1) XSS - тип атаки на веб-системы, который состоит в том, что в выдаваемую страницу внедряется вредоносный JS-код.

Для ее предотвращения надо обезопасить страницу, отфильтровав информацию из формы. Функция htmlspecialchars преобразует специальные символы в HTML-сущности.

Применим данную функцию к данным, вводимым пользователем, перед тем, как вносить их в базу данных:

```
$name = htmlspecialchars($_POST['name']);
$email = htmlspecialchars($_POST['email']);
$birthday = htmlspecialchars($_POST['birthday']);
$gender = htmlspecialchars($_POST['gender']);
$limbs = htmlspecialchars($_POST['limbs']);
$biography = htmlspecialchars($_POST['biography']);

$stmt = $db->prepare("UPDATE users6 SET name = ?, email = ?, birthday = ?, gender = ?, limbs = ?, biography = ? WHERE usr_id = ?");
$stmt -> execute(array($name, $email, $birthday, $gender, $limbs, $biography, $_SESSION['uid']));
$stmt = $db->prepare('DELETE FROM powers6 WHERE usr_id = ?');
$stmt->execute([$_SESSION['uid']]);
$stmt = $db->prepare("INSERT INTO powers6 SET usr_id = ?, superpower = ?");
foreach ($_POST['superpowers'] as $pw) {
    $stmt -> execute(array($_SESSION['uid'], htmlspecialchars($pw)));
}
```

2) SQL Injection - способ взлома, основанный на внедрении в запрос к базам данных нежелательного SQL-кода. Внедрение SQL позволяет хакеру выполнить произвольный запрос к базе данных (прочитать содержимое любых таблиц, удалить, изменить или добавить данные).

Для защиты используем подготовленные запросы (видно на изображении выше), а также проверяем входные данные на соответствие формату и используем фильтрацию с помощью функции в п.1:

```
else if (!preg_match("/^[A-Z][a-z]+$/", $_POST['name'])){
    setcookie('name_error', 'incorrect', time() + 24 * 60 *60);
    $errors = TRUE;
}
else {
    setcookie('name_value', $_POST['name'], time() + 365 * 24 * 60 * 60);
}

if (empty($_POST['email'])) {
    setcookie('email_error', 'empty', time() + 24 * 60 *60);
    $errors = TRUE;
  }
else if (!preg_match("/^[A-Za-z0-9][A-Za-z0-9_\-.]+@[A-Za-z0-9][A-Za-z0-9_\-.]+\.[A-Za-z]+$/",$_POST['email'])){
    setcookie('email_error', 'incorrect', time() + 24 * 60 *60);
    $errors = TRUE;
}
```

3) CSRF - вид атак на посетителей веб-сайтов, использующий недостатки протокола HTTP. Если жертва заходит на сайт, созданный злоумышленником, от её лица тайно отправляется запрос на другой сервер, осуществляющий некую вредоносную операцию.

Способ защиты – специальное значение (токен), которое генерируется случайным образом при авторизации и сохраняется в сессии посетителя.

Создаем токен при входе в сессию:

```
session_start();
if (empty($_SESSION['token']))
    $_SESSION['token'] = bin2hex(random_bytes(15));
```

И затем используем его в скрытой строке полей с методом POST:

```
<form action="" method="POST">

    <input type="hidden" name="token" value=$_SESSION['token'] />

    <label>
```

4) Include, Upload уязвимости могут возникнуть при загрузке и скачивании файлов, соответственно. Так как в нашей работе такие возможности отсутствуют, менять в коде нечего.