Лабораторная работа – уязвимости JWT

В данной лабораторной работе вам предлагается попробовать свои силы в решении задач, схожих с реальными случаями проведения тестирования на проникновение.

Ваша задача – обнаружение и эксплуатация уязвимостей при использовании JWT токенов.

Вам необходимо сделать **любое количество** из предложенных заданий для получения балла.

ВНИМАНИЕ: максимальный балл за эту работу — 10 (если вы выполните заданий больше, чем требуется, максимальная оценка все равно будет 10). Вы можете выбрать любые из предложенных заданий для получения максимальной оценки. Максимальный балл за каждое задание приведен рядом с заданием.

Часть 1 – Необходимые инструменты

Обязательные требования:

- Для выполнения **всех** заданий вам потребуется инструмент Burp Suite Community Edition — это интегрированная платформа для тестирования безопасности веб-приложений. Скачать Burp Suite Community Edition можно по ссылке:
  https://portswigger.net/burp/communitydownload
  Для загрузки потребуется ввести вашу почту.
- Для выполнения Заданий вам потребуется зарегистрироваться на сайте PortSwigger.
- Перед выполнением задания мы рекомендуем вам ознакомится, как работать с JWT токенами в Burp Suite [https://portswigger.net/burp/documentation/desktop/testing-workflow/session-management/jwts].

Часть 2 - Требования к сдаче заданий и к оформлению отчета

Обязательные требования:

- В решении требуется описать способ атаки или обхода защитных механизмов и приложить доказательства выполнения (скриншоты/логи).
- Все шаги выполнения атаки должны быть подтверждены соответствующими скриншотами и текстовыми комментариями к ним.
- На **каждом** скриншоте должна быть информация о выполнившем работу студенте в формате **Фамилия_Имя_Группа** (Пример: Иванов_Иван_БИБ221).
- На **каждом** скриншоте должна быть видна панель задач **с датой и временем** выполнения данного скриншота.
- Из отчета должна быть понятна последовательность ваших рассуждений при проведении атак.
- Используйте шаблон отчета, который прикреплен к заданию.

Часть 3 – Ссылки на задания

Внимание. Все упражнения в лабораторной работе должны выполняться только на специально предоставленных лабораториях (PortSwigger Labs) или других выделенных тестовых средах. Любые попытки применять описанные техники против сторонних ресурсов, не принадлежащих вам или вашей организации, являются незаконными и неприемлемыми. Выполнение заданий вне изолированной лаборатории недопустимо.

- **Задание 1 – PortSwigger — Задание 1 – JWT: Обход аутентификации через не проверяемую подпись (уровень – Apprentice) (2,5 балла)**

  Лабораторная работа демонстрирует критическую уязвимость: сервер принимает JWT-токены *без проверки подписи*, полагаясь только на содержимое payload. Это позволяет злоумышленнику подделать токен и повышать свои привилегии.
  Постановка задачи: Авторизуйтесь в своём аккаунте (`wiener:peter`), извлеките JWT-токен из сессионной cookie и измените поле `sub` (subject) с `wiener` на `administrator`. Используйте поддельный токен для доступа к админ-панели `/admin` и удалите пользователя `carlos`.

  https://portswigger.net/web-security/jwt/lab-jwt-authentication-bypass-via-unverified-signature

- **Задание 2 – JWT: Обход аутентификации через некорректную проверку подписи (`alg=none`) (уровень – Apprentice) (2,5 балла)**

  Лабораторная работа иллюстрирует классическую уязвимость конфигурации JWT: сервер принимает токены с алгоритмом `none`, интерпретируя их как *подписанные*, хотя на самом деле подпись отсутствует. Это позволяет подделать токен, удалив подпись и указав `alg: none`.

  Постановка задачи: Авторизуйтесь в аккаунте `wiener:peter`, извлеките JWT из сессионной cookie, измените значение поля `sub` на `administrator`, установите алгоритм `none` в заголовке и *удалите подпись* (оставив точку после payload). Используйте модифицированный токен для доступа к `/admin` и удаления пользователя `carlos`.

  https://portswigger.net/web-security/jwt/lab-jwt-authentication-bypass-via-flawed-signature-verification

- **Задание 3 – JWT: Обход аутентификации через слабый ключ подписи (уровень – Practitioner, средняя сложность) (3,5 балла)**

  Лабораторная работа моделирует ситуацию, при которой JWT подписывается с использованием *слабого секретного ключа* (например, из списка распространённых паролей). Такой ключ поддаётся брутфорсу, после чего злоумышленник может подписать произвольный токен и выполнить атаку повышения привилегий.

  Постановка задачи: Авторизуйтесь в аккаунте `wiener:peter`, извлеките JWT-токен и с помощью брутфорса (например, через `hashcat` и словарь распространённых секретов) определите ключ, использованный для подписи. Затем сформируйте и подпишите новый токен с `sub: administrator`, отправьте запрос к `/admin`, получите доступ к админ-панели и удалите пользователя `carlos`.

  Совет: Перед выполнением задания рекомендуется ознакомиться с работой JWT в Burp Suite. Для подбора ключа рекомендуется использовать hashcat с режимом 16500 (JWT HMAC) (см. дополнительно https://portswigger.net/web-security/jwt#brute-forcing-secret-keys-using-hashcat).

  https://portswigger.net/web-security/jwt/lab-jwt-authentication-bypass-via-weak-signing-key

- **Задание 4 – JWT: Обход аутентификации через внедрение ключа в заголовок (`jwk`) (уровень – Practitioner, средняя сложность) (3,5 балла)**

  Лабораторная работа демонстрирует уязвимость, возникающую при неконтролируемом использовании параметра `jwk` в заголовке JWT: сервер *доверяет* ключу, встроенному непосредственно в токен, и проверяет подпись с его помощью — без проверки источника или доверия ключа. Это позволяет злоумышленнику сгенерировать собственную пару ключей, подписать поддельный токен и обойти аутентификацию.

  Постановка задачи: Авторизуйтесь в аккаунте `wiener:peter`, извлеките JWT-токен и создайте новую RSA-пару. Измените поле `sub` на `administrator`, добавьте в заголовок токена параметр `jwk` со своим *публичным* ключом и подпишите токен *приватным* ключом. Отправьте модифицированный токен к `/admin`, получите доступ к админ-панели и удалите пользователя `carlos`.

  https://portswigger.net/web-security/jwt/lab-jwt-authentication-bypass-via-jwk-header-injection

- **Задание 5 – JWT: Обход аутентификации через path traversal в заголовке `kid` (уровень – Practitioner, средняя сложность) (3,5 балла)**

Лабораторная работа демонстрирует уязвимость, связанную с некорректной обработкой параметра `kid` (Key ID) в заголовке JWT: сервер использует его *непосредственно* как путь к файлу с ключом на файловой системе, не проводя валидацию. Это открывает возможность атаки path traversal и подмены ключа (например, через `/dev/null`), что позволяет подписать токен «нулевым» или предсказуемым ключом.

Постановка задачи: Авторизуйтесь в аккаунте `wiener:peter`, проанализируйте JWT-токен и идентифицируйте уязвимость в обработке `kid`. С помощью path traversal укажите некорректный путь к ключу, сформируйте и подпишите поддельный токен с `sub: administrator`, получите доступ к `/admin` и удалите пользователя `carlos`.

https://portswigger.net/web-security/jwt/lab-jwt-authentication-bypass-via-kid-header-path-traversal

- **Задание 6 – JWT: Обход аутентификации через Algorithm Confusion (уровень – Expert) (5,0 балла)**

  Лабораторная работа иллюстрирует одну из самых коварных уязвимостей в реализациях JWT: атаку *algorithm confusion* (RS256 → HS256). Сервер ожидает JWT, подписанные асимметричным алгоритмом (RSA), но некорректно проверяет `alg` в заголовке и может интерпретировать публичный ключ как *симметричный секрет*. Если публичный ключ доступен (например, через `/jwks.json`), злоумышленник может подписать поддельный токен **публичным ключом**, и сервер ошибочно примет его как валидный.

  Постановка задачи: Авторизуйтесь в аккаунте `wiener:peter`, найдите и извлеките публичный ключ сервера (например, из `/jwks.json`). Создайте симметричный ключ на его основе, измените заголовок JWT на `alg: HS256`, установите `sub: administrator`, подпишите токен *публичным ключом как секретом* и получите доступ к `/admin`. Удалите пользователя `carlos`.

  https://portswigger.net/web-security/jwt/algorithm-confusion/lab-jwt-authentication-bypass-via-algorithm-confusion

Часть 4 – Чеклист для самопроверки содержания отчета:

Обязательно для всех заданий:

- Для оформления отчета, просьба использовать шаблон оформления, прикрепленный к заданию

- Наличие скриншотов, соответствующих шагам выполнения задания, с информацией о студенте и временными отметками.
  - Допускается добавление подписи вида `Фамилия_Имя_Группа` в угол изображения.
  - Рекомендуется использовать встроенные средства Burp Suite: например, сохранение запросов/ответов из Repeater с временной меткой в HTTP history (Burp автоматически фиксирует время запроса в колонке `Time`), либо скриншот вкладки `Proxy > HTTP history` с выделенным запросом и временной меткой.

Техническое содержание:

- Указаны исходные данные: адрес лаборатории, учётные данные (если даны)
- Если используются сторонние утилиты (hashcat, jwt-cli, openssl), укажите их название и версию — если они критичны для воспроизведения атаки
- Подробное воспроизводимое пошаговое описание эксплуатации (шаги должны быть достаточно конкретны, чтобы третье лицо могло воспроизвести результат):
  - Исходный и модифицированный JWT (в raw-формате и/или декодированный JSON), с выделением изменённых полей;
  - Как и где был изменён заголовок/полезная нагрузка/подпись (вручную, через Burp Repeater, через JWT Editor);
  - Какой алгоритм указан в `alg`, и как это соотносится с ожидаемым сервером;
  - Если использовался `jwk` или `kid` — привести их значение и пояснить, почему сервер его принял;
  - Если применялся брутфорс/генерация ключа — привести команду и результат (например, найденный секрет `secret1`).»

Доказательства и воспроизводимость

- Снимки экрана/логи для ключевых шагов:
  - исходная страница входа / форма;
  - примеры запросов/ответов (включая заголовки и тело ответа);

- Скриншоты должны содержать контекст выполнения: URL, заголовки запроса/ответа, тело JWT (в декодированном или raw-виде), и/или вывод инструментов (например, hashcat, jwt.io, Burp Repeater)
- Скриншот/лог декодирования JWT на jwt.io или в Burp Inspector (до и после модификации);
- Для атак с подменой алгоритма/внедрением ключа — скриншот заголовка токена с изменённым `alg`, `kid`, `jwk`;
- Для `alg=none` — подтверждение отсутствия подписи (т.е. токен оканчивается на `.` без третьей части);
- Для брутфорса — вывод hashcat с найденным ключом.
- Наличие скриншота с отметкой Lab Solved с главной страницы каждого задания (ссылки на задания в данной методичке – это ссылки на главные страницы заданий).

Анализ уязвимости и риск

Попробуйте оценить потенциальный риск данной уязвимости.

- Является ли нарушение следствием ошибки **валидации** (не проверяется `alg`, `iss`, `aud`, `exp`) или **логики проверки подписи** (принимается `none`, доверяется `jwk`, не фиксируется алгоритм);
- Возможность горизонтального/вертикального эскалирования (например, замена `sub: wiener` → `administrator` — вертикальный; `sub: wiener` → `carlos` — горизонтальный);
- Потенциал массовой эксплуатации (например, утечка секрета → компрометация всех токенов за период его действия)

.

Чек-лист для самопроверки

№	Элемент отчёта	Обязателен?	Примечание
1	Этическое подтверждение: «Работа выполнена только в лабораторной среде»	Да	Короткий текст в начале отчёта
2	Указаны: выбранные задания (номера) и краткий результат	Да	Кратко: какие задания сделаны и итог (например, /admin доступ или Lab Solved)
3	Инструменты + версии (Burp, hashcat, jwt-cli, openssl, ОС и т.д.)	Да	Пример: Burp Suite Community 2024.10, hashcat 6.2.6

4	Исходный JWT (raw) — скрин или текст	Да	Показываем оригинал до модификации
5	Исходный JWT (decoded JSON) — скрин или текст	Да	Для удобства проверяющего
6	Модифицированный JWT (raw) — скрин/текст	Да	Тот, который использовали для атаки
7	Модифицированный JWT (decoded JSON) — скрин/текст	Да	Выделите изменённые поля (например, sub: administrator)
8	Шаги воспроизведения — пошагово (команды/инструменты/какие поля изменяли)	Да	Должно быть достаточно, чтобы третье лицо повторило
9	Скриншоты/логи HTTP-запросов и ответов (Proxy/Repeater/History) с временной меткой	Да	Допустимо: скрин History с выделенным запросом; Burp фиксирует Time
10	Скриншот «Lab Solved» или иной итоговый артефакт (если есть)	Если применимо	Для PortSwigger labs — обязательно приложить
11	Для brute-force: вывод hashcat / команда и найденный секрет	Только для соответствующих заданий	Укажите словарь/режим hashcat
12	Для jwk/kid/alg=none: скрин заголовка токена (header) с изменёнными полями	Да, если применимо	Покажите header: alg/kid/jwk
13	Анализ уязвимости и рекомендации по исправлению (1–2 параграфа)	Да	Что пропущено и как фиксить (конкретные меры)
14	Приложены все артефакты (экспорт Repeater, текстовые файлы, логи)	Опционально	ZIP с артефактами приветствуется
15	Отчёт форматирован по шаблону (титул, содержание, шаги, выводы)	Да	Используйте предоставленный шаблон отчёта