Отчёт.

1. **CSRF (Межсайтовая подделка запроса)** – это вид атаки, при которой вражеский сайт выдаёт себя за доверенного пользователя и отсылает на сайт нежелательные команды. Для защиты от CSRF атак используют CSRF токены и проверку refer. Ниже приведён пример кода:

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

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

```php
if (parse_url($_SERVER['HTTP_REFERER'], PHP_URL_HOST) !== 'u54409.kubsu-dev.ru')
{
  die('Invalid referer');
}
```

```php
    if ($_SESSION['csrf_token'] !== $_POST['token']) {
  die('Invalid CSRF token');
}
```

2. **XSS (Межсайтовый скриптинг) –** это вид атаки, при которой злоумышленники могут внедрять вредоносный код через веб-сайт в браузеры других пользователей. Для защиты от XSS атак программисты фильтруют входные данные и экранируют вывод. Ниже приведён пример кода:

2.1. Фильтрация входных данных:

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

2.2. Экранирование вывода:

```php
function filter_output_data($output){
    return htmlentities($output, ENT_QUOTES, 'UTF-8');
}
```

3. **SQL injection –** это вид атаки, во время которой вредоносный код вставляется в строки, которые позже будут переданы на экземпляр SQL Server для анализа и выполнения. Для защиты от SQL injection используют заготовленные запросы и фильтрацию входных данных. Ниже приведён пример кода:

```
$stmt = $db->prepare("INSERT INTO forma SET
name=?,email=?,birthday=?,sex=?,limbs=?,biographiya=?");
$stmt ->
execute(array(filter_input_data($_POST['name']),filter_input_data($_POST
['email']),filter_input_data($_POST['birthdayear']),filter_input_data($_
POST['gen']),filter_input_data($_POST['body']),filter_input_data($_POST[
'biographiya'])));
```

4. **Upload –** это вид атаки, во время которой на сервер загружаются вредоносные файлы, которые в будущем могут красть данные пользователя. Для защиты от Upload используют проверку типа, и размера загружаемого файла, и уникальные имена файлов. Ниже приведён пример кода:

```
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);
```

5. **Include** – это вид атаки, во время которой на сервер подключается вредоносный код из других файлов, которые в будущем могут красть данные пользователя. Для защиты от Include используют относительные пути и проверку наличия файлов. Ниже приведён пример кода:

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