**Курсовая работа**

Отчёт 16 с., 7 рис., 6 ч., 5 источников.

РЕАЛИЗАЦИЯ ДВУХФАТОРНОЙ АУТЕНТИФИКАЦИИ.

Цель работы — реализовать технологию двухфакторной аутентификации на примере простейшего сайта.

В процессе работы были изучены механизм работы с двухфакторной аутентификацией, а также фреймворк и библиотеки, помогающие ее использовать.

В результате была изучен принцип работы технологии двухфакторной аутентификации, а также такие библиотеки Python, как pyotp и фреймворк Flask.

4

**СОДЕРЖАНИЕ**

[ОПРЕДЕЛЕНИЯ 5](#_Toc154599555)

[ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 6](#_Toc154599556)

[ВВЕДЕНИЕ 7](#_Toc154599557)

[ОСНОВНАЯ ЧАСТЬ 8](#_Toc154599558)

[1 Этапы выполнения работы 8](#_Toc154599559)  
[2 Примерный механизм работы нашей реализации 2FA 9](#_Toc154599559)

[3 Выбор инструментов для реализации 9](#_Toc154599563)

[4 Реализация механизма 2FA 10](#_Toc154599559)  
[5 Реализация среды (сайта) 11](#_Toc154599559)

[ЗАКЛЮЧЕНИЕ 13](#_Toc154599566)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16](#_Toc154599566)

# ОПРЕДЕЛЕНИЯ

**Общие понятия:**

* **Двухфакторная аутентификация –** метод защиты учетных записей, который требует двух различных способов подтверждения личности пользователя перед предоставлением доступа к учетной записи.
* **Аккаунт (учетная запись)** – это набор данных, хранящих информацию, необходимую для идентификации и аутентификации конкретного пользователя или сущности в информационной системе. Этот набор данных обычно включает в себя уникальное имя пользователя (логин) и сопоставленный с ним защищенный пароль или другие средства аутентификации, такие как ключи или сертификаты.
* **Аутентификация** – процесс проверки подлинности представленных учетных данных (например, логина и пароля), с целью установления того, имеет ли пользователь или сущность право на доступ к определенным ресурсам или функционалу информационной системы.

# ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **2FA (англ.)** | – | (Two-Factor Authentication) сокращение для двухфакторной аутентификации. |
| **БД** | – | База Данных. |
| **URI (англ.)** | – | (Uniform Resource Identifier) последовательность символов, идентифицирующая абстрактный или физический ресурс. |
| **TOTP (англ.)** | – | (Time-based One-Time Password) 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. |
|  |  |

# ВВЕДЕНИЕ

При появлении первых реализаций персональных услуг в интернете появились и первые учетные записи. И тут же встал вопрос о защите этих персональных данных от взлома и утечки данных. Изначально для доступа к аккаунту пользователя использовался один уровень идентификации, один ключ безопасности – пароль пользователя, связанный непосредственно с его логином, придуманным им самим при регистрации на ресурсе. Однако практика показала, что из-за совершенствующихся каждый день технологий и методов взлома и кражи информации, а также из-за цифровой неграмотности большинства пользователей, одного ключа безопасности не хватает.

Решением этой проблемы стало введение второго ключа безопасности – скрытого индивидуального ключа безопасности, неизвестного (грубо говоря) никому, кроме программы, которая его присваивает каждому пользователю. Также для усложнения подбора этого пароля придумали его преобразовывать и менять некоторым сложным образом с течением времени (так изначальный ключ становится чем-то вроде исходного ключа генерации или сида (от англ. seed)). Таким образом, взломать аккаунт пользователя становится крайне затруднительной задачей, ведь просчитать такой пароль практически невозможно, т. к. он сложным образом изменяется во времени каждые (в нашем случае) 30 секунд, а утечка его практически невозможна, так как сам пользователь не знает исходного ключа, а действующий временный ключ безопасности ему известен, но быстро сменяется, и обладание таким временным ключом быстро теряет актуальность. Такая система и стала называться системой двухфакторной аутентификации.

При этом то, каким образом пользователь получает временный пароль, генерируемый по второму ключу безопасности – выбирает разработчик. Обычно это либо отправка по некоторым контактным данным приватного письма пользователю с временным паролем, либо способ синхронизации по времени этого пароля на устройстве с помощью приложения. Понятно, что первый способ будет требовать доступ к сотовой связи, либо к интернету, либо чего-то еще, и потому является не таким доступным и простым, как второй способ, который не требует ничего, после привязывания сида к приложению – дальше временный пароль на сервере и в этом отдельном приложении всегда будут синхронизированы и будут изменяться одинаково, ведь оба будут зависеть лишь от времени, к которому любое устройство всегда будет иметь доступ, и пользователь в любой момент при любых обстоятельствах сможет посмотреть этот временный пароль у себя в этом приватном приложении. По этим причинам мы будем реализовывать 2FA именно вторым способом.

# ОСНОВНАЯ ЧАСТЬ

**1 Этапы выполнения работы:**

1. Изучение механизма работы технологии двухфакторной аутентификации.
2. Подбор платформы для реализации.
3. Реализация абстрагированного механизма 2FA.
4. Написание простого сайта для демонстрации работы технологии.
5. Добавление базы данных и работающего механизма регистрации, завязанном на принципах 2FA.

**2 Примерный механизм работы нашей реализации 2FA**

В введении были описаны основные принципы работы нашей программы, так что подведем итог того, что бы мы хотели получить в результате: механизм регистрации, который запрашивал бы у пользователя логин, пароль и временный ключ 2FA, который он сможет узнать в одном из многочисленных приложений для 2FA (при тестировании использовалось бесплатное приложение “Microsoft Authenticator”).

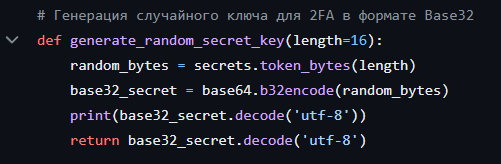
При регистрации пользователю будет выдан QR код, который будет содержать информацию о втором ключе идентификации, и после его сканирования в некотором из приложений, о которых было сказано ранее, никто другой не сможет им воспользоваться, ведь ключ уникален, так что эта система будет безопасна, а у пользователя всегда будет конфиденциальный независимый доступ к этой информации.

## 3 Выбор инструментов для реализации

Наименее трудоемко будет реализовывать 2FA на языке Python с помощью библиотеки с открытым кодом “pyotp”. Для генерации по сиду QR кода будем использовать библиотеку с открытым кодом “qrcode”. Страницы для сайта будем писать на чистом HTML, а бэкэнд реализуем с помощью фреймворка Flask. Базу данных привяжем с помощью библиотеки “Flask-SQLAlchemy”. Наш сайт будет хоститься локально на компьютере. Все используемые инструменты находятся в открытом доступе, и любой желающий сможет запустить сайт на своей машине. Как это сделать подробно описано в файле “README.md” в репозитории проекта.

## 4 Реализация механизма 2FA

Будем генерировать случайный ключ 2FA в формате Base64 с помощью встроенной библиотеки “base64”, и затем преобразовывать его в Base32 с помощью функции .b32encode() из упомянутой выше библиотеки. Таким образом получаем случайных ключ в формате b32, который присвоим пользователю, и позже будем хранить в БД в виде сгенерированного по нему URI.

  
рис. 1

Теперь, когда у нас есть ключ генерации можем воспользоваться библиотекой pyotp для генерации уникального URI (по которому с помощью библиотеки qrcode будем создавать QR), затем, с помощью метода .TOTP() будем получать временный код аутентификации.

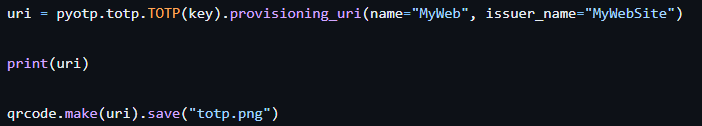
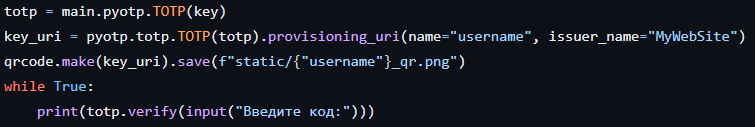


рис. 2

Теперь пользователь сможет отсканировать QR с помощью приложения 2FA, и далее будет всегда знать временный код аутентификации. Реализуем теперь проверку ввода кода 2FA, зная ключ генерации. Делается это следующим образом с помощью метода .verify():

  
рис. 3

Реализация 2FA готова. Осталось только реализовать среду, в которой она будет использоваться.

## 5 Реализация среды (сайта)

Реализуем простой сайт с помощью фреймворка Flask на Python. Сайт будет следовать следующей конструкции: при запуске сайта будет отображена страница, где пользователю будет предложено выбрать что он хочет сделать – зарегистрироваться на сайте, либо войти в уже созданный аккаунт (см. рис. 4).

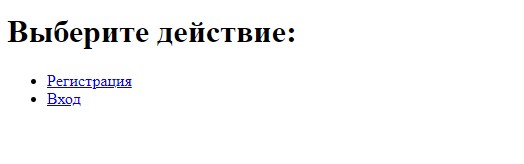


рис. 4

Если выбрана регистрация, то мы перекидываем его сначала на первый этап регистрации (ввод логина и пароля):

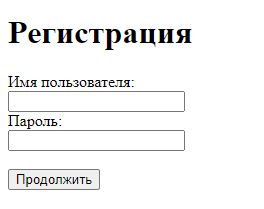


рис. 5

После ввода пользователя направит на второй этап регистрации (установка и проверка кода 2FA):



рис. 6

На этом этапе мы интегрируем уже имеющуюся реализацию установки и проверки 2FA, и ассоциируем ее с интерфейсом. Если код будет введен успешно – в базу данных добавятся данные о пользователе в формате:

Id / username / password / 2FA key

Если же пользователь захочет войти в свой аккаунт, то ему нужно будет ввести все ранее созданные ключи идентификации:

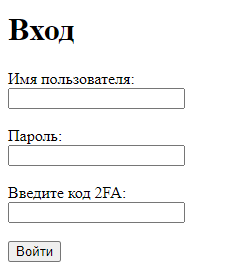


рис. 7

Если все введено верно, пользователь сможет войти в свой аккаунт.

**ЗАКЛЮЧЕНИЕ**

В ходе данной работы мы реализовали двухфакторную аутентификацию веб-приложения с использованием Flask, SQLAlchemy и библиотек для работы с QR-кодами и одноразовыми паролями.

Результаты:

1. Разработка функционала регистрации и входа в систему:
   1. Созданы страницы для регистрации и входа в систему.
   2. Реализована проверка учетных данных пользователя при входе.
   3. При регистрации пользователей данные сохраняются в базе данных.
2. Внедрение двухфакторной аутентификации:
   1. В процессе регистрации генерируется секретный ключ для TOTP.
   2. Создан QR-код, содержащий секретный ключ, для сканирования приложением аутентификации.
   3. Реализовано ввод и проверка одноразового пароля (TOTP) при входе пользователя в систему.
3. Улучшение безопасности системы:
   1. Двухфакторная аутентификация повышает безопасность системы за счет использования дополнительного фактора, отличного от стандартного пароля.
   2. Реализация TOTP обеспечивает генерацию временных и одноразовых паролей, которые труднее подделать или перехватить.

Выводы:

1. Безопасность:
   1. Внедрение двухфакторной аутентификации значительно повышает уровень безопасности системы, обеспечивая дополнительный уровень защиты от несанкционированного доступа.
   2. Использование TOTP и QR-кодов обеспечивает надежный и удобный механизм аутентификации пользователей.
2. Пользовательский опыт:
   1. Реализованный функционал не только усиливает безопасность, но и сохраняет удобство использования для конечных пользователей, минимизируя сложности при регистрации и входе.
3. Дальнейшие улучшения:
   1. Для дальнейшего совершенствования системы можно рассмотреть добавление дополнительных методов аутентификации, таких как SMS-коды или аутентификация по биометрическим данным.
   2. Также возможно расширение функционала администрирования пользователей и управления их правами доступа.

Итоги:

Реализация двухфакторной аутентификации является важным шагом в обеспечении безопасности веб-приложений. Полученные результаты подтверждают эффективность и удобство использования данного метода аутентификации. Дальнейшее развитие и улучшение системы поможет обеспечить еще более высокий уровень безопасности и комфортного пользовательского опыта

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аутентификация: от паролей к открытым ключам – Ричард Э. Смит [Книга]. (дата издания: 2002)
2. ДВУХФАКТОРНАЯ АУТЕНТИФИКАЦИЯ КАК МЕТОД ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ [Электронный ресурс]. – Режим доступа: URL: [https://cyberleninka.ru/article/n/dvuhfaktornaya-autentifikatsiya-kak-metod-obespecheniya-informatsionnoy-bezopasnosti](https://cyberleninka.ru/article/n/dvuhfaktornaya-autentifikatsiya-kak-metod-obespecheniya-informatsionnoy-bezopasnosti%20) (дата обращения: 27.12.2023)
3. Двухфакторная аутентификация [Электронный ресурс]. – Режим доступа: URL: [https://cyberleninka.ru/article/n/dvuhfaktornaya-autentifikatsiya](https://cyberleninka.ru/article/n/dvuhfaktornaya-autentifikatsiya%20) (дата обращения: 27.12.2023)
4. Two-Factor Authentication: A Comprehensive Study – David A. Horsman, Zachary K. Baker, Charles V. Wright [Статья]. (дата издания: 2015)
5. A Survey on Two-Factor Authentication Solutions – Chandan Kumar, Azad Kumar, and Shampa Chakraverty (2016) [Статья]. (дата издания: 2016)