CVE-2020-13405: MicroWeber Unauthenticated User Database Disclosure

Hunter Stanton

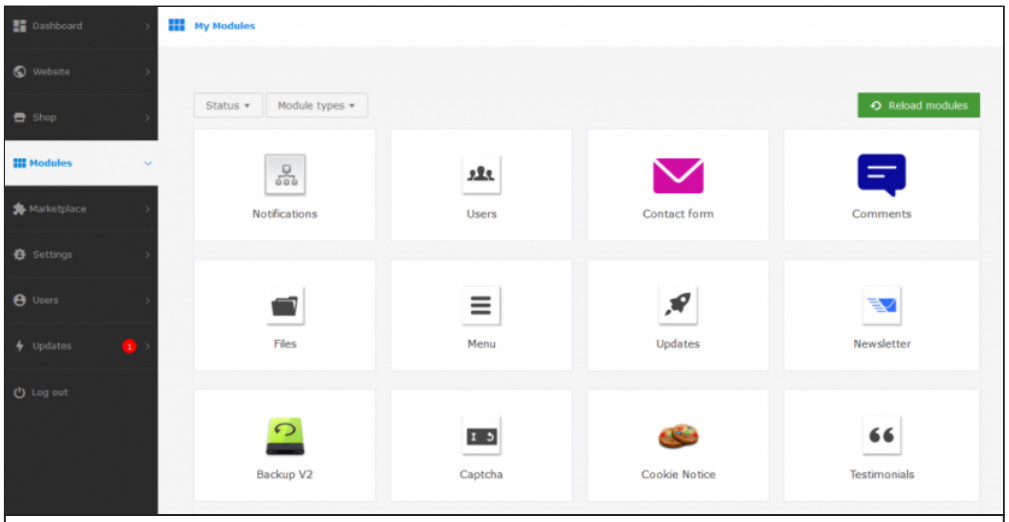
Introduction to Vulnerability Findings in MicroWeber

MicroWeber is an open-source Content Management System (CMS) written in PHP. It allows web administrators to easily build a website by dragging and dropping components where they want them to be. It is a popular choice among those looking to start a website that is both easy to set up and is very customizable.

In this blog post, I will be detailing a critical vulnerability I discovered in MicroWeber where an attacker could disclose the entire users database, which includes admin password hashes and emails. No authentication was required to exploit the vulnerability, making it very high impact.

MicroWeber Internals and Script Customization

Being a PHP-based CMS, MicroWeber calls upon multiple PHP scripts to handle different parts of its functionality. This allows MicroWeber to be customizable and users can plug in their own scripts or easily modify existing ones to change how the CMS operates.



The Modules section of the MicroWeber Admin Panel

The MicroWeber Modules System

One such piece of this customization functionality is the aptly named modules system. Modules allow for different functions like embedding Tweets or a search utility to be added into MicroWeber. This allows web administrators to easily extend the functionality of their MicroWeber site by finding or writing a module that adds the functionality they need.

Identifying The Vulnerability

The vulnerability was discovered in the “controller.php” script, which is part of MicroWeber’s users module.

```
<?php
dd(User::all());
```

The PHP code for controller.php

This PHP script does nothing but run Laravel's dump and die function on the users database. Dump and die simply prints the contents of the entire PHP variable (in this case, the users database) out to HTML and then halts execution of the script, hence the "dump and die".

Invoking Module Functionality

For a module's functionality to be invoked by a user, a user must submit a POST request to the /modules/ endpoint which has to contain a few arguments, the most important of which is the "module" parameter. This parameter is the path to the PHP script for the particular module that is being invoked.

To trigger the vulnerability, a user just needs to submit the following POST request to the /modules/ endpoint:

```
module=/modules/users/controller
```

When this is submitted, the "controller.php" script will be executed, and the entire users database will appear in the response. This requires no authentication as MicroWeber allows modules to be invoked without being authenticated.

The users database being returned in the web browser after successful exploitation.

Post Exploitation

When the vulnerability is successfully exploited, you have access to the entire users database, including administrator users. Every bit of information that is stored about an individual user by MicroWeber is revealed.

Cracking Password Hashes

MicroWeber does not store passwords as plain-text, so the hashes must be cracked. By default, MicroWeber is using bcrypt for hashing. MicroWeber administrators are free to implement another hash algorithm if they want, but the only supported option out of the box is bcrypt. These hashes are crackable with Hashcat under the default configuration, making it feasible for an attacker to obtain administrator passwords by cracking the hash.

The MicroWeber Fix

MicroWeber staff fixed this vulnerability by removing controller.php from the MicroWeber source code. According to a MicroWeber developer, Controller.php was a leftover from the early days of MicroWeber's development. Now that controller.php has been removed from the source, this vulnerability is no longer possible to exploit.

Conclusion

Disclosing this vulnerability was overall a positive experience. The MicroWeber team was prompt and responsive in getting it fixed up and I thank them for that. I will definitely be researching MicroWeber more in the future and look forward to our next interaction.

Thanks for reading! Check back frequently for more blog posts, research, and tool releases. In the meantime, follow us on Twitter for news and updates: @RhinoSecurity, @Hun10sta

Timeline of Events

04-27-2020 – Vulnerability Disclosed to MicroWeber

05-22-2020 – MicroWeber staff confirms the vulnerability and begins working on a fix. The CVE is assigned CVE-2020-13405 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-13405)

06-22-2020 – MicroWeber 1.1.20 is released which fixes the vulnerability

(https://rhinosecuritylabs.com)

Related Resources

(https://rhinosecuritylabs.com/research/cve-2022-26113-forticlient-arbitrary-file-write-as-system/)

CVE-2022-26113: FortiClient Arbitrary File Write As SYSTEM

(https://rhinosecuritylabs.com/research/cve-2021-41577-evga-precision-x1/)

CVE-2021-41577: MITM to RCE in EVGA Precision X1

(https://rhinosecuritylabs.com/research/cve-2020-5377-dell-openmanage-server-administrator-file-read/)

CVE-2020-5377: Dell OpenManage Server Administrator File Read

Interested in more information?

20603

Contact Us Today

ABOUT US

Rhino Security Labs is a top penetration testing and security assessment firm, with a focus on cloud pentesting (AWS, GCP, Azure), network pentesting, web application pentesting, and phishing. With manual, deep-dive engagements, we identify security vulnerabilities which put clients at risk.

Endorsed by industry leaders, Rhino Security Labs is a trusted security advisor to the Fortune 500.