Региональный конкурс научно-технологических проектов

“Большие вызовы”

Секция: Информационная безопасность

“Создание расширения для браузера для автоматического обнаружения подозрительных скриптов на сайтах”

Работу выполнил:

Комаров Святослав Михайлович

Руководитель проекта**:**

Городилов Александр Алексеевич

Новосибирск, 2026

Оглавление

1. Введение

В современном мире интернет является неотъемлемой частью жизни человека. Ежедневно миллионы пользователей посещают сайты, используют онлайн-сервисы, социальные сети и веб-приложения. Однако вместе с удобством и доступностью возрастает и количество угроз, связанных с безопасностью в сети Интернет. Одной из наиболее распространённых проблем является использование злоумышленниками вредоносных JavaScript-скриптов, которые могут собирать личные данные пользователей, перенаправлять на опасные сайты или выполнять скрытые действия без ведома пользователя.

В связи с этим особую актуальность приобретает разработка инструментов, способных анализировать содержимое веб-страниц и предупреждать пользователя о потенциальных угрозах. Одним из таких инструментов являются браузерные расширения, которые позволяют в реальном времени контролировать работу сайтов и обеспечивать дополнительный уровень защиты.

В рамках данного проекта было разработано браузерное расширение, предназначенное для анализа загружаемых скриптов и выявления потенциально опасного поведения. Проект сочетает в себе знания в области веб-разработки, программирования и основ информационной безопасности.

1.1 Цель

Создать расширение для браузера, которое будет динамически анализировать выполняемые на странице скрипты, выявлять потенциально опасные и предупреждать пользователя о возможных угрозах.

1.2 Задачи

1. Изучить существующие подходы к анализу и детекции вредоносных скриптов.

2. Разработать архитектуру браузерного расширения для мониторинга и анализа загружаемых скриптов.

3. Подготовить набор данных и обучить простую модель машинного обучения для классификации скриптов.

4. Интегрировать механизм анализа и уведомления пользователя о потенциальных угрозах.

5. Провести тестирование расширения и оценить точность выявления подозрительных скриптов.

1.3 Актуальность

Актуальность данного проекта обусловлена постоянным ростом числа интернет-угроз. Вредоносные скрипты используются для кражи персональных данных, отслеживания действий пользователей и распространения вредоносного программного обеспечения. Большинство обычных пользователей не обладает достаточными знаниями для самостоятельного анализа кода веб-страниц, что делает их уязвимыми.

Разработанное расширение позволяет автоматизировать процесс анализа и предоставить пользователю понятную информацию о потенциальных рисках. Таким образом, проект может быть полезен как для начинающих пользователей, так и для тех, кто интересуется вопросами кибербезопасности.

2. Теоретическая часть

2.1 Ознакомление и изучение необходимых языков программирования

1. HTML – используется для структуры интерфейса расширения.
    - Создаёт каркас всплывающего окна, страницы настроек и панели отображения результатов анализа.
2. JavaScript — основной язык разработки браузерного расширения.
    - Отвечает за анализ и обработку загружаемых скриптов.
    - Реализует основную логику мониторинга и фильтрации
    - Работает с DOM – структурой страниц, перехватывая подозрительные скрипты до их выполнения
3. CSS – отвечает за визуальное отображение расширения (цвет, шрифт текста, отступы и бортики)
    - Придаёт современный и аккуратный вид окну расширения, кнопкам и таблице анализа

4. JSON – применяется для хранения настроек и правил анализа, а также необходим для загрузки расширения в браузере Google Chrome и последующей его работоспособности.
5. Python – используется для модели ML* (ML – machine learning).

Примечание*

*Машинное обучение используется в проекте для повышения точности обнаружения подозрительных скриптов.

В процессе работы была рассмотрена задача классификации, при которой JavaScript-скрипты разделяются на безопасные и потенциально опасные. Для этого анализируются различные характеристики скриптов, такие как структура кода, наличие определённых функций, частота обращений к внешним ресурсам и другие признаки.

Использование Python для обучения модели обусловлено наличием большого количества библиотек для работы с данными и машинным обучением, а также простотой реализации и наглядностью кода. Даже простая модель позволяет повысить эффективность анализа и сделать систему более гибкой.

2.2 Принципы анализа JavaScript-скриптов

JavaScript-скрипты могут выполнять как полезные, так и вредоносные действия. Анализ скриптов может включать:

- поиск подозрительных конструкций кода,
- выявление попыток скрытого сбора данных,
- анализ обращений к внешним ресурсам,
- определение потенциально опасных функций.

Результаты анализа представляются пользователю в удобной форме, что позволяет принять решение о дальнейшем использовании сайта.

2.3 Проблема безопасности JavaScript в современном вебе

JavaScript является одним из наиболее распространённых языков программирования в веб-среде. Практически каждый современный сайт

использует JavaScript для реализации интерактивных элементов, динамической загрузки данных и взаимодействия с пользователем. Однако именно из-за своей гибкости и широких возможностей JavaScript часто становится инструментом для реализации вредоносных действий.

Злоумышленники могут использовать JavaScript для скрытого выполнения кода, перенаправления пользователя на фишинговые сайты, внедрения рекламных скриптов или сбора персональных данных без согласия пользователя. Особую опасность представляют скрипты, загружаемые с внешних ресурсов, так как пользователь не имеет прямого контроля над их содержимым.

В связи с этим задача автоматического анализа JavaScript-кода становится особенно актуальной. Использование браузерных расширений позволяет выполнять такой анализ непосредственно в момент загрузки страницы, что повышает уровень безопасности и снижает риск негативных последствий для пользователя.

3. Практическая часть

3.1 Реализация расширения

Разработка проекта велась с использованием среды разработки Visual Studio Code. Основной упор был сделан на практическую реализацию логики анализа и удобство пользовательского интерфейса.

В ходе разработки:

- был создан файл конфигурации расширения,
- реализованы скрипты анализа содержимого страниц,
- разработан интерфейс отображения результатов,
- проведено тестирование работы расширения на различных сайтах.

3.2 Оформление (дизайн)

Для оформления расширения использовались HTML и CSS. Интерфейс выполнен в минималистичном стиле, что позволяет пользователю быстро

понять результаты анализа. Вся информация отображается структурировано и наглядно.

3.3 Тестирование и анализ результатов

Для проверки корректности работы расширения было проведено тестирование на различных веб-сайтах, содержащих как стандартные, так и более сложные JavaScript-скрипты. В процессе тестирования оценивалась стабильность работы расширения, корректность анализа и наглядность отображаемой информации.

Особое внимание уделялось тому, чтобы расширение не влияло на корректную работу сайтов и не вызывало ошибок в отображении контента. Результаты тестирования показали, что разработанное расширение успешно выявляет скрипты и почти с 90% вероятностью выявляет их уровень опасности (безопасный, подозрительный, опасный), а также своевременно уведомляет пользователя о потенциально подозрительной активности.

Полученные результаты подтверждают работоспособность предложенного решения и возможность его дальнейшего развития.

4. Заключение

В ходе выполнения проекта были получены практические навыки разработки браузерных расширений, работы с языками веб-программирования и основами информационной безопасности. Проект позволил на практике применить теоретические знания и лучше понять принципы защиты пользователей в сети Интернет.

Разработанное расширение может быть использовано в учебных целях, а также послужить основой для дальнейшего развития, включая добавление и последующее расширение набора эвристик, улучшение модели машинного обучения и добавление новых функций анализа. Таким образом, поставленные цель и задачи проекта были успешно достигнуты.

4.1 Литература и интернет-источники

1. https://developer.mozilla.org/ru/docs/Web/JavaScript (Официальная документация по JavaScript)
2. https://developer.mozilla.org/ru/docs/Web/HTML (Официальная документация по HTML)
3. https://developer.mozilla.org/en-US/docs/Web/CSS (Официальная документация по CSS)
4. https://cloud.ru/blog/mashinnoye-obucheniye-na-python?ysclid=mj85xit02c777840454 (Документация по созданию модели машинного обучения на Python)
5. https://developer.mozilla.org/ru/docs/Mozilla/Add-ons/WebExtensions (Документация по разработке браузерных расширений)
6. https://elar.urfu.ru/bitstream/10995/73899/3/978-5-7996-2677-8_2019.pdf?ysclid=mj85ofhatn827090188 (Материалы по основам информационной безопасности)

4.2 Пример работы проекта

https://disk.yandex.ru/d/WQiHv73GEgToAw (Ссылка на видео с работой проекта, которое лежит на Яндекс диске)

https://github.com/W1RCHER/PopupProjectBC.git (Ссылка на репозиторий на github.com)