

SECURITY FLAWS AND PROTECTION ON WEB BASED SYSTEM

ASSIGNMENT-02



Submitted to:

Md. Iftekhar Alam Efat

Lecturer

IIT, NSTU

Submitted by:

Saifur Rahman

Roll: ASH1825031M

Batch: BSSE 1st

IIT, NSTU

05-05-2021

Security Flaws and Protection on Web based System

Scenario: There are 3 types of users for this system: Teacher, Director/Chairman, Exam Controller

Teacher will have a UI (dashboard) where the teacher will input marks for a particular course and multiple students OR a particular student's multiple course's marks. S/he can edit these marks many times. There will be a client-side script which will restrict the teacher to input marks more than the boundary (limit). After inserting the marks, s/he will submit it to the Director/Chairman by clicking an appropriate button.

Director/Chairman will have an UI (non-editing) where it will show either there is any inconsistency or boundary value problem with the marks. Also, s/he can give comment/feedback on marks and resend it to the TEACHER as well. However, if there is no problem, Director/Chairman will approve the marksheet, which will then transfer to Exam Controller.

Exam Controller can view and print individual student marksheet from his/her UI (dashboard).

Based on the scenario I design web application and try to prevent cyber security attacks.

1. SQL injection:

At first we can see that in the scenario there are 3 users so I design a log in page and when each user gives correct username or password, then they can enter the system. For example, when teacher gives correct username or password then teacher can enter the system and can submit student's marks on the system. Otherwise he/she cannot submit marks or enter the system.

So, Attacker try to bypass the login page to use sql injection attack on the login page. Attacker inject malicious code to the username and password field.

Sign In

[Forgot password?](#)

1'or'1'='1

••••••••

Faculty Member ▼

☐ Remember me

Login

Don't have an account! [Sign Up Here](#)

Here we can see that attacker try to inject sql query which return always true value (1). So, to prevent SQL injection in the login page

- a) Strong Password: Password must be at least 8 characters where must be at least one upper character, one lower character, one special character, one digit.
- b) MD5 Encryption: Password must be encrypting by md5 encryption mechanism which will prevent web system from common security attack.

```
$password=md5($_POST['password']);
```

user_password

81dc9bdb52d04dc20036dbd8313ed055

81dc9bdb52d04dc20036dbd8313ed055

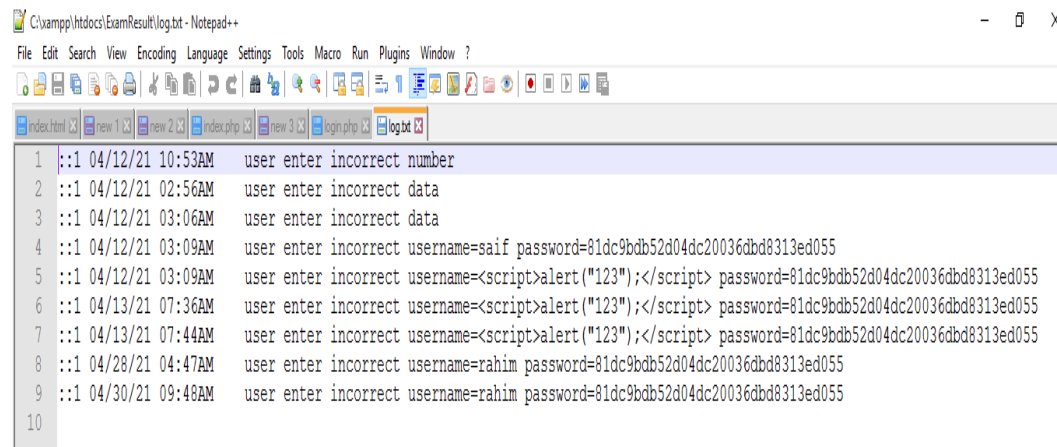
81dc9bdb52d04dc20036dbd8313ed055

81dc9bdb52d04dc20036dbd8313ed055

81dc9bdb52d04dc20036dbd8313ed055

c) Log file:

Log file will save attacker all behavior in a file. It also helps us to know about attacker behavior. Log file will save time, entered malicious code etc.



```

C:\xampp\htdocs\ExamResult\log.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.html new 1 new 2 index.php new 3 login.php log.txt
1 ::1 04/12/21 10:53AM user enter incorrect number
2 ::1 04/12/21 02:56AM user enter incorrect data
3 ::1 04/12/21 03:06AM user enter incorrect data
4 ::1 04/12/21 03:09AM user enter incorrect username=saif password=81dc9bdb52d04dc20036dbd8313ed055
5 ::1 04/12/21 03:09AM user enter incorrect username=<script>alert("123");</script> password=81dc9bdb52d04dc20036dbd8313ed055
6 ::1 04/13/21 07:36AM user enter incorrect username=<script>alert("123");</script> password=81dc9bdb52d04dc20036dbd8313ed055
7 ::1 04/13/21 07:44AM user enter incorrect username=<script>alert("123");</script> password=81dc9bdb52d04dc20036dbd8313ed055
8 ::1 04/28/21 04:47AM user enter incorrect username=rahim password=81dc9bdb52d04dc20036dbd8313ed055
9 ::1 04/30/21 09:48AM user enter incorrect username=rahim password=81dc9bdb52d04dc20036dbd8313ed055
10

```

When three fields are match then the user can enter the system. Three fields are username, password, and user position. If user don't select position, then also user can't enter the system.

2. **Cross Site Scripting:** Cross-site Scripting (XSS) is a client-side code injection attack. The attacker aims to execute malicious scripts in a web browser of the victim by including malicious code in a legitimate web page or web application. The actual attack occurs when the victim visits the web page or web application that executes the malicious code. A web page or web application is vulnerable to XSS if it uses unsanitized user input in the output that it generates. To Prevent XSS I use

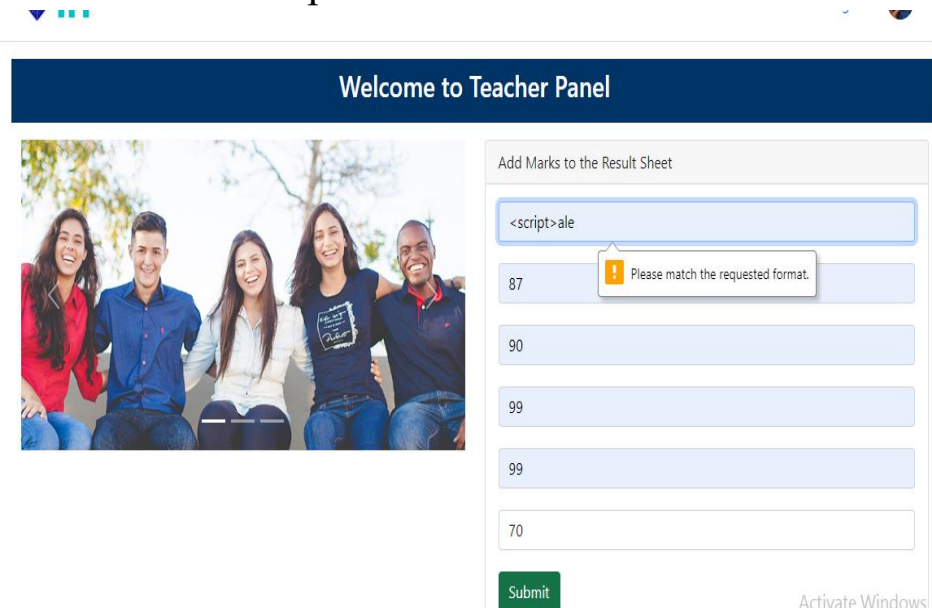
- ❖ Sanitized User input: PHP filters is used to validate and filter data coming from insecure sources, like user input. `FILTER_SANITIZE_SPECIAL_CHARS` Removes special characters from user input. When user give input to the input field then `FILTER_SANITIZE_SPECIAL_CHARS` Removes special characters from user input. This method sanitizing the user input.

```

$bangla= filter_input(INPUT_POST,'bangla',FILTER_SANITIZE_SPECIAL_CHARS);
$english=filter_input(INPUT_POST,'english',FILTER_SANITIZE_SPECIAL_CHARS);
$math=filter_input(INPUT_POST,'math',FILTER_SANITIZE_SPECIAL_CHARS);
$physics=filter_input(INPUT_POST,'physics',FILTER_SANITIZE_SPECIAL_CHARS);
$chemistry=filter_input(INPUT_POST,'chemistry',FILTER_SANITIZE_SPECIAL_CHARS);

```

- ❖ Use Pattern in input: Input text field where users give appropriate value based on this values system also response. Input field is a large area where attacker try to inject XSS payload. So for this reason, I use pattern in the input field which allow user if user input match to the given pattern otherwise it can't accept the value.



Welcome to Teacher Panel

Add Marks to the Result Sheet

<script>ale

87

90

99

99

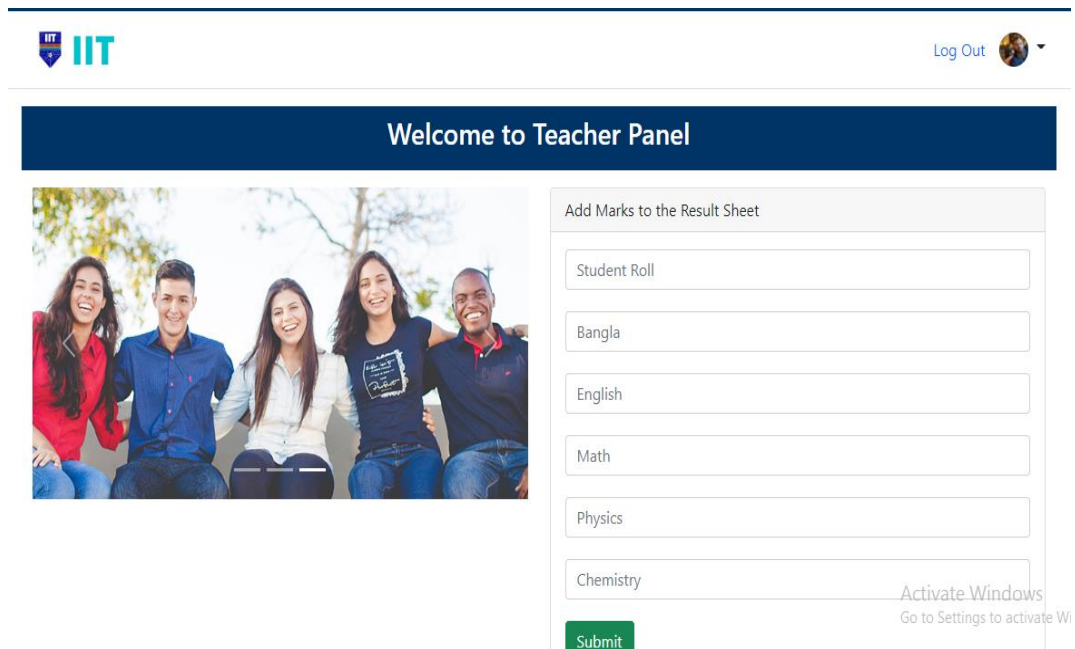
70

Submit

Please match the requested format.

In the picture we can see that attacker try to inject XSS payload but input field do not accept input value and it's also show warning message "Please match the requested format". Pattern is a most useful in this case.

- ❖ Set type of input field: After teacher enter to the system he/she can see the dashboard.



IT IIT

Log Out

Welcome to Teacher Panel

Add Marks to the Result Sheet

Student Roll

Bangla

English

Math

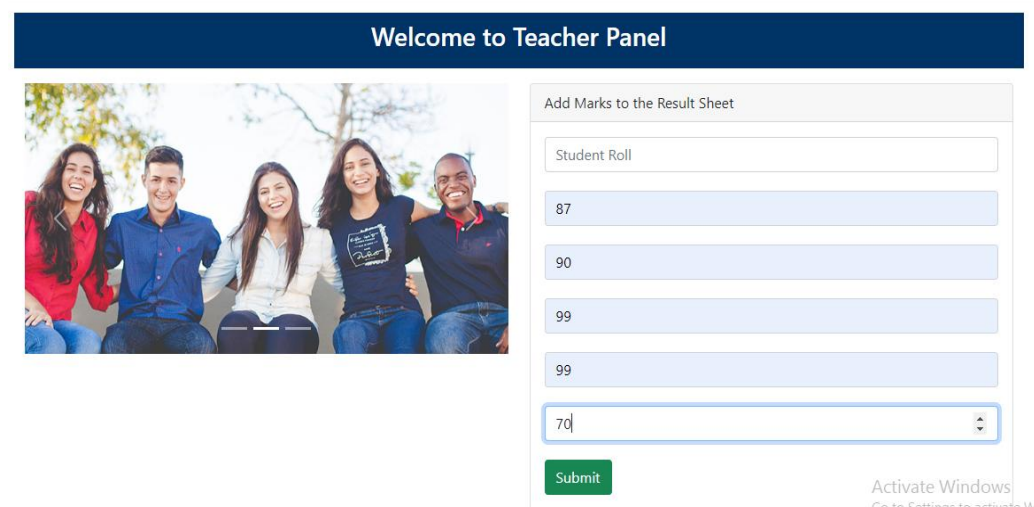
Physics

Chemistry

Submit

Activate Windows
Go to Settings to activate Wi

Here teacher can submit student's marks based on their roll ID. We know marks are number so I set all marks input fields types are number. For this reason, attacker can't submit XSS payload to the marks fields. Marks fields always can accept number values. I also set minimum number value 0 and maximum marks value 100. It will fix negative values effects on the system.



Welcome to Teacher Panel

Add Marks to the Result Sheet

Student Roll

87

90

99

99

70

Submit

Activate Windows
Go to Settings to activate Wi

- ❖ `mysql_real_escape_string()` function: The `real_escape_string()` / `mysql_real_escape_string()` function escapes special characters in a string for use in an SQL query,

taking into account the current character set of the connection. This function is used to create a legal SQL string that can be used in an SQL statement. If any user uses special characters in a string, then this function helps to escape special characters in user input.

```
$roll=$_POST['roll'];
$rol= mysqli_real_escape_string($con, $roll);
//Example: $roll=1; $roll=1;
```

3. Invalidated Redirects

The web application uses few methods to redirect users to other pages for an intended purpose.

If there is no proper validation while redirecting to other pages, attackers can make use of this and can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

<http://localhost/ExamResult/office.php>

but here attack modified url is

<http://localhost/ExamResult/director.php>

Surprisingly, attacker also on director panel. But he/she also log in as an exam-controller.

Prevention: Restrict user access

Now attacker can't redirector one page to another page.


4. Web Parameter Tampering attack:

The Web Parameter Tampering attack is based on the manipulation of parameters exchanged between client and server in order to modify application data, such as user credentials and permissions.

Parameter tampering can often be done with:

- Cookies form fields
- HTTP headers
- URL Manipulation
- URL Manipulation:

When the URL passes sensitive values through parameters, the attacker can tamper this query string and perform malicious actions.



localhost/ExamResult/resultprint.php?r=27

IIT Log Out

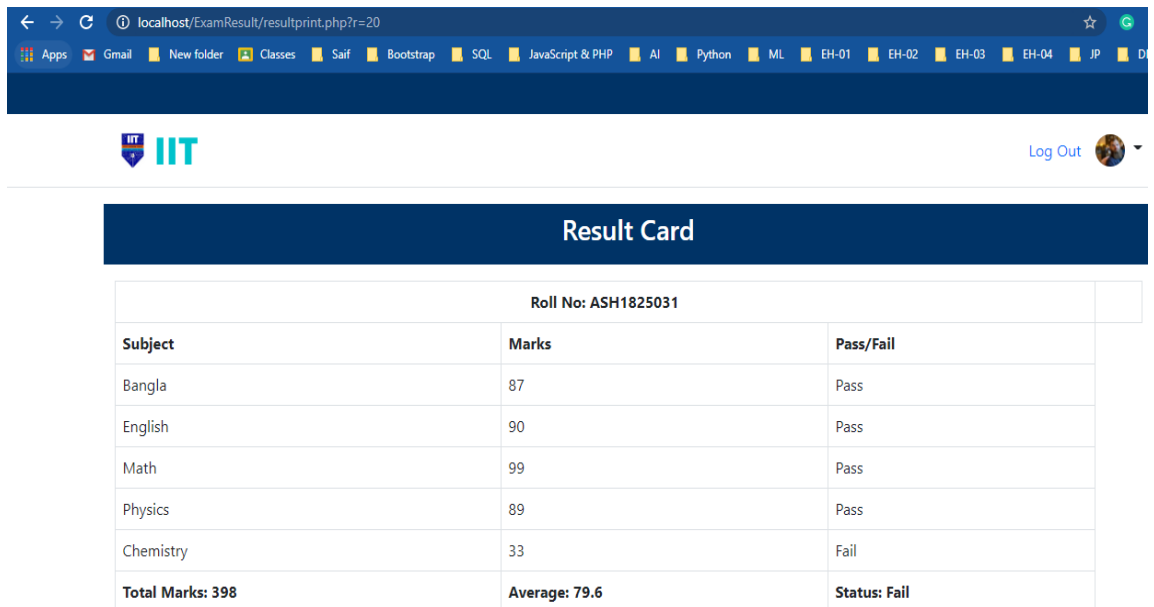
Result Card

Roll No: ASH1825001

| Subject | Marks | Pass/Fail |
|-------------------------|--------------------|---------------------|
| Bangla | 77 | Pass |
| English | 87 | Pass |
| Math | 90 | Pass |
| Physics | 78 | Pass |
| Chemistry | 98 | Pass |
| Total Marks: 430 | Average: 86 | Status: Pass |

Back PRINT Activate Windows

After tempering parameter...



localhost/ExamResult/resultprint.php?r=20

IIT Log Out

Result Card

Roll No: ASH1825031

| Subject | Marks | Pass/Fail |
|-------------------------|----------------------|---------------------|
| Bangla | 87 | Pass |
| English | 90 | Pass |
| Math | 99 | Pass |
| Physics | 89 | Pass |
| Chemistry | 33 | Fail |
| Total Marks: 398 | Average: 79.6 | Status: Fail |

Attacker modified parameter and result card also response with parameter means result card also change. One can print another result card.

Prevention: Parameter encryption

After using this, attacker doesn't find real parameter. Real parameter also hide by some extra string. So Attacker can't do Web parameter temper attack.

END