New issue

Reflected Cross-Site Scripting (XSS) in /admin/Add notice/batch-notice.php via $_SERVER['PHP_SELF'] #238

Open

duckpigdog opened 2 weeks ago

Reflected Cross-Site Scripting (XSS) in /admin/Add notice/batch-notice.php via $_SERVER['PHP_SELF']

Describe the bug
A reflected Cross-Site Scripting (XSS) vulnerability exists in `/admin/Add notice/batch-notice.php` at line 190. The script uses the unsanitized `$_SERVER['PHP_SELF']` variable as the form action attribute, allowing an attacker to inject arbitrary JavaScript code through a crafted URL.

Steps to reproduce

1. Log in to the admin panel using the preset admin account (username: `admin`, password: `admin123`).
2. Access the following crafted URL in a browser (the vulnerable page is only accessible after login):

http://127.0.0.1:3000/admin/Add%20notice/batch-notice.php/%22%3E%3Cscript%3Ealert('XSS_POC')%3C/script%3E

3. The browser will execute the injected JavaScript and display an alert box with the text `XSS_POC`, confirming the vulnerability.

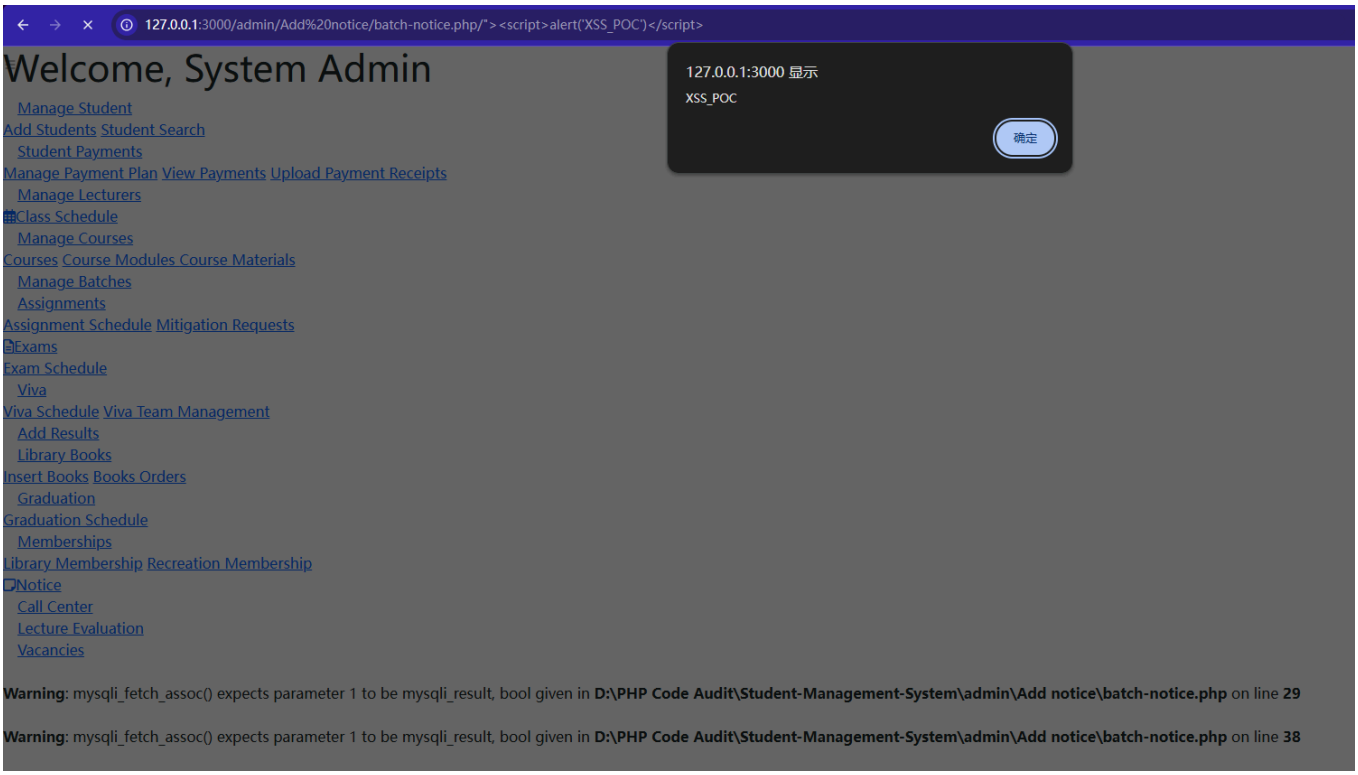
The malicious code can be replaced with more harmful scripts (e.g., to steal cookies, perform phishing, or hijack the admin session).

Describe the solution
To fix this vulnerability, the `action` attribute should not directly output user-controllable input. Instead, use a hardcoded relative URL or properly sanitize the output.

Recommended fix:

- Replace `<?php echo $_SERVER['PHP_SELF']; ?>` with a static value, such as `""` (post to same page) or `"batch-notice.php"`.
- If dynamic values are necessary, use `htmlspecialchars()` to encode output:
  `<?php echo htmlspecialchars($_SERVER['PHP_SELF'], ENT_QUOTES, 'UTF-8'); ?>`

Screenshots



duckpigdog 2 weeks ago Author • • •

During the audit, it was found that no *.sql table creation scripts are provided in the project repository (the result of executing Glob **/*.sql is empty), and README.md only contains project introduction and video links, without providing database import steps. To complete the system deployment, the database initialization script was generated by manually analyzing the code logic. Copy the following code, save it as a .sql file, and import it into MySQL for use.

```
-- Database initialization script for Student Management System
-- Generated based on code analysis

CREATE DATABASE IF NOT EXISTS `student_management_system` DEFAULT CHARACTER SET utf8mb4 COLLA
USE `student_management_system`;

-- 1. Authentication & Users
CREATE TABLE IF NOT EXISTS `login_tbl` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `username` VARCHAR(100) NOT NULL UNIQUE,
  `password` VARCHAR(255) NOT NULL,
  `student_name` VARCHAR(255),
```

```
  `course` VARCHAR(100),
  `batch_number` VARCHAR(50),
  `gender` VARCHAR(20),
  `dob` DATE,
  `nic` VARCHAR(50),
  `email` VARCHAR(100),
  `contact` VARCHAR(20),
  `awarding_uni` VARCHAR(100),
  `uni_number` VARCHAR(100),
  `lec` VARCHAR(100),
  `role` VARCHAR(20) DEFAULT 'Student'
);

CREATE TABLE IF NOT EXISTS `admin_login_tbl` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `username` VARCHAR(100) NOT NULL UNIQUE,
  `password` VARCHAR(255) NOT NULL,
  `name` VARCHAR(255),
  `gender` VARCHAR(20)
);

CREATE TABLE IF NOT EXISTS `lecturers` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `username` VARCHAR(100) NOT NULL UNIQUE,
  `password` VARCHAR(255) NOT NULL,
  `name` VARCHAR(255),
  `department` VARCHAR(100),
  `email` VARCHAR(100),
  `dob` DATE,
  `gender` VARCHAR(20),
  `nic` VARCHAR(50),
  `contact` VARCHAR(20)
);

-- 2. Academics & Courses
CREATE TABLE IF NOT EXISTS `course_tbl` (
  `course_id` INT AUTO_INCREMENT PRIMARY KEY,
  `course_name` VARCHAR(255) NOT NULL,
  `award_uni` VARCHAR(255)
);

CREATE TABLE IF NOT EXISTS `modules` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `course` VARCHAR(100),
  `module_name` VARCHAR(255),
  `module_code` VARCHAR(50),
  `date` DATE,
  `duration` VARCHAR(50),
  `num_assignments` INT DEFAULT 0
);

CREATE TABLE IF NOT EXISTS `course_materials` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `module_name` VARCHAR(255),
  `module_code` VARCHAR(50),
  `topic` VARCHAR(255),
```

```
  `batch_number` VARCHAR(50),
  `course` VARCHAR(100),
  `download` VARCHAR(255)
);

CREATE TABLE IF NOT EXISTS `class_schedule` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `course` VARCHAR(100),
  `batch` VARCHAR(50),
  `module` VARCHAR(100),
  `lecturer` VARCHAR(100),
  `date` DATE,
  `time` VARCHAR(50),
  `hall` VARCHAR(100),
  `notes` TEXT
);

-- 3. Assignments & Exams
CREATE TABLE IF NOT EXISTS `assignment_schedule` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `course` VARCHAR(100),
  `batch_number` VARCHAR(50),
  `module_name` VARCHAR(255),
  `module_code` VARCHAR(50),
  `assignment_name` VARCHAR(255),
  `date_of_issue` DATE,
  `date_of_submit` DATE,
  `view` BOOLEAN DEFAULT 1,
  `status` VARCHAR(50),
  `feedback` TEXT,
  `mitigation_request` BOOLEAN DEFAULT 0,
  `allow_submission` BOOLEAN DEFAULT 1
);

CREATE TABLE IF NOT EXISTS `assignments` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `username` VARCHAR(100),
  `batch_number` VARCHAR(50),
  `module_name` VARCHAR(255),
  `module_code` VARCHAR(50),
  `assignment_name` VARCHAR(255),
  `submission_date` DATETIME DEFAULT CURRENT_TIMESTAMP,
  `feedback` TEXT,
  `file_path` VARCHAR(255),
  `results` DECIMAL(5,2)
);

CREATE TABLE IF NOT EXISTS `exam_schedule` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `course` VARCHAR(100),
  `module_name` VARCHAR(255),
  `module_code` VARCHAR(50),
  `batch_number` VARCHAR(50),
  `exam_name` VARCHAR(255),
  `date` DATE,
  `time` VARCHAR(50),
```

```
  `location` VARCHAR(100),
  `hours` VARCHAR(50),
  `allow_submission` BOOLEAN DEFAULT 1
);

CREATE TABLE IF NOT EXISTS `exam_submission` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `username` VARCHAR(100),
  `batch_number` VARCHAR(50),
  `module_name` VARCHAR(255),
  `module_code` VARCHAR(50),
  `exam_name` VARCHAR(255),
  `file_path` VARCHAR(255),
  `results` DECIMAL(5,2),
  `submission_date` DATETIME DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE IF NOT EXISTS `viva_schedules` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `course` VARCHAR(100),
  `module_name` VARCHAR(255),
  `module_code` VARCHAR(50),
  `batch_number` VARCHAR(50),
  `viva_name` VARCHAR(255),
  `date` DATE,
  `location` VARCHAR(100)
);

-- 4. Notices & Misc
CREATE TABLE IF NOT EXISTS `notice` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `subject` VARCHAR(255),
  `added_date` DATE,
  `view_link` VARCHAR(255)
);

CREATE TABLE IF NOT EXISTS `vacancy_tbl` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `title` VARCHAR(255),
  `content` TEXT,
  `image` VARCHAR(255)
);

CREATE TABLE IF NOT EXISTS `vacancy_apply_tbl` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `job_id` INT,
  `job_title` VARCHAR(255),
  `student_id` VARCHAR(100),
  `name` VARCHAR(255),
  `batch_num` VARCHAR(50),
  `gender` VARCHAR(20),
  `email` VARCHAR(100),
  `contact` VARCHAR(20)
);

CREATE TABLE IF NOT EXISTS `payment_summary_tbl` (
```

```
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `username` VARCHAR(100),
  `tot_course_fee` DECIMAL(10,2) DEFAULT 0.00,
  `amount_paid` DECIMAL(10,2) DEFAULT 0.00,
  `outstanding` DECIMAL(10,2) DEFAULT 0.00
);

CREATE TABLE IF NOT EXISTS `payment_receipts` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `student_id` VARCHAR(100),
  `student_name` VARCHAR(255),
  `payment_date` DATE,
  `payment_amount` DECIMAL(10,2),
  `file_path` VARCHAR(255),
  `remark` TEXT,
  `status` VARCHAR(50) DEFAULT 'pending'
);

CREATE TABLE IF NOT EXISTS `library_mem` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `username` VARCHAR(100),
  `student_name` VARCHAR(255),
  `email` VARCHAR(100),
  `date` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  `status` VARCHAR(50) DEFAULT 'active'
);

CREATE TABLE IF NOT EXISTS `recreation_mem` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `username` VARCHAR(100),
  `student_name` VARCHAR(255),
  `email` VARCHAR(100),
  `date` TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE IF NOT EXISTS `lecture_evaluation` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `lec_name` VARCHAR(255),
  `username` VARCHAR(100),
  `title` VARCHAR(255),
  `evaluation` TEXT
);

CREATE TABLE IF NOT EXISTS `manuals` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `manual` VARCHAR(255),
  `filename` VARCHAR(255),
  `filedata` LONGBLOB
);

-- 5. Insert Default Data for Testing
INSERT INTO `admin_login_tbl` (`username`, `password`, `name`, `gender`) VALUES
('admin', 'admin123', 'System Admin', 'Male');

INSERT INTO `lecturers` (`username`, `password`, `name`, `department`, `email`, `contact`) VA
('lec01', 'lec123', 'John Doe', 'Computer Science', 'john.doe@example.com', '1234567890');
```

```
INSERT INTO `login_tbl` (`username`, `password`, `student_name`, `course`, `batch_number`, `r
('student01', 'stu123', 'Alice Smith', 'BSc Computer Science', 'Batch-01', 'Student');

INSERT INTO `course_tbl` (`course_name`, `award_uni`) VALUES
('BSc Computer Science', 'University of Tech');

INSERT INTO `modules` (`course`, `module_name`, `module_code`, `date`, `duration`, `num_assig
('BSc Computer Science', 'Web Development', 'CS101', '2023-01-01', '6 Months', 3);

INSERT INTO `notice` (`subject`, `added_date`, `view_link`) VALUES
('Welcome to the New Semester', '2023-09-01', '#');
```


Metadata

Assignees

No one assigned

Labels

No labels

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

No branches or pull requests

Participants