CVE-2020-35748

Author: Arcangelo Saracino (github @arkango) (twitter @Arcange17622696)
Company: Hacktive Security Srl

During one of our research activities we discovered an authenticated stored cross site scripting in the FV Flowplayer Video Player Wordpress plugin.

The FV Flowplayer Video Player Wordpress plugin boasts over 40,000 active installations and an excellent reputation in terms of reviews, easy-to-use, and complete solution for embedding FLV or MP4 videos into posts or pages.[1]

Cross site scripting (XSS) is a common attack vector that injects client-side malicious code into a vulnerable web application. XSS differs from other web attack vectors (e.g., SQL injections), in that it does not directly target the application itself. Instead, the users of the web application are the ones at risk.
Stored cross-site scripting (also known as second-order or persistent XSS) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way.
An attacker able to exploit an xss can create a phishing scenario through a fake login page and retrieve valid credentials, extraction of cookies not protected by the `HttpOnly` flag and consequent exfiltration of a valid session, simple defacement.

Tested on:

- V7.4.37.727 (latest)

Authenticated Stored Cross Site Scripting

The FV Flowplayer Video Player Wordpress plugin allows you to embed FLV or MP4 videos into posts or pages.

It receives two inputs:
- a video source
- a cover image.

In the video source parameter an attacker can be able to inject JavaScript code in order to execute a Stored Cross-Site Scripting.

[1] https://wordpress.org/plugins/fv-wordpress-flowplayer/

The POST request to create/update the videoplayer contains the `data` parameter, which contains a json string and the vulnerable field fv_wp_fvvideoplayer_src to the injection.

Function used to save the field, without escaping or check.

So we can confirm from the github source code[2] the vulnerability still exists.

Now we show how an attacker could be able to trigger the vulnerability from the wordpress frontend.

[2] https://github.com/foliovision/fv-wordpress-flowplayer

Payload execution after the video player has been saved.