DOI 10.37882/2223-2966.2025.02.35

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА ПЛАГИНОВ ДЛЯ УСТРОЙСТВ ИНТЕРНЕТА ВЕЩЕЙ

SECURITY, SCALABILITY AND EFFICIENT RESOURCE ALLOCATION ANALYSIS ON INTERNET OF THINGS MANAGEMENT PLATFORMS

D. Frolov
A. Kharitonov

Summary. This article analyzes opensource platforms for smart homes considering the requirements for security, scalability, and resource management. The proposed solution introduces the concept of a microservices infrastructure, where each plugin for an Internet of Things device is represented by a separate microservice. This approach enables a high level of security through microservices isolation and provides flexibility in scalability.

Keywords: Internet of Things (IoT), openHAB, Home Assistant, security, code analysis.

Фролов Данил Александрович
Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»
frolovdanil@list.ru
Харитонов Антон Юрьевич
Кандидат технических наук, инженер, доцент, Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»
akharitonov@itmo.ru

Аннотация. В данной статье проводится анализ платформ с открытым кодом для умного дома с учётом требований к безопасности, масштабируемости и управления ресурсами. В качестве решения выдвигается концепция микросервисной инфраструктуры, где каждый плагин для устройства Интернета вещей представлен отдельным микросервисом. Этот подход позволяет обеспечить высокий уровень безопасности благодаря изоляции микросервисов и предоставляет возможность для гибкой масштабируемости.

Ключевые слова: микросервисная архитектура, безопасность, Интернет вещей, платформа управления Интернет вещей.

С постоянным развитием технологий Интернета вещей системы автоматизации дома, такие как openHAB [9] и Home Assistant [10], становятся все более популярными в использовании. Эти платформы предоставляют пользователям удобное управление различными устройствами Интернета вещей: от освещения и отопления до систем развлечений. Однако, несмотря на большую популярность данных платформ, существует большой пласт проблем, связанных с:

1. безопасностью. С ростом масштабов рынка Интернет вещей компании, выпускающие умные устройства, стремятся выпускать продукты как можно быстрее, а на разработку защиты и безопасности устройств уделяют меньше внимания [1, с. 102]. В связи с этим информационная безопасность устройств Интернета вещей является одной из главных проблем, так как устройства системы управления являются достаточно уязвимыми перед различными векторами вредоносных атак. Это может привести к потере контроля, доступности работы, а также нарушению конфиденциальности данных компаний и информации о клиентах [1, с. 95];
2. масштабируемостью. С ростом числа подключенных устройств, включая сенсоры, умные устройства и промышленное оборудование, возникает необходимость эффективного управления и координации этой инфраструктурой. Вопросы, связанные со сбором, передачей, анализом данных, обновлением программного обеспечения, обеспечением безопасности и общим управлением становятся более сложными при больших масштабах;
3. использованием ресурсов. Существует необходимость при любой начальной инфраструктуре настроить оптимизацию вычислительных мощностей, памяти, сетевой пропускной способности и других ресурсов для обеспечения бесперебойной и эффективной работы масштабированных систем управления устройствами Интернета вещей. С ростом числа подключенных устройств и количества обрабатываемых данных возникают проблемы с балансировкой нагрузки. Эти проблемы возникают ввиду того, что большинство существующих систем управления устройствами Интернета вещей не обладают достаточной гибкостью.

Для решения вышеперечисленных проблем авторами данной статьи была разработана микросервисная инфраструктура, где плагин для каждого устройства Интернета вещей будет представлен отдельным микросервисом. Благодаря изоляции микросервисов предполагается обеспечение высокого уровня безопасности. Также эта система способна обеспечивать балансировку нагрузки, что приведёт к гибкой и эффективной работе системы. Кроме того, использование такой инфраструктуры предполагает процессы непрерывного развертывания, которые позволяют автоматизировать развертывание и своевременное обновление плагинов.

Для оценки состояния безопасности и масштабируемости современных платформ устройств Интернета вещей авторами данной статьи был выполнен анализ ряда научных статей [1, с. 99; 2, с. 271; 3; 4, с. 1247] и открытого исходного кода платформ openHAB и Home Assistant. В статье [4, с. 1249] был проведён статический анализ кода плагинов, доступных на официальном GitHub репозитории openHAB [3]. В результате анализа открытого исходного кода, авторами была составлена Таблица 1, которая предоставляет обзор результатов после анализа.

В результате анализа, авторами было найдено 3019 багов, что является крайне большим значением для платформы. Было выяснено, что добавление большого числа плагинов к работающей системе процесс очень простой, но учитывая низкое качество кода некоторых

Таблица 1.

Результат анализ кода плагинов устройств openHAB

Количество плагинов	153
Количество найденных багов	3019
Количество плагинов с прямым взаимодействием с шиной	9
Количество плагинов, не проверяющих пользовательский ввод	103

из плагинов, это может привести к сбоям внутренних процессов openHAB [9].

Помимо проблемы низкого качества кода для модулей Интернета вещей, существует проблема, связанная с избыточными правами плагинов. Например, плагин openHAB exec binding позволяет выполнение системных сценариев или команд напрямую из модуля без необходимых разрешений. Эта уязвимость позволяет как локальному, так и удаленному пользователю выполнять команды без какого-либо контроля доступа, представляя потенциальную угрозу безопасности.

Большинство из 153 обнаруженных в результате анализа плагинов платформы openHAB позволяют пользователю управлять ими, предоставляя входные аргументы без какой-либо валидации. Некоторые плагины также взаимодействуют непосредственно с шиной сообщений openHAB, что может привести к утечке конфиденциальной информации, такой как PIN-коды и пароли.

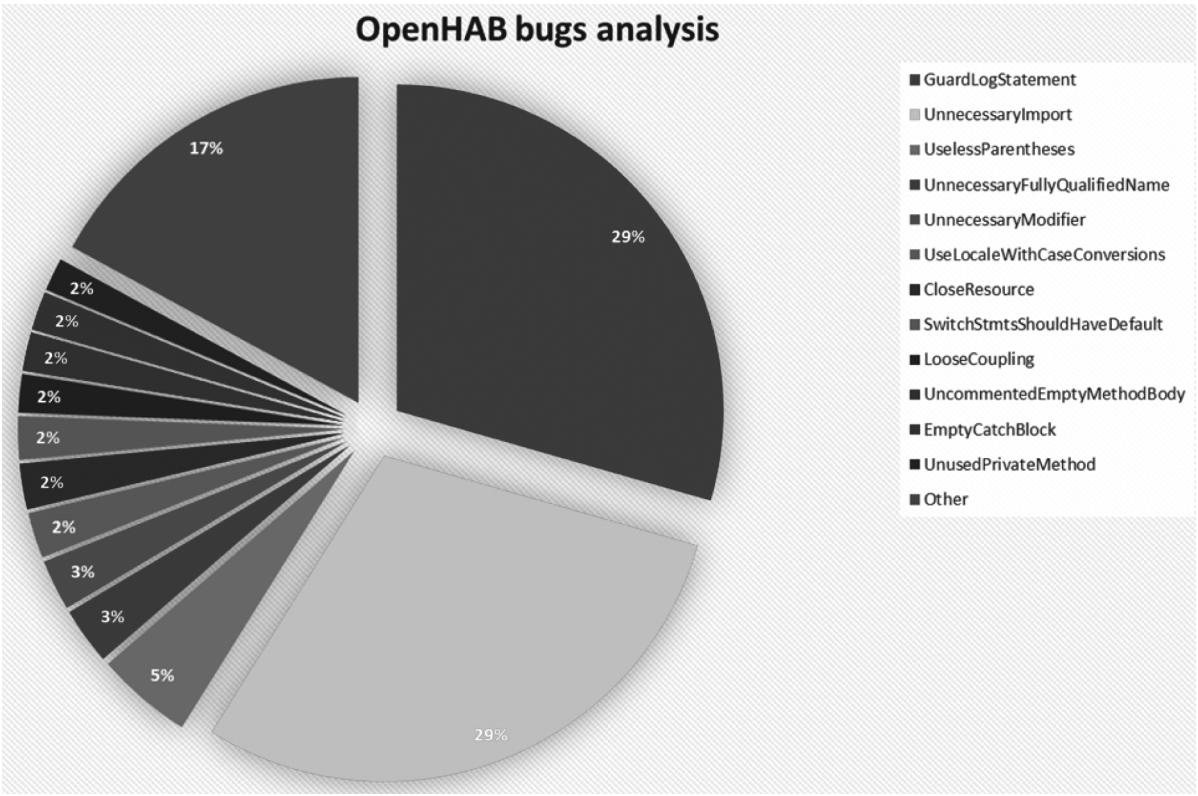

Рис. 1. Результаты анализа кода openHAB [9] плагинов с помощью PMD Code Analyzer [11]

В дополнение к результатам статьи [4, с. 1246], был проанализирован код плагинов с помощью инструмента PMD Code Analyzer [11], построена диаграмма (рис. 1), благодаря которой были сделаны выводы о масштабируемости и эффективном использовании ресурсов системы.

В частности, обнаружено множество ошибок, связанных с неправильным использованием логирования, такая как отсутствие контроля уровня лога (GuardLogStatement), может привести к избыточному выводу информации или даже игнорированию важных сообщений, что затрудняет отслеживание и анализ состояния системы при её масштабировании.

Ошибки, найденные в результате анализа кода платформы openHAB [9] не только увеличивают сложность поддержки и разработки openHAB, но и создают потенциальные уязвимости и риски неправильной работы системы при её масштабировании, особенно в случае использования платформы на маломощных устройствах.

Анализируя ситуацию с не менее популярной платформой для управления устройствами Интернета вещей — Home Assistant [10], были получены следующие результаты (рис. 2):

Несмотря на то, что количество багов и уязвимостей у Home Assistant меньше в сравнении с openHAB, опасности, которые несут эти уязвимости, существенны.

Так, например были найдены проблемы, связанные с открытым хранением паролей или использованием слабого шифрования для них. Образец использования представлен в распечатке 1.

```
ENVIRONMENT_URLS = {
    GeocachingApiEnvironment.Staging: GeocachingOAuthApiUrls(
    authorize_url="https://staging.geocaching.com/oauth/authorize.aspx",
    token_url="https://oauth-staging.geocaching.com/token",
),
```

Распечатка. 1. Образец кода небезопасного хранения пароля при анализе инфраструктуры Home Assistant [10]

Ошибка, связанная с привязкой ко всем интерфейсам (включая внешние и внутренние) в приложении, может создать потенциальную уязвимость безопасности, так как плагин может быть доступен извне сети.

Учитывая вышеописанные проблемы, авторами была разработана собственная микросервисная инфраструктура для плагинов Интернета вещей.

Микросервисная архитектура Интернета вещей

Чтобы обеспечить надёжность, безопасность и эффективную разработку такой архитектуры, авторами был разработан комплексный подход, включающий в себя непрерывные процессы интеграции и доставки, мониторинг кода, а также системы тестирования. Реализованная авторами структура платформы для управления устройствами Интернета вещей, а также структура для разработки плагинов Интернета вещей представлена на рис. 3.

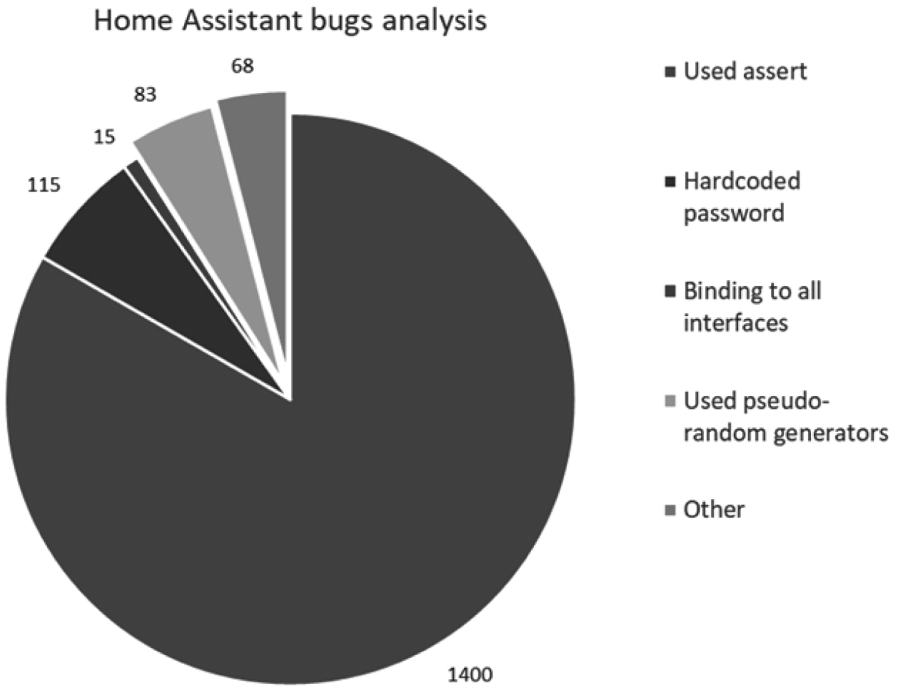

Рис. 2. Результаты анализа кода Home Assistant плагинов с помощью инструмента bandit [12]

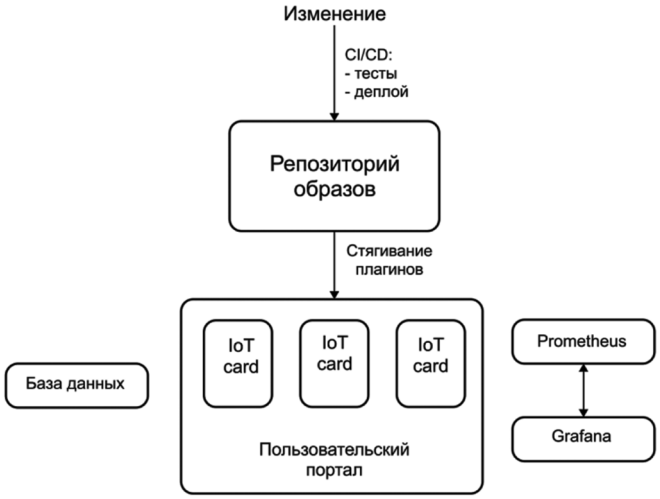

Рис. 3. Структура платформы для управления устройствами Интернета вещей

Подход к разработке плагинов устройств

Использование подхода непрерывной интеграция/непрерывная доставки, а также системы тестирования — это важный и эффективный подход к разработке программного обеспечения, используемый авторами для обеспечивания автоматизации процессов разработки, тестирования и развёртывания [5]. Автоматизированные процессы позволяют обнаруживать и устранять ошибки на ранних этапах разработки кода плагинов для интеграции устройств Интернета вещей, обеспечивая быструю обратную связь между разработчиками и повышать качество кода. Благодаря концепции непрерывной интеграции/непрерывной доставки ускоряются поставки новых функций, обеспечивается лёгкость отката изменений, стандартизация окружений и непрерывный мониторинг. Особое внимание уделяется тестам приложений, так как большинство уязвимостей в коде плагинов для интеграции систем openHAB [9] и Home Assistant [10] существуют ввиду отсутствия систем тестирования и сканирования кода на уязвимости. В распечатке 2 представлен разработанный авторами данной статьи сценарий непрерывной интеграции/непрерывной доставки.

```
stages:
  - test
  - build

unit_tests:
  stage: test
  image: python:3.12
  tags:
    - runner
  script:
    - pip install -r requirements.txt
    - python3 unit.py

secure_tests:
  stage: test
  image: python:3.12
  tags:
    - runner
  script:
    - pip install bandit
    - bandit -r app.py
  allow_failure: true

build:
  stage: build
  image: python:3.12
  tags:
    - runner
  script:
    - docker login -u $CI_REGISTRY_USER -p $CI_REGISTRY_PASSWORD
    - docker build --build-arg BULB_USER_PASSWD=$BULB_USER_PASSWD -t $IMAGE
    - docker push $IMAGE
```

Распечатка 2. Разработанный сценарий непрерывной интеграции/непрерывной доставки для одного из плагинов устройств

Подход для разработки платформы для управления устройствами Интернета вещей

Одним из самых важных аспектов при разработке платформы является возможность её использования как в облачной среде, так и на собственных вычислительных мощностях. Для этих целей авторами была разработана платформа для заказа облачной инфраструктуры для управления устройствами Интернета вещей, основанная на подходе «Инфраструктура как код». «Инфраструктура как код» применяет декларативный метод для настройки, управления серверами и автоматизации задач. Это значительно упрощает процесс администрирования инфраструктуры и повышает эффективность развертывания и обслуживания системы [5].

Написанная авторами платформа производит заказ облачной микро сервисной платформы, а также инфраструктуры к ней.

На распечатке 3 указан один из разработанных авторами сценариев для развёртывания базы данных.

Результаты

В рамках исследования был разработан и предложен подход, основанный на непрерывной интеграции и доставке (CI/CD), в сочетании с системой тестирования и анализа кода.

Кроме того, было проведено сравнение эффективности для микро сервисной инфраструктуры (разработанной авторами) и системы openHAB: Для плагина «микро сервис управления освещением» было реализовано нагрузочное правило. Это правило включает и выключает умную лампочку с некоторыми интервалами. Результаты снятых метрик процессорного времени представлены на рис. 4 и 5.

Согласно вышеприведенному анализу, можно сделать вывод, что разработанная микро сервисная архи-

```
---
- hosts: infra
  become: true

  vars_files:
    - db_creds.yml

  pre_tasks:
    - name: Install required system packages
      pip:
        name: psycopg2-binary
        state: latest

    - name: "Check if PostgreSQL is installed"
      command: "dpkg -l postgresql"
      register: postgres_installed
      changed_when: false
      ignore_errors: true

    - name: "Install PostgreSQL if not installed"
      apt:
        name: "{{ item }}"
        state: present
      when: postgres_installed.rc != 0
      loop:
        - postgresql
        - postgresql-server

  tasks:
    - name: "Check if PostgreSQL is initialized"
      ansible.builtin.stat:
        path: "/etc/postgresql/{{ postgres_version }}/main/pg_hba.conf"
      register: postgres_data

    - name: "Initialize PostgreSQL if not initialized"
      shell: "pg_ctlcluster {{ postgres_version }} main start"
      when: not postgres_data.stat.exists
```

Распечатка 3. Разработанный сценарий развёртывания части инфраструктуры (а именно базы данных) портала

тектура обладает меньшей ресурсозатратностью, а как следствие лучшей масштабируемости в сравнении с рассматриваемыми системами управления Интернет вещей с открытым исходным кодом.

Обсуждение

Основные результаты исследования подтверждают, что недостатки, связанные с безопасностью и эффективностью в работе платформы openHAB связаны с качеством кода плагинов для устройств Интернета вещей. Предложенный подход с использованием конвейера непрерывной интеграции/непрерывной доставки и микро сервисной архитектуры представляет собой значимое решение для исправления вышеперечисленных проблем. Однако, необходимо учитывать ограничения связанных с необходимостью разработки большого количества плагинов для микро сервисной архитектуры. Переход к такой архитектуре требует значительных усилий и времени для разработки, тестирования и поддержки каждого плагина устройства IoT. Дальнейшие исследования могут сосредоточиться на разработке более эффективных алгоритмов обработки данных для устройств Интернета вещей, а также на автоматизации процесса тестирования и анализа кода плагинов.

Заключение

Исследование безопасности и эффективности openHAB в контексте подключения устройств Интернета вещей выявило значительные проблемы, связанные с качеством написанных плагинов и использованием ресурсов. Недостаточная оптимизация кода, отсутствие обновлений и уязвимости в библиотеках создают потен-

Рис. 4. График изменения значения CPU openHAB во время работы нагрузочного правила

Рис. 5. График изменения значения CPU для микросервисной архитектуры во время работы нагрузочного правила

циальные угрозы для системы. Однако, предложенный подход, основанный на концепции непрерывной интеграции/непрерывной доставки, совместно с системой тестирования и анализа кода, демонстрирует эффективное решение в вопросе безопасности. Кроме того, неэффективно написанный код плагинов, неоптимизированные запросы к базе данных и отсутствие кэширования могут привести к избыточной нагрузке на систему. В данном контексте архитектура микросервисов представляет собой перспективное решение. Гибкое распределение ресурсов между плагинами позволяет оптимизировать использование памяти и процессорного времени, что способствует более эффективной работе системы.

Учитывая вышесказанное, можно сделать вывод о том, что микросервисная система для устройств Интернета вещей, предоставляет более безопасное, масштабируемое и эффективное решение для управления системами Интернета вещей по сравнению с традиционными платформами.

ЛИТЕРАТУРА

1. Н.А. Наралиев, Д.И. Самаль Обзор и анализ стандартов и протоколов в области Интернет вещей. Современные методы тестирования и проблемы информационной безопасности IoT // International Journal of Open Information Technologies ISSN. 2019. №8. С. 94–104.
2. A. Cyril Jose, R. Malekian Smart Home Automation Security // Smart Computing Review. 2015. №5. С. 269–285.
3. GitHub репозиторий: [openHAB] URL: https://github.com/openhab (дата обращения: 10.02.2024).
4. MIPRO: International Convention on Information and Communication Technology, Electronics and Microelectronics: Security Analysis of Open Home Automation Bus System: материалы междунар. науч. конф., Хорватия, Опатия, май 22–26, 2017г. / University of Zagreb, Milan Ramljak, стр. 1245–1250
5. 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence): Comparison of Different CI/CD Tools Integrated with Cloud Platform: материалы междунар. науч. конф., Ноида, Индия, январь 10–11, 2019 г. / Charanjot Singh, Nikita Seth Gaba, Manjot Kaur, Bhavleen Kaur
6. V. Garousi, B. Küçük Smells in software test code: A survey of ¨ knowledge in industry and academia // Journal of Systems and Software. 2018. №138. С. 52–81.
7. 2016 IEEE Symposium on Security and Privacy (SP): Security Analysis of Emerging Smart Home Applications: материалы междунар. науч. конф., США, Сан-Хосе, май 22–26 2016 г. / University of Michigan, E. Fernandes, J. Jung and A. Prakash, стр. 636–654
8. Proceedings of the International Conference on Human Factors in Computing Systems: Home Automation in the Wild: Challenges and Opportunities: материалы междунар. науч. конф., Ванкувер, Канада, май 7–12, 2011 г. / University of Washington, A.J. Bernheim Brush, Bongshin Lee
9. Интернет-ресурс: [openHAB] URL: https://www.openhab.org/
10. Интернет-ресурс: [Home Assistant] URL: https://www.home-assistant.io/
11. GitHub репозиторий: [PMD Code Analyzer] URL: https://pmd.github.io/
12. GitHub репозиторий: [bandit] URL: https://github.com/PyCQA/bandit
13. GitHub репозиторий: [app] URL: https://github.com/Koban1dze/app

Журнал «Современная наука: актуальные проблемы теории и практики»