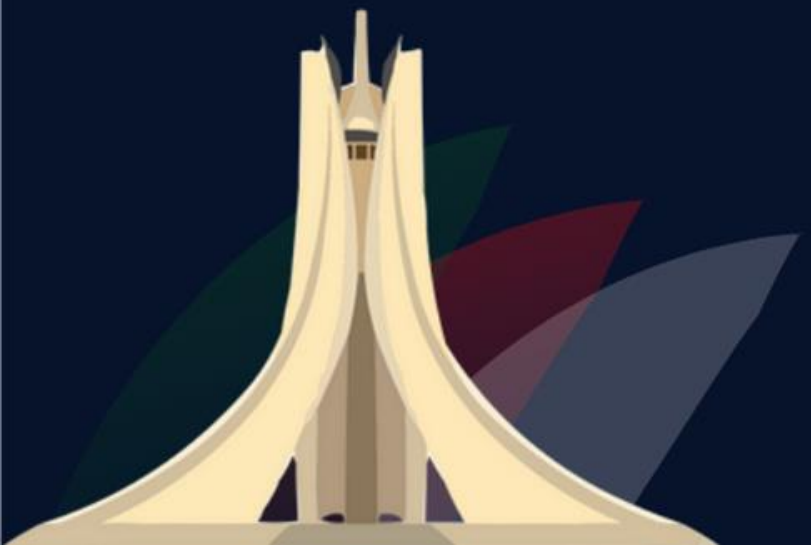
OWASP ALGIERS



OWASP ALGIERS

SPEAKER

Ahmed Hassan

Cyber Security Consultant & Penetration Tester AUSTRIA (Vienna)

- Cyber Security Consultation since 6 years (more in the Self-Introduction section)

Agenda

1. Self-Introduction
2. What is a CVE?
3. Where can you find a suitable open-source Application for Testing?
4. Installing the open-source Application
5. Identification of a Vulnerability in the open-source Application
6. Reporting Identified Vulnerabilities: Methods and Platforms
7. CVE Acceptance and Publication
8. Questions from the audience & further Explanations (Q&A)?

Self-Introduction

Working since almost 6 years as a Cyber Security Engineer & Penetration Tester @ Condignum, AUSTRIA

Speaker at various international security conferences & Universities (Egypt, UAE, Saudi Arabia, Austria etc.)

Securing the Clients environment (Web, API, Active Directory, Mobile, Infrastructure etc.)

BSIDES COLORADO SPRINGS

جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

black hat MIDDLE EAST AND AFRICA

Constra Tech

HACK RED CON

2023 Arab Security Conference & Exhibition

Faculty of Computer and Information Sciences 1995

أكاديمية ربدان
Rabdan Academy

AOU
الجامعة العربية المفتوحة
Arab Open University
Egypt

OWASP ALGIERS

Qualifications & professional Certifications

studied and working in Austria in the Cyber Security Field as Cyber Security Engineer, Penetration Tester and Bug Bounty Hunter

Certifications for advanced IT professionals

Certified EC-council Instructor

Certified Ethical Hacker

Offensive Security Certified Professional

Offensive Security Web Assessor

OWASP ALGIERS

Identified Vulnerabilities & 0-day Vulnerabilities

- **Identified more than 52 CVEs. Some as an example:**
- **My GitHub Repo-Link: https://github.com/ahmedvienna/CVEs-and-Vulnerabilities**

  - CVE-2024-0351
  - CVE-2024-0350
  - CVE-2024-0349
  - CVE-2024-0348
  - CVE-2024-0347
  - CVE-2024-0262
  - CVE-2024-1972
  - CVE-2024-1922
  - CVE-2024-1919
  - CVE-2024-3735
  - CVE-2024-7466

Hall of Fame announcement and CVE assignment, for identifying a Zero-day Vulnerability in CISCO's Devices.

What is a CVE ?

CVE-Explanation

The Common Vulnerabilities and Exposures (CVE) system is a framework operated by the U.S. National Cybersecurity FFRDC and maintained by the Mitre Corporation. It provides standardized identification and naming of publicly known security vulnerabilities and other weaknesses in computer systems.

The primary goal of the CVE system is to prevent multiple naming of the same threats by different organizations and institutions. Each known vulnerability is assigned a unique identifier, which consists of the prefix CVE, the year of discovery, and a sequential number (e.g., CVE-2020-1234). This ensures the consistent identification of vulnerabilities and facilitates smooth information exchange between the various databases maintained by individual vendors.

Criteria for requesting a CVE

Detailed Process how CVEs are reported and assigned: https://github.com/CVEProject/cveproject.github.io/blob/master/requester/reservation-guidelines.md

The Common Vulnerabilities and Exposures (CVE) identifier can be requested when a security vulnerability in a software product or system meets specific criteria.

This process typically applies to different applications such as Desktop applications or even Web applications. It is essential to first clarify with the vendor whether they are authorized to request or assign CVEs.

The basic process for reserving a CVE ID is as follows:

1. Determine if a CVE ID is needed and appropriate. If yes,
2. Contact a vendor whose product is affected to disclose a vulnerability (coordinated disclosure).
3. Determine whether the request should be made to a vendor CNA. If no,
4. Determine whether the request should be made to a third party coordinator CNA, or to a disclosure mailing list. If no,
5. Request a CVE ID from DWF
6. Request a CVE ID from MITRE using the CVE Request web form.
7. Provide the required information in the request.
8. Receive a confirmation email with a reference number and save it for your records.
9. Provide follow-up information as needed.
10. Receive a CVE ID (or an explanation if a CVE ID was not provided)
11. Share the CVE ID with all parties.
12. Include the CVE ID in the announcement of the vulnerability.
13. Notify MITRE that the vulnerability has been made public using the CVE Request web form, and selecting "Notify CVE about a Publication."

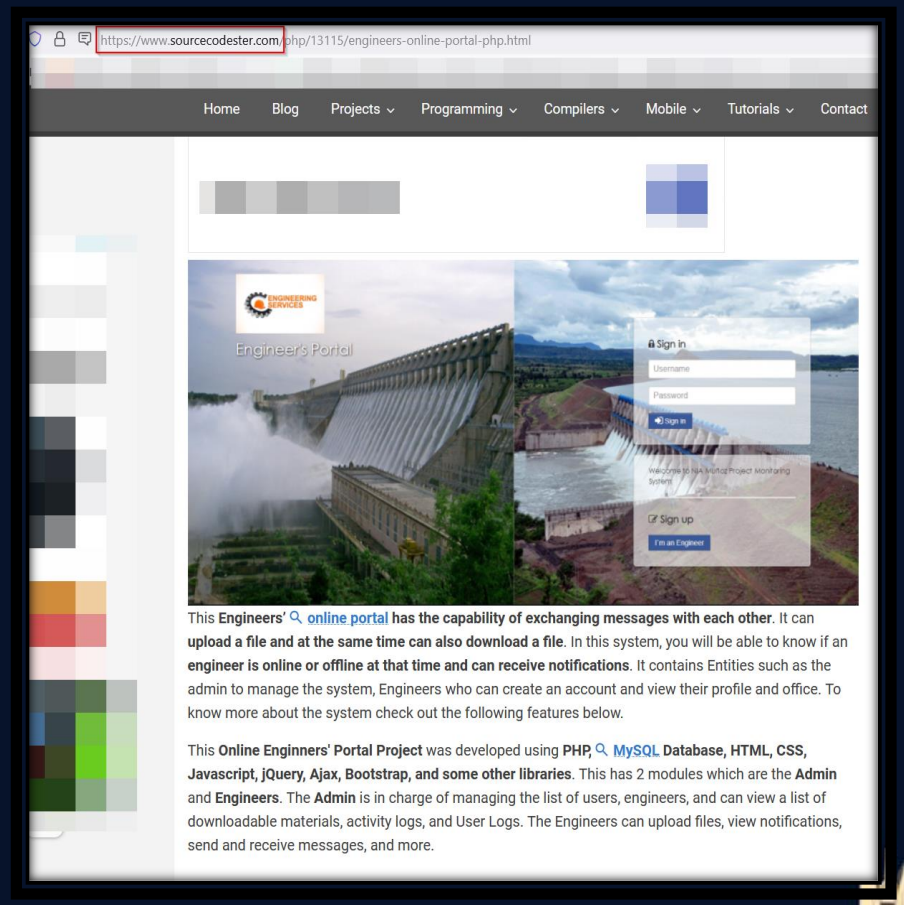
The CVE is then published by MITRE and will appear on the CVE List.

Workflow for CVE Assignment

This outlines the complete workflow, from identifying the vulnerability to receiving a CVE assignment.

Where can you find a suitable open-source Application for Testing?

1. This website hosts a vast collection of open-source applications, of which 99% are legitimate candidates for requesting a CVE.
2. The Website link: https://www.sourcecodester.com/



Installing the open-source Application

The installation of the application is well-documented, making it easy and straightforward to download.

- **Download** and **Install** any **local web server** such as **XAMPP/WAMP.**
- **Download** the provided source code **zip** file. (*download button is located below*)

Installation/Setup

1. **Open** your **XAMPP/WAMP's Control Panel** and start the `Apache` and `MySQL` .
2. **Extract** the **downloaded source code zip file**.
3. If you are using **XAMPP**, **copy** the extracted source code folder and **paste** it into the **XAMPP's "htdocs" directory**. And If you are using **WAMP**, **paste** it into the **"www" directory.**
4. **Browse** the `PHPMyAdmin` in a **browser**. i.e. `http://localhost/phpmyadmin`
5. **Create** a **new database** naming `capstone` .
6. **Import** the provided `SQL` file. The file is known as `capstone.sql` located inside the **db** folder.
7. **Browse** the **Engineers Online Portal Project** in a **browser**. i.e. `http://localhost/nia_munoz_monitoring_system` and `http://localhost/nia_munoz_monitoring_system/admin` for the admin side.

Default Admin Access

Username: **admin**
Password: **admin**

Identification of a Vulnerability in the open-source Application

Assuming we have identified a vulnerability in this open-source application, it is important to thoroughly document all details before proceeding to the reporting phase.

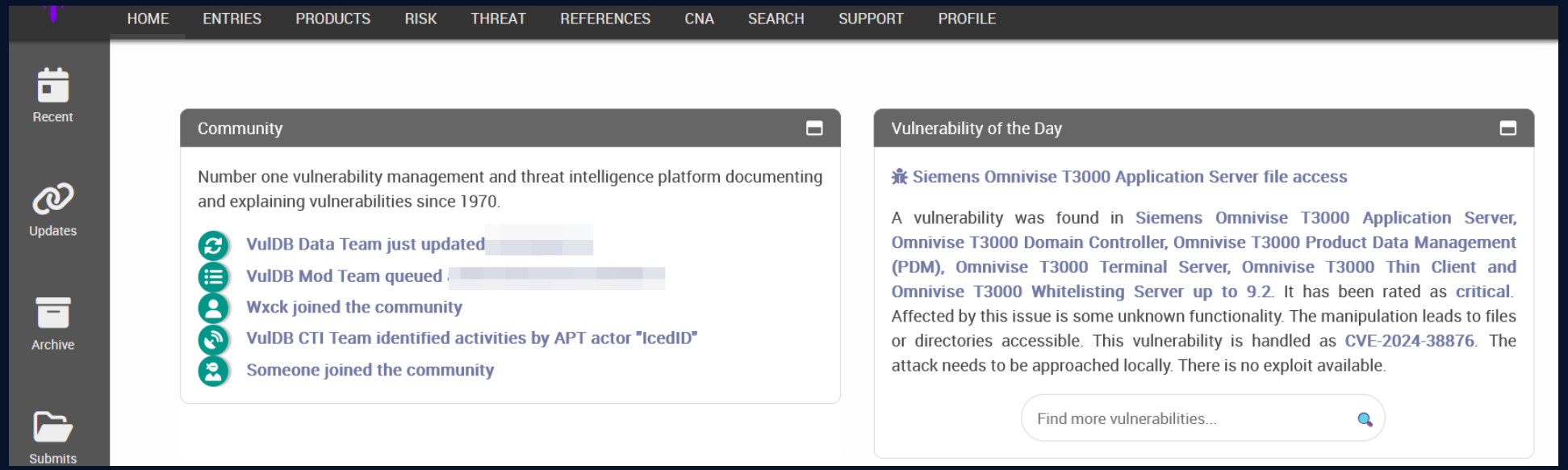
In this phase, we need to collect screenshots, videos, Payloads used and both the request and response data to ensure we provide the vendor with all necessary information.

Reporting identified Vulnerabilities: Methods and Platforms

If you can locate the vendor details, such as those for Cisco, and contact their security team, this should be your first step. They may respond and have the capability to assign a CVE.

In cases where you cannot find the security team or if the company shows no interest in the security aspects of their products, you will need to reach out to a recognized CVE authority for assistance in assigning a CVE after reviewing your vulnerability.

The recognized CVE authority Website: https://vuldb.com/

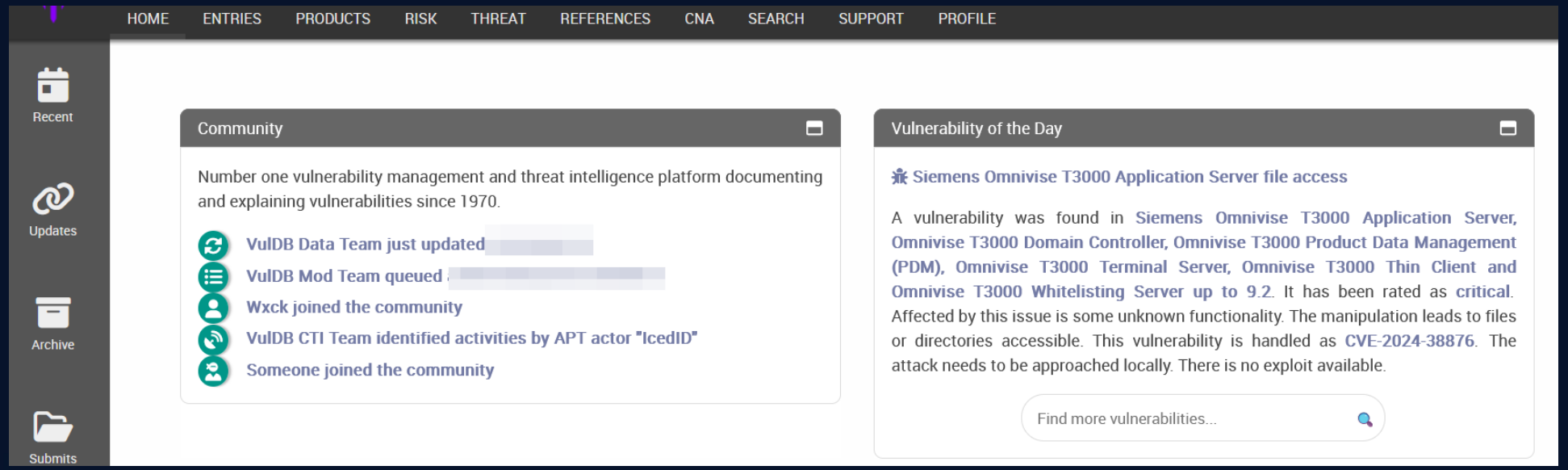


Detailed Procedure for reporting a Vulnerability

If you can locate the vendor details, such as those for Cisco, and contact their security team, this should be your first step. They may respond and have the capability to assign a CVE.

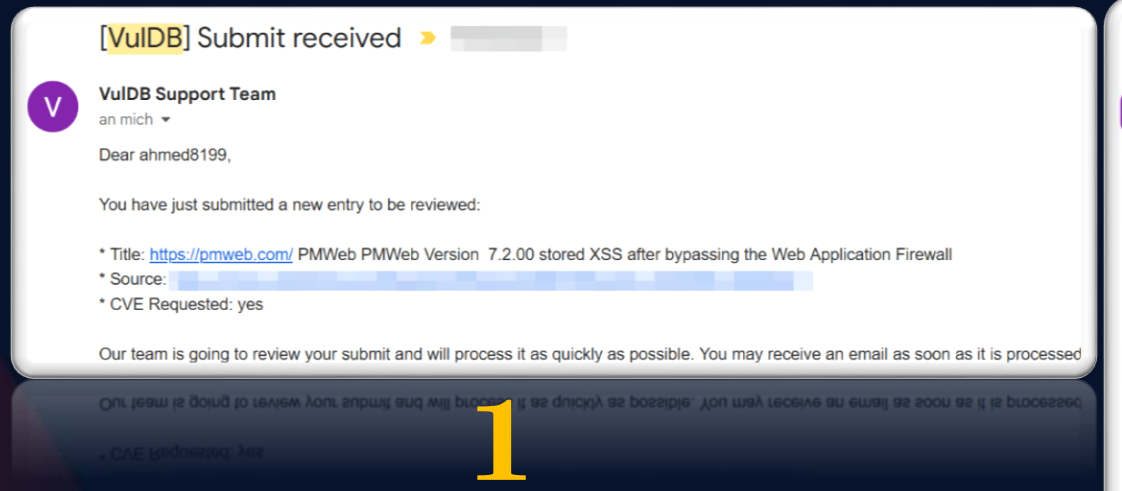
In cases where you cannot find the security team or if the company shows no interest in the security aspects of their products, you will need to reach out to a recognized CVE authority for assistance in assigning a CVE after reviewing your vulnerability.

The recognized CVE authority Website: https://vuldb.com/

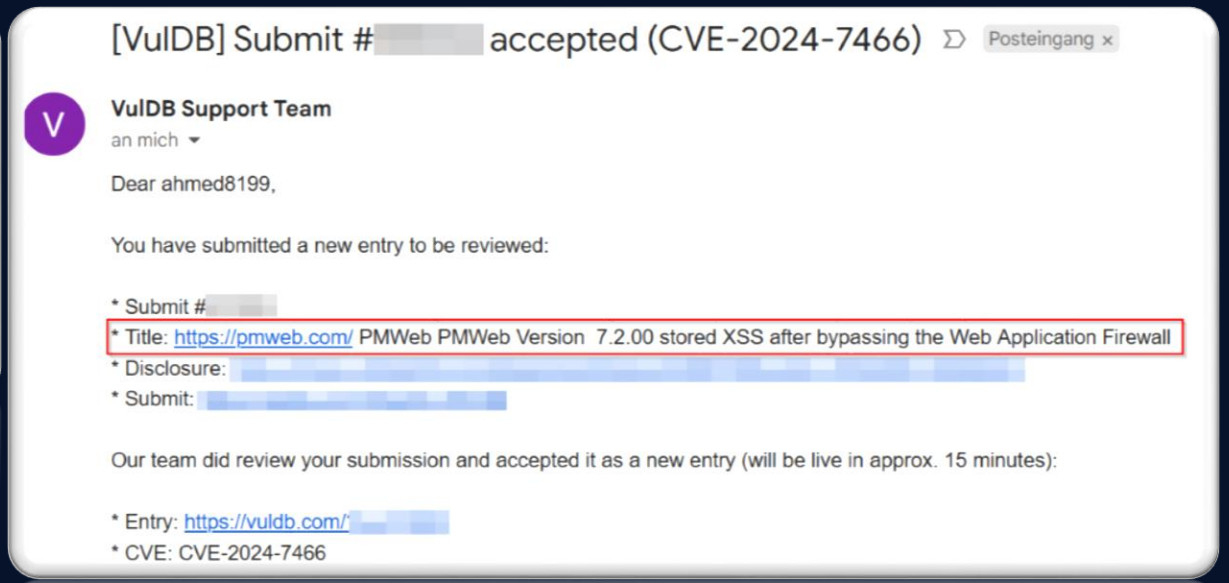


Submitting the form and receiving the Response

Here we can review the message received after submitting the CVE request, along with the response confirming the approval and issuance of a new, valid CVE.

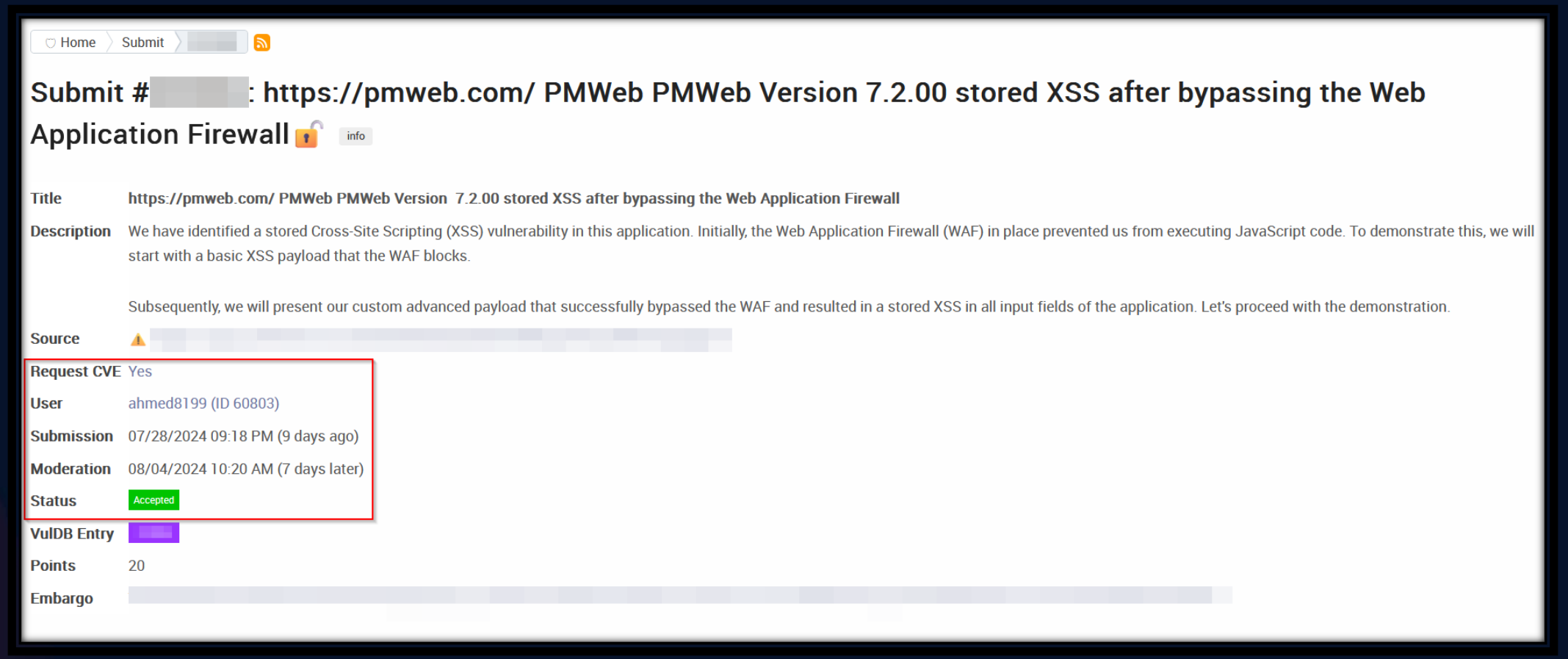


1



2

Successful CVE Acceptance and Publication



Home › Submit

Submit #[redacted]: https://pmweb.com/ PMWeb PMWeb Version 7.2.00 stored XSS after bypassing the Web Application Firewall info

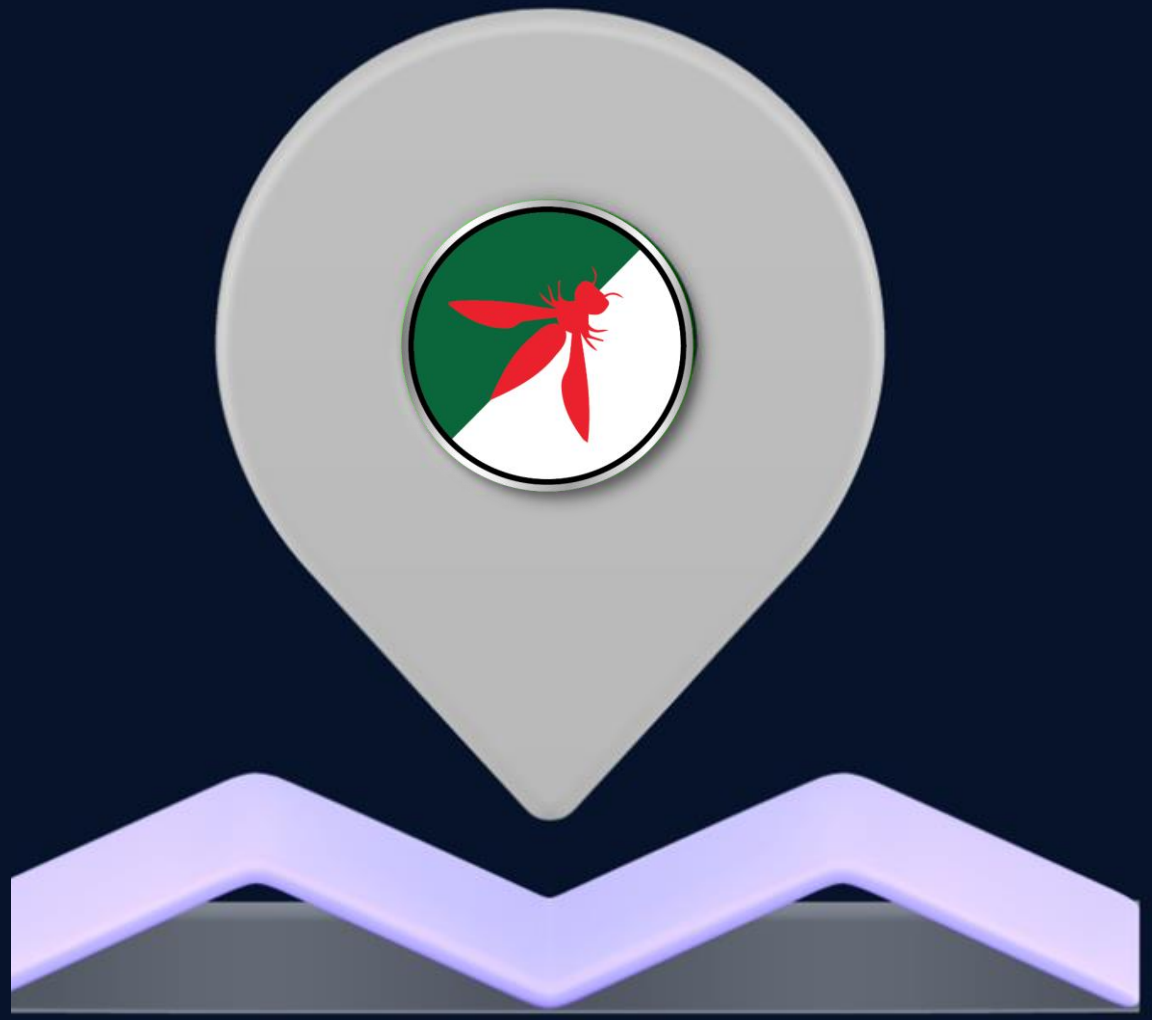
Title	https://pmweb.com/ PMWeb PMWeb Version 7.2.00 stored XSS after bypassing the Web Application Firewall
Description	We have identified a stored Cross-Site Scripting (XSS) vulnerability in this application. Initially, the Web Application Firewall (WAF) in place prevented us from executing JavaScript code. To demonstrate this, we will start with a basic XSS payload that the WAF blocks. Subsequently, we will present our custom advanced payload that successfully bypassed the WAF and resulted in a stored XSS in all input fields of the application. Let's proceed with the demonstration.
Source	[redacted]
Request CVE	Yes
User	ahmed8199 (ID 60803)
Submission	07/28/2024 09:18 PM (9 days ago)
Moderation	08/04/2024 10:20 AM (7 days later)
Status	Accepted
VulDB Entry	[redacted]
Points	20
Embargo	[redacted]

THANK YOU

My LinkedIn: https://www.linkedin.com/in/ahmed-hassan-79559487/

OWASP ALGIERS

Contact us



ALGIERS-LEADERS@OWASP.ORG

https://owasp.org/www-chapter-algiers/