Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра «Информационная безопасность»

Направление подготовки/ специальность: 10.03.01 Информационная безопасность

ОТЧЕТ

по проектной практике

Студент: Ильин Кирилл Александрович Группа: 241-353

Место прохождения практики: Московский Политех, кафедра «Информационная безопасность»

Отчет принят с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики: Гневшев Александр Юрьевич

Москва 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ…………………………………………………………………………………………………………………………………...2

Общая информация о проекте3

Название проекта3

Цели и задачи проекта3

Описание задания по проектной практике4

Описание достигнутых результатов по проектной практике5

ЗАКЛЮЧЕНИЕ 17

СПИСОК ЛИТЕРАТУРЫ…………………………………………………………………………………………………………17

ПРИЛОЖЕНИЯ 17

**ЗАДАНИЕ ПО ПРОЕКТНОЙ ПРАКТИКЕ**

Проектная практика включала базовую и вариативную части, общей трудоёмкостью 72 академических часа. Работа выполнялась в составе команды из двух человек: Ильин К.А. (241-353) и Кондраков В.И. (241-371).

Для контроля версий использовалась система Git, документация оформлялась с применением языка разметки Markdown, а для создания статического веб-сайта применялись HTML и CSS. Репозитории размещались на платформе [GitHub](https://github.com).

Проект состоял из двух частей: обязательной (базовой) для всех студентов и вариативной, задание на которую было выдано ответственным за практику представителем выпускающей кафедры.

1. Базовая часть задания

1. Настройка Git и репозитория:

Создать групповой репозиторий на [GitHub](https://github.com/)  на основе предоставленного [шаблона](https://github.com/mospol/practice-2025-1).

Освоить базовые команды Git: клонирование, коммит, пуш и создание веток.

Регулярно фиксировать изменения с осмысленными сообщениями к коммитам.

Примерное время: 5 часов.

1. Написание документов в Markdown:

Все материалы проекта (описание, журнал прогресса и др.) оформить в формате Markdown.

Изучить синтаксис Markdown и подготовить необходимые документы.

Примерное время: 5 часов.

1. Создание статического веб-сайта:

Для создания сайта необходимо использовать только HTML и CSS.

Создать новый сайт об основном проекте по дисциплине «Проектная деятельность» (INVADE). Оформление и наполнение сайта должны быть уникальны.

Сайт должен включать:

Домашнюю страницу с аннотацией проекта.

Страницу «О проекте» с описанием проекта.

Страницу «Участники» с описанием личного вклада каждого участника группы в проект по «Проектной деятельности».

Страницу «Журнал» с минимум тремя постами (новостями, блоками) о прогрессе работы.

Страницу «Ресурсы» со ссылками на полезные материалы.

Оформить страницы сайта графическими материалами (фотографиями, схемами, диаграммами, иллюстрациями)

Примерное время: изучение и настройка — 14 часов, дизайн и наполнение — 8 часов.

2. Вариативная часть задания:

В качестве вариативной части нашей группе было дано следующее задание:

**Тема задания:**

"Настроить систему логирования и базового анализа событий безопасности веб-сервера."

**Задачи задания:**

Установить и настроить веб-сервер (Apache/Nginx).

Включить и настроить ведение логов доступа и ошибок.

Обработать логи: фильтрация атак, brute-force, SQL-инъекции, DoS-атаки (и другие на усмотрение студентов).

Оформить шаблон отчета ИБ-инцидента (карточки инцидента) на основе логов.

*Примечание: рекомендуется проводить все манипуляции в изолированном сегменте*.

**Примерное время:** 32-40 часов

**ДОСТИГНУТЫЕ РЕЗУЛЬТАТЫ ПО ПРОЕКТНОЙ ПРАКТИКЕ**

**Изучен язык разметки HTML** для формирования базовой структуры веб-сайта (Затраченное время: 22 часа)

**Настроен групповой репозиторий** на платформе GitHub, выполнено добавление и отслеживание изменений проекта с использованием системы контроля версий Git;( Затраченное время: 5 часов)

**Установлен и сконфигурирован веб-сервер Apache в изолированной среде** (локальный сегмент), обеспечена доступность по адресу 127.0.0.1; (Затраченное время: 8 часов)

**Активированы и проверены механизмы логирования** событий на веб-сервере, включая ведение файлов access.log и error.log; (Затраченное время: 3 часа)

**Проведены и зафиксированы три практических теста по моделированию атак:**

**Brute-force атака** на форму входа с использованием инструмента Hydra, приведшая к успешному подбору пароля;

**SQL-инъекция** с применением утилиты sqlmap, позволившая извлечь данные из уязвимой базы SQLite;

**DoS-атака типа Slowloris**, направленная на исчерпание доступных соединений веб-сервера, результатом которой стало достижение лимита MaxRequestWorkers и нарушение доступности;

(Затраченное время: 14 часов)

В результате работы получен практический опыт в **использовании HTML, настройки репозитория, по развёртыванию веб-сервера, организации логирования, реализации тестовых атак.** Все действия проводились в рамках изолированной среды и сопровождались регистрацией действий для последующего документирования.

Также я исполнял обязанности тимлида нашей команды и осуществлял проверку всех файлов, подготовленных сокомандником.  
(Анализ требований к работе занял 3 часа, редактирование текста на сайте — 4 часа, создание и наполнение GitHub-репозитория, а также его проверка — 9 часов, участие во всех организационных онлайн-собраниях — 8 часов).

Вариативная часть:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 1. Установка Apache*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 2. Проверка работоспособности*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 3. Проверяем локальную доступность*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 4. Настройка логирования, запрос страницы*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 5. Анализ логов*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 6. Установка инструментов для тестирования*

Brute-force атака

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 7. Создаём простую форму входа на сайтие*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 8. Создаём простую форму входа на сайтие*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 9. Запуск атаки*

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 10. Как выглядит работа*

SQL-инъекция

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 11. Создаём уязвимый скрипт*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 12. Пишем уязвимый скрипт*

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 13. Пример работы уязвимого скрипта*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 14. Эмулируем атаку*

*--batch — не задавать вопросы*

*--level=3 --risk=3 — расширенный анализ*

*--random-agent — имитирует разные браузеры*

*SQLmap начнёт проверку уязвимостей. Даже если реальной БД нет — он всё равно будет писать в лог* ***попытки SQL-инъекции*** *(через URL-параметры)*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 15. Вид как работает атака*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 16. Конечный результат*

Изображение выглядит как снимок экрана, текст

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 17. Вид Логов*

DoS-Атака

Изображение выглядит как текст, снимок экрана, Графическое программное обеспечение, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 18. Запускаем атаку*

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 19. Страница бесконечно пытается обновиться, значит атака работает*

Изображение выглядит как текст, снимок экрана, Шрифт, информация

Контент, сгенерированный ИИ, может содержать ошибки.

*Рисунок 20. Логи*

**Заключение**

В ходе проектной практики были достигнуты следующие результаты:

Настройка **GitHub репозитория**

**Изучение HTML**

**Установлен и настроен веб-сервер Apache** в изолированной среде (локальный сегмент 127.0.0.1);

Включено ведение логов доступа и ошибок (access.log и error.log);

**Проведены три успешные имитации атак:**

**Brute-force** с использованием утилиты hydra;

**SQL-инъекция** через sqlmap с реальной базой данных SQLite;

**DoS-атака (Slowloris)** с перегрузкой Apache;

Таким образом, поставленная цель — была успешно достигнута, задачи выполнены в полном объёме, результаты задокументированы и подтверждены логами и практическими тестами и коммитами на GitHub. Общее затраченное время: 76 часов

Список литературы:  
1. [HTML основы](https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content)

2. [Настройка GitHub репозитория](https://github.com/mospol/practice-2025-1/blob/master/task/git_structure.md)

3. [Настройка веб-сервера](https://itsecforu.ru/2022/01/14/%F0%9F%90%89-%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0-http-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%B0-kali-linux/)  
Приложения:  
[GitHub команды](https://github.com/swelock/PRACTIC_1_KURS/tree/main)