**МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГИМНАЗИЯ № 1**

**ИНДИВИДУАЛЬНЫЙ ИТОГОВЫЙ ПРОЕКТ**

**Тема**

**Программа для защиты от мошеннических сайтов**

**Выполнил:**

**Бородин Денис**

**Ученик 10 класса «Б»**

**Куратор проекта:**

**г.о. Жуковский**

**Введение**

В настоящее время наблюдается стремительно увеличение числа киберпреступлений. Самым распространенным видом киберпреступления является мошенничество. В этом случае жертва зачастую добровольно оставляет конфиденциальную информацию, которую мошенники впоследствии используют для нанесения материального вреда. Получение информации чаще всего происходит посредством фишинга – вида мошенничества в интернете.

С каждым годом количество фишинговых атак увеличивается, а их методы модернизируются. Кроме того, на эффективность атак влияет человеческий фактор, чем мошенники удачно пользуются, применяя социальную инженерию. Одним из самых популярных методов фишинга является обман с помощью фишинговых (поддельных) сайтов.

В данной работе будет затронут теоретический материал по теме фишинга и фишинговых сайтов, проведен анализ возможных угроз и их решений, а также будут разработаны практические методы противодействия мошенникам в виде компьютерной программы.

**Гипотеза**: возможно разработать программу, которая закроет потребности в защите большинства пользователей от вредоносных сайтов.

Задачи для доказательства гипотезы:

1. Проанализировать методы защиты от мошеннических сайтов и выбрать подходящие.
2. Выбрать оптимальные инструменты для разработки решения проблемы.
3. Разработать интерфейс понятный рядовому пользователю.
4. Разработать программу с упором на маленький вес и быструю работу.
5. Провести опрос про удобство получившейся программы.

**Цель работы**: Создание программы по защите от вредоносных сайтов/ссылок, удобной для работы массовому пользователю.

**План работы:**

1. Изучение информации по Фишингу и фишинговым сайтам.
2. Рассмотрение возможных угроз, которые могут нести фишинговые сайты.
3. Выбор оптимальных методов и инструментов для разработки решения проблемы.
4. Описание процесса разработки программы.
5. Оценка полученных результатов.

Объектом исследования являются фишинговые сайты.

Предметом исследования являются угрозы мошеннических сайтов.

При выполнении работы были применены следующие методы исследования:

1. Теоретические: анализ литературы, практическое моделирование.

Практическая значимость проекта состоит в возможности использования результатов работы в повседневной жизни и рабочей обстановке.

**Глава 1. Исследование фишинга и его составляющих**

* 1. **Фишинг и его разновидности**

Фишинг – вид интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей – логинам, паролям, персональным данным и т.д. Выдавая себя за человека или говоря от имени организации, которым пользователь доверяет, мошенник легко может узнать реквизиты банковской карты или данные паспорта. Другими словами, при помощи методов социальной инженерии он ловит пользователя на наживку доверия, чтобы получить ценную информацию.

Для задания курса проекта стоит рассмотреть наиболее известные вариации фишинга:

1. Email/spam – наиболее распространенный вид фишинга. Поддельное письмо отправляется огромному количеству пользователей, в надежде на то, что фишинг атака закончится успехом.
2. Spear phishing – похож на спам, но отличается тем, что атака становится более целенаправленной. Злоумышленники знают кого преследуют, и заранее исследуют цель для подготовки наиболее удачной атаки. Может затрагивать как обычных пользователей, так и организацию.
3. Whaling – метод не сильно отличающийся от spear phishing, но целевая группа становится более специфичной и ограниченной. Этот метод нацелен на руководящие должности.
4. Vishing (Voice phishing) – метод, в котором мошенник осуществляет телефонный звонок жертве с целью выполнения жертвой определенных действий. В данном типе мошенники выдают себя за реально существующую компанию/личность для получения доверия со стороны жертвы, либо же для оказания давления на нее.
5. Clone phishing – метод, в котором настоящее и ранее отправленное электронное письмо, содержащее вложение или ссылку, используется для создания идентичного письма, в котором настоящие вложения/ссылки будут заменены на вредоносные.

Из этих вариантов фишинговых атак можно понять, что большинство из них имеет связь с поддельными сайтами или основывается на них.

* 1. Актуальность решения проблемы фишинга

Анализ актуальности проблемы фишинга будет основан на отчёте “Лаборатории Касперского” по спаму и фишингу за 2023 год.[[1]](#footnote-1) Основной целью фишинговых атак в последние года является почта – 96% от всех атак.[[2]](#footnote-2) На Рис.1. мы можем видеть диаграмму, на которой показана доля спама в почтовом трафике российского интернета. В среднем на протяжении 2023 года она составляла 46,59%, что является крайне большим показателем и указывает на то, что почти каждое второе письмо является спамом. Но больше всего нас интересуют Рис.2. на котором изображена диаграмма количества срабатываний антивируса на попытки открытия вредоносных файлов во вложениях к письмам и Рис.3. на котором изображена диаграмма количества срабатываний антивируса на попытки перехода по вредоносным ссылкам в письмах. Всего попыток открыть вредоносные файлы было 136 миллионов, а попыток перейти по опасным ссылкам – 710 миллионов. По этим данным мы можем судить, что количество попыток открыть вредоносные ссылки преобладает более чем в 5 раз. Соответственно, проблема фишинговых сайтов актуальней и подлежит скорейшему решению.

**Глава 2. Программа для защиты от мошеннических сайтов**

**2.1 Обучение или программа**

Для построения защиты от фишинговых атак можно выделить два основных направления:

- Проведение обучения пользователей.

- Реализовать программное средство защиты для недопущения атак/предупреждения их.

Можно обучать пользователей основам сетевой безопасности с помощью программного обеспечения, но данный способ будет намного более затратен по времени со стороны пользователей, а также потребует от них постоянно сохранять бдительность. Защита же пользователей с помощью приложения позволит им меньше задумываться об опасности, так как они будут постоянно защищены в фоновом режиме.

**2.2 Разработка приложения**

**2.2.1 Выбор методов и инструментов для создания решения**

От фишинга страдают все слои населения, от детей до пожилых людей, вследствие чего было решено делать программу, имеющую нулевой порог вхождения, эстетически приятную, а также имеющую функционал под каждого пользователя.

В качестве фреймворка был выбран Qt, а языком для работы с ним был выбран Python с библиотекой PySide. Выбор фреймворка был обусловлен несколькими факторами: Кроссплатформенностью, удобством работы с фреймворком, скоростью разработки программ, а также скоростью программ, написанных на нём. Среди фреймворков также были рассмотрены такие как варианты, как Electron и Flutter.

Pyside был выбран потому, что PyQt распространяется по лицензии GPL, которая ограничивает использование в коммерческих проектах, а также требует открытого исходного кода. В то же время PySide распространяется по лицензии LGPL, в которой такие ограничения отсутствуют.

Среди методов можно выделить 3 основных актуальных:

1. Анализ сайта с помощью машинного обучения по совокупности его характеристик. При таком методе рассматриваются абсолютно разные характеристики – от регистрационных данных сайта до характеристик его URL адреса. Проблема этого метода состоит в том, что он очень сложно алгоритмически реализуется, а также не поддается стандартизации. Соответственно, данный вариант нам не подходит.
2. Ручная проверка ссылки с использованием базы со списком адресов фишинг-сайтов.
3. Использование автоматической проверки сайтов, на которые заходит пользователь с помощью DNS.

Из этих трёх вариантов для нашей задачи нам подходит только автоматическая и ручная проверка. Можно было бы ограничиться только автоматической проверкой, но мы решили сделать приложение, подходящее для всех – в том числе и для более продвинутых пользователей, соответственно реализуем мы оба этих метода.

В качестве базы данных для ручной проверки ссылок был выбран API от VirusTotal, так как он предоставляет крайне обширный набор данных, а также имеет наиболее высокую репутацию среди себе подобных. Для автоматической проверки ссылок я выбрал Яндекс DNS.

**2.2.2 Процесс создания программы**

Для начала, я решил сделать тело программы, а к её логике перейти уже когда будет готовый интерфейс. Так как мне предстояло написать свою первую программу с полноценным графическим интерфейсом – мне надо было обучиться фреймворку Qt. В Qt разработка приложения может производиться как полностью через код, так и через комбинацию графического интерфейса Qt designer и написания кода. Это позволит частично разделить логику программы от её графической части. Обучался я по официальной документации Pyside и руководству.[[3]](#footnote-3) Главной сложностью при обучении Qt designer было то, что для нормального создания интерфейса у каждого элемента должно быть своё статическое место в интерфейсе, из-за чего элементы нельзя свободно двигать по экрану, а размеры элемента регулируются через коэффициенты. Но зато это позволяет сделать адаптивный интерфейс, который не будет съезжать при изменении размера окна и будет меняться под него. На Рис.4 изображено как выглядит интерфейс моей программы, но сразу видно, чего ей не хватает – стилизации, так что это было моим следующим шагом.

Стилизация также с помощью кода, но с помощью языка CSS (англ. Cascading Style Sheets – “Каскадные Таблицы Стилей”). Был выбран минималистичный стиль для максимального удобства работы с программой. На Рис.5 показан тот же самый интерфейс, но уже после стилизации.

Дальше было начато создание логики программы. С самого начала было добавлено взаимодействие c API VirusTotal. После, была добавлена функция, которая не позволяет нажать на кнопку проверки ссылки пока ссылка не будет верной.

Для прототипа программы был сделан 1 режим – ручная проверка ссылки с упрощением под массового пользователя. Итоговый вид прототипа показан на Рис.6.

**Заключение**

Я считаю, что смог создать прототип приложения, которое сможет удовлетворить потребности людей в защите от мошеннических сайтов. У программы есть много векторов развития, от простого добавления новых функций, как поддержка DNS, так и создание сайта/мобильного приложения.

Приложения

Рис. 1. Доля спама в почтовом трафике российского интернета

Рис. 2. Количество срабатываний почтового антивируса на попытки открыть вредоносный файл

Рис. 3. Количество срабатываний системы “Антифишинг” на попытки перехода по мошеннической ссылке

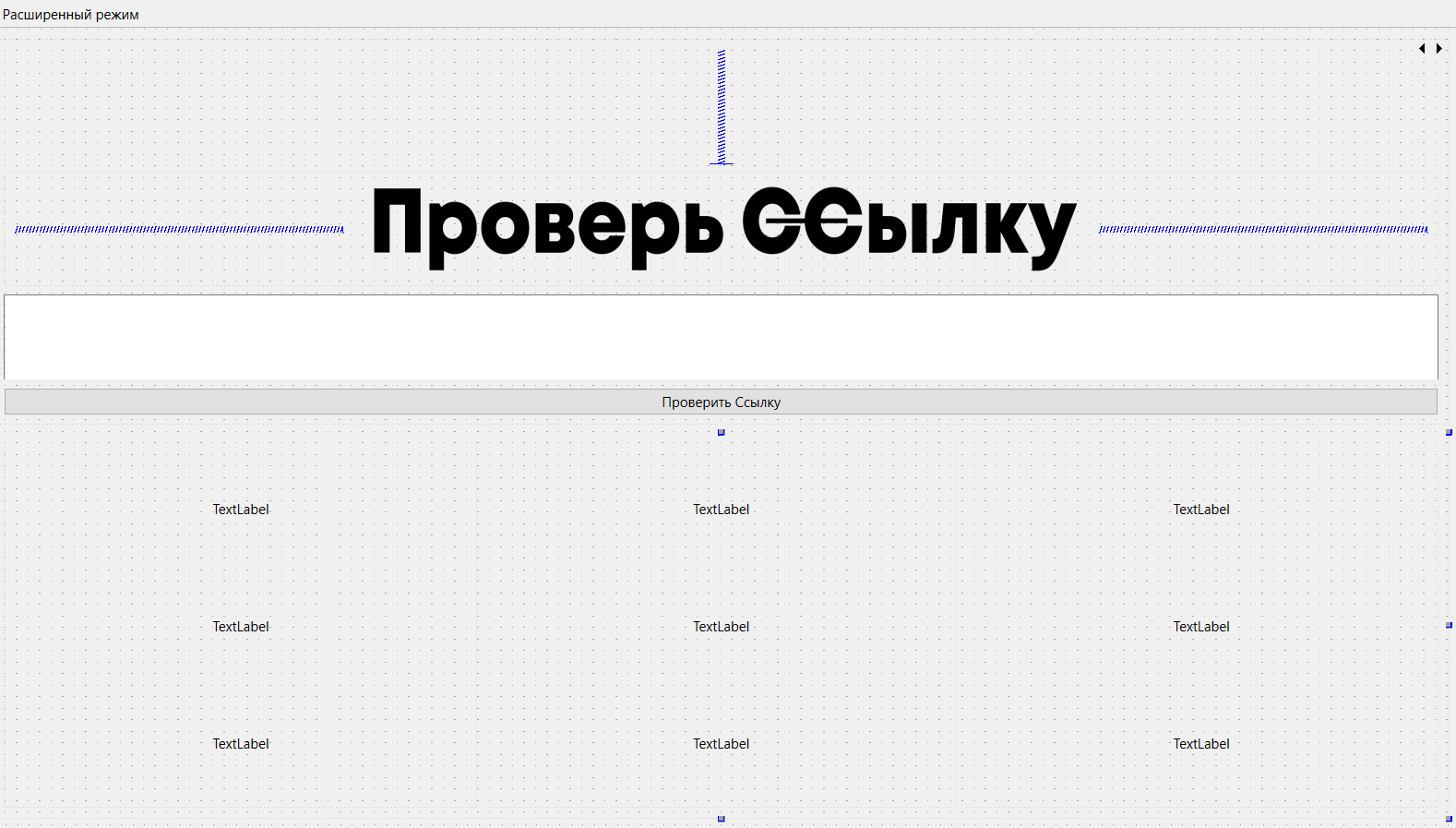


Рис. 4. Начальный интерфейс программы без стилизации (на скриншоте присутствуют системные элементы)



Рис. 5. Начальный интерфейс программы со стилизацией (на скриншоте присутствуют системные элементы)

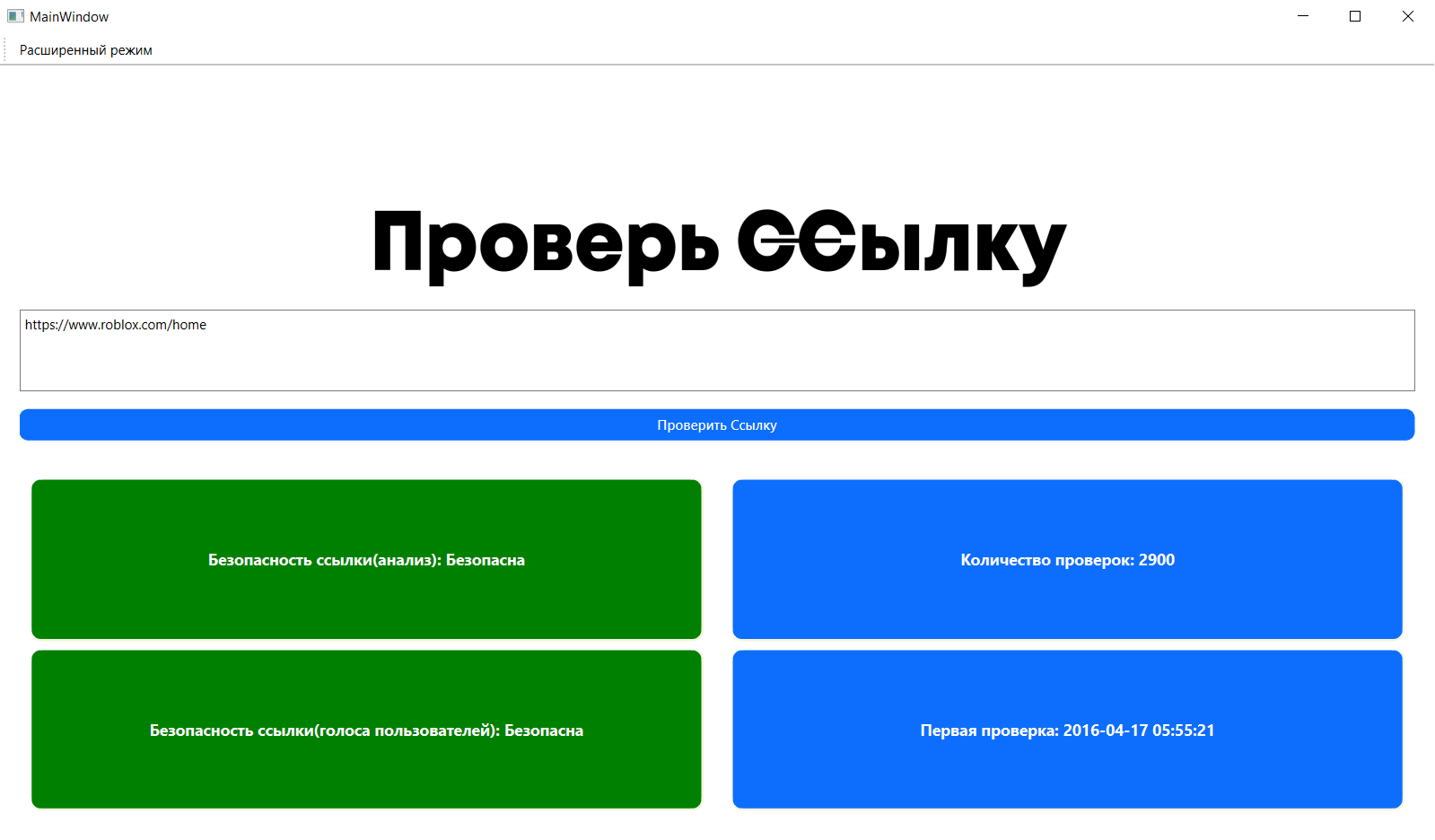


Рис.6. Итоговый вид прототипа программы

**Список Литературы**

1. Данько О.С Медведа Т.С. Исследование техник фишинга и методов защиты от него [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/issledovanie-tehnik-fishinga-i-metodov-zaschity-ot-nego/viewer>
2. Фишинговые письма: как их распознать и не стать их жертвой [Электронный ресурс]. URL: <https://www.kaspersky.ru/resource-center/preemptive-safety/phishing-prevention-tips>
3. Отчёт “Лаборатории Касперского” по спаму и фишингу за 2023 год [Электронный ресурс]. URL: <https://securelist.ru/spam-phishing-report-2023/109104/>
4. Анализ угроз информационной безопасности при использовании фишинговых сайтов [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/analiz-ugroz-informatsionnoy-bezopasnosti-pri-ispolzovanii-fishingovyh-saytov/viewer>
5. Pyside6 Tutorial [Электронный ресурс]. URL: <https://www.pythonguis.com/pyside6-tutorial/>

1. Отчёт “Лаборатории Касперского” по спаму и фишингу за 2023 год [Электронный ресурс]. URL: <https://securelist.ru/spam-phishing-report-2023/109104/> [↑](#footnote-ref-1)
2. Анализ угроз информационной безопасности при использовании фишинговых сайтов [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/analiz-ugroz-informatsionnoy-bezopasnosti-pri-ispolzovanii-fishingovyh-saytov/viewer> [↑](#footnote-ref-2)
3. Pyside6 Tutorial [Электронный ресурс]. URL: https://www.pythonguis.com/pyside6-tutorial/ [↑](#footnote-ref-3)