Multiple Vulnerabilities in DiveBook WordPress Plugin (v.1.1.4)

2020-09-15

Background

The DiveBook plugin for WordPress was prone to a multiple vulnerabilities including SQL injection, cross-site scripting, and improper authorization (CVE-2020-14205, CVE-2020-14206, CVE-2020-14207). An attacker could leverage these issues to dump the database including administrative user credentials, to steal cookie-based authentication credentials, or launch other attacks. Other versions are likely affected, though they were not tested.

Download Link: DiveBook

Vulnerable version: v.1.1.4.

This WordPress plugin is a solutions for dive enthusiasts to collaboratively log and track their dives.
From the WordPress website: The DiveBook plugin consists of three main functionality areas:

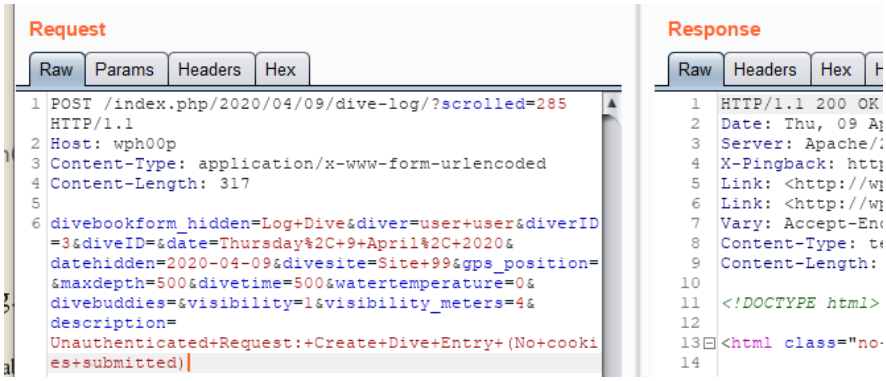
1. The main part of the plugin can be injected in pages or posts (statistics about logged dives, input form to easily log new dives and edit existing dives, and list and information about all logged dives)
2. Settings page for the WordPress Administrator where general settings for the plugin can be stored (not used in Beta version).
3. Sidebar widget that will show key information about the most recent logged dives. You can specify how many dives to be shown in the widget.

Vulnerability - Improper Authorization Check (CVE-2020-14205)

An authorization issue is present in the DiveBook "Add New Dive" feature. An anonymous user may create a new dive entry with a crafted HTTP POST.

Steps:

1. Identify a WordPress application with the DiveBook (v.1.1.4) plugin installed (hxxps://example.wordpress.com).
2. Locate a page with the DiveBook content included. Note: within the WordPress page editor interface, "[divebook]divebook_display();[/divebook]" will be included.
3. Submit a crafted POST request containing the required parameters (see HTTP request below).
4. Observe that a new dive has been logged within the database, which is visible on the page.



The screenshot above contains the necessary parameters to create a new dive entry without supplying any cookies.

Vulnerability - Unauthenticated Reflected XSS (CVE-2020-14206)

A reflected cross-site scripting vulnerability exists within the DiveBook log's filter functionality. Arbitrary URL parameters are reflected into the application's response, rendered by the browser as HTML or JavaScript. An attacker may abuse this functionality by sending a victim a crafted link containing JavaScript, which will execute within the context of the victim's browser. The "scrolled" parameter is also vulnerable.

Steps:

1. Identify a WordPress application with the DiveBook (v.1.1.4) plugin installed (hxxps://example.wordpress.com).
2. Locate a page with the DiveBook content included. Note: within the WordPress page editor interface, "[divebook]divebook_display();[/divebook]" will be included.
3. Browse to a URL triggering the "filter" functionality and include an arbitrary parameter containing HTML. Example: hxxps://example.wordpress.com/index.php/2020/06/15/hello-world/?filter_diver=0&filter_divesite=site&divelog_page=1&scrolled=798&foobar"<>script>alert`XSS+in+arbitrary+URL+parameter`</script>"<
4. Observe that JavaScript executes within the web page (an alert box appears).

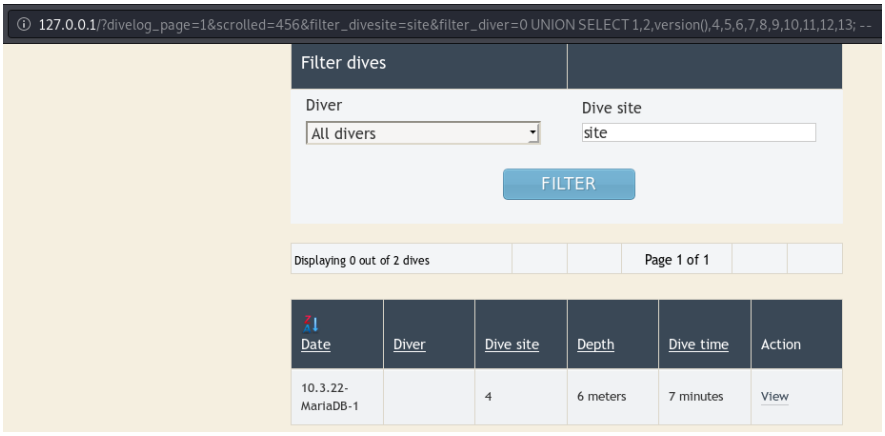
The screenshot above demonstrates the attack (request, response, and result). Observe that a JavaScript "alert" box appears with arbitrary content. This payload may be modified to execute arbitrary JavaScript in the victim's browser.

Vulnerability - SQL Injection (CVE-2020-14207)

A SQL injection vulnerability exists within the DiveBook log's filter functionality. Though the plugin escapes some user input (quotes), complete compromise of the application database is possible by injecting the "filter_diver" GET parameter. Because the WordPress administrative credentials are stored within the database, this may result in complete or partial application compromise.

Steps:

1. Identify a WordPress application with the DiveBook (v.1.1.4) plugin installed (hxxps://example.wordpress.com).
2. Locate a page with the DiveBook content included. Note: within the WordPress page editor interface, "[divebook]divebook_display();[/divebook]" will be included.
3. Browse to a URL triggering the "filter" functionality and injecting into the back-end SQL statement: hxxps://example.wordpress.com/?divelog_page=1&scrolled=456&filter_diver=0%20UNION%20SELECT%201,2,version(),4,5,6,7,8,9,10,11,12,13;%20--
4. Observe that the MySQL version is returned within the page.



The screenshot above demonstrates the attack and MySQL version returned in the page. A few more crafted queries would result in the password hashes and usernames of all WordPress users for the instance.

Testing Conditions:

- Firefox v68.9.0esr
- WordPress 5.4.2

Vulnerability Disclosure Policy

Hooper Labs takes security issues seriously. We believe in working with relevant stakeholders to achieve coordinated disclosure within a reasonable period of time. We also adhere to the industry-standard 90-day disclosure deadline, where vendors are notified of vulnerabilities immediately, with details shared to the public after 90 days (or sooner if the issues are resolved earlier).

Common Vulnerabilities and Exposures (CVEs) are an industry standard for identifying vulnerabilities (link). This system is a method for reference and tracking of publicly-known exposures. A CVE is a way to uniquely reference vulnerabilities across systems and Mitre Corporation is the primary CVE Numbering Authority (CNA) for the program. We believe that users have a right to know their exposures in order to make informed risk decisions.

Hooper Labs does not participate in bug bounty programs, but instead relies on responsible disclosure (link). Effectively communicating vulnerabilities and risks to the vendor, users, and public ensure that risk can be documented, calculated, and mitigated. We hope that through this process the Information Domain may be marginally safer.

Contact Us

If you have any questions, suggestions, or concerns, please reach out on Twitter (@nopantrootdance). Feel free to contribute to any of the projects on GitHub.