Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

Факультет информационных технологий и анализа больших данных

РАЗРАБОТКА И ВЕРИФИКАЦИЯ МОДЕЛИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ АНАЛИЗА ЗАЩИЩЕННОСТИ WEB-ПРИЛОЖЕНИЙ НА ОСНОВЕ МЕТОДОЛОГИИ OWASP TOP 10



Докладчик: студент группы УИБ25-1м, направление подготовки 10.04.01 Информационная безопасность (Управление информационной безопасностью в кредитно-финансовой сфере)

Матевосян Гурген Арменович

Актуальность и проблема

Критичные классы уязвимостей (OWASP Top 10)

- A01:2021 — Broken Access Control;
- A03:2021 — Injection;
- A05:2021 — Security Misconfiguration

Недостатки существующих сканеров (Acunetix, Nessus, OpenVAS, Nikto)

- Высокий уровень ложных срабатываний;
- Отсутствие контекста бизнес-логики (сканер не понимает, что страница «404 Not Found» не является SQL-инъекцией)



Цель и формальная постановка задачи

Цель: представление формальной модели автоматизированной системы тестирования безопасности, снижающей вероятность ложных срабатываний

Формальная постановка:

WEB-приложение $W = \{E_1, E_2, \dots, E_n\}$, где $E_i = (URL_i, Method_i, Params_i, Headers_i)$

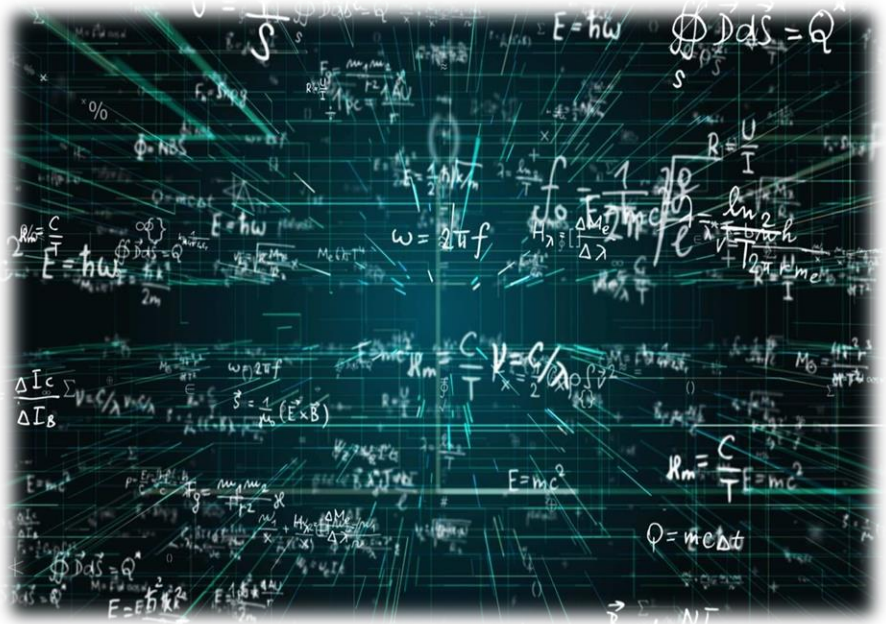
Множество уязвимостей $U = \{U_1, U_2, \dots, U_{10}\}$

Классический сканер $S_{classic}$: проверка $\forall E_i$ на $\forall U_j \rightarrow$ $\forall U_j \rightarrow$ сложность $O(n * m)$

Требуется построить модель ММ:
$Result = \{(E_i, U_j, risk) \mid F_c(E_i, W) = True \wedge V(E_i, U_j) = True\}$

F_c — функция фильтрации контекста

V — функция верификации



Архитектура модели

1. Модуль рекогносцировки (Spider/Crawler)

- Обход приложения с учетом графа переходов;
- Поддержка сессий (JWT/Bearer tokens) и форм с CSRF-защитой

2. Базовый модуль активного анализа (OWASP Core)

- Библиотека пэйлоадов (SQLi, XSS, LFI, SSTI);
- Трехфазный тест: эталонный запрос → атакующий запрос → верификационный запрос

3. Модуль контекстной фильтрации (ключевое новшество)

Срабатывание отвергается, если одновременно:

- HTTP-код ответа совпадает с базовым;
- Длина ответа отличается от базовой не более чем на 5%;
- Пэйлоад не вызвал структурных изменений в DOM (сравнение через LCS).

Исключение: Blind Injection — критерий $\Delta t > 5000$ мс (анализ времени отклика).

4. Модуль оценки рисков

- Учет критичности эндпоинта (PII, данные авторизации);
- Вычисление итогового скоринга

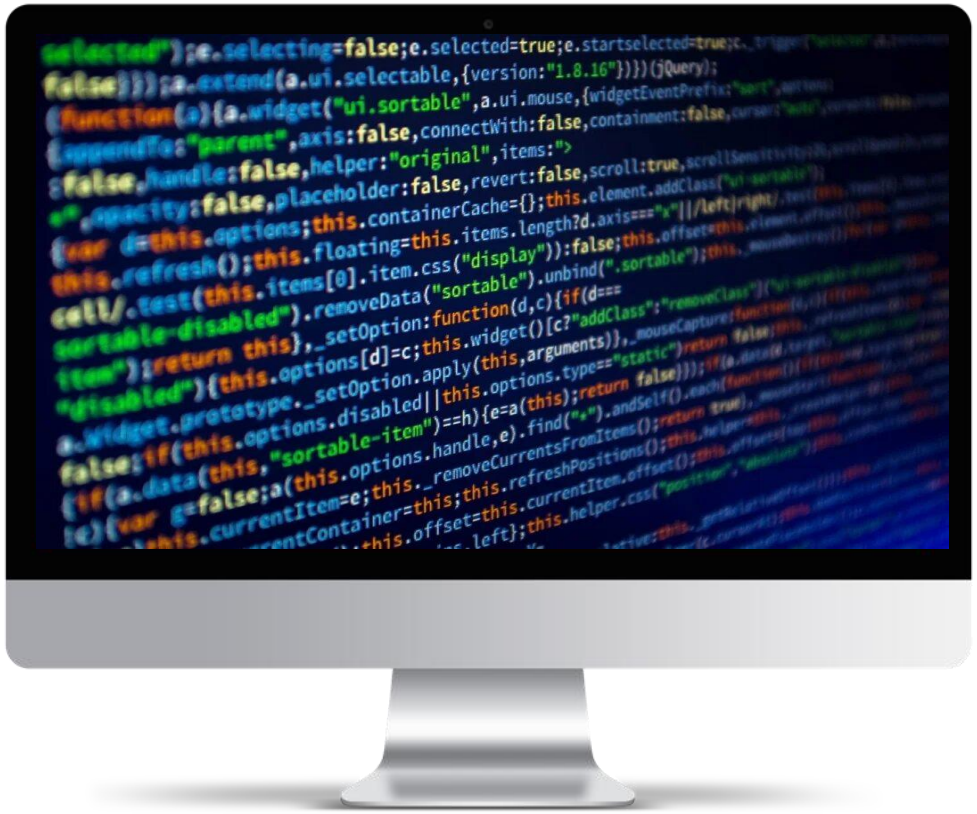
Пример работы фильтрации

Типичная ошибка ZAP

Сообщает об XSS на странице поиска, если в ответе возвращается введенная строка без экранирования.

Представленная модель

- Отправляет <script>alert(1)</script>;
- Получает тот же HTML;
- Отправляет нейтральную строку "test123";
- Визуально сравнивает DOM-деревья (алгоритм LCS);
- Отвергает уязвимость, если структура не изменилась.



Заключение

Основные результаты (теоретическая разработка):

- Формализована задача снижения ложных срабатываний через введение контекстной фильтрации;
- Разработан алгоритм трёхфазной верификации (база — атака — нейтрализация);
- Предложена архитектура модели, включающая модуль контекстной фильтрации и модуль оценки рисков на основе CVSS 3.1

Границы исследования:

- Работа носит теоретический характер: модель описана формально, экспериментальная верификация — предмет дальнейших исследований;
- Для практической реализации требуется программная реализация прототипа и его проверка на тестовом полигоне

Возможные перспективы развития:

- Добавление модуля анализа бизнес-логики (IDOR — Insecure Direct Object References);
- Интеграция с системами управления уязвимостями (Defect Dojo);
- Предложение: апробировать модель на реальном приложении

ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

Спасибо за внимание!