УДК 004.056.5:681.518

Конюк Дмитрий Александрович студент
2, курс Томский государственный университет систем управления и радиоэлектроники (ТУСУР)
Россия, г. Томск

ПОДХОД К ЗАЩИТЕ API УСТРОЙСТВ ИНТЕРНЕТА ВЕЩЕЙ НА ОСНОВЕ СОВМЕСТНОГО АНАЛИЗА СТРУКТУРЫ ЗАПРОСОВ И ПОВЕДЕНЧЕСКИХ ХАРАКТЕРИСТИК УСТРОЙСТВ

Аннотация: В статье рассматривается проблема обеспечения безопасности прикладных интерфейсов устройств Интернета вещей. Показано, что традиционные средства защиты API, основанные преимущественно на сигнатурном анализе запросов и статических правилах фильтрации, обладают ограниченной эффективностью при обнаружении атак, использующих легитимные механизмы взаимодействия с приложением. Предложен подход к защите API устройств Интернета вещей, основанный на совместном использовании структурного анализа запросов и анализа поведенческих характеристик устройств. Разработана архитектура системы защиты, включающая механизмы структурного анализа API-запросов, контроля доступа, формирования профиля нормального поведения устройств и выявления аномалий. Для оценки эффективности предложенного подхода проведены экспериментальные исследования на стенде, моделирующем работу распределённой IoT-инфраструктуры. Результаты испытаний показали среднюю эффективность выявления атак на уровне 93 %, эффективность обнаружения атак типа API Flood на уровне 98,6 %, а также низкую долю ложных срабатываний, не превышающую 0,7 %. Полученные результаты подтверждают возможность применения предложенного подхода для защиты API устройств Интернета вещей, робототехнических комплексов и других распределённых систем,

использующих API в качестве основного механизма взаимодействия компонентов.

Ключевые слова: Интернет вещей, API, безопасность API, информационная безопасность, Web Application Firewall, поведенческий анализ, обнаружение аномалий, IoT Security, API Security, защита веб-приложений.

AN APPROACH TO PROTECTING INTERNET OF THINGS APIs BASED ON COMBINED ANALYSIS OF REQUEST STRUCTURE AND DEVICE BEHAVIORAL CHARACTERISTICS

Annotation: The paper addresses the problem of securing application programming interfaces (APIs) used in Internet of Things (IoT) environments. Traditional API protection solutions based primarily on signature analysis and static filtering rules are shown to have limited effectiveness against attacks that exploit legitimate API interaction mechanisms. An approach to IoT API protection based on the combined use of request structure analysis and device behavioral analysis is proposed. The developed architecture includes modules for API request validation, access control, behavioral profiling, anomaly detection and risk assessment. Experimental evaluation performed on a distributed IoT testbed demonstrated an average attack detection rate of 93%, an API Flood detection rate of 98.6%, and a false positive rate below 0.7%. The obtained results confirm the applicability of the proposed approach for securing IoT platforms, robotic systems and distributed infrastructures relying on APIs as their primary communication mechanism.

Key words: Internet of Things, API, API Security, Web Application Firewall, IoT Security, Behavioral Analysis, Anomaly Detection, Information Security.

Введение

Технологии Интернета вещей (Internet of Things, IoT) являются одним из наиболее активно развивающихся направлений цифровой трансформации современной экономики. Устройства Интернета вещей широко применяются в промышленности, энергетике, транспортной инфраструктуре, системах умного города, логистических комплексах и робототехнических системах. Увеличение количества подключённых устройств сопровождается ростом объёмов передаваемых данных и усложнением взаимодействия между компонентами распределённых информационных систем [1].

Одним из ключевых механизмов обмена данными в современных IoT-платформах являются прикладные программные интерфейсы (Application Programming Interface, API). Посредством API осуществляется передача телеметрической информации, удалённое управление устройствами, настройка параметров функционирования, получение статистических данных и взаимодействие между отдельными сервисами распределённой инфраструктуры. В робототехнических системах API используются для передачи управляющих команд исполнительным механизмам, получения данных от датчиков и организации взаимодействия между программными и аппаратными компонентами системы.

Рост количества API и расширение их функциональности сопровождаются увеличением числа угроз информационной безопасности. Согласно OWASP API Security Top 10, прикладные интерфейсы являются одной из наиболее привлекательных целей для злоумышленников [2]. Наиболее распространёнными угрозами остаются нарушения контроля доступа на уровне объектов (Broken Object Level Authorization), ошибки авторизации, злоупотребление функциональными возможностями API, чрезмерное потребление ресурсов и атаки на бизнес-логику приложений. Особенностью подобных атак является то, что во многих случаях они

выполняются посредством корректно сформированных запросов, не содержащих явных признаков вредоносной активности.

Для систем Интернета вещей последствия успешной атаки на API могут выходить за пределы информационной среды и оказывать непосредственное влияние на физические процессы. Получение несанкционированного доступа к интерфейсам управления способно привести к изменению параметров работы устройств, подмене телеметрической информации, нарушению функционирования технологических процессов и отказу отдельных компонентов инфраструктуры [3]. В робототехнических системах компрометация API потенциально может повлиять на выполнение управляющих операций и безопасность эксплуатации оборудования.

В настоящее время для защиты прикладных интерфейсов широко применяются межсетевые экраны веб-приложений (Web Application Firewall, WAF), системы контроля доступа и механизмы ограничения интенсивности запросов. Большинство подобных решений ориентировано на анализ структуры HTTP-запросов и выявление известных шаблонов атак. Несмотря на высокую эффективность при обнаружении распространённых угроз, такие системы обладают ограниченными возможностями по выявлению злоупотреблений, использующих легитимные механизмы взаимодействия с API. В частности, атаки, основанные на последовательном переборе идентификаторов объектов, нехарактерной активности устройств или нарушении логики взаимодействия компонентов системы, зачастую остаются незамеченными традиционными средствами защиты.

Одной из особенностей IoT-инфраструктуры является высокая регулярность взаимодействия устройств с прикладными интерфейсами. Большинство устройств функционирует в рамках заранее определённых сценариев и использует ограниченный набор API-методов. Это позволяет

формировать профиль нормального поведения устройства и использовать его в качестве дополнительного источника информации при оценке безопасности выполняемых запросов. В последние годы методы поведенческого анализа активно применяются в задачах обнаружения аномалий и выявления угроз информационной безопасности [4]. Однако использование исключительно поведенческого анализа может приводить к увеличению числа ложных срабатываний и не позволяет эффективно обнаруживать некоторые типы атак, связанных с передачей вредоносного содержимого.

Таким образом, возникает необходимость разработки подходов к защите API устройств Интернета вещей, сочетающих преимущества структурного анализа запросов и поведенческого анализа устройств. Совместное использование данных механизмов позволяет учитывать как характеристики конкретного запроса, так и контекст взаимодействия устройства с информационной системой, что потенциально обеспечивает более высокую эффективность обнаружения атак по сравнению с применением каждого из подходов по отдельности.

Целью настоящей работы является разработка подхода к защите API устройств Интернета вещей на основе совместного анализа структуры запросов и поведенческих характеристик устройств. Для достижения поставленной цели предлагается система защиты, включающая механизмы структурного анализа API-запросов, контроля доступа, формирования профиля нормального поведения устройств и выявления аномалий. Эффективность предложенного подхода оценивается на экспериментальном стенде, моделирующем функционирование распределённой IoT-инфраструктуры.

Научная новизна работы заключается в разработке подхода к защите API устройств Интернета вещей, основанного на интеграции структурного анализа запросов и анализа поведенческих характеристик устройств в

рамках единой модели принятия решений. Практическая значимость исследования определяется возможностью применения предложенного подхода для защиты IoT-платформ, робототехнических комплексов, автоматизированных складских систем, систем умного города и других распределённых инфраструктур, использующих API в качестве основного механизма взаимодействия компонентов.

Анализ существующих подходов к защите API

В настоящее время безопасность прикладных программных интерфейсов является одним из наиболее активно развивающихся направлений информационной безопасности. Распространение микросервисных архитектур, облачных платформ и устройств Интернета вещей привело к тому, что API стали основным механизмом взаимодействия между компонентами современных информационных систем. В результате прикладные интерфейсы всё чаще становятся объектом целенаправленных атак со стороны злоумышленников [5].

Наиболее распространённым средством защиты API являются межсетевые экраны веб-приложений (Web Application Firewall, WAF). Принцип работы подобных решений основан на анализе HTTP-трафика и выявлении признаков известных атак с использованием сигнатур, правил фильтрации и механизмов сопоставления шаблонов [6]. Современные WAF способны эффективно обнаруживать SQL-инъекции, межсайтовое выполнение сценариев, попытки обхода механизмов аутентификации и другие распространённые угрозы прикладного уровня.

Несмотря на широкое распространение сигнатурного подхода, его эффективность существенно зависит от наличия актуальных правил обнаружения атак. Как отмечается в работе Ahmed и соавторов [7], сигнатурные механизмы демонстрируют ограниченные возможности по выявлению ранее неизвестных атак и злоупотреблений, использующих

корректно сформированные запросы. Особенно актуальна данная проблема для API-интерфейсов, поскольку значительная часть атак на них не связана с передачей вредоносного содержимого и осуществляется посредством легитимных функций приложения.

Отдельное направление развития средств защиты связано с механизмами контроля доступа. Их основная задача заключается в ограничении полномочий пользователей и устройств в соответствии с установленными политиками безопасности. Для этого используются различные модели авторизации, включая Role-Based Access Control (RBAC) и Attribute-Based Access Control (ABAC) [8]. Подобные механизмы позволяют снизить риск несанкционированного доступа к ресурсам приложения, однако не способны выявлять случаи компрометации легитимных учётных данных или злоупотребления разрешёнными функциями API.

В последние годы всё большее внимание уделяется использованию методов поведенческого анализа и обнаружения аномалий. В отличие от сигнатурных решений, такие системы ориентированы не на поиск известных шаблонов атак, а на выявление отклонений от нормального режима функционирования пользователей, приложений или устройств. Согласно исследованию Chandola и соавторов [9], аномалии могут выявляться на основе статистических характеристик трафика, временных зависимостей, последовательности выполняемых действий и других признаков.

Особый интерес представляет применение поведенческого анализа в системах Интернета вещей. Как отмечается в работах Sicari и соавторов [10] и Roman и соавторов [11], большинство IoT-устройств функционирует в рамках заранее определённых сценариев взаимодействия и использует ограниченный набор операций. Благодаря этому появляется возможность

формировать профиль нормального поведения устройства и использовать его для выявления подозрительной активности.

Однако использование исключительно поведенческого анализа также имеет ряд ограничений. Изменение режима работы устройства может быть связано с обновлением программного обеспечения, изменением конфигурации или особенностями эксплуатации. В подобных случаях существует риск увеличения количества ложных срабатываний и ошибочного ограничения доступа [12]. Кроме того, поведенческий анализ не всегда позволяет выявлять атаки, основанные на передаче вредоносного содержимого или эксплуатации известных уязвимостей прикладного уровня.

Проведённый анализ показывает, что существующие подходы обладают как преимуществами, так и недостатками. Сигнатурные механизмы обеспечивают эффективное обнаружение известных атак, но обладают ограниченными возможностями по выявлению злоупотреблений легитимными API-вызовами. Поведенческий анализ позволяет обнаруживать ранее неизвестные угрозы и аномалии, однако подвержен ложным срабатываниям. Механизмы контроля доступа обеспечивают разграничение полномочий, но не способны выявлять компрометацию доверенных устройств или пользователей.

В связи с этим перспективным направлением является разработка комплексного подхода, объединяющего структурный анализ запросов и анализ поведенческих характеристик устройств в рамках единой системы принятия решений [13–15]. Совместное использование данных механизмов позволяет учитывать как особенности конкретного API-запроса, так и контекст взаимодействия устройства с информационной системой, что потенциально обеспечивает более высокую эффективность защиты по сравнению с использованием каждого из подходов по отдельности.

В качестве предлагаемого решения в настоящей работе рассматривается подход к защите API устройств Интернета вещей, основанный на совместном анализе структуры запросов и поведенческих характеристик устройств. Концепция предлагаемого подхода представлена на рисунке 1.

Рисунок 1. Концепция совместного анализа структуры API-запросов и поведенческих характеристик устройств

Предлагаемый подход к защите API устройств Интернета вещей

Проведённый анализ существующих подходов показал, что ни сигнатурные механизмы защиты, ни системы контроля доступа, ни методы поведенческого анализа по отдельности не обеспечивают достаточной эффективности при защите API устройств Интернета вещей. Сигнатурные средства демонстрируют высокую эффективность при обнаружении известных атак, однако испытывают трудности при выявлении злоупотреблений легитимными функциями API. В свою очередь, поведенческий анализ способен обнаруживать аномальную активность, но подвержен ложным срабатываниям и не всегда позволяет выявлять атаки,

связанные с передачей вредоносного содержимого. В связи с этим в настоящей работе предлагается подход, основанный на совместном использовании структурного анализа запросов и анализа поведенческих характеристик устройств.

Основная идея предлагаемого подхода заключается в том, что решение о безопасности запроса принимается не только на основании его структуры и содержимого, но и с учётом контекста взаимодействия устройства с информационной системой. Таким образом, анализируется не только сам запрос, но и соответствие поведения устройства ранее сформированному профилю нормального функционирования.

В рамках предлагаемого решения каждый входящий API-запрос проходит несколько последовательных этапов обработки. На первом этапе выполняется структурный анализ запроса. Проверяется корректность URI, используемого HTTP-метода, наличие обязательных параметров и соответствие структуры запроса спецификации API. Дополнительно анализируются заголовки запроса и характеристики передаваемых данных. Целью данного этапа является выявление известных атак и попыток нарушения установленного формата взаимодействия с приложением.

Следующим этапом является контроль доступа. На данном этапе выполняется проверка полномочий пользователя или устройства на выполнение запрашиваемой операции. Дополнительно контролируется возможность обращения к конкретным ресурсам приложения и выполняется проверка соответствия запроса действующим политикам безопасности. Это позволяет предотвращать нарушения авторизации и ограничивать доступ к чувствительным функциям API.

После завершения структурного анализа и контроля доступа запрос передаётся в модуль поведенческого анализа. В отличие от предыдущих этапов данный механизм ориентирован не на содержимое отдельного запроса, а на анализ поведения устройства в целом. Для этого используются

сведения о ранее выполненных API-вызовах и накопленная статистика взаимодействия с системой.

Ключевым элементом предлагаемого подхода является использование набора признаков, характеризующих типовое поведение устройства. В процессе исследования были выделены следующие основные группы признаков:

– частота обращений к API;

– набор используемых endpoint;

– распределение HTTP-методов;

– временные интервалы между запросами;

– последовательность вызовов API;

– размеры запросов и ответов;

– характеристики ответов сервера.

Использование одновременно нескольких групп признаков позволяет уменьшить влияние случайных отклонений и повысить устойчивость системы к изменению отдельных параметров работы устройства.

Результаты структурного и поведенческого анализа объединяются в модуле оценки риска. Для каждого запроса формируется интегральный показатель риска, отражающий степень соответствия запроса установленным политикам безопасности и профилю нормального поведения устройства. При расчёте учитываются как признаки нарушения структуры запроса, так и выявленные отклонения от типового сценария взаимодействия.

В зависимости от полученного значения система принимает одно из трёх решений: разрешение обработки запроса, применение ограничений либо блокировка запроса. Использование нескольких уровней реагирования позволяет снизить вероятность необоснованной блокировки легитимного трафика и обеспечить более гибкое управление политиками безопасности.

Предлагаемый подход позволяет одновременно использовать преимущества сигнатурных механизмов защиты и методов поведенческого анализа. В результате обеспечивается возможность выявления как известных атак, связанных с нарушением структуры запросов, так и злоупотреблений, выполняемых посредством корректных API-вызовов и не содержащих явных признаков вредоносной активности.

Последовательность обработки запросов в разработанной системе защиты представлена на рисунке 2.

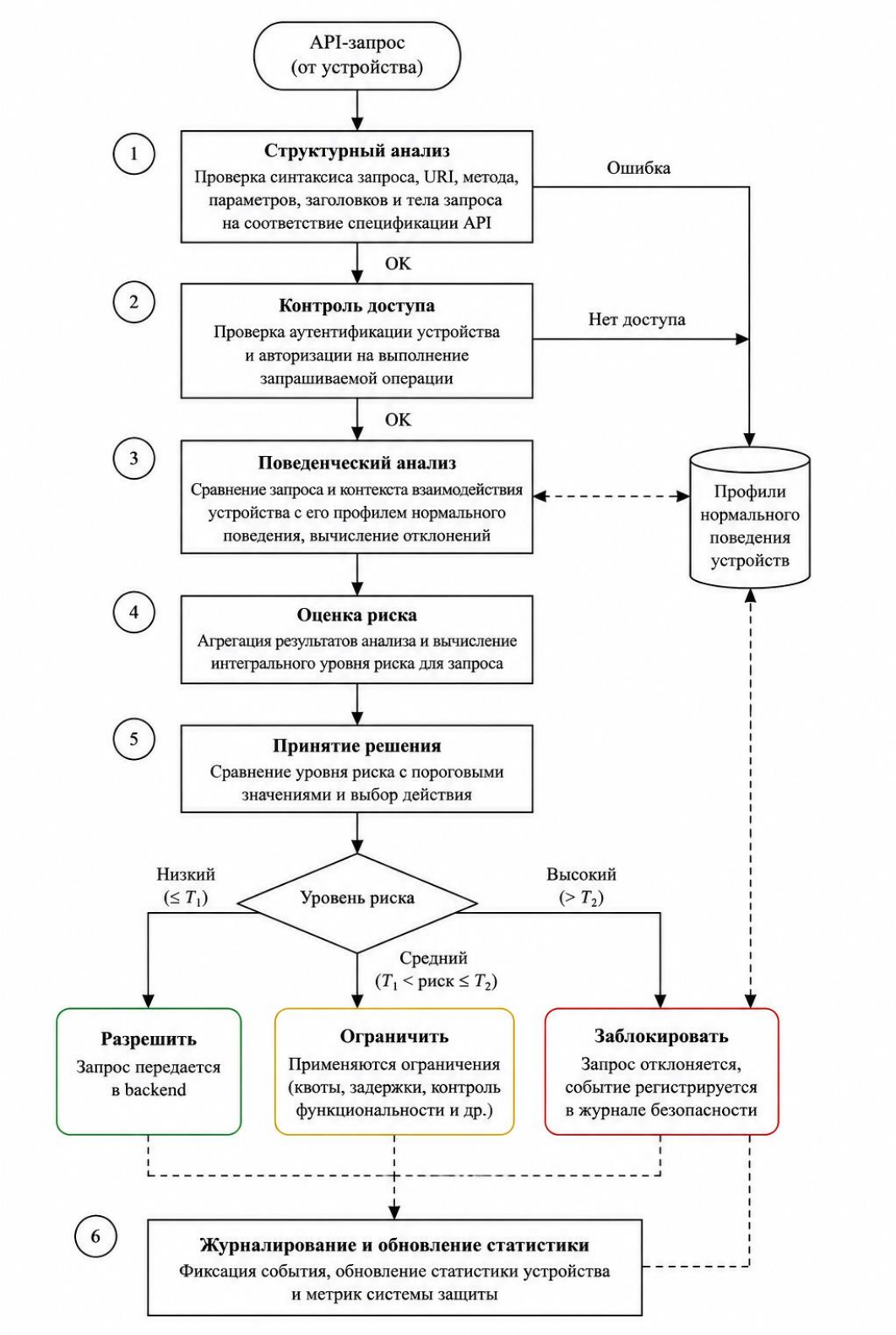

Рисунок 2. Алгоритм обработки API-запроса в системе защиты

Формирование профиля нормального поведения устройства

Ключевым элементом предлагаемого подхода является механизм формирования профиля нормального поведения устройства. Его задача заключается в накоплении статистической информации о характере взаимодействия устройства с API и последующем использовании полученных данных для выявления аномалий. В отличие от традиционных сигнатурных механизмов защиты, ориентированных на анализ содержимого отдельных запросов, данный подход позволяет учитывать долгосрочные особенности функционирования устройства и выявлять отклонения от типового сценария работы.

Формирование профиля начинается со сбора статистической информации о взаимодействии устройства с API. На этапе накопления данных регистрируются сведения обо всех запросах, поступающих от устройства, включая используемые endpoint, HTTP-методы, время обращения, объём передаваемых данных и характеристики ответов сервера. Сбор статистики осуществляется непрерывно в процессе эксплуатации системы и не требует внесения изменений в логику работы защищаемого приложения.

Одним из наиболее значимых признаков является частота обращений к API. Для каждого устройства определяется характерная интенсивность взаимодействия с системой. В процессе функционирования большинство устройств выполняет ограниченный набор операций с относительно стабильной периодичностью. Например, датчик температуры может передавать телеметрию через равные промежутки времени, а система видеонаблюдения — выполнять периодическую передачу служебной информации. Резкое увеличение количества запросов может свидетельствовать о попытке реализации атаки типа API Flood, переборе ресурсов приложения либо компрометации устройства.

Следующей характеристикой является набор используемых конечных точек API. Практика эксплуатации IoT-систем показывает, что каждое устройство обычно взаимодействует с ограниченным количеством методов и ресурсов приложения. Например, датчик мониторинга может использовать только методы передачи телеметрии и получения конфигурации. Появление обращений к новым endpoint, ранее не использовавшимся устройством, рассматривается как потенциальный индикатор подозрительной активности и требует дополнительного анализа.

Дополнительно анализируется распределение HTTP-методов. Для большинства устройств характерно использование ограниченного набора операций, таких как GET и POST. Использование нетипичных методов либо существенное изменение соотношения между ними может свидетельствовать о попытке выполнения действий, не предусмотренных логикой работы устройства.

Важным признаком также являются временные интервалы между запросами. Для многих типов IoT-устройств характерна высокая регулярность взаимодействия с информационной системой. Существенное отклонение от установленного временного шаблона может указывать на изменение режима работы устройства, попытку автоматизированного перебора ресурсов либо использование устройства злоумышленником.

Отдельное внимание уделяется анализу последовательности вызовов API. В процессе эксплуатации устройства выполняют определённые сценарии взаимодействия с системой. Например, после процедуры аутентификации может выполняться получение конфигурации, а затем передача телеметрии. Нарушение типичной последовательности вызовов API рассматривается как дополнительный фактор риска и может свидетельствовать о попытке злоупотребления функциональностью приложения.

Помимо характеристик взаимодействия анализируются размеры запросов и ответов. Значительное увеличение объёма передаваемых данных может указывать на попытку массового извлечения информации либо выполнение нехарактерных для устройства операций. Аналогично изменение типичного размера ответов сервера может свидетельствовать о попытках доступа к дополнительным ресурсам приложения.

На основе накопленной статистики формируется профиль нормального поведения устройства, содержащий диапазоны допустимых значений для каждого признака. При поступлении нового запроса выполняется сравнение его характеристик с параметрами сформированного профиля. Для каждого признака определяется степень отклонения от нормального поведения, после чего рассчитывается совокупная оценка аномальности.

Использование нескольких независимых признаков позволяет повысить устойчивость механизма обнаружения аномалий и снизить вероятность ложных срабатываний. В отличие от решений, основанных на анализе одного параметра, предложенный подход учитывает комплекс характеристик взаимодействия устройства с API, что обеспечивает более точное выявление потенциально опасной активности.

Общая схема формирования профиля нормального поведения устройствма представлена на рисунке 3.

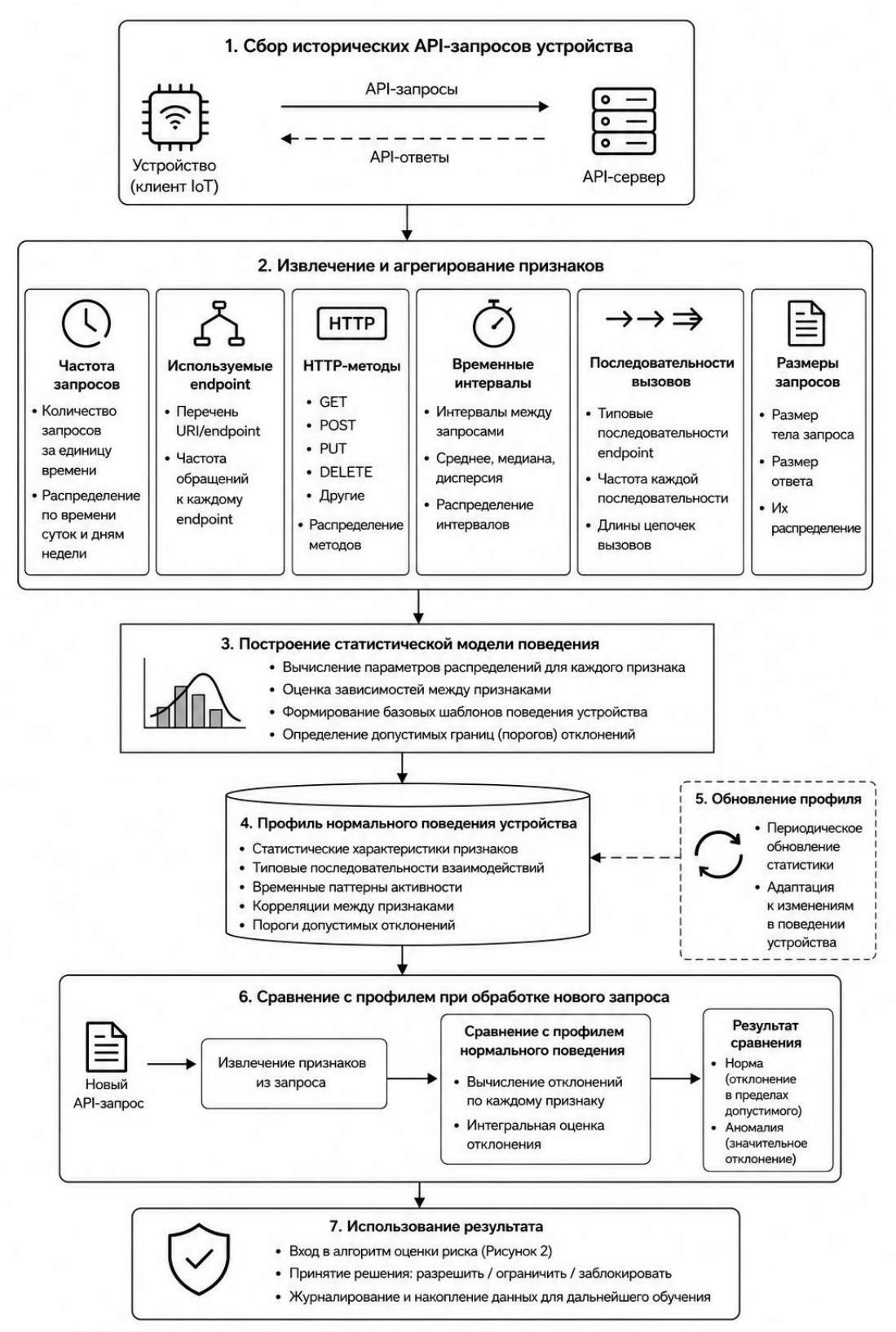

Рисунок 3. Формирование профиля нормального поведения устройства на основе анализа API-взаимодействий

Предлагаемый механизм позволяет выявлять атаки, выполняемые посредством легитимных API-запросов и не содержащие признаков нарушения структуры данных. Благодаря этому появляется возможность обнаружения компрометации устройств, злоупотребления разрешёнными функциями API и других угроз, которые сложно выявить исключительно средствами сигнатурного анализа.

Архитектура системы защиты API устройств Интернета вещей

Практическая реализация предложенного подхода выполнена в виде специализированной системы защиты API, интегрируемой на уровне reverse proxy перед защищаемым приложением. Подобная архитектура позволяет анализировать запросы до их передачи в backend-инфраструктуру и принимать решение о возможности дальнейшей обработки на ранних этапах взаимодействия.

В качестве точки интеграции системы защиты используется обратный прокси-сервер, через который проходит весь входящий API-трафик. Такое решение обеспечивает прозрачность внедрения и не требует модификации логики защищаемого приложения. Кроме того, размещение системы защиты на уровне reverse proxy позволяет централизованно контролировать доступ к API и выполнять анализ запросов независимо от внутренней архитектуры информационной системы.

Разработанная архитектура включает несколько функциональных компонентов, каждый из которых отвечает за выполнение отдельного этапа анализа.

Первым компонентом является модуль структурного анализа запросов. Его задачей является проверка корректности URI, используемых HTTP-методов, параметров запроса и структуры передаваемых данных. На данном этапе выявляются попытки нарушения спецификации API и известные атаки прикладного уровня.

Следующим компонентом является модуль контроля доступа. Он выполняет проверку полномочий пользователей и устройств на выполнение запрашиваемых операций, а также обеспечивает применение политик безопасности, определяющих допустимые сценарии взаимодействия с API.

Центральным элементом системы является модуль поведенческого анализа. Данный компонент осуществляет сбор статистики о взаимодействии устройств с API, формирование профиля нормального поведения и выявление отклонений от типовых сценариев работы. Именно этот модуль обеспечивает обнаружение атак, использующих легитимные механизмы взаимодействия с приложением.

Результаты структурного и поведенческого анализа поступают в модуль оценки риска. Его задачей является агрегирование полученных данных и вычисление интегрального показателя риска для каждого запроса. Использование единой модели оценки позволяет учитывать результаты нескольких механизмов защиты одновременно и принимать более обоснованные решения.

На основании рассчитанного уровня риска модуль принятия решений определяет дальнейшую судьбу запроса. В зависимости от степени потенциальной опасности запрос может быть разрешён, ограничен либо заблокирован. Дополнительно информация о выявленных событиях безопасности передаётся в подсистему журналирования и мониторинга для последующего анализа.

Предлагаемая архитектура обеспечивает возможность масштабирования системы защиты и её применения в различных типах инфраструктур, включая IoT-платформы, робототехнические комплексы, системы промышленной автоматизации и распределённые сервисные платформы. Разделение функций между отдельными модулями упрощает дальнейшее развитие системы и позволяет адаптировать её под особенности конкретной предметной области.

Архитектура разработанной системы защиты представлена на рисунке 4.

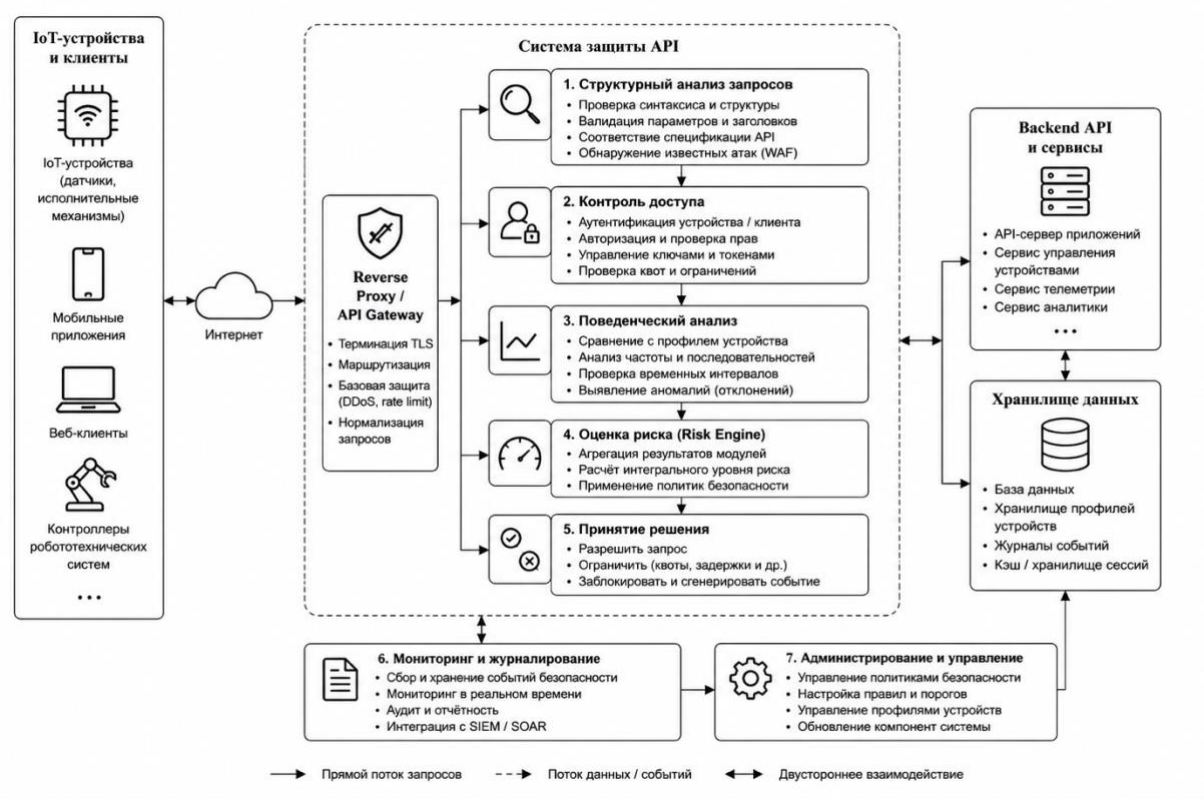

Рисунок 4. Архитектура системы защиты API устройств Интернета вещей

Экспериментальные исследования и оценка эффективности

Для оценки эффективности предложенного подхода был разработан экспериментальный стенд, моделирующий функционирование распределённой IoT-инфраструктуры. Основной целью исследований являлась проверка способности системы выявлять атаки на API, использующие как нарушения структуры запросов, так и легитимные механизмы взаимодействия с приложением. Дополнительно оценивалось влияние разработанного решения на производительность защищаемой инфраструктуры.

Экспериментальный стенд включал набор эмулируемых IoT-устройств, взаимодействующих с backend-приложением посредством HTTP API. В ходе испытаний моделировалась работа устройств различного назначения, выполняющих периодическую передачу телеметрии,

получение конфигурации и выполнение сервисных операций. Для обеспечения реалистичности сценариев использовались типовые шаблоны взаимодействия устройств с прикладными интерфейсами.

В рамках исследования были сформированы четыре основные группы атак:

– атаки типа API Flood, направленные на исчерпание вычислительных ресурсов системы за счёт генерации большого количества запросов;

– перебор идентификаторов объектов и устройств с целью получения доступа к данным других пользователей;

– злоупотребление логикой приложения посредством выполнения нетипичных последовательностей API-вызовов;

– обращения к запрещённым конечным точкам API.

Для каждой группы атак выполнялась серия испытаний, в ходе которых оценивалась эффективность их обнаружения средствами разработанной системы защиты. Дополнительно проводились испытания на легитимном трафике для определения уровня ложных срабатываний.

Результаты проведённых исследований показали высокую эффективность предложенного подхода. Средняя эффективность выявления атак составила 93 %. Наиболее высокий результат был достигнут при обнаружении атак типа API Flood, эффективность выявления которых составила 98,6 %. Эффективность обнаружения перебора идентификаторов устройств достигла 93,5 %, а эффективность выявления нарушений логики взаимодействия составила 87,5 %. Попытки обращения к запрещённым конечным точкам API обнаруживались в 100 % случаев.

Отдельное внимание было уделено оценке влияния системы защиты на легитимный трафик. В ходе испытаний было обработано более 320 тысяч легитимных запросов. Доля ложных срабатываний составила 0,7 %, что свидетельствует о возможности практического применения предложенного подхода без существенного влияния на работу пользователей и устройств.

Для оценки производительности проводились испытания при различных уровнях нагрузки. Результаты показали, что разработанное решение сохраняет работоспособность при интенсивности трафика до 2000 запросов в секунду. При максимальной нагрузке задержка обработки запросов не превышала 150 мс, что позволяет использовать систему в большинстве практических сценариев эксплуатации устройств Интернета вещей и робототехнических комплексов.

Дополнительно было выполнено сравнение предложенного подхода с традиционными средствами защиты API. Анализ результатов показал, что использование совместного структурного и поведенческого анализа обеспечивает более высокую эффективность выявления злоупотреблений легитимными API-вызовами по сравнению с решениями, основанными исключительно на сигнатурных механизмах обнаружения атак.

Полученные результаты подтверждают работоспособность предложенного подхода и демонстрируют возможность его практического применения для защиты прикладных интерфейсов устройств Интернета вещей. Использование профилей нормального поведения позволяет выявлять угрозы, которые сложно обнаружить средствами традиционного сигнатурного анализа, при сохранении приемлемых показателей производительности и низкого уровня ложных срабатываний.

Заключение

В работе рассмотрена проблема обеспечения безопасности прикладных интерфейсов устройств Интернета вещей. Проведённый анализ показал, что существующие средства защиты API, основанные преимущественно на сигнатурном анализе запросов и механизмах контроля доступа, обладают ограниченной эффективностью при обнаружении злоупотреблений, использующих легитимные механизмы взаимодействия с приложением.

Для решения данной проблемы предложен подход к защите API устройств Интернета вещей, основанный на совместном использовании структурного анализа запросов и анализа поведенческих характеристик устройств. В отличие от традиционных решений, предлагаемый подход учитывает не только содержимое отдельных запросов, но и особенности взаимодействия устройства с информационной системой в течение длительного периода времени.

В рамках исследования разработан механизм формирования профиля нормального поведения устройства, включающий анализ частоты обращений к API, используемых endpoint, распределения HTTP-методов, временных интервалов между запросами и последовательности вызовов API. На основе накопленной статистики выполняется выявление отклонений от типового сценария функционирования устройства и формируется оценка уровня риска выполняемых запросов.

Для практической реализации предложенного подхода разработана архитектура системы защиты API, интегрируемая на уровне reverse proxy и обеспечивающая выполнение структурного и поведенческого анализа до передачи запросов в backend-инфраструктуру. Предложенная архитектура позволяет централизованно контролировать доступ к API и обнаруживать угрозы на ранних этапах обработки трафика.

Результаты экспериментальных исследований подтвердили эффективность разработанного решения. Средняя эффективность выявления атак составила 93 %. Эффективность обнаружения атак типа API Flood достигла 98,6 %, эффективность выявления перебора идентификаторов устройств составила 93,5 %, а эффективность обнаружения нарушений логики взаимодействия достигла 87,5 %. При этом обращения к запрещённым конечным точкам API выявлялись в 100 % случаев. Доля ложных срабатываний не превысила 0,7 %, что свидетельствует о возможности практического применения предложенного подхода в реальных условиях эксплуатации.

Проведённые нагрузочные испытания показали, что разработанная система сохраняет работоспособность при интенсивности трафика до 2000 запросов в секунду. Максимальная задержка обработки запросов при данной нагрузке не превышала 150 мс, что подтверждает возможность использования предложенного решения без существенного снижения производительности защищаемой инфраструктуры.

Полученные результаты подтверждают возможность эффективной защиты API устройств Интернета вещей и робототехнических систем за счёт совместного использования структурного и поведенческого анализа. Практическая значимость работы заключается в возможности применения разработанного подхода для защиты IoT-платформ, робототехнических комплексов, систем умного города, автоматизированных складских систем и других распределённых инфраструктур, использующих API в качестве основного механизма взаимодействия компонентов.

Список литературы

1. Lin J., Yu W., Zhang N., Yang X., Zhang H., Zhao W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy and Applications // IEEE Internet of Things Journal. – 2017. – Vol. 4, No. 5. – P. 1125–1142.
2. OWASP Foundation. OWASP API Security Top 10 – 2023 [Электронный ресурс]. – Режим доступа: https://owasp.org/API-Security (дата обращения: 15.06.2026).
3. Roman R., Zhou J., Lopez J. On the Features and Challenges of Security and Privacy in Distributed Internet of Things // Computer Networks. – 2013. – Vol. 57, No. 10. – P. 2266–2279.
4. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey // ACM Computing Surveys. – 2009. – Vol. 41, No. 3. – P. 1–58.
5. Gartner. API Security: What You Need to Do to Protect APIs [Электронный ресурс]. – Режим доступа: https://www.gartner.com (дата обращения: 15.06.2026).
6. Grossman J., Hansen R., Petkov P., Rager A., Fogie S. XSS Attacks: Cross Site Scripting Exploits and Defense. – Burlington: Syngress, 2007. – 488 p.
7. Ahmed M., Mahmood A.N., Hu J. A Survey of Network Anomaly Detection Techniques // Journal of Network and Computer Applications. – 2016. – Vol. 60. – P. 19–31.
8. Hu V.C., Ferraiolo D., Kuhn R. Assessment of Access Control Systems. – Gaithersburg: NIST, 2006. – 126 p.
9. Chandola V., Banerjee A., Kumar V. Anomaly Detection for Discrete Sequences: A Survey // IEEE Transactions on Knowledge and Data Engineering. – 2012. – Vol. 24, No. 5. – P. 823–839.

10. Sicari S., Rizzardi A., Grieco L.A., Coen-Porisini A. Security, Privacy and Trust in Internet of Things: The Road Ahead // Computer Networks. – 2015. – Vol. 76. – P. 146–164.

11. Roman R., Najera P., Lopez J. Securing the Internet of Things // Computer. – 2011. – Vol. 44, No. 9. – P. 51–58.

12. Sommer R., Paxson V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection // Proceedings of the IEEE Symposium on Security and Privacy. – 2010. – P. 305–316.

13. OWASP Foundation. OWASP Internet of Things Project [Электронный ресурс]. – Режим доступа: https://owasp.org/www-project-internet-of-things (дата обращения: 15.06.2026).

14. Ring M., Wunderlich S., Grüdl D., Landes D., Hotho A. A Survey of Network-Based Intrusion Detection Data Sets // Computers & Security. – 2019. – Vol. 86. – P. 147–167.

15. ENISA. Threat Landscape for the Internet of Things [Электронный ресурс]. – Режим доступа: https://www.enisa.europa.eu (дата обращения: 15.06.2026).