Отчёт: аудит безопасности на примере формы из 4 задания.

1. XSS уязвимость:

```php
if (empty($_POST['fio'])) {
    setcookie('err_fio','1',time()+3600*24*365);
    $errors = TRUE;
}
else{
    setcookie('err_fio','');
}
if (empty($_POST['year'])) {
    setcookie('err_date','1',time()+3600*24*365);
    $errors = TRUE;
}
else{

    setcookie('err_date','');
}
if (empty($_POST['email'])) {
    setcookie('err_email','1',time()+3600*24*365);
    $errors = TRUE;
}
else{
    setcookie('err_email','');
}
if (empty($_POST['biography'])) {
    setcookie('err_biography','1',time()+3600*24*365);
    $errors = TRUE;
}
else{
    setcookie('err_biography','');
}
if (!preg_match('/^[a-zA-Za-яA-я\s]+$/u', $_POST['fio'])) {
    $errors = TRUE;
    setcookie('err_fio2','1',time()+3600*24*365);
```

```php
$name = $_POST["fio"];
$ye = $_POST["year"];
$em = $_POST["email"];
$bio = $_POST["biography"];
$limbs = $_POST["limbs"];
$gen=$_POST["gender"];
```

Устранение:

```php
$name = htmlspecialchars($_POST["fio"], ENT_QUOTES, 'UTF-8');
$ye = htmlspecialchars($_POST["year"], ENT_QUOTES, 'UTF-8');
$em = htmlspecialchars($_POST["email"], ENT_QUOTES, 'UTF-8');
$bio = htmlspecialchars($_POST["biography"], ENT_QUOTES, 'UTF-8');
$limbs = htmlspecialchars($_POST["limbs"], ENT_QUOTES, 'UTF-8');
$gen = htmlspecialchars($_POST["gender"], ENT_QUOTES, 'UTF-8');
$abil = array_map('intval', $_POST["Abilities"]);
```

2. Использование подготовительных запросов для защиты от SQL injection:

```php
$user = 'u53851';
$pass = '6440273';
$db = new PDO('mysql:host=localhost;dbname=u53851', $user, $pass, [PDO::ATTR_PERSISTENT => true]);
try {
  $stmt = $db->prepare("INSERT INTO form2(fio, email, biography, limbs, year, gender) VALUES (:fio,:email,:biography,:limbs,:year,:gender)");
```

```php
  $stmt -> execute(['fio'=>"$name", 'email'=>"$em", 'biography'=>"$bio",'limbs'=>"$limbs", 'year'=>"$ye","gender"=>"$gen"]);
 $id_connection=$db->lastInsertId();
  $stmt = $db->prepare("INSERT INTO connection2(id_ability, id_person) VALUES (:id_ability,:id_person)");
foreach ($abil as $abilities){
    if ($abilities!=false){
    $stmt-> execute(['id_ability'=> $abilities, 'id_person'=>$id_connection]);
    }
  }
```

3. CSRF

```php
5   // Генерируем токен CSRF и сохраняем его в сессию
6   if (!isset($_SESSION['csrf_token'])) {
7       $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
8   }
```

```php
88  // Проверяем токен CSRF
89  if ($_POST && $_POST['csrf_token'] !== $_SESSION['csrf_token']) {
90      die('Invalid token');
91  }
```

```php
102 try {
103     // Генерируем новый токен CSRF и сохраняем его в сессию
104     $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
105
```

```php
12  <body>
13  <div class="con">
14  <form action="index.php"
15          method ="POST"><div>
16          <input type="hidden" name="csrf_token" value="<?php echo $_SESSION['csrf_token']; ?>">
```

4. include и upload уязвимости: