Cross site scripting in CASAP Automated Enrollment System v1.0

Multiple cross-site scripting vulnerabilities are present in CASAP Automated Enrollment System v1.0. The following is detailed information about these vulnerabilities:

Following (table_name -> property) are prone to stored XSS because application code do not sanitise the value before storing into database.

activity_log -> username
activity_log -> action
aprjun -> class
class -> class_name
class -> category
janmar -> class
Julsep -> class
octdec -> class
Students -> Firstname
Students -> Middlename
Students -> lastname
Students -> address
Students -> class
Students -> gfirstname
Students -> gmiddlename
Students -> glastname
Students -> rship
Students -> status
Students -> transport
Students -> route
users -> username
users -> password
users -> firstname
users -> lastname
users -> status

Attacker can insert malicious input to forms and code allows to store input into database, later resulting into XSS while rendering data from database. Following are the few examples, of XSS attacks on above mentioned database properties.

File: save_class.php

Parameters: user_username, category
This file saves into the db using the unsanitized code. The variables are being plugged into a sql query and stores exploitable data. This is also true for other files in application that saves users, students, and classes.

File: student_table.php

Parameters: firstname, class, status
Displays the exploitable data from database without any sanitization. All three variables are being displayed in each entry on the table.

File: add_class1.php

Parameters: category, class_name
Takes the malicious input from the add_class.php file and sends it to be stored in the db in save_class.php. Data is not sanitized while processing the form. Therefore class_name can be easily used to store JS and catagory can be changed to have malicious values via inspect elements.

File: add_student.php

Parameters: fname, mname, lname, address, class, gfname, gmname, glname, rship, status, transport, route
Almost every input in this file is can be used to send malicious code into the database and in future, gets executed when a new student is added. All variables listed can store JS value.

File: save_stud.php

Parameters: fname, mname, lname, address, class, fgname, gmname, glname, rship, status, transport, route
Just like the above examples, the variables are stored into the database without sanitization.

File: add_user.php

Parameters: status, fname, lname
Similar to students and class as above, the users information can be used to inject malicious code in parameters to execute exploitable code when the tables is displays.

File: users.php

Parameters: username, firstname, status
Renders the malicious data from database without sanitization.

File: save_user.php

This vulnerability was detected as part of the DARPA CHESS program[https://www.darpa.mil/program/computers-and-humans-exploring-software-security]