Insecure Direct Object Reference (IDOR) in /viva/update.php allows unauthorized modification of team members' names #236

Open

duckpigdog opened 2 weeks ago

Insecure Direct Object Reference (IDOR) in /viva/update.php allows unauthorized modification of team members' names

Describe the bug
An Insecure Direct Object Reference (IDOR) vulnerability exists in the `/viva/update.php` file.
An attacker can modify the `name` field for a user in the `team_members` table without any authentication, simply by knowing the target username and sending a simple HTTP POST request.

Steps to reproduce:

1. Ensure the `team_members` table exists in the database and contains a record with `username='student01'` where the `name` is initially set to `Alice`.
2. Send the following POST request (using Burp Suite, curl, or Postman):

POST /viva/update.php HTTP/1.1
Host: 127.0.0.1:3000
Content-Type: application/x-www-form-urlencoded

username=student01&name=HACKED

3. Query the database again; the `name` for `student01` will have been changed to `HACKED`.
4. Replace `username` with any other existing username (e.g., `admin`, `lec01`) – the `name` for that user will also be modified.

This vulnerability affects all users of the system. Attackers can maliciously alter the `name` field in `team_members` for other users (including administrators), potentially causing data inconsistency or enabling further attacks.

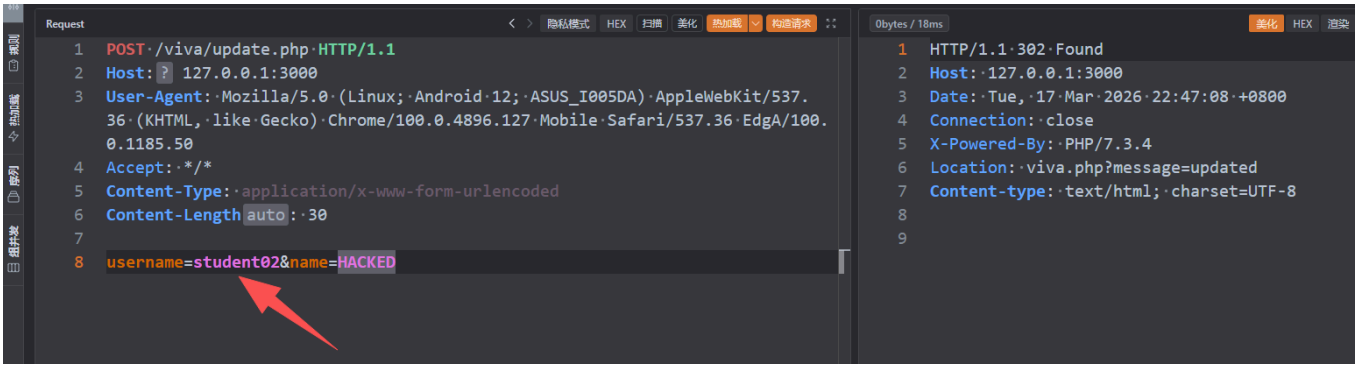
Describe the solution

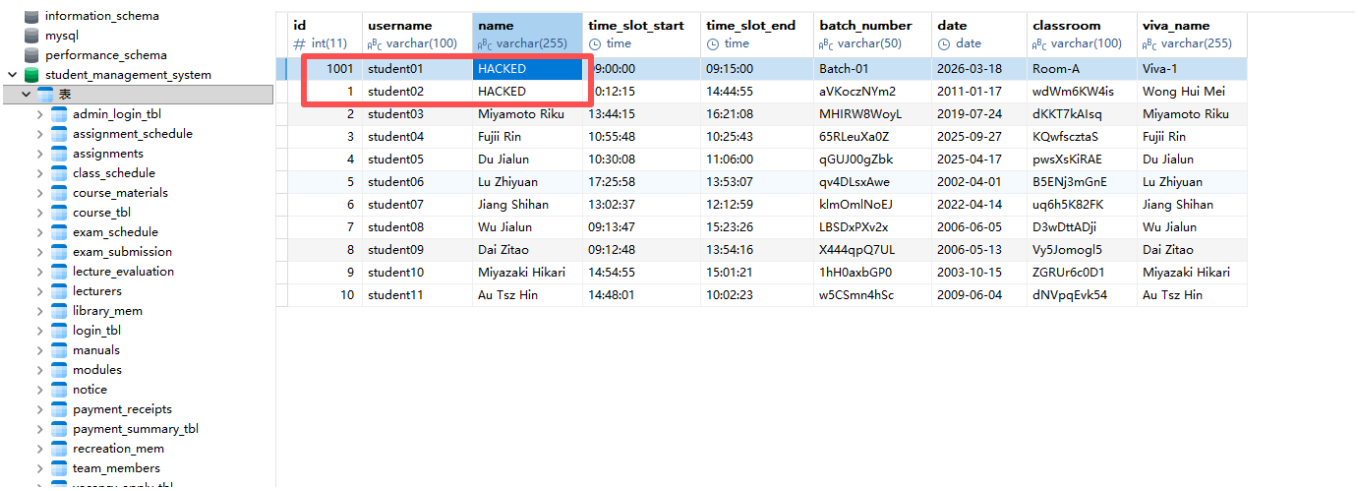
1. In `/viva/update.php`, first verify that the user is logged in (e.g., by checking session or token).
2. Instead of directly using the `username` supplied by the client as the update condition, retrieve the current user's identifier from the session and allow modifications only for their own record.
3. If administrator-level modifications are required, implement role-based access control and ensure that only authorized users can perform such operations.

Suggested fix example:

- Obtain the current username from `$_SESSION['username']` rather than using `$_POST['username']`.
- Alternatively, if cross-user modification must be supported, verify that the current user is an administrator and log the operation.

Screenshots





Are you working on it

No

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', '#');
```

duckpigdog 2 weeks ago Author •••

To trigger this vulnerability, you need to first create the team_members table in the database and insert test data. Execute the following SQL statement:

```
USE student_management_system;

CREATE TABLE IF NOT EXISTS team_members (
  id INT NOT NULL,
  username VARCHAR(100) NOT NULL,
  name VARCHAR(255) NOT NULL,
  time_slot_start TIME NULL,
  time_slot_end TIME NULL,
  batch_number VARCHAR(50) NULL,
  date DATE NULL,
  classroom VARCHAR(100) NULL,
  viva_name VARCHAR(255) NULL,
  PRIMARY KEY (id, username),
```

```
  KEY idx_username (username),
  KEY idx_viva_name (viva_name)
);

INSERT INTO team_members
(id, username, name, time_slot_start, time_slot_end, batch_number, date, classroom, viva_name
VALUES
(1001, 'student01', 'Alice', '09:00:00', '09:15:00', 'Batch-01', '2026-03-18', 'Room-A', 'Viv
ON DUPLICATE KEY UPDATE name = VALUES(name);
```


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