Министерство образования Российской Федерации Министерство науки и высшего образования Российской Федерации

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н.Э. БАУМАНА

Факультет: Информатика и системы управления
Кафедра: Информационная безопасность (ИУ8)

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Домашнее задание

Вариант 3

Преподаватель:
Ванин М.В.

Цирлов В.Л.

Студент:
Шапран А.В.

Группа:
ИУ8-74

Москва 2019

СОДЕРЖАНИЕ

УСЛОВИЕ ЗАДАЧИ

Реализовать для iOS или Android мобильное приложение, вырабатывающее TOTP-коды для входа на тестовый Gmail-аккаунт.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

TOTP (Time-based One-Time Password Algorithm, RFC 6238) – OATH-алгоритм создания одноразовых паролей для защищенной аутентификации, являющийся улучшением HOTP (HMAC-Based One-Time Password Algorithm). Является алгоритмом односторонней аутентификации – сервер удостоверяется в подлинности клиента. Главное отличие TOTP от HOTP – это генерация пароля на основе времени, то есть время является параметром. При этом обычно используется не точное указание времени, а текущий интервал с установленными заранее границами (например, 30 секунд).

Принцип работы

TOTP является вариантом HOTP алгоритма, в котором в качестве значения счетчика подставляется величина, зависящая от времени.

Обозначим:

- T – дискретное значение времени, используемое в качестве параметра;
- X – интервал времени, в течение которого действителен пароль. (По умолчанию 30 сек.);
- T_0 – начальное время, необходимое для синхронизации сторон. (По умолчанию — время от начала UNIX эры);
- K – разделяемый секрет;
- $CurrentTime$ – текущее время.

Тогда:

$$T = \frac{CurrentTime - T_0}{X}$$

$$HOTP(K,T) = Truncate\big(HMAC - SHA1(K,T)\big)$$

$$TOTP = HOTP(K,T),$$

где:

- $HMAC - SHA1(K,T)$ – генерация 20-ти байт на основе секретного ключа и времени с помощью хеш-функции SHA1;
- $Truncate$ – функция выбора определенным способом 4 байт.

Обозначим $String$ – результат $HMAC - SHA1(K,T)$; $OffsetBits$ – младшие 4 бита строки $String$; $Offset = StringToNumber(OffsetBits)$ и результатом $Truncate$ будет строка из четырёх символов – $String[Offset] \dots String[Offset + 3]$.

Надежность алгоритма

Концепция одноразовых паролей вкупе с современными криптографическими методами может использоваться для реализации надежных систем удаленной аутентификации. TOTP достаточно устойчив к криптографическим атакам, однако вероятности взлома есть, например, возможен такой вариант атаки «человек посередине»:

Прослушивая трафик клиента, злоумышленник может перехватить посланный логин и одноразовый пароль (или хеш от него). Затем ему достаточно блокировать компьютер «жертвы» и отправить аутентификационные данные от собственного имени. Если он успеет это сделать за промежуток времени X, то ему удастся получить доступ. Именно поэтому X стоит делать небольшим. Но если время действия пароля сделать слишком маленьким, то в случае небольшой рассинхронизации клиент не сможет получить доступ.

Также существует уязвимость связанная с синхронизацией таймеров сервера и клиента, так как существует риск рассинхронизации информации о времени на сервере и в программном и/или аппаратном обеспечении

пользователя. Поскольку TOTP использует в качестве параметра время, то при не совпадении значений все попытки пользователя на аутентификацию завершатся неудачей. В этом случае ложный допуск чужого также будет невозможен. Стоит отметить что вероятность такой ситуации крайне мала.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Приложение было разработано на языке C++ с помощью фреймворка Qt. Приложение является кроссплатформенным и может быть развернуто на операционных системах Windows, Linux, Android.

Установка приложения

Исходный код приложения доступен из репозитория на github по следующей ссылке:

- https://github.com/alexshapran-as/TotpAuthy

Для установки данного приложения на Android необходимо скачать apk файл расположенный по следующей ссылке:

- https://github.com/alexshapran-as/TotpAuthy/blob/master/apk/debug/android-build-debug.apk

Скачав данный файл, его необходимо открыть и следовать стандартным пунктам установки приложений на Android.

На рисунке 1 изображен процесс установки приложения под Android.

На рисунке 2 изображен внешний вид приложения, запущенного на Android.

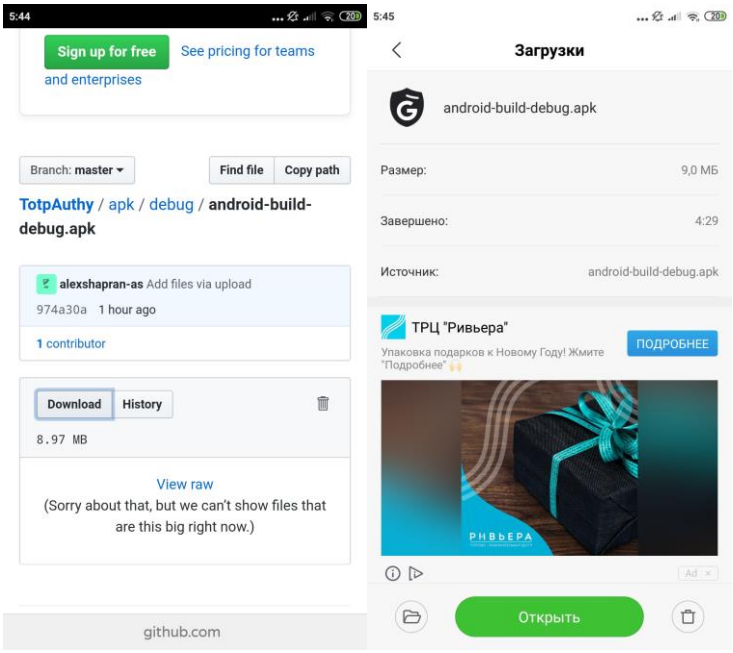

Рисунок 1 – процесс установки приложения под Android

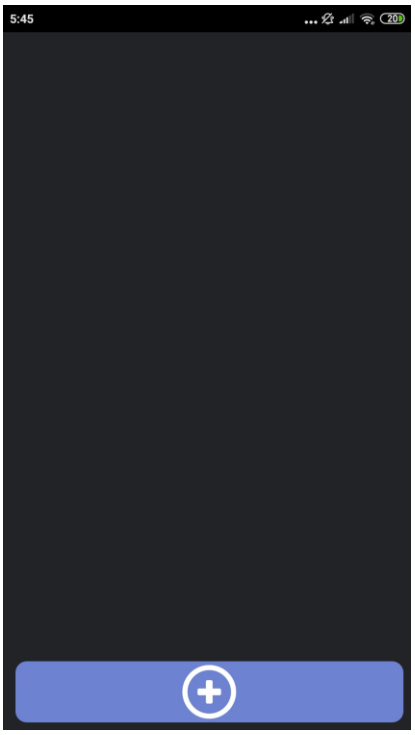

Рисунок 2 – внешний вид приложения

Для установки приложения под операционные системы Windows и Linux необходимо иметь установленный и настроенный фреймворк Qt. Настройка фреймворка Qt подразумевает наличие установленных компиляторов Qt MinGW или Qt MSVC.

В Qt достаточно произвести импорт проекта из репозитория на github. И запустить сборку проекта.

На рисунке 3 представлены этапы импорта проекта и результат его компиляции под Windows.

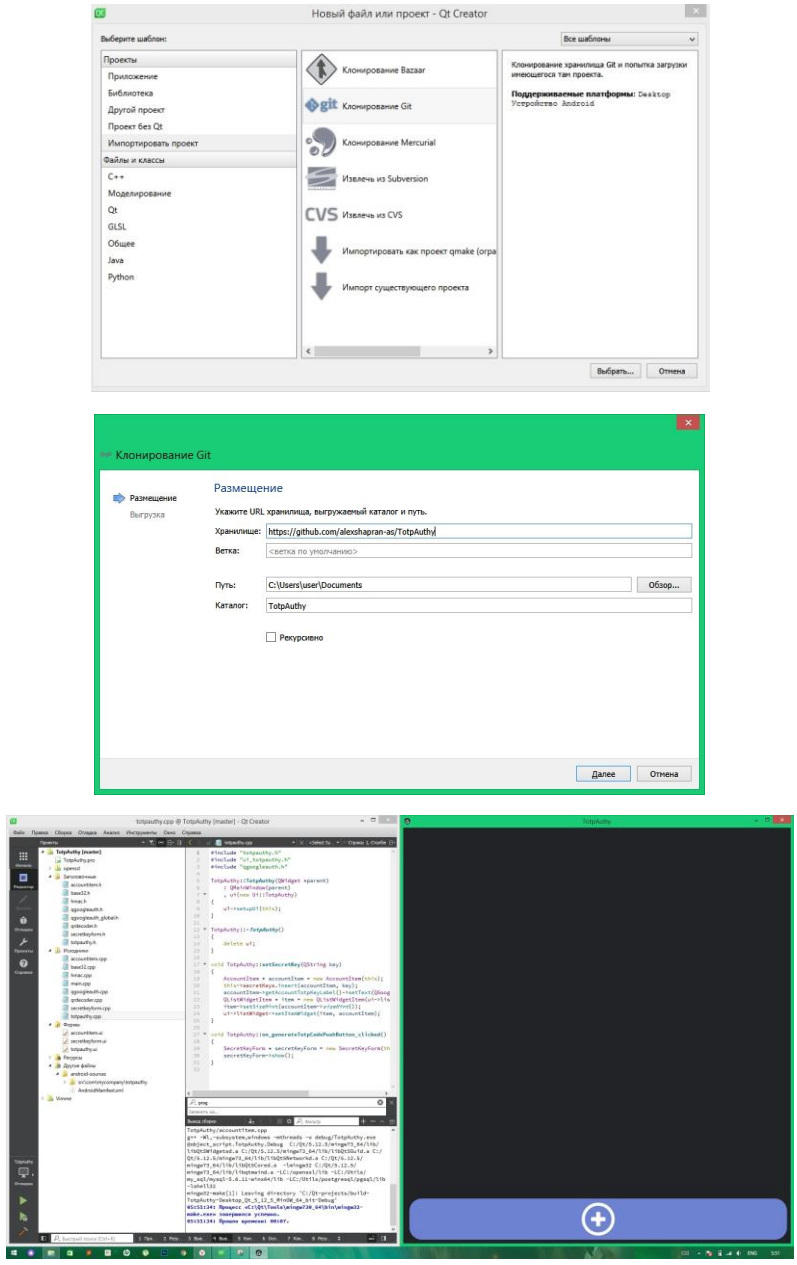

Рисунок 3 – этапы импорта проекта и результат его компиляции под Windows

Функционал приложения

Для создания нового TOTP-кода необходимо нажать на кнопку плюс и ввести секретный ключ, который создается при настройке двухфакторной

авторизации в Google аккаунте. После чего на экране появится TOTP-код, который будет обновляться каждые 30 секунд.

Более подробное описание работы приложения и этапы двухфакторной авторизации с помощью данного приложения можно посмотреть на видео так же размещенного в репозитории на github.