1. Для защиты от вставки вредоносного кода в веб-страницы и кражи данных пользователя, необходимо использовать фильтрацию входных данных и экранирование вывода, чтобы предотвратить межсайтовый скриптинг (XSS).
2. Для защиты от вставки вредоносного SQL-кода в запросы к базе данных и кражи данных пользователя, необходимо использовать подготовленные запросы и фильтрацию входных данных, чтобы предотвратить SQL Injection.
Данные методы помогут в защите от этих уязвимостей

```php
function filter_input_data($input){
    return htmlspecialchars(trim($input),ENT_QOUTES,'UTF-8');
}
function filter_output_data($output){
    return htmlspecialchars($output,ENT_QOUTES,'UTF-8');
}
```

3. Для защиты от подделки запросов, которые отправляются с других сайтов и могут привести к изменению данных пользователя, необходимо использовать токены CSRF и проверку referer, чтобы предотвратить межсайтовую подделку запроса (CSRF).

```php
    if($_SESSION['csrf_token']!==$_POST['token']){
        die('Invalid CSRF token');
    }
if(parse_url($_SERVER['HTTP_REFERER'],PHP_URL_HOST)!=='u54446.kubsu-dev.ru'){
    die('Invalid refrer');
}
```

```php
if(!isset($_SESSION['csrf_token'])){
    $_SESSION['csrf_token']=bin2hex(random_bytes(32));
}
$token=$_SESSION['csrf_token'];
```

```php
<form action="" method="POST">
<input type="hidden" name="token" value="<?= $token;?>">
```

4. Для защиты от включения вредоносного кода из внешних файлов и кражи данных пользователя, необходимо использовать только относительные пути и проверку наличия файла, чтобы предотвратить Include.

```php
    if(file_exists('form.php')){
    include('form.php');
    }
```

5. Для защиты от загрузки вредоносных файлов на сервер и кражи данных пользователя, необходимо проверять тип и размер загружаемого файла, а также использовать уникальные имена файлов при загрузке. В данном случае не требуется загрузка файлов на сервер, но в целом рекомендуется учитывать эти меры безопасности при работе с загрузкой файлов. Пример:

```
// Проверка типа и размера файла
if ($_FILES['file']['type'] !== 'image/jpeg' || $_FILES['file']['size'] > 1000000)
{ die('Invalid file type or size'); }
// Генерация уникального имени файла
```

```
$filename = uniqid() . '.jpg';
move_uploaded_file($_FILES['file']['tmp_name'], 'uploads/' . $filename);
```