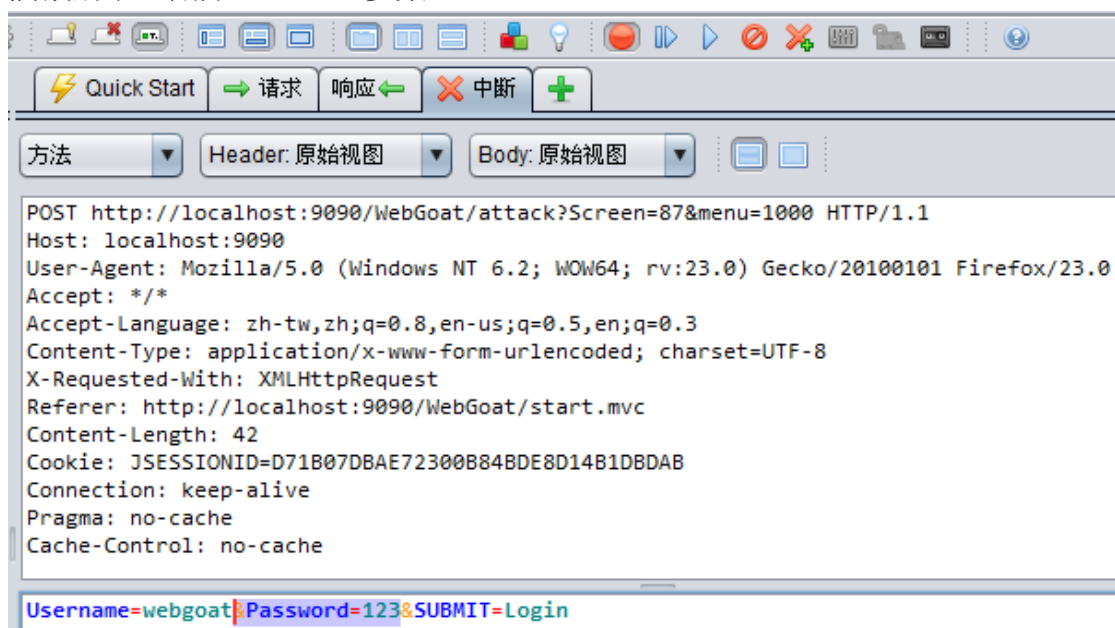
Sign in

Please sign in to your account. See the OWASP admin if you do not have an account.

*Required Fields

*User Name	<input type="text" value="webgoat"/>
*Password	<input type="password" value="..."/>
<input type="button" value="Login"/>	

攔截請求，刪除 Password 參數



* **Congratulations. You have successfully completed this lesson.**

Welcome, webgoat





You have been authenticated with Fail Open Error Handling

[Logout](#)

[Refresh](#)

Injection Flaws-Numeric SQL Injection

按下 Go!送出

 **Numeric SQL Injection**   

[Java \[Source\]](#) [Solution](#) [Lesson Plan](#) [Hints](#) [Restart Lesson](#)

SQL injection attacks represent a serious threat to any database-driven site. The methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks, an incredible number of systems on the internet are susceptible to this form of attack.

Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can easily be prevented.

It is always good practice to sanitize all input data, especially data that will be used in OS command, scripts, and database queries, even if the threat of SQL injection has been prevented in some other manner.

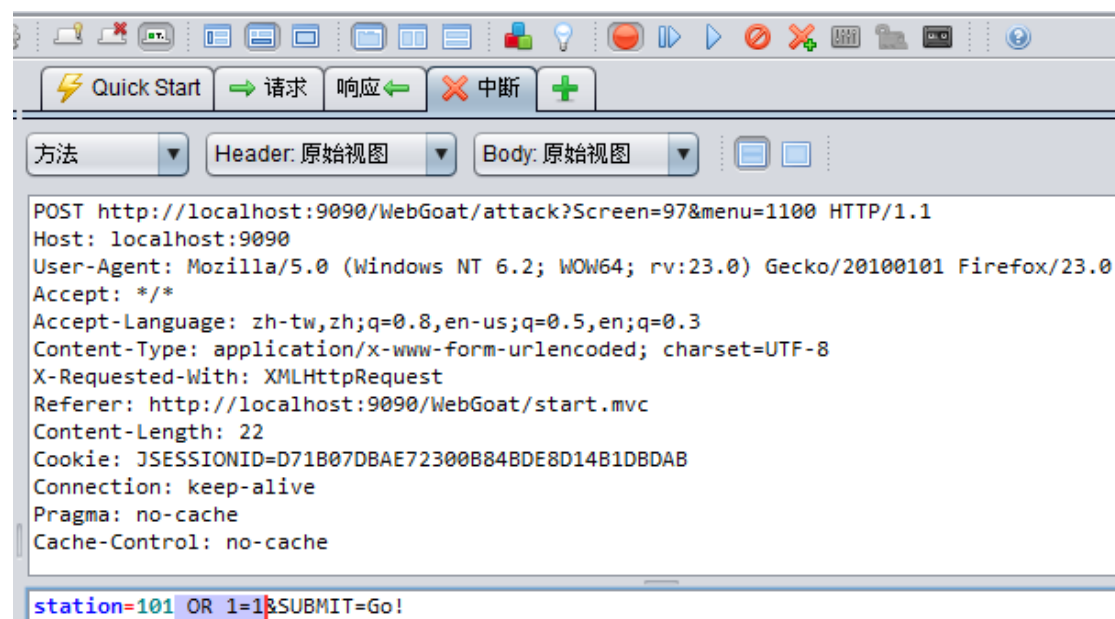
General Goal(s):

The form below allows a user to view weather data. Try to inject an SQL string that results in all the weather data being displayed.

Select your local weather station:

```
SELECT * FROM weather_data WHERE station = [station]
```

攔截請求，在 station 後補充 OR 1=1



Quick Start 请求 响应 中断

方法 Header: 原始视图 Body: 原始视图

```
POST http://localhost:9090/WebGoat/attack?Screen=97&menu=1100 HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:23.0) Gecko/20100101 Firefox/23.0
Accept: */*
Accept-Language: zh-tw,zh;q=0.8,en-us;q=0.5,en;q=0.3
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://localhost:9090/WebGoat/start.mvc
Content-Length: 22
Cookie: JSESSIONID=D71B07DBAE72300B84BDE8D14B1DBDAB
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache

station=101 OR 1=1&SUBMIT=Go!
```

就變成了 `SELECT * FROM weather_data WHERE station = 101 OR 1=1`，由於 `1=1` 恆成立，所以會列出資料庫中對應的所有資料

*** Congratulations. You have successfully completed this lesson.**

*** Bet you can't do it again! This lesson has detected your successful attack and has now switched to a defensive mode. Try again to attack a parameterized query.**

Select your local weather station:

Go!

```
SELECT * FROM weather_data WHERE station = 101 OR 1=1
```

STATION	NAME	STATE	MIN_TEMP	MAX_TEMP
101	Columbia	MD	-10	102
102	Seattle	WA	-15	90
103	New York	NY	-10	110
104	Houston	TX	20	120
10001	Camp David	MD	-10	100
11001	Ice Station Zebra	NA	-60	30