ПРОГРАМНА РЕАЛІЗАЦІЯ БЕЗПЕКИ ЗАСТОСУВАНЬ ДЛЯ ОБМІНУ ІНФОРМАЦІЄЮ

Міхневич І.Д.
Науковий керівник – к.т.н., доцент Голян В.В.
Харьківський національний університет радіоелектроніки
(61166, м. Харків, пр. Науки, 14, каф.Програмної інженерії,
тел. (057) 702-14-46)

In this work, the main object of research is the development of application software for the transmission and encryption of text messages. The methods of making language based on object-oriented direction C # development environment and Microsoft Visual Studio, using object-oriented programming approach. The result of the work done is functioning application that can send and receiving messages.

У двадцять першому столітті все більше уваги приділяється інформаційним технологіям. Віртуальне спілкування увійшло у звичку. Але разом з цим підвищився рівень кіберзлочинності. Мало хто знає що будь-яку розмову в мережі можна прослухати або прочитати при великій необхідності. В зв'язку з цим багато мессенджерів та соціальних мереж розроблюють нові засоби захисту персональних сторінок та передачі даних. Метою даної роботи є реалізація та тестування мікросервісу, за допомогою якого можна підвищити рівень безпеки додатку, що його буде використовано при передачі текстових повідомлень. Монолітні додатки можуть бути успішними, але все більше людей розчаровуються в них, особливо в світлі того, що все більше додатків розгортаються в хмарі. Будь-які зміни, навіть самі невеликі, вимагають перебудування і розгортання всього моноліту. З плином часу, стає важче зберігати гарну модульну структуру, зміни логіки одного модуля мають тенденцію впливати на код інших модулів. Масштабувати доводиться додаток цілком, навіть якщо це потрібно тільки для одного модуля цього додатка.[1-3]

Мікросервісна архітектура робить великий акцент на моніторинг додатків в режимі реального часу, перевірці як технічних елементів (наприклад, як багато запитів в секунду отримує база даних), так і бізнес-метрик (наприклад, як багато замовлень в хвилину отримує додаток). Семантичний моніторинг може надати системі раннє попередження проблемних ситуацій, дозволяючи команді розробці підключитися до дослідження проблеми на самих ранніх стадіях. Команди мікросервісов, як правило, створюють витончені системи моніторингу і логування для кожного індивідуального сервісу. Прикладом може служити консоль, що показує статус (он-лайн / офлайн) сервісу і різні технічні та бізнес-метрики: поточна пропускна здатність, час обробки запиту і т.п.[4-5]

Наша основна мета полягала в тому, щоб пояснити основні ідеї і принципи мікросервісної архіктури. Мікросервісний стиль - важлива ідея, що розглядається для enterprise додатків. Не так давно ми розробили кілька

систем, використовуючих цей стиль і знаємо кілька інших команд, які використовують цей підхід.[6-7]

Необхідно реалізувати мікросервіс який буде виконувати роль безпеки, та підвищить рівень захищеності додатку, що його буде використовувати. У даному випадку мікросервіс буде окремим додатком який буде реалізований на об`єктно орієнтовній мові C# та з використанням середовища розробки Visual Studio. Основна мета розробки мікросервісів взагалі – це розбиття програмного додатку на окремі модулі, що в свою чергу зменшує зв'язність компонентів системи і спрощують масштабування системи і заміну модуля, не торкаючись при цьому інших компонентів додатку.

Додаток буде реалізований за допомогою RabbitMQ framework який знаходиться у свободному доступі, та використовується для обміну повідомленнями, також до нього буде доданий алгоритм шифрування повідомлень для підвищення рівня безпеки. Алгоритм був обраний RC4 у зв’язку з легкістю його реалізації.

Перелік посилань

1. Сєм Ньюмен – создание микросервисов /Питер, 2014 – 304с.
2. Martinfawler.com blog about microsevices [Електронний ресурс] / режим доступу: http://martinfowler.com/articles/microservices.html 06.06.2016р. - Загол. з екрану.
3. Visual Studio Magazine Deploying microservices [Електронний реурс] / режим доступу: https://visualstudiomagazine.com/articles/2015/09/30/microservices-csharp.aspx 06.06.2016р. - Загол. з екрану.
4. Мартин , Р. Чистый код. Создание, анализ и рефакторинг / Питер, 2013 – 464 с.
5. Бэк, К. Экстремальное программирование: разработка через тестирование / Питер 2014. – 224с.
6. Хейлсберг А., Язык программирования C#. Классика Computers Science. 4-е изд. / Хейлсберг А., Торгерсен М., Вилтамут С., Голд – Питер, 2015 – 784с.
7. Сэм Канер Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / Сэм Канер, Джек Фолк – Парад, 2013 – 544с.