Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет прикладной информатики (ФПИН)

Лабораторная работа №6

по дисциплине: Основы кибербезопасности

Автор: доцент практики, кандидат технических наук

Кравчук Алексей Владимирович

Санкт-Петербург
2025

Тема занятия: Формирование CVSS-векторов и плана устранения уязвимостей, включая компенсирующие меры.

Цель работы:

- Закрепить навыки анализа отчётов сканеров безопасности (на примере Acunetix).
- Получить навыки формирования векторов оценки уязвимостей по стандарту CVSS v4.0.
- Разработать план устранения выявленных проблем, включая компенсирующие меры (mitigation)

Задание:

1. Ознакомьтесь с отчётом Acunetix, полученным при сканировании уязвимого веб-приложения OWASP Juice Shop в ходе выполнения лабораторной работы № 5.

2. Выберите три уязвимости различного уровня критичности (High, Medium, Low). Для каждой уязвимости:

- Определите её тип, место возникновения и потенциальное воздействие.
- Сформируйте базовый CVSS v4.0-вектор с пояснением выбранных метрик.
- Рассчитайте итоговую оценку с использованием официального калькулятора FIRST (https://www.first.org/cvss/calculator/4.0).

4. Выберите 3 типа уязвимостей (CWE, Common Weakness Enumeration), представьте в отчете их краткое описание и поясните, как этот тип уязвимостей реализуется в веб-приложении Juice Shop.

5. Составьте итоговый план устранения уязвимостей, включая компенсирующие меры, **в табличной форме**: №п/п (номер шага), приоритет, уязвимость, действия для устранения, компенсирующие меры, ответственный (DevOps, разработчик Backend, разработчик Frontend, системный администратор, руководитель).

Краткие теоретические сведения

CVSS (Common Vulnerability Scoring System) — это международный стандарт оценки степени опасности уязвимостей. Он позволяет унифицировано оценивать риск по набору метрик, отражающих сложность эксплуатации, уровень требуемых привилегий, влияние на конфиденциальность, целостность и доступность системы.

CVSS-вектор – это строка, описывающая значения метрик уязвимости. Например:

CVSS:4.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:L/A:N

Где:

AV — Attack Vector (вектор атаки)

AC — Attack Complexity (сложность атаки)

PR — Privileges Required (требуемые привилегии)

UI — User Interaction (взаимодействие с пользователем)

S — Scope (область воздействия)

C/I/A – Confidentiality, Integrity, Availability (влияние на конфиденциальность, целостность, доступность)

Практическая часть

Ниже приведены примеры типичных уязвимостей, соответствующих им CVSS-векторов и краткие рекомендации по устранению уязвимостей.

Пример 1. Reflected XSS

CVSS-вектор: CVSS:4.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:L/A:N

Пояснение: уязвимость эксплуатируется по сети, проста в исполнении, не требует авторизации, требует взаимодействия пользователя. Может привести к утечке данных (C:H), нарушению целостности (I:L).

Рекомендации: фильтрация пользовательского ввода, внедрение Content Security Policy, использование безопасных API для вывода данных.

Пример 2. SQL Injection

CVSS-вектор: CVSS:4.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

Пояснение: эксплуатация возможна удалённо без авторизации, может привести к чтению, изменению и удалению данных. Высокое воздействие на конфиденциальность, целостность и доступность.

Рекомендации: использовать параметризованные запросы (prepared statements), ORM, фильтрацию входных данных.

Пример 3. Information Disclosure

CVSS-вектор: CVSS:4.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

Пояснение: данные могут быть раскрыты без аутентификации, например через ошибки сервера или лог-файлы. Высокое влияние на конфиденциальность, но не на целостность или доступность.

Рекомендации: отключить отладочный вывод, ограничить информацию об ошибках, использовать централизованное логирование.

Пример 4. Broken Access Control (IDOR)

CVSS-вектор: CVSS:4.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N

Пояснение: пользователь с низкими правами может получить доступ к чужим ресурсам. Высокое воздействие на конфиденциальность и целостность.

Рекомендации: проверка полномочий на каждый запрос, принцип наименьших привилегий.

Пример 5. План устранения уязвимостей — последовательность и фазы

В этой секции приведена рекомендуемая методика поэтапного устранения уязвимостей. Последовательность выбирается исходя из критичности (CVSS score), вероятности эксплуатации и потенциального воздействия на бизнес.

Общие принципы приоритезации:

1. Сначала — Critical / High (уязвимости, которые позволяют удалённое выполнение кода, SQLi, RCE, серьёзные утечки данных).
2. Далее — Medium (XSS, CSRF, некоторые конфигурационные проблемы).
3. В конце — Low (информационные утечки низкой чувствительности, directory listing и т.п.).

Фазы устранения (шаблон):

1. Оперативные компенсирующие меры (Immediate/Short-term): быстро внедряемые меры для снижения риска.
2. Быстрое исправление (Short-term Fix): исправления, которые можно сделать за 1–7 дней.
3. Техническое решение (Medium-term): архитектурные изменения, патчи, изменение конфигураций (1–4 недели).
4. Долгосрочные меры (Long-term): улучшения процесса разработки, CI/CD, обучение, мониторинг (>4 недель).