Текст выступления

Здравствуйте, уважаемые жюри, сегодня я бы хотел представить вам свой проект, а именно программу для защиты от вредоносных сайтов.

В настоящее время наблюдается стремительное увеличение числа киберпреступлений, и фишинг является самым распространённым из них. Фишинг это вид киберпреступления, при котором жертва неосознанно передаёт свои конфиденциальные данные мошенникам.

С каждым годом фишинговые атаки модернизируются, а их методы меняются, но цели фишинговых атак зачастую остаются неизменными. Например, основной целью фишинга за 2023 год являлась почта, занимая 96% от общего числа атак.

Кратко опишу основные методы фишинговых атак:

Спам – этот вид фишинга подразумевает под собой отправку письма с вредоносной вложением/ссылкой огромному числу пользователей почти без разбора.

Спир фишинг – он почти идентичен спаму, но мошенники знают кого преследуют, из-за чего атака становится персонализированной.

Вэйлинг – также почти идентичен спир фишингу, но нацелен на руководящие должности.

Как мы видим, основные виды мошенничества на почте почти одинаковы и подразумевают под собой наличие опасного сайта/вложения.

(Актуальность проблемы)

Для начала работы нужно понять актуальность проблемы, ведь от него зависит востребованность продукта. Для этого было проведено исследование как на глобальном уровне, так и на локальном.

Был проведён опрос среди 10 классов, из которого следует, что более 50% опрошенных сталкиваются с фишингом на постоянной основе.

Также были составлены графики на базе отчёта лаборатории касперского, из которых следует, что почти 50% писем являлись спамом, но больше всего нас интересуют эти два графика. На первом изображено количество вредоносных вложений, а на втором количество опасных ссылок в письмах. Вредоносных вложений было всего 130млн., а опасных ссылок 710млн. По этим данным мы можем судить, что проблема ссылок является намного более актуальной.

(Цели и задачи проекта)

**Цель работы**: Создание программы для защиты от вредоносных сайтов, удобной для работы массовому пользователю

Задачи:

1. Провести анализ потребностей целевой аудитории;
2. Выбрать оптимальные методы и инструменты для разработки решения проблемы;
3. Рассмотреть аналоги продукта;
4. Разработать рабочий прототип программы;
5. Составить план дальнейшего развития проекта;
6. Провести опрос про удобство рабочего прототипа;

(Целевая аудитория)

Для полноценной конкуренции с конкурентами наша программа должны быть рассчитана на массового пользователя, соответственно обязана обладать интуитивно понятным интерфейсом, портативностью и низкой нагрузкой на устройство.

(Выбор инструментов)

Я считаю, что выбор фреймворка является крайне важным шагом, отчего я решил составить таблицу сравнения фреймворком для подбора подходящего мне. Qt был выбран потому, что вне зависимости от операционной системы он будет обладать высокой производительностью и нормальным весом конечной программы.

(Выбор методов)

Для реализации в программе было рассмотрено 3 метода:

Анализ сайта с помощью машинного обучения, Ручная проверка ссылки с использованием API/базы данных, автоматическая защита с помощью днс.

В качестве API был выбран VirusTotal

(Сравнение с аналогами)

Для сравнения нашей программы с аналогами была составлена таблица, на которой мы можем увидеть, что будущему функционалу нашей программы соответствует только десктопная версия касперского. Но как мы можем видеть на опросе справа, более 84% не готовы платить за антивирус, а касперский в свою очередь стоит 1800 руб в год, в то время как наша программа как минимум в первое время будет распространяться бесплатно.

(Этап разработки)

Для начала было решено сделать набросок программы без кода и стилизации, вид программы на этом этапе показан на экране.

(стилизация)

Для стилизации был использован язык CSS

(разработка/пример работы)

Было решено писать код на питон обёртке Qt, а именно на PySide. Для того чтобы соответствовать потребностям разных кругов лиц также были реализованы упрощённый режим обработки программы и расширенный. Апи вирустотала отправляет данные в виде словарей с числовыми значениями, что может быть непонятно массовому пользователю, вследствие чего в упрощённом режиме информация будет считаться и выдаваться в простом виде, в словах безопасно, опасно и т.д., а в расширенном режиме информация будет выводиться в изначальном виде, но всё равно будет содержать только самое важное.

(тест программы)

Был проведён тест ручного режима проверки в 7 разных ситуациях, и программа успешно прошла все тесты. Скриншоты к итогам тестов находятся на гитхабе.

(Дальнейшее развитие проекта 1)

На данный момент реализован только вариант с прямой отправкой запроса на апи вирутотала, но для избежания ограничений накладываемых вирустоталом на использование API необходимо создать систему, при которой сайт сначала будет искаться в базе данных, а при отсутствии сайта запрос уже будет идти на апи.