Date: Sun, 15 Nov 2020 19:10:59 +0000
From: Jonathan Gregson via Fulldisclosure <fulldisclosure@...lists.org>
To: "fulldisclosure@...lists.org" <fulldisclosure@...lists.org>
Subject: [FD] Fancy Product Designer for WooCommerce - Unrestricted File
Upload

About Fancy Product Designer for WooCommerce
Fancy Product Designer for WooCommerce is a WordPress plugin which allows users to design custom products in a vendor's WooCommerce store. It is sold through the third-party marketplace "Envato Market" and boasts over 15,000 sales.

Unrestricted File Upload
Fancy Product Designer for WooCommerce before and including version 4.5.1 contains an Unrestricted File Upload vulnerability.

An unauthenticated attacker is able to upload any type of file to an affected WooCommerce store by exploiting a Time of Check, Time of Use (TOCTOU) weakness in custom-image-handler.php's `url` parameter. However, the file will be saved with one of the following extensions on the server: jpeg, png, or svg.

Fancy Product Designer for WooCommerce provides an option to require users to log in before uploading images. However, an attacker is able to access the custom-image-handler.php file directly and upload arbitrary files without authentication.

Details
The custom-image-handler.php file provides an interface where unauthenticated users can provide the URL of an image and have that image saved on the server. Before saving the file, custom-image-handler.php first checks the MIME type of the file and confirms that it is one of the following: jpeg, png, or svg. Once the file's MIME type has been verified, custom-image-handler.php downloads the file a second time and saves the most recent copy to the filesystem with an extension corresponding to the MIME type of the originally verified file.

This vulnerability can be exploited by sending a POST request to the following URL of an affected WooCommerce site:

    /wp-content/plugins/fancy-product-designer/inc/custom-image-handler.php

With this request, the following POST parameters should be sent as form data:
 - saveOnServer: 1
 - uploadsDirURL: https://[affected site]/wp-content/uploads/fancy_products_uploads/
 - uploadsDir: [full path to where the file should be saved on the server]
 - url: [attacker-controlled URL]

The attacker-controlled URL should initially point to a valid file of type: jpeg, png, or svg. Once a request has been made for the file, the attacker can replace the file at the attacker-controlled URL with a malicious file, and the malicious file will be saved on the server. The URL of the uploaded file will then be returned to the attacker in response to the POST request.

Note: The default values for the `uploadsDirURL` and `uploadsDir` parameters can be found by searching for the same strings in the source of any page using an affected version of Fancy Product Designer for WooCommerce. Attackers are also able to provide local file paths, which will cause FPD to copy the specified file into the uploads directory if the file is one of the expected types and the server is running with sufficient permissions to read the file.

Impact
The fact that the uploaded file will have one of the previously mentioned extensions greatly mitigates the impact of this vulnerability, as none of the allowed extensions will be executed by the server. The following scenarios are plausible by abusing this vulnerability:
 - Stored XSS by uploading an SVG containing a malicious JavaScript payload
 - Malware distribution by uploading malicious binaries and other payloads

Proof of Concept
 - Exploit code: [poc.php](https://github.com/jdgregson/Disclosures/blob/master/fancy-product-designer/unrestricted-file-upload/poc.php)
 - Demo video: [unrestricted-file-upload.mp4](https://raw.githubusercontent.com/jdgregson/Disclosures/master/fancy-product-designer/unrestricted-file-upload/unrestricted-file-upload.mp4)

Disclosure Timeline
 - 10/11/2020: issue reported via ticket on developer's support forum
 - 10/11/2020: ticket closed by developer with no response
 - 10/20/2020: developer released an update which did not address the issue
 - 10/26/2020: developer released an update which did not address the issue
 - 11/14/2020: full disclosure