СТАТЬИ

УДК 004.5

АРХИТЕКТУРА МОДУЛЯ ЦЕНТРАЛИЗОВАННОГО УПРАВЛЕНИЯ МНОГОФУНКЦИОНАЛЬНЫМ МЕЖСЕТЕВЫМ ЭКРАНОМ В СООТВЕТСТВИИ С ТРЕБОВАНИЯМИ ФЕДЕРАЛЬНОЙ СЛУЖБЫ ПО ТЕХНИЧЕСКОМУ И ЭКСПОРТНОМУ КОНТРОЛЮ

Ширяев А. И., Нажимова Н. А.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Нижегородский государственный технический университет имени Р. Е. Алексеева», Дзержинский филиал, Дзержинск, Российская Федерация, e-mail: aledjohierra@gmail.com

Ключевой задачей в обеспечении безопасности современных корпоративных и государственных сетей является эффективное управление распределенными сетевыми экранами, усложненное необходимостью соблюдения жестких нормативных требований. Статья посвящена решению проблемы централизованного, согласованного и безопасного администрирования множества многофункциональных межсетевых экранов в гетерогенной среде. Цель исследования – разработка и обоснование архитектуры модуля централизованного управления, соответствующей стандартам Федеральной службы по техническому и экспортному контролю (ФСТЭК) России. В работе использованы современные методы интеграции: прямое проксирование HTTP-запросов и REST API в сочетании с протоколом WebSocket для мониторинга и оповещений в реальном времени, что обеспечивает оперативное реагирование на угрозы. Безопасность передачи данных обеспечивается взаимной аутентификацией на основе TLS и JWT-токенов, применяемых для авторизации запросов и контроля доступа**.** Сквозное шифрование реализуется протоколом TLS 1.3**.** В результате предложены и сравнительно проанализированы два подхода к управлению. Прямое проксирование показало свою эффективность для сценариев, требующих минимальной задержки, таких как оперативное администрирование. Командный метод, основанный на постановке задач в очередь, продемонстрировал преимущества для унификации управления разнородными устройствами, централизованного аудита и агрегации данных. Разработанная архитектура подтверждает свою практическую значимость, повышая общую безопасность инфраструктуры за счет криптографической защиты каналов и снижая операционные риски благодаря автоматизации применения политик. Перспективы развития системы видятся в интеграции модулей аналитики больших данных для прогнозирования угроз и создания более интуитивного веб-интерфейса.

Ключевые слова: многофункциональный межсетевой экран, централизованное управление, безопасность, ФСТЭК, REST API, WebSocket, JWT, TLS

ARCHITECTURE OF A CENTRALIZED MANAGEMENT MODULE FOR A MULTIFUNCTIONAL NETWORK FIREWALL COMPLIANT WITH FEDERAL SERVICE FOR TECHNICAL AND EXPORT CONTROL REQUIREMENTS

Shiryaev A. I., Nazhimova N. A.

Federal State Budgetary Educational Institution of Higher Education "Nizhny Novgorod State Technical University named after R. E. Alekseev", Dzerzhinsk branch, Dzerzhinsk, Russian Federation, e-mail: aledjohierra@gmail.com

A critical challenge in securing modern corporate and state networks is the efficient management of distributed firewalls, complicated by the need to comply with stringent regulatory requirements. This article addresses the problem of centralized, consistent, and secure administration of multiple multifunctional firewalls in a heterogeneous environment. The research aims to design and justify an architecture for a centralized management module that complies with the standards of the Russian Federal Service for Technical and Export Control (FSTEC). The study employs modern integration methods: direct HTTP request proxying and REST API combined with the WebSocket protocol for real-time monitoring and alerts, ensuring prompt response to threats. Data transmission security is ensured by a two-tier protection system: JWT token-based authentication and end-to-end encryption using TLS. As a result, two management approaches are proposed and comparatively analyzed. Direct proxying proved effective for scenarios requiring minimal latency, such as operational administration. The command-based method, which relies on task queuing, demonstrated advantages for unifying control over heterogeneous devices, centralized auditing, and data aggregation. The developed architecture confirms its practical value by enhancing overall infrastructure security through cryptographic channel protection and reducing operational risks through policy automation. Future development of the system is envisioned in the integration of big data analytics modules for threat prediction and the creation of a more intuitive web interface.

Keywords: multifunctional firewall, centralized management, FSTEC, cybersecurity, REST API, WebSocket, JWT, TLS

Введение

Многофункциональный межсетевой экран (ММЭ) уровня сети – это программно-аппаратный комплекс, предназначенный для защиты информационных систем путем фильтрации сетевого трафика, обнаружения и блокирования угроз, а также обеспечения контроля доступа [1]. Согласно требованиям ФСТЭК, такие экраны должны соответствовать строгим требованиям, включая сертификацию по классам защиты (4–6), и объединять функции классического межсетевого экрана с дополнительными модулями безопасности, такими как:

− Глубокая проверка пакетов (DPI) – технология глубокого анализа сетевых пакетов, позволяющая идентифицировать, классифицировать и управлять трафиком на основе его содержимого в реальном времени [2].

− Системы обнаружения/предотвращения вторжений (IDS/IPS) – системы, анализирующие трафик для выявления атак, используя сигнатуры или аномалии. IDS только предупреждает об угрозах, а IPS также блокирует их [3].

− Антивирусная защита.

− Веб-фильтрация и контроль приложений.

В состав ММЭ должны входить компоненты, обеспечивающие возможность централизованного управления несколькими экземплярами ММЭ, эксплуатируемых в одной информационной (автоматизированной) системе, в соответствии с ролями пользователей ММЭ. Основным компонентом является система управления [4].

Система управления многофункциональным межсетевым экраном – это комплексное программное решение, предназначенное для централизованного контроля и администрирования функций межсетевого экрана. Оно обеспечивает согласованное взаимодействие между пользовательскими интерфейсами и модулями безопасности, отвечающими за фильтрацию трафика, мониторинг угроз и применение политик доступа. Система выступает в роли посредника, обрабатывая пользовательские команды и настройки и пересылая их модулям безопасности, а также собирает и анализирует данные о работе экрана для последующей отчетности.

Согласно требованиям ФСТЭК: «В случае удаленного доступа к ММЭ от имени пользователей ММЭ защита информации должна обеспечиваться путем защиты каналов связи от несанкционированного физического доступа (подключения) к ним и (или) применения в соответствии с законодательством РФ средств криптографической защиты информации или иными методами».

Цель исследования – разработка модуля централизованного управления, соответствующего требованиям ФСТЭК, для системы управления многофункциональным межсетевым экраном.

Материалы и методы исследования

В ходе экспериментального исследования решались следующие задачи:

1. Сравнение задержки и пропускной способности при управлении одним и группой (N = 1, 5, 10) подчиненных ММЭ.

2. Тестирование устойчивости WebSocket-соединений при разрывах канала и восстановлении связи.

3. Оценка масштабируемости (увеличение числа управляемых узлов до 20) и отказоустойчивости (имитация отказа управляющего ММЭ).

Эксперименты проводились на виртуальной инфраструктуре на базе VMware vSphere 7.0:

− управляющий ММЭ: 4 vCPU, 8 ГБ RAM, Astra Linux Special Edition версии 1.7;

− подчиненные ММЭ (до 20 экз.): 2 vCPU, 4 ГБ RAM, Astra Linux Special Edition версии 1.7;

− клиентское рабочее место: генерация нагрузки через утилиты wrk2 и k6;

− все узлы объединены в изолированную VPN-сеть (WireGuard) с ограничением полосы 1 Гбит/с, RTT 10 мс [5].

Для сбора метрик производительности использовались wrk2 (задержки, RPS) и k6 (сценарное тестирование). Профилирование потребления ресурсов CPU и памяти выполнялось штатными средствами ОС (perf, py-spy), а верификация защищенных соединений (TLS, WebSocket) проводилась путем анализа трафика в Wireshark.

Нормативно-правовые требования к реализации централизованного управления ММЭ определены в Требованиях Федеральной службы по техническому и экспортному контролю по безопасности информации к многофункциональным экранам уровня сети [1].

Разработана модель угроз и матрица верификации, фрагмент которой представлен в таблице.

− Активы: конфигурации ММЭ, журналы аудита, ключевая информация, трафик управления.

− Нарушитель: внешний (попытка перехвата/модификации трафика) и внутренний низкопривилегированный администратор.

− Векторы атак: перехват HTTP-сессий, подделка JWT, разрыв WS-соединений, DoS управляющего модуля.

Матрица верификации требований ФСТЭК

Требование ФСТЭК	Архитектурное решение	Метод проверки	Артефакт
Защита каналов связи	TLS 1.3, взаимная аутентификация	Анализ handshake в Wireshark	Отчет о тестировании TLS
Разграничение доступа	JWT + Ролевая модель	Тестирование API с разными ролями	Журнал аудита
Регистрация событий	Логирование в формате ГОСТ Р 59548 [6]	Инспекция БД аудита	Снимок экрана журнала
Централизованное управление	Прокси / Командный API	Сравнительное нагрузочное тестирование	Графики latency/rps

Методика измерений и критерии успешности (SLO):

− латентность (P99): не более 100 мс для 95 % запросов при 1000 RPS;

− пропускная способность: не менее 5000 запросов/с;

− устойчивость WebSocket: автоматическое переподключение не более 3 с при обрыве связи;

− масштабируемость: рост потребления CPU не более O(log N) при $N \leq 20$;

− повторяемость: каждый тест выполнялся 5 раз, доверительный интервал 95 %.

Для разработки централизованного управления были использованы следующие технологии:

– Протокол HTTP и его защищенная версия HTTPS – сетевые протоколы прикладного уровня, с помощью которых осуществляется обмен данных в формате Запрос – Ответ. HTTPS поддерживает шифрование протоколов TLS в целях повышения безопасности [7].

– Протокол WebSocket (WS) и его защищенная версия WebSocket Secure (WSS) работают поверх TCP. Установление защищенного соединения и обмен сертификатами выполняются на этапе инициализации HTTPS-соединения с использованием TLS [8], что обеспечивает шифрование на транспортном уровне модели OSI [9].

– JSON Web Token – стандарт создания токена доступа, хранящего данные в формате JSON. JWT используется для аутентификации мастер-пользователя на управляемом ММЭ [10].

Для организации централизованного прямого управления применяются проксирование HTTP-запросов и установление цепочки WebSocket-соединений по схеме «Клиент → Управляющий ММЭ → Управляемый ММЭ». Для командного управления организуется REST API [11] и цепочка WS-соединений «Управляемый ММЭ → Управляющий ММЭ → Клиент». Взаимодействие между управляющим и подчиненным организуется по схеме «клиент-сервер», где клиентом выступает головная система управления, а сервером – подчиненная СУ [12].

Результаты исследования и их обсуждение

В ходе реализации проекта была разработана и протестирована архитектура модуля централизованного управления многофункциональными межсетевыми экранами, соответствующая требованиям ФСТЭК.

В качестве сервера, обрабатывающего пользовательские запросы, выступает другой многофункциональный экран, оснащенный собственной системой управления. Аутентификация и авторизация запросов выполняются при помощи JWT, при этом конфиденциальность и целостность передачи обеспечивается протоколом TLS 1.3. Токен хранит те же данные, что и JWT при стандартном подключении к системе управления. Хранение ключей и информации о подчиненных ММЭ осуществляется в защищенном виде в базе данных с применением необратимого хэширования с «солью» для паролей и шифрованием для симметричных ключей [13].

Реализация командного управления на СУ решает задачи получения статистики и состояния работы подчиненных ММЭ и управления их политиками. Получение состояния модулей безопасности и собранной с них статистики в реальном времени осуществляется с помощью WebSocket-соединения.

Новый вариант с прямым подключением к подчиненному ММЭ представляет собой проксирование запросов на управляющей системе управления [14]. Этот режим подключения предоставляет полноценный доступ к конфигурированию подчиненного ММЭ, как если бы пользователь управлял в интерфейсе основной СУ.

1. Прямое проксирование запросов (прозрачное управление)

После авторизации в системе управления многофункционального межсетевого экрана пользователь может подключиться к подчиненному ММЭ, используя данные лицензии последнего.

Рис. 1. Прозрачное управление подчиненным ММЭ
Примечание: составлен авторами по результатам данного исследования

На основе идентификатора пользователя и лицензии подчиненного ММЭ формируется токен доступа, хранящийся на управляющем сервере. При каждом новом запросе пользователя управляющий инициализирует проверку, находится ли администратор в режиме управления подчиненным ММЭ. Если да, то проксирует его запрос на подчиненный сервер. Получив ответ от управляемого сервера, управляющий сервер передает его пользователю без дополнительной обработки. Вариант организации централизованного управления ММЭ через прямое проксирование запросов представлен на рис. 1.

При прямом проксировании управляющий ММЭ выступает прозрачным HTTP-прокси, перенаправляя запросы GUI/REST API к подчиненному узлу без обработки. Это обеспечивает минимальную задержку (latency P99 ≤ 50 мс при 1000 RPS), но требует строгой синхронизации API между узлами. Подход не позволяет централизованно валидировать команды и усложняет аудит.

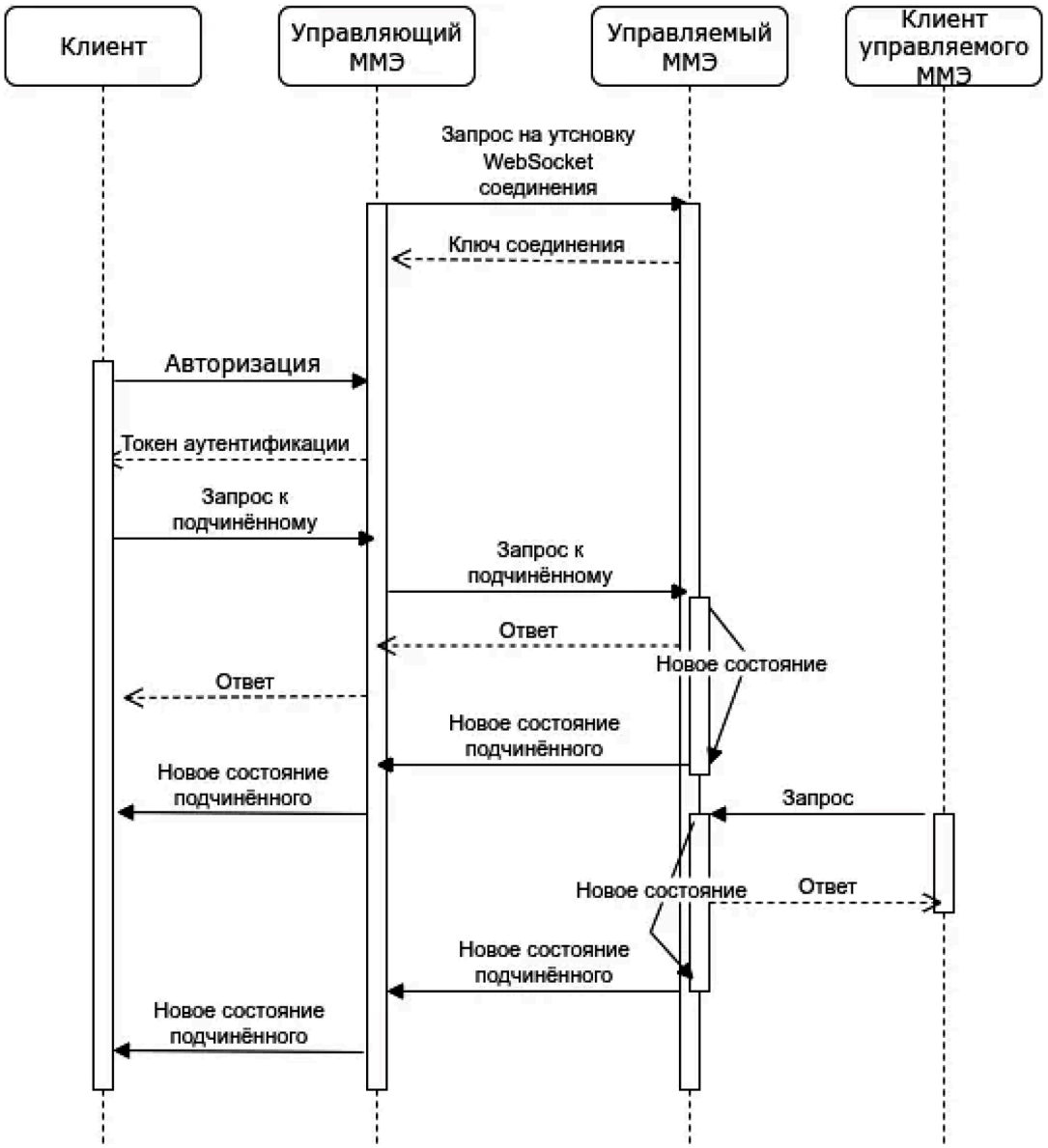

Рис. 2. Управление подчиненным ММЭ при помощи команд
Примечание: составлен авторами по результатам данного исследования

2. Командный подход (абстрактное управление через эндпоинты)

Управляемый ММЭ при регистрации управляемого многофункционального межсетевого экрана устанавливает с ним WebSocket-соединение, по которому передается статистика и состояние подчиненного в реальном времени. Для передачи политик безопасности и команд на изменение состояния модулей безопасности используется REST API управляющего ММЭ, который отправляет HTTP-запросы подчиненному. Диаграмма последовательности архитектуры абстрактного управления подчиненными ММЭ изображена на рис. 2.

При командном подходе управляющий ММЭ предоставляет унифицированные REST-эндпоинты, инкапсулирующие логику взаимодействия с подчиненными узлами. Пользователь отправляет команды в стандартизированном формате, а управляющий узел преобразует их в API-вызовы конкретных ММЭ. Это обеспечивает единый интерфейс для GUI [15] и позволяет реализовать централизованную валидацию, кеширование, агрегацию данных и аудит. Однако цена унификации – дополнительная задержка: latency P99 возрастает до 120 мс (против 50 мс при проксировании) при 1000 RPS за счет обработки команд на управляющем узле.

Заключение

В ходе проведенного исследования была разработана архитектура модуля централизованного управления многофункциональным межсетевым экраном, соответствующая требованиям ФСТЭК. Предложенное решение основано на двух ключевых подходах: прямом проксировании запросов и командном управлении через REST API. Оба метода обеспечивают безопасное и эффективное администрирование распределенных ММЭ, но различаются по степени абстракции и гибкости.

Прямое проксирование позволяет минимизировать задержки за счет сквозной передачи запросов между управляющим и подчиненными ММЭ, что особенно важно для оперативного управления в реальном времени. Однако этот подход требует строгой синхронизации API и усложняет централизованный аудит изменений. В свою очередь, командный подход обеспечивает унифицированный интерфейс для управления разнородными ММЭ, позволяя агрегировать данные, валидировать команды и вести единый журнал событий. Это повышает удобство администрирования, но может вносить дополнительную задержку из-за необходимости преобразования запросов.

Важным аспектом реализации является обеспечение безопасности. Взаимодействие между узлами защищается с помощью JWT-аутентификации, HTTPS-шифрования и WebSocket-соединений с TLS, что соответствует требованиям регуляторов. Кроме того, хранение ключей и конфигураций в зашифрованном виде снижает риск утечки критичных данных.

Разработанная система упрощает администрирование крупных сетевых инфраструктур за счет:

− централизованного управления политиками безопасности,

− мониторинга состояния подчиненных ММЭ в реальном времени,

− автоматизации рутинных задач через REST API.

В перспективе исследование методов централизованного управления будет продолжено с акцентом на следующее:

1. Повышение функциональности – интеграцию новых модулей анализа угроз и машинного обучения для прогнозирования атак.

2. Улучшение пользовательского интерфейса – разработку интуитивно понятных дашбордов с расширенной визуализацией данных.

3. Усиление безопасности – внедрение постквантовой криптографии и более строгих механизмов аутентификации.

Список литературы

1. Ермаков Г. С., Пантелеев Н. Н. Организация защиты информации с помощью NTA, NGFW и WAF в контексте защиты веб-приложений // Наукосфера. 2024. № 4–2. С. 22–26. EDN: CXNUMT. DOI: 10.5281/zenodo.11101435.

2. Воронин В. В. Анализ технологии Deep Packet Inspection // Вестник Воронежского института высоких технологий. 2018. № 3 (26). С. 40–43. EDN: YLWHGX.

3. Губарев В. Д. Системы обнаружения и предотвращения вторжений // Научный аспект. 2024. Т. 20. № 5. С. 2700–2704. EDN: BMCMFG.

4. Федеральная служба по техническому и экспортному контролю. Требования по безопасности информации к многофункциональным экранам уровня сети от 07.03.2023 № 44. 07.03.2023.

5. Фадеев Н. В., Лукьяненко А. В. Платформа ipsec VPN: создание сети типа Site-To-Site VPN на примере ос Cisco // Молодежь. Наука. Инновации. 2023. Т. 1. С. 206–210. EDN: LADZZX.

6. ГОСТ Р 59548-2022. Защита информации. Регистрация событий безопасности. Введ. 2022-09-01. М.: Стандартинформ, 2022. 28 с.

7. RFC 8446. The Transport Layer Security (TLS) Protocol Version 1.3 / E. Rescorla. August 2018. URL: https://datatracker.ietf.org/doc/html/rfc8446 (дата обращения: 19.08.2025).

8. Федоренков Р. В., Ничушкина Т. Н. Интерактивный веб-сервис WebSocket // Инженерный вестник. 2015. № 1. С. 3. EDN: TQMOAX.

9. Yuryev M. V. OSI model // Молодежь. Общество. Современная наука, техника и инновации. 2023. № 22. P. 67–68. EDN: PHPKVF.

10. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT) [RFC 7519] May 2015. Standards Track. ISSN 2070–1721. URL: https://datatracker.ietf.org/doc/html/rfc7519 (дата обращения: 19.08.2025).

11. Аникин Д. А. Анализ методов авторизации и аутентификации REST API // Международный журнал информационных технологий и энергоэффективности. 2023. Т. 8. № 5–2 (31). С. 120–124. EDN: DZKSMQ.

12. Лиманова Н. И., Селезнев И. А. Анализ эффективности клиент-серверной архитектуры // Бюллетень науки и практики. 2022. Т. 8. № 7. С. 392–396. DOI: 10.33619/2414-2948/80/37. EDN: AONLOB.

13. Головинский С. А., Маслова М. А., Лагуткина Т. В. Использование хеш-таблиц в механизме зашиты от DOS-атак на примере языка Python // Научный результат. Информационные технологии. 2025. Т. 10. № 2. С. 49–56. DOI: 10.18413/2518-1092-2025-10-2-0-5. EDN: ROGRBA.

14. Лазарев С. А., Демидов А. В. Применение технологии обратного проксирования в рамках системы управления информационным обменом сети корпоративных порталов // Информационные системы и технологии. 2011. № 6 (68). С. 131–136. EDN: OIJPWV.

15. Бобровская Р. М., Заяц А. М., Кечеруков А. Р., Вагизов М. Р. Разработка web-сервиса по мониторингу окружающей среды // Инновационное приборостроение. 2025. Т. 4. № 2. С. 77–83. DOI: 10.31799/2949-0693-2025-2-77-83. EDN: IGQNRK.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.