ISSN 3033-9359
Выпуск 1 (005), 2025

НАУЧНЫЙ ЖУРНАЛ ПРОФЕССИОНАЛИТЕТ

Москва

ПРОФЕССИОНАЛИТЕТ

№ 1 (005), 2025

Научный журнал

Основан в 2023 году

Зарегистрирован федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций

Номер свидетельства ЭЛ № ФС 77-84640

Дата регистрации 01.02.2023

Учредитель:

Уймин А.Г.

Адрес редакции:

Адрес редакции: 115432, г. Москва, ул. 5-я Кожуховская, д. 18, корп. 1, пом. 12.

ПРОФЕССИОНАЛИТЕТ, 2025, № 1 (005)

PROFESSIONALITET

No.1 (005), 2025

Scientific Journal

Founded in 2023

Registered with the Federal Service for Supervision of Communications, Information Technology and Mass Media

Certificate of Registration: EL No. FS 77-84640

Registration Date: 01.02.2023

Founder:

Uymin A.G.

Editorial Office:

Editorial Office Address: Premises 12, 18, Building 1, 5th Kozhukhovskaya St., Moscow, 115432, Russia.

СОДЕРЖАНИЕ

CONTENTS

Рубрика 1. Информатика и информационные процессы

УДК *004.056.53*

В. С. Скоромников [1], А. П. Андрюхина [1]

[1]ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина»,г. Москва, Российская Федерация

ИССЛЕДОВАНИЕ РЕАЛИЗАЦИИ IPsec IKEv2 НА БАЗЕ vESR

Аннотация: Статья посвящена экспериментальному исследованию реализации протокола IPsec IKEv2 на платформе виртуального расширенного коммутатора маршрутизации vESR в условиях развёртывания виртуализированных сетей NFV/SDN. Актуальность работы обусловлена необходимостью обеспечения защищённого туннелирования между распределёнными сетевыми узлами при ограниченных вычислительных ресурсах виртуальной инфраструктуры. В качестве основной проблемы рассматривается влияние выбора криптографических алгоритмов и количества одновременно устанавливаемых туннелей на производительность, загрузку CPU и устойчивость работы управляющей плоскости vESR. В ходе исследования был развернут тестовый стенд, включающий виртуальный маршрутизатор vESR, пир с реализацией IPsec IKEv2 и генератор сетевой нагрузки. Для оценки производительности применялось нагрузочное тестирование с использованием различных криптографических профилей, включая AES-128, AES-256, режимы GCM и CBC. Дополнительно анализировалось поведение системы при увеличении количества одновременных IPsec-туннелей. Полученные результаты показали наличие выраженного компромисса между уровнем криптографической защиты и производительностью: более стойкие алгоритмы повышают нагрузку на процессор и снижают запас масштабируемости. Установлено, что при росте числа туннелей нагрузка на управляющую плоскость увеличивается нелинейно, что ограничивает стабильную работу системы. На основе результатов сформулированы практические рекомендации по выбору оптимальной конфигурации IPsec IKEv2 на платформе vESR для достижения баланса между безопасностью, функциональностью и эффективным использованием ресурсов. Ограничения интерпретации результатов связаны с единичными измерениями и конкретной конфигурацией виртуального стенда.

Ключевые слова: *IPsec IKEv2; vESR; NFV/SDN; нагрузочное тестирование; криптографический алгоритм; пропускная способность; загрузка CPU.*

Введение

Актуальность работы обусловлена необходимостью обеспечения безопасности туннелей между распределёнными узлами при переходе на виртуальные сетевые функции (NFV/SDN). vESR как виртуальный маршрутизатор является ключевым элементом таких сетей. Протокол IKEv2 в связке с IPsec является стандартным решением для защищённого туннелирования [1]. Архитектурные ограничения IPsec и практические рекомендации по развёртыванию учитывались по RFC 4301 и NIST SP 800-77 Rev. 1 [5, 7].

Основные исследования IPsec и IKEv2 направлены на криптостойкость и производительность. Проблемы интеграции с виртуализацией изучены поверхностно.

Известны: архитектура IKEv2, методы оптимизации для аппаратного ускорения, базовые реализации в ПО (StrongSwan) [2]. Неизученным остаётся поведение и производительность стека IPsec IKEv2 на платформе vESR в условиях ограничения ресурсов ВМ.

Объектом исследования является обеспечение информационной безопасности в виртуализированных сетях передачи данных с использованием протоколов защищённого туннелирования.

Предмет исследования заключается в реализации и функционировании протокола обмена ключами IKEv2 в составе стека IPsec на программной платформе виртуального расширенного коммутатора маршрутизации (vESR).

Цель исследования – провести комплексное исследование реализации IPsec IKEv2 на платформе vESR, оценить её производительность и безопасность, а также разработать оптимизированные профили конфигурации для типовых сценариев развёртывания [4].

Методы исследования

Исследование является прикладным экспериментальным исследованием с элементами нагрузочного тестирования (бенчмаркинга). Цель – эмпирическая оценка производительности и поведения конкретной системы (IPsec IKEv2 на vESR) в контролируемых условиях.

К характеристикам сбора данных относятся vESR 8.2 (2 vCPU, 4 ГБ RAM, интерфейс eth1: 192.168.10.1/24). Пир (StrongSwan 5.9) с аналогичными ресурсами (eth1: 192.168.10.2/24). Генератор нагрузки (Ubuntu 22.04) с iperf3.

Для сравнения и формирования туннелей используются заранее подготовленные и настроенные образы виртуальных роутеров (например, на базе StrongSwan/Libreswan или другого vESR), выступающие в роли пиров (равноправных узлов).

Тестовая нагрузка. Сгенерированный сетевой трафик (TCP/UDP потоки) с использованием инструмента iperf3.

Мониторинг ресурсов для сбора данных. Сбор метрик потребления ресурсов vESR (загрузка CPU по ядрам, потребление оперативной памяти) средствами гипервизора (например, virt-top) и встроенными утилитами самой vESR.

Настройка первой фазы протокола IKEv2 (см. Приложение А).

Первая фаза отвечает за установление защищённого управляющего канала между маршрутизаторами. В рамках данной фазы были заданы следующие параметры:

- версия протокола: IKEv2;
- алгоритм шифрования: AES-256;
- алгоритм хэширования: SHA-256;
- метод аутентификации: Pre-Shared Key (PSK);
- время жизни ассоциации безопасности (SA).

Для аутентификации узлов в процессе установления IKE-соединения был использован метод предварительно распределенного ключа (Pre-Shared Key). Одинаковый ключ был задан на обоих виртуальных маршрутизаторах, что позволило успешно пройти процедуру взаимной аутентификации.

Настройка второй фазы протокола IKEv2 (см. Приложение В).

После успешной настройки первой фазы была выполнена настройка второй фазы IPsec, предназначенной для защиты пользовательского трафика.

В рамках Phase 2 были определены:

- протокол защиты: ESP (Encapsulating Security Payload); Выбор AES-GCM и AES-CBC сопоставлялся с актуальными требованиями к алгоритмам ESP [6].
- алгоритм шифрования данных: AES-256;
- алгоритм контроля целостности: SHA-256;
- параметры времени жизни IPsec-ассоциации.

На следующем этапе созданная IPsec-политика была привязана к соответствующему интерфейсу виртуального маршрутизатора. После применения политики инициировался процесс установления IPsec-туннеля между узлами. В результате была сформирована защищённая логическая связь между двумя удаленными локальными сетями.

Проверка состояния ассоциаций безопасности

На завершающем этапе была выполнена проверка состояния ассоциаций безопасности IKE и IPsec. Были получены таблицы IKE SA и IPsec SA, подтверждающие, что обе фазы протокола успешно установлены и находятся в активном состоянии.

Активное состояние IKE SA и IPsec SA подтверждено командами платформы; в таблице 1 приведён фрагмент захвата ESP и сопутствующего служебного трафика.

Таблица 1 – Фрагмент захвата ESP и сопутствующего служебного трафика в Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
18	39.219728	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
20	39.226789	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
21	39.231723	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)

22	39.237195	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
24	39.247124	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
25	39.248372	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
26	39.254083	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
27	39.259122	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
28	39.266499	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
30	39.273128	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
31	39.280137	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
32	39.285124	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
33	39.286452	101.1.1.2	101.1.1.2	ESP	182	ESP (SPI=0x0d830193)
34	40.017135	aa:bb:cc:00:10:10	aa:bb:cc:00:10:10	LOOP	60	Reply
35	40.973158	aa:bb:cc:00:10:10	224.0.0.5	OSPF	90	Hello Packet
36	40.172154	aa:bb:cc:00:10:10	aa:bb:cc:00:10:10	CDP/VTP/DTP/ PAgP/UDLD	254	Device ID: ESPI - Port ID: Ethernet0/1
37	40.193640	aa:bb:cc:00:10:10	aa:bb:cc:00:10:10	LOOP	60	Reply
38	40.624894	aa:bb:cc:00:10:10	aa:bb:cc:00:10:10	LOOP	60	Reply

Методы обработки данных

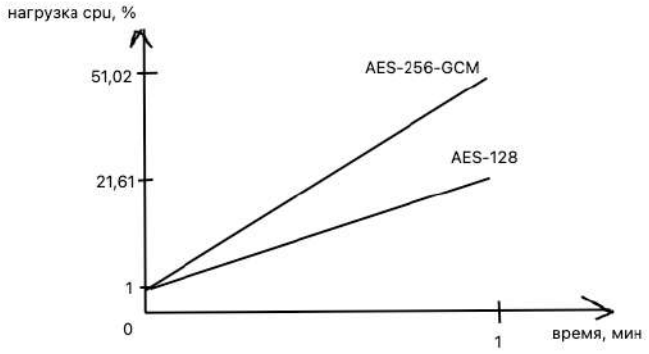


Рис. 1. Тест производительности AES-128 и AES-256-GCM

Результаты сравнения AES-128 и AES-256-GCM представлены в таблице 2.

Таблица 2 – Сравнительная таблица AES-128 vs AES-256-GCM

Параметр	AES-128	AES-256-GCM
Длина ключа	128 бит	256 бит
Тип режима	Обычный блочный (CBC/ECB)	AEAD (шифр + аутентификация)
Защита целостности	SHA2-256	Да (128-битный тег)

Устойчивость к анализу гистограммы	Средняя (зависит от режима, ECB уязвим)	Высокая — структура полностью скрыта
Производительность	• Загрузка CPU: 21,61 % • Средний RTT: 0,431 ms • Потери: 0%	• Загрузка CPU: 51,02 % • Средний RTT: 0,523 ms • Потери: 0%
Устойчивость к перебору	2^{128}	2^{256}
Применение	Устаревающие системы	Современные протоколы (TLS 1.3, IPsec)

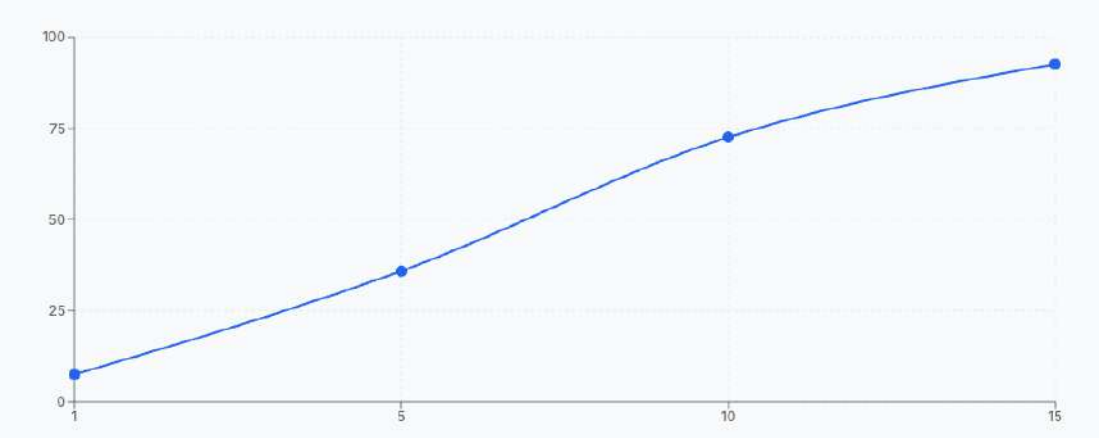


Рис. 2. Загрузка CPU (1, 5, 10, 15 туннелей)

Результаты тестирования производительности путем изменения количества туннелей представлены в таблице 3.

Таблица 3 – Влияние количества одновременных туннелей на производительность vESR

Кол-во туннелей	Загрузка CPU (управляющая плоскость, %)	Потребление памяти (MB)	Скорость установления (тунн./сек)
1	7,55	122	-
5	35,88	207	3,5
10	72,58	267	2,8
15	92,58	380	1,4

Обсуждение результатов

В рамках исследования была проведена комплексная экспериментальная оценка реализации стека IPsec IKEv2 на платформе виртуального расширенного коммутатора маршрутизации (vESR). Методология включала нагрузочное тестирование и сравнительный анализ различных криптографических алгоритмов (AES-128, AES-256, GCM, CBC). На специально развернутом стенде оценивались загрузка центрального процессора, RTT, потери и скорость установления туннелей; отдельная сводная метрика пропускной способности в представленные таблицы не включена.

1. Влияние алгоритмов шифрования. Единичные замеры CPU, RTT и потерь показывают различия между сравниваемыми профилями, однако не позволяют утверждать о статистически значимом снижении пропускной способности на 25 %, поскольку throughput, число повторов и разброс не приведены. В условиях данного стенда профиль AES-128 показал меньшую нагрузку CPU, чем AES-256-GCM; результат следует считать предварительным из-за одновременного изменения длины ключа и режима шифрования.

2. Масштабируемость и нагрузка на управляющую плоскость. Данные (Таблица 3) выявили нелинейную зависимость нагрузки от числа туннелей. Рост количества одновременных подключений с 1 до 15 приводит к увеличению загрузки CPU управляющей плоскости с 7,55% до 92,58%, а скорость установления новых туннелей падает с 3,5 до 1,4 тунн./сек.

3. Определение оптимальной конфигурации. На основе эмпирических данных была подтверждена гипотеза о существовании специфичной для vESR оптимальной конфигурации. Результаты показывают, что для достижения баланса между безопасностью, функциональностью и эффективным использованием ресурсов необходимо сместить выбор в сторону менее ресурсоёмких, но современных алгоритмов и жестко ограничивать масштабирование.

Заключение

Проведенное исследование позволило подтвердить ключевые гипотезы и сформулировать следующие выводы и рекомендации для практического применения IPsec IKEv2 на платформе vESR:

1. Существует выраженный компромисс между уровнем безопасности и производительностью. В условиях ограниченных виртуальных ресурсов (2 vCPU, 4 ГБ RAM) использование алгоритма AES-128-GCM является оптимальным для большинства стандартных сценариев.

2. Критическим параметром для стабильности системы является нагрузка на управляющую плоскость. Во избежание деградации сервиса и чрезмерных задержек рекомендуется ограничивать количество одновременных туннелей таким образом, чтобы устойчивая загрузка CPU не превышала 80-85%, что для тестового стенда соответствует приблизительно 10 туннелям.

3. Сформулированная оптимальная конфигурация (AES-128-GCM, группа Диффи-Хеллмана 19/14, контроль числа туннелей) обеспечивает разумный баланс, отличный от подходов, применяемых для аппаратных маршрутизаторов.

Направление дальнейших исследований:

1. Оценка влияния аппаратного ускорения шифрования (AES-NI) на производительность и поведение системы при миграции ВМ между хостами.

2. Анализ отказоустойчивости и времени переключения (failover) IPsec-туннелей в кластерных конфигурациях vESR.

3. Исследование безопасности реализации IKEv2 на vESR методами фаззинга для выявления потенциальных уязвимостей.

Библиографический список

1. Уймин, А. Г. Применение отечественного сетевого оборудования Eltex и EcoRouter в рамках специальности 09.02.06 "Сетевое и системное администрирование". Вопросы импортозамещения и подготовки квалифицированных кадров в сетевом оборудовании / А. Г. Уймин, И. М. Толмачев // Автоматизация и информатизация ТЭК. – 2025. – № 11(628). – С. 58–62. – EDN DMHQJU.
2. Kaufman, C. Internet Key Exchange Protocol Version 2 (IKEv2) [Электронный ресурс] : RFC 7296 / C. Kaufman, P. Hoffman, Y. Nir [и др.]. – IETF, 2014. – URL: https://datatracker.ietf.org/doc/html/rfc7296 (дата обращения: 12.12.2025).
3. Nir, Y. IPsec Cluster Problem Statement [Электронный ресурс] : RFC 6027 / Y. Nir. – IETF, 2010. – URL: https://datatracker.ietf.org/doc/html/rfc6027 (дата обращения: 12.12.2025).
4. Sheffer, Y. Additional EAP Methods for IKEv2 [Электронный ресурс] : RFC 5106 / Y. Sheffer, S. Fluhrer. – IETF, 2008. – URL: https://datatracker.ietf.org/doc/html/rfc5106 (дата обращения: 12.12.2025).
5. Kent, S. Security Architecture for the Internet Protocol [Электронный ресурс] : RFC 4301 / S. Kent, K. Seo. – IETF, 2005. – URL: https://www.rfc-editor.org/rfc/rfc4301.html (дата обращения: 31.03.2025).
6. Wouters, P. Cryptographic Algorithm Implementation Requirements and Usage Guidance for ESP and AH [Электронный ресурс] : RFC 8221 / P. Wouters, D. Migault, J. Mattsson, Y. Nir, T. Kivinen. – IETF, 2017. – URL: https://www.rfc-editor.org/rfc/rfc8221.html (дата обращения: 31.03.2025).
7. Barker, E. Guide to IPsec VPNs [Электронный ресурс] : NIST SP 800-77 Rev. 1 / E. Barker, Q. Dang, S. Frankel, K. Scarfone, P. Wouters. – NIST, 2020. – DOI: 10.6028/NIST.SP.800-77r1 (дата обращения: 31.03.2025).

References

1. Uymin, A. G. The use of domestic Eltex and EcoRouter network equipment in the framework of specialty 09.02.06 "Network and system administration". Issues of import substitution and training of qualified personnel in network equipment / A. G. Uymin, I. M. Tolmachev // Automation and informatization of the fuel and energy complex. – 2025. – № 11(628). – Pp. 58-62. – EDN DMHQJU.
2. Kaufman, C. Internet Key Exchange Protocol Version 2 (IKEv2) [Electronic resource] : RFC 7296 / C. Kaufman, P. Hoffman, Y. Nir [et al.]. – IETF, 2014. – URL: https://datatracker.ietf.org/doc/html/rfc7296 (date of request: 12.12.2025).
3. Nir, Y. IPsec Cluster Problem Statement [Electronic resource] : RFC 6027 / Y. Nir. – IETF, 2010. – URL: https://datatracker.ietf.org/doc/html/rfc6027 (date of request: 12.12.2025).
4. Sheffer, Y. Additional EAP Methods for IKEv2 [Electronic resource] : RFC 5106 / Y. Sheffer, S. Fluhrer. – IETF, 2008. – URL: https://datatracker.ietf.org/doc/html/rfc5106 (date of request: 12.12.2025).
5. Kent, S., & Seo, K. (2005). Security architecture for the Internet Protocol (RFC 4301). IETF. https://www.rfc-editor.org/rfc/rfc4301.html
6. Wouters, P., Migault, D., Mattsson, J., Nir, Y., & Kivinen, T. (2017). Cryptographic algorithm implementation requirements and usage guidance for ESP and AH (RFC 8221). IETF. https://www.rfc-editor.org/rfc/rfc8221.html
7. Barker, E., Dang, Q., Frankel, S., Scarfone, K., & Wouters, P. (2020). Guide to IPsec VPNs (NIST SP 800-77 Rev. 1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-77r1

Информация об авторах

Скоромников Виталий Сергеевич — студент 2-го курса, группы КА-24-06 Факультета КБ ТЭК, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: skoromnikov.vita@yandex.ru

Андрюхина Анастасия Павловна — студентка 2-го курса, группы КА-24-06 Факультета КБ ТЭК, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: Anastashia33@yandex.ru

INVESTIGATION OF IPsec IKEv2 IMPLEMENTATION BASED ON vESR

V. S. Skoromnikov[1], A. P. Andryukhina[1]

[1]National University of Oil and Gas «Gubkin University»

Abstract. The article presents an experimental study of the IPsec IKEv2 protocol implementation on the virtual extended routing switch platform, vESR, in the context of NFV/SDN-based virtualized network deployment. The relevance of the study is determined by the need to provide secure tunneling between distributed network nodes under the limited computing resources of a virtual infrastructure. The main research problem is the influence of cryptographic algorithm selection and the number of simultaneously established tunnels on throughput, CPU utilization, and the stability of the vESR control plane. During the study, a test environment was deployed, including a vESR virtual router, a peer node with IPsec IKEv2 support, and a network traffic generator. Load testing was used to evaluate performance under different cryptographic profiles, including AES-128, AES-256, GCM, and CBC modes. The behavior of the system was also analyzed when the number of simultaneous IPsec tunnels was increased. The obtained results demonstrated a significant trade-off between cryptographic strength and performance: stronger algorithms increase CPU load and reduce scalability margins. It was found that the control plane load grows nonlinearly as the number of tunnels increases, which limits stable system operation. Based on the experimental data, practical recommendations are proposed for selecting an optimal IPsec IKEv2 configuration on the vESR platform in order to balance security requirements, functionality, and efficient resource utilization.

Keywords: IPsec IKEv2; vESR; NFV/SDN; load testing; cryptographic algorithm; bandwidth; CPU utilization.

Information about the authors

Skoromnikov Vitaly Sergeevich — 2nd year student of the KA-24-06 group of the Faculty of the Fuel and Energy Complex Design Bureau, Gubkin Russian State University of Oil and Gas (NRU), Moscow, e-mail: skoromnikov.vita@yandex.ru

Andryukhina Anastasia Pavlovna — student of the 2nd year, group KA-24-06 Faculty of the Fuel and Energy Complex Design Bureau, Gubkin Russian State University of Oil and Gas (NRU), Moscow, e-mail: Anastashia33@yandex.ru

ПРОФЕССИОНАЛИТЕТ, 2025, № 1 (005)

УДК *004.624*

Д. С. Андрианова [1]

[1]*РГУ нефти и газа (НИУ) имени И.М. Губкина, г. Москва, Российская Федерация*

МИГРАЦИЯ ВИРТУАЛЬНЫХ МАШИН С VIRTUAL APPLIANCE (OVA) НА ZVIRT. ВОПРОСЫ НАДЁЖНОСТИ, БЕЗОПАСНОСТИ

Аннотация: *В статье рассматривается миграция виртуальных машин из формата Virtual Appliance (OVA), применяемого в гипервизорах VMware и VirtualBox, на отечественную платформу виртуализации zVirt версии 4.5. Актуальность исследования обусловлена необходимостью импортозамещения зарубежных средств виртуализации, а также обеспечением надёжности и безопасности при переносе виртуальной инфраструктуры. Цель работы заключается в экспериментальной проверке возможности холодной миграции виртуальной машины в среду zVirt 4.5 с сохранением её работоспособности и основных параметров безопасности.*

В ходе исследования был разработан и развёрнут экспериментальный стенд, включающий хост виртуализации, NFS-хранилище и менеджер управления. На стенде была выполнена миграция виртуальной машины под управлением операционной системы Microsoft Windows 10 из OVA-формата в среду zVirt. Особое внимание уделено конвертации виртуального диска из формата VMDK в целевой формат, корректности импорта, сохранению сетевых параметров и проверке функционирования виртуальной машины после переноса.

По результатам эксперимента подтверждена возможность успешной холодной миграции виртуальной машины на платформу zVirt 4.5. Для оценки надёжности и безопасности были выполнены сравнение хеш-сумм исходного и перенесённого OVA-образа, анализ журналов событий, а также проверка сохранения прав доступа NTFS/ACL. Полученные результаты показывают, что при корректной подготовке инфраструктуры миграция может быть выполнена с сохранением целостности данных, работоспособности виртуальной машины и контролируемости процесса переноса.

Ключевые слова: *zVirt, Virtual Appliance, OVA, хеш-сумма, надёжность, безопасность, миграция.*

Введение

В современном мире миграция виртуальных машин (далее – ВМ) играет ключевую роль в обеспечении гибкости и масштабируемости информационных технологий. Миграция представляет собой перенос виртуальных машин с исходного узла на целевой. Она применяется для балансировки нагрузки между серверами, проведения технического обслуживания без остановки сервисов, аварийного восстановления при сбоях, а также для переноса ВМ между различными платформами виртуализации в целях импортозамещения.

Актуальность исследования обусловлена необходимостью внедрения и обеспечения надёжности и безопасности миграции ВМ средствами отечественного программного продукта для виртуализации и управления виртуальной инфраструктурой zVirt версии 4.5. Это позволит повысить безопасность, надёжность и управляемость ВМ, с учётом современных требований к платформам виртуализации и отечественному программному обеспечению.

Проблема, решаемая в данной работе, заключается в отсутствии экспериментальных исследований, верифицирующих процедуру холодной миграции виртуальных машин из формата Virtual Appliance (OVA) на отечественную платформу виртуализации zVirt 4.5, учитывая обеспечение критериев безопасности и надёжности. Анализ существующих публикаций показывает, что основное внимание уделяется либо теоретическим аспектам миграции виртуальных машин на отечественную платформу zVirt 4.5, либо настройке на зарубежном оборудовании (Proxmox, Hyper-V) [1–2].

Целью работы является исследование и экспериментальная реализация миграции виртуальной машины с Virtual Appliance (OVA) на отечественную платформу виртуализации zVirt 4.5 с учётом требований к надёжности и безопасности.

Объектом исследования выступает процесс миграции виртуальной машины с Virtual Appliance (OVA) на отечественную платформу виртуализации zVirt 4.5.

Предметом исследования являются методы и процедуры настройки, реализации и обеспечения безопасности и надёжности миграции из Virtual Appliance (OVA) в среду

программного обеспечения zVirt 4.5. В рамках работы изучаются особенности настройки параметров виртуальной инфраструктуры, а также вопросы надёжности и безопасности.

Метод основан на экспериментальном подходе и сравнительном анализе полученных результатов с критериями надежности и безопасности.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области миграции ВМ из формата Virtual Appliance (OVA) и его особенности на основе современных подходов к виртуализации, требований к безопасности и надёжности.

2. Проанализировать особенности формата Virtual Appliance (OVA).

3. Спроектировать и развернуть экспериментальный стенд, имитирующий инфраструктуру миграции на основе программного решения zVirt 4.5.

4. Обеспечить надёжность и безопасность процесса миграции и эксплуатации ВМ.

5. Провести тестирование корректности работы виртуальной машины после миграции на платформу zVirt 4.5.

Литературный обзор

Проблема, рассматриваемая в работе Л.Е. Попок [3], заключается в недостаточной разработанности комплексных методик миграции виртуальных машин с платформы VMware vSphere на Microsoft Hyper-V. Автор рассматривает подходы к конвертации виртуальных машин между гипервизорами VMware и Hyper-V, а также совместимость дисков (VMDK в VHDX). Однако Л.Е. Попок не рассматривает реализацию миграции виртуальных машин на отечественных платформах виртуализации.

Работа Цыганковой А.А. и Климченко К.П. [4] посвящена сравнительному анализу отечественных платформ виртуализации Брест и zVirt, проведённому на основе критериев, важных для дальнейшего внедрения в образовательный процесс. Однако данные исследования носят теоретический характер и не затрагивают практическую реализацию и тестирование на отечественных платформах виртуализации.

В контексте вопросов безопасности интерес представляет работа Н.В. Корнеева и А.Б. Дикого [5], посвященная разработке паттерна для обеспечения безопасности информационной инфраструктуры при миграции образов виртуальных машин. Авторы предлагают решение на основе микросервисной архитектуры, развёрнутой в контейнерах Docker, целью которого является защита от DoS-атак в процессе переноса данных. Данное исследование носит общий характер и не фокусируется на отечественной платформе виртуализации zVirt 4.5 и особенностях работы с форматом OVA.

Проведённый анализ источников свидетельствует об отсутствии экспериментальной работы, в ходе которой была бы произведена миграция виртуальных машин с Virtual Appliance (OVA) на отечественную платформу с учётом аспектов надёжности и безопасности.

Анализ предметной области

Миграция виртуальной машины представляет собой процесс перемещения работающей или остановленной виртуальной машины с одного хоста на другой. Данный процесс включает полное сохранение состояния машины, включая в себя содержимое оперативной памяти, состояние процессорных регистров, конфигурацию подключённых устройств и данные виртуальных дисков. Операция миграции виртуальной машины является критической для современных центров обработки данных, поскольку позволяет обеспечивать непрерывность работы приложений и балансировать нагрузку между физическими серверами.

Существует два типа миграции. Первый предполагает миграцию без остановки виртуальной машины – горячая миграция [6]. Процесс данной миграции в реальном времени переносит работающую ВМ с одного узла на другой. Все свойства ВМ, в том числе IP-адреса, метаданные, память, состояние сетей, операционных систем и приложений, остаются неизменными. Выбираемый для миграции целевой узел должен быть активен [7]. Второй тип — с остановкой ВМ: виртуальная машина недоступна на время миграции. Данный тип миграции называется холодной миграцией [6]. Этот процесс копирует дисковые образы, конфигурацию и метаданные на целевой узел без передачи оперативной памяти, поскольку она очищается при

выключении. Миграция с Virtual Appliance (OVA) на платформу zVirt 4.5 относится ко второму типу. Это можно объяснить тем, что ВМ полностью останавливается для создания экспортного образа, затем выполняется перенос и конвертация. На последнем шаге виртуальная машина импортируется и запускается на новой платформе.

Формат OVA (Open Virtualization Appliance) представляет собой каталоги в формате OVF – стандартном формате программ виртуальных устройств. Отличительная черта данного формата состоит в том, что все компоненты сохраняются в одном архиве. Для этого используется метод упаковки TAR. Это облегчает распространение данных виртуальных машин [9]. Платформа zVirt 4.5 использует собственные форматы виртуальных дисков (QCOW2, RAW), имея собственные требования к конфигурации виртуальных устройств. Эти особенности создают необходимость преобразования импортируемых образов, корректировки параметров виртуальных машин и проверки их работоспособности после переноса [10].

Надёжность миграции заключается в корректности виртуальной машины после переноса. Целостность виртуальных дисков является важным фактором при конвертации из исходного формата в целевой [8]. Ошибки при конвертации могут привести к невозможности загрузки операционной системы (далее – ОС). Формат Virtual Appliance (OVA) состоит из нескольких файлов, и повреждение или изменение их может привести к некорректной работе виртуальной машины, что обусловливает необходимость проверки целостности исходного образа. Для обеспечения надёжности выполняется тестовый запуск виртуальной машины, проверка работоспособности сервисов, корректности сетевых настроек.

В Российской Федерации ключевым документом, устанавливающим требования к средствам виртуализации, является Приказ ФСТЭК России № 187 от 27.10.2022 «Об утверждении требований по безопасности информации к средствам виртуализации» [11]. Требования приказа направлены на обеспечение конфиденциальности, целостности и доступности информации.

При соблюдении конфиденциальности данные мигрированной виртуальной машины не должны быть раскрыты. Злоумышленник может получить доступ к виртуальной машине, если на системе не настроены права доступа (NTFS/ACL) или они были сброшены в процессе миграции.

При обеспечении целостности данные и состояние ВМ не должны быть изменены, повреждены или подменены в процессе миграции. Повреждение файлов мигрирующей машины может произойти из-за аппаратных сбоев или ошибок в программных инструментах конвертации и передачи данных, приводя к изменению структуры OVA-образа.

Доступность определяется тем, что сервисы, работающие на виртуальной машине, должны оставаться доступными для пользователя после процесса миграции. Неудачная попытка миграции виртуальной машины с Virtual Appliance (OVA) в среду zVirt 4.5 приводит к изменению OVA-образа. Данный фактор может привести к неработоспособности виртуальной машины после процесса миграции.

Безопасность миграции заключается в защите данных переносимой виртуальной машины. Значимую роль в вопросе безопасности играют файлы с журналами событий, используемые для аудита и анализа событий.

Вопросы надежности и безопасности миграции виртуальной машины с Virtual Appliance (OVA) в среду zVirt 4.5 включают проверку целостности данных, а также верификацию работоспособности сервисов, контроль корректности сетевых настроек, просмотр журналов с логами.

Проведение эксперимента

Для реализации процесса миграции виртуальных машин с Virtual Appliance (OVA) в среду zVirt 4.5 был выполнен эксперимент, включающий развертывание среды zVirt 4.5, настройку конфигурации процедуры миграции виртуальных машин из формата Virtual Appliance (OVA). Данный эксперимент направлен на тестирование холодной миграции с аспектами на вопросы надёжности и безопасности.

Были использованы виртуальные машины (далее – ВМ) с конфигурацией, представленной в таблице 1.

Таблица 1 – Конфигурация используемых виртуальных машин

Объект	ОС	RAM, ГБ	CPU, МБ	Размер диска, ГБ	IP-адрес	FQDN
Host 1	zVirt Node 4. 5	6	5	150	192.168.105.123/24	host1.vlab.local
NFS-хранилище	zVirt Node 4.5	4	4	150	192.168.105.162/24	nfs.vlab.local
CLI	РЕД 8.0.2	4	2	100	192.168.105.185/24	cli.vlab.local
Менеджер управления	-	4	4	61	192.168.105.13/24	engine.vlab.local
Мигрирующая виртуальная машина	Microsoft Windows 10	4	4	30	192.168.105.182/24	-

Для проведения эксперимента был развернут тестовый стенд на базе платформ ОС zVirt Node 4.5 и ОС РЕД 8.0.2. Архитектура стенда построена под задачу отработать миграцию виртуальной машины с Virtual Appliance (OVA) в среду zVirt 4.5. Выделены три основные машины. Первая Host 1 – машина с гипервизором, где работают виртуальные машины и развёрнут менеджер управления. Вторая NFS-хранилище – машина, на которой развёрнуто сетевое хранилище для хранения данных о виртуальных машинах. Третья машина выступает в роли клиента и обеспечивает доступ к веб-интерфейсу, так как первая и вторая машины имеют консольный режим (рис. 1). Миграция виртуальной машины win1 будет происходить с машины Admin-PC с платформы VMware Workstation Pro 25H2 [12].

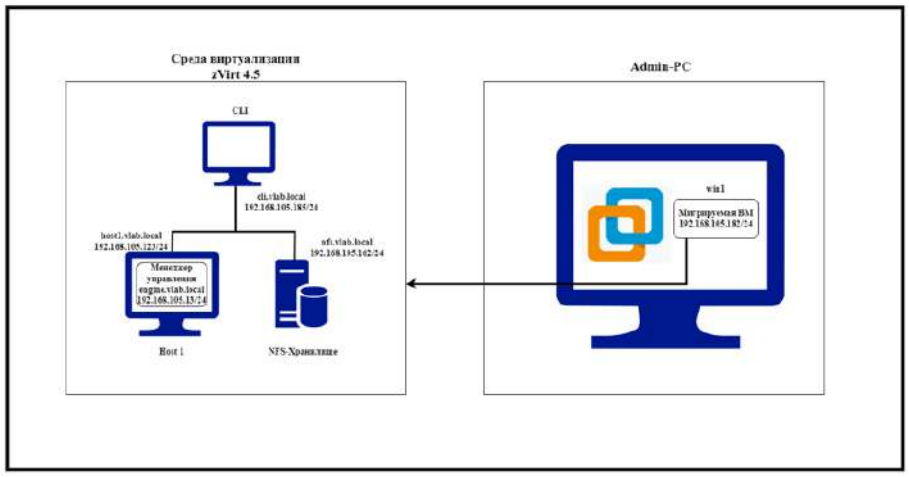


Рис. 1. Структура экспериментального стенда для миграции виртуальной машины в среду zVirt 4.5

Для начала эксперимента производится назначение IP-адресов и FQDN на устройствах согласно таблице 1 (рис.2 – рис.4).

Рис. 2. Назначение IP-адреса и FQDN на узле host1.vlab.local

Рис. 3. Назначение IP-адреса и FQDN на узле nfs.vlab.local

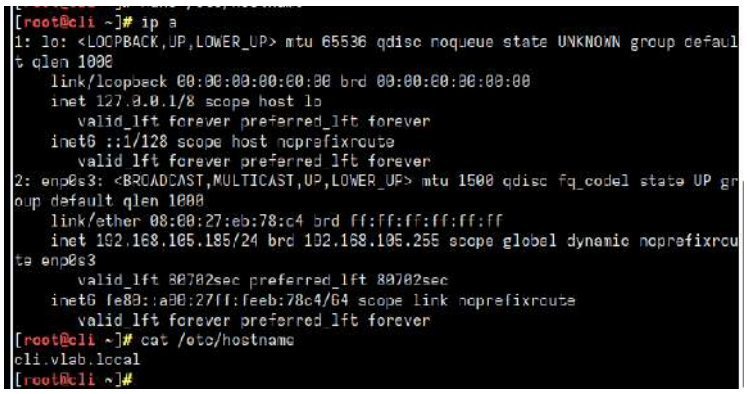


Рис. 4. Назначение IP-адреса и FQDN на клиентской машине cli.vlab.local

На каждой машине прописывается FQDN в файл /etc/hosts (рис. 5).

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.105.123 host1.vlab.local
192.168.105.162 mfs.vlab.local
192.168.105.13  engine.vlab.local
```

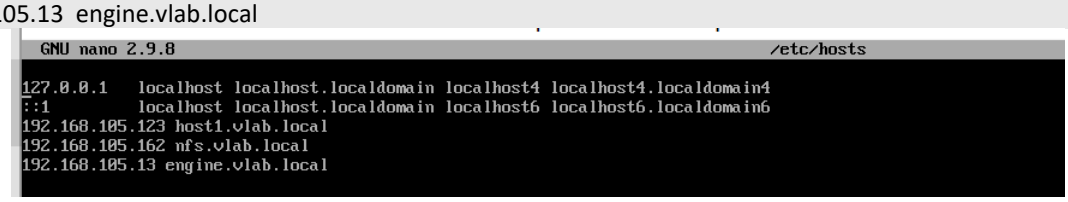


Рис. 5. Заполнение файла /etc/hosts для сопоставления IP-адресов и FQDN узлов стенда

Для подготовки хранилища происходит создание каталога, в котором должно быть больше 55 ГиБ свободного места, а также установка прав доступа (рис. 6).

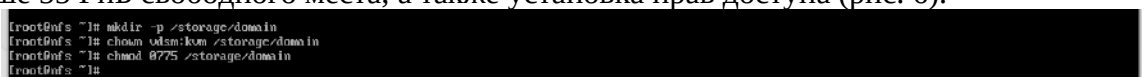


Рис. 6. Создание каталога NFS-хранилища и настройка прав доступа

Далее добавляется запись /storage/domain *(rw,anonuid=36,anongid=36) для NFS-сервера. Затем происходит запуск необходимых сервисов и создание правила межсетевого экрана.

На машине, выполняющей роль хоста, производится настройка репозиториев и развертывание менеджера управления. Затем с помощью команды hosted-engine --deploy начинается процесс установки менеджера управления. При развёртывании менеджера управления указываются его параметры согласно таблице 1.

Далее при корректной установке менеджера управления можно получить доступ к веб-интерфейсу по адресу https://engine.vlab.local/ovirt-engine/ (рис.7).

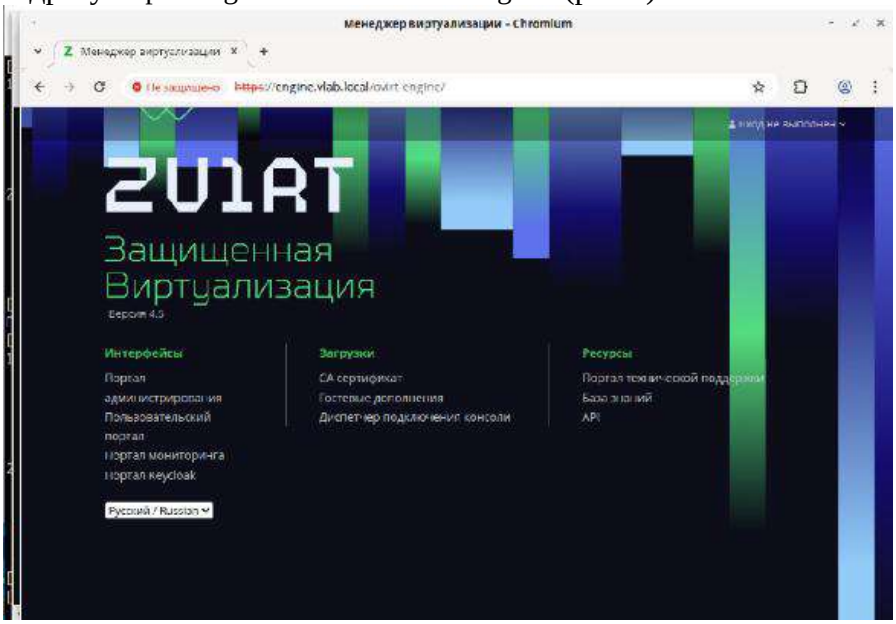


Рис. 7. Веб-интерфейс менеджера управления zVirt 4.5

После развёртывания менеджера управления происходит добавление хостов и хранилища.

Для импорта виртуальной машины из Virtual Appliance (OVA) в среду zVirt 4.5 дисковым форматом является VMDK. Перед тем, как импортировать виртуальную машину win1 в среду zVirt 4.5, делается экспорт из платформы VMware Workstation Pro 25H2.

Далее каталогу /data присваиваются права доступа 660 и владельцы:группа 36:36. В данном каталоге находится образ формата OVA (рис.8).

```
cd /data/
ls
cd /
chown 36:36 /data/win1.ova
chmod 660 /data/win1.ova
```

Рис. 8. Настройка прав доступа к каталогу с OVA-файлом виртуальной машины

Затем через менеджер управления выполняется импорт машины в среду виртуализации zVirt 4.5. На первом шаге импорта происходит указание хоста и пути к OVA-файлу на NFS-хранилище (рис. 9).

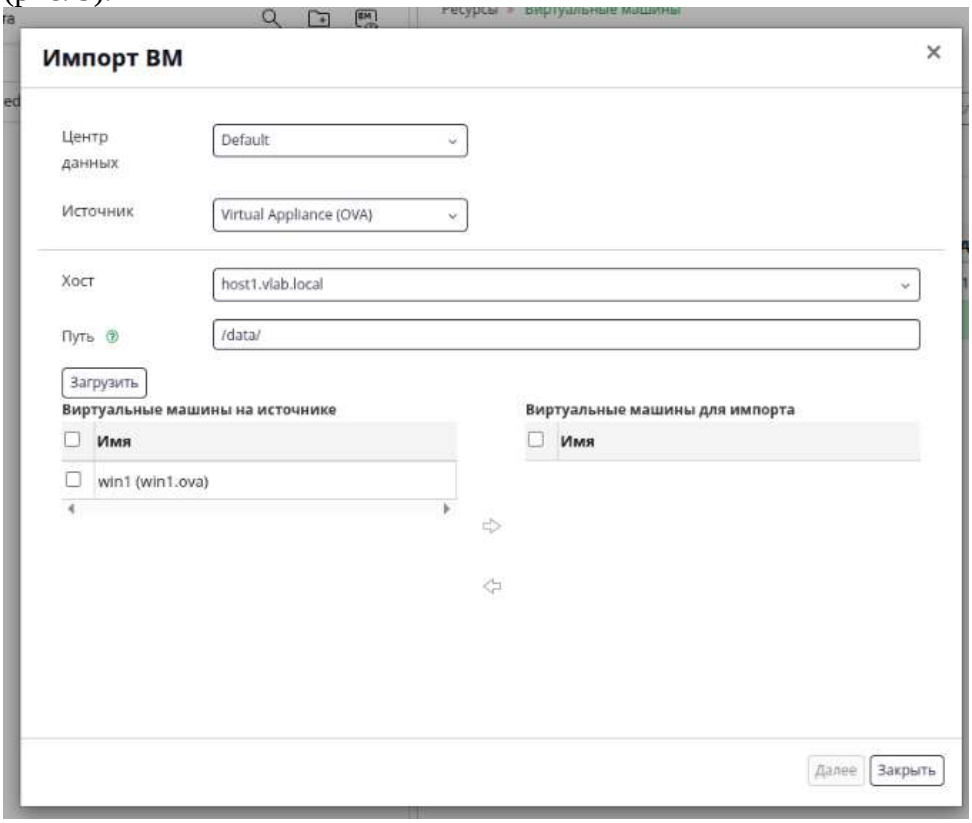


Рис. 9. Выбор хоста и пути к OVA-файлу при импорте виртуальной машины

На втором шаге задаются параметры импортируемой машины: целевой домен хранения hostedstorage, целевой кластер и профиль центрального процессора (далее – ЦП) по умолчанию, имя win1, а также выбирается импортируемая операционная система Windows 10. Остальные параметры были оставлены по умолчанию (рис. 10 – рис. 12).

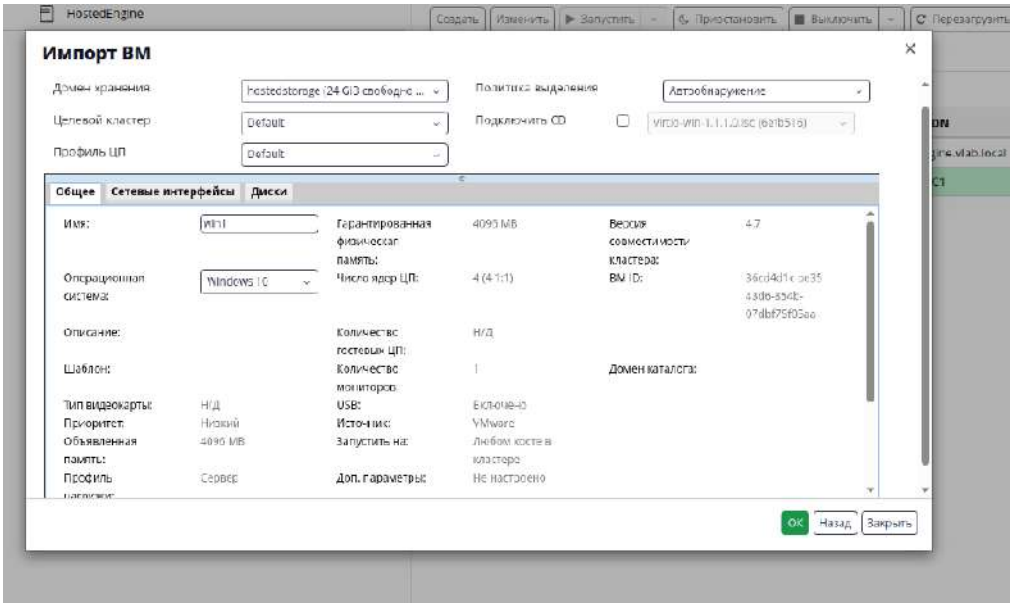


Рис. 10. Настройка целевого домена хранения при импорте виртуальной машины

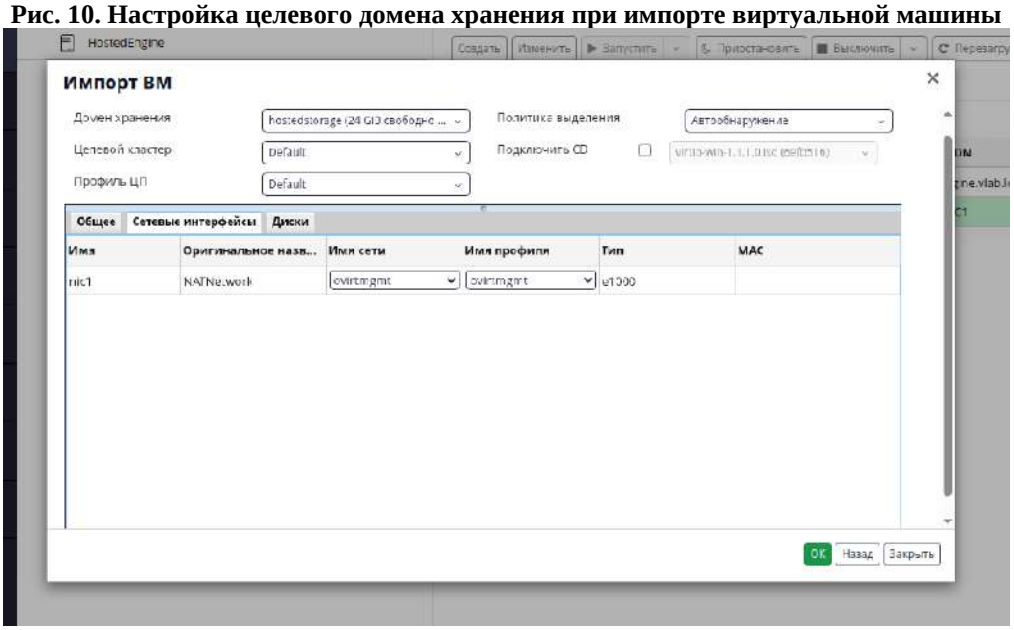


Рис. 11. Настройка параметров кластера и профиля процессора при импорте виртуальной машины

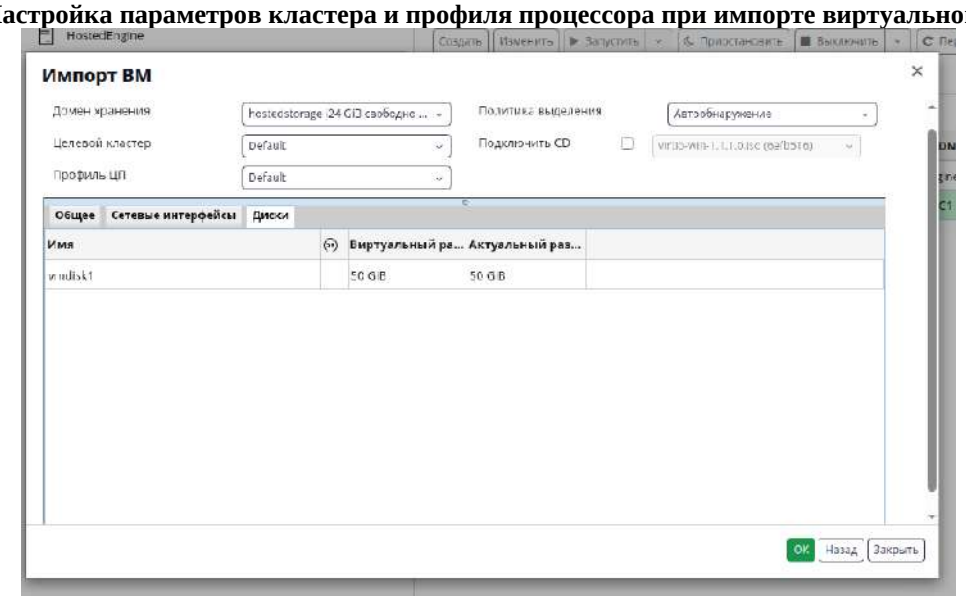


Рис. 12. Выбор операционной системы и имени импортируемой виртуальной машины

Импорт OVA-файла происходит напрямую в zVirt 4.5 через веб-интерфейс на хосте host1.vlab.local, с автоматической конвертацией дисков VMDK в QCOW2/RAW и настройкой интерфейсов VirtIO. Служба управления хостами zVirt 4.5 (VDSM) вызывает qemu-img convert для конвертации дисков из VMDK в QCOW2/RAW с VirtIO; диски импортируются в домен хранения без ручного ввода команд. Виртуальная машина win1 была сконвертирована в формат RAW (рис. 13).

Рис. 13. Результат импорта виртуальной машины в среду zVirt 4.5

В ходе эксперимента была успешно мигрирована ВМ с операционной системой Microsoft Windows 10. После запуска в среде отечественной платформы виртуализации zVirt 4.5 работоспособность операционной системы, сетевые подключения и базовые сервисы были полностью сохранены (рис. 14–15).

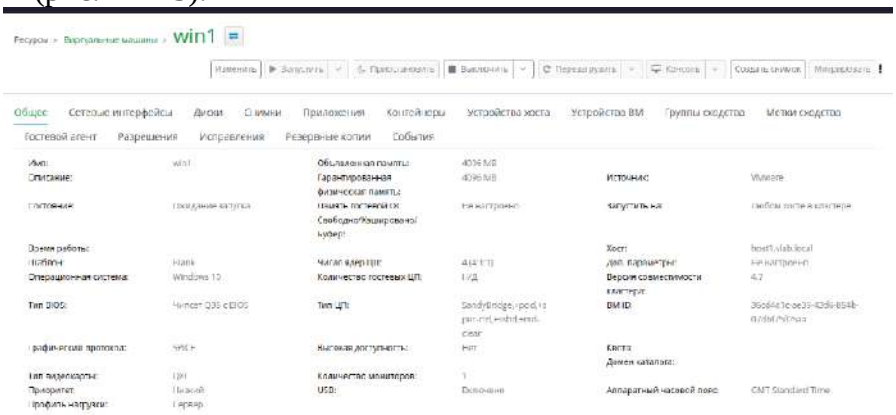


Рис. 14. Параметры виртуальной машины после запуска в среде zVirt 4.5

Рис. 15. Проверка работоспособности виртуальной машины после миграции

Стоит отметить, что во вкладке «События» содержатся сообщения об успешном импорте машины win1 (рис.16).

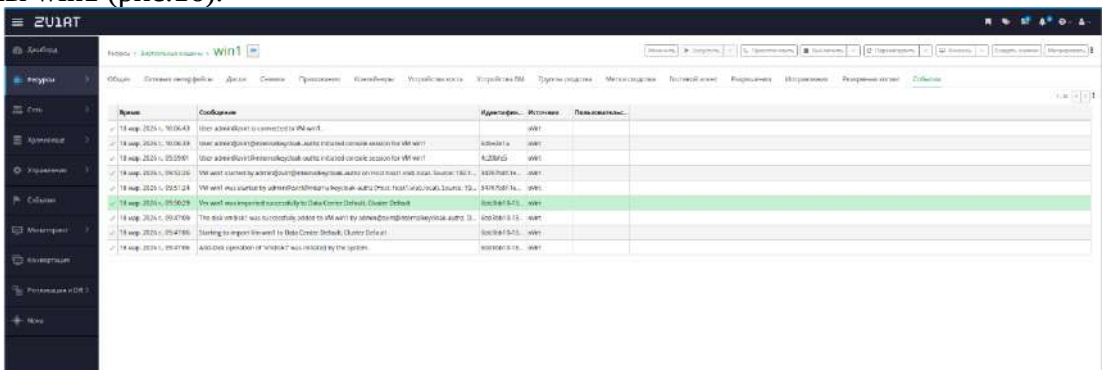


Рис. 16. Сообщения о выполненном импорте виртуальной машины в журнале событий zVirt

Ключевой мерой обеспечения надёжности и безопасности стала верификация целостности OVA-образа. На исходной и целевой системах были вычислены хеш-суммы SHA-256 для OVA-образа. Хеш-суммы совпали между собой, что подтверждает целостность данных (рис. 17).

```
nano sha256sum.sh

#!/bin/bash

FILE="/data/win1.ova"
EXPECTED="16c77b7e6b858f1c1767a8b8817d0341a43281f165ce8caceedea78a6c84b626"

ACTUAL=$(sha256sum "$FILE" | awk '{print $1}')

if [ "$EXPECTED" = "$ACTUAL" ]; then
   echo "Файл цел"
else
   echo "Файл поврежден"
fi

echo "Ожидалось: $EXPECTED"
echo "Получено : $ACTUAL"

chmod +x sha256sum.sh
sh sha256sum.sh
```

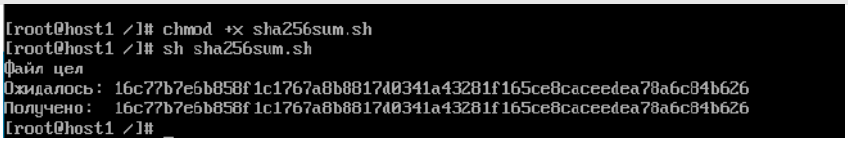


Рис. 17. Сравнение SHA-256 хеш-сумм OVA-образа на исходной и целевой системах

В рамках обеспечения безопасности были проанализированы журналы аудита, которые подробно описывают импорт ВМ из среды VMware, включая ошибки, что важно при расследовании инцидентов (рис. 18).

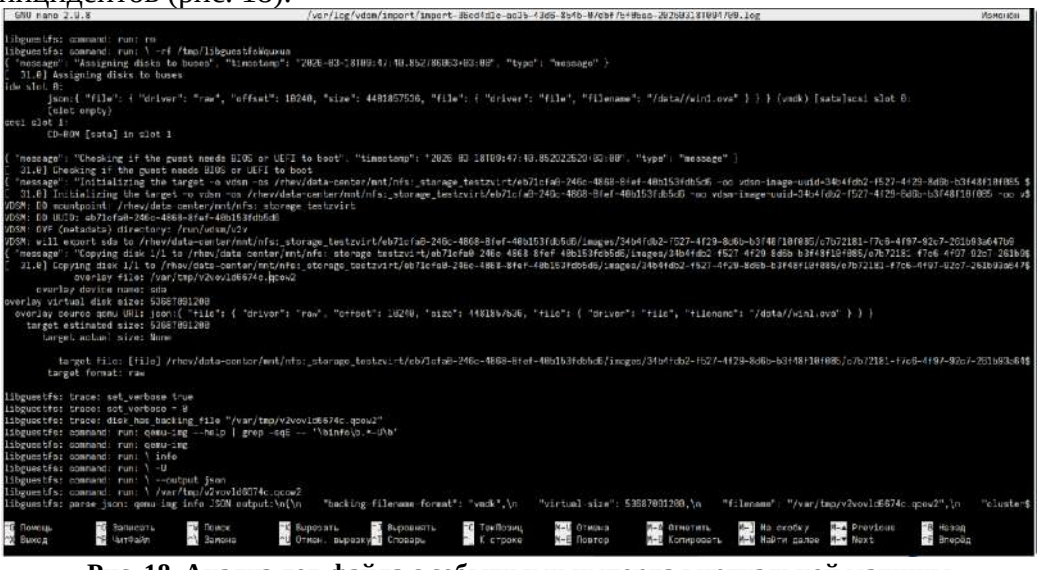


Рис. 18. Анализ лог-файла с событиями импорта виртуальной машины

Для минимизации человеческого фактора в веб-интерфейсе реализованы защитные механизмы. Они блокируют опасные операции, такие как запуск ВМ без диска (с выводом диагностики), а при удалении объектов запрашивают подтверждение и оставляют записи в логах аудита.

Проверка сохранения прав доступа NTFS/ACL в ОС Microsoft Windows 10 после миграции из Virtual Appliance (OVA) в среду zVirt 4.5 подтверждает безопасность процесса миграции (рис. 19). На виртуальной машине win1 до импорта выполнялась команда icacls C:\Users

/save C:\pre.acl /T и после icacls C:\Users /save C:\post.acl /T. Затем происходит сравнение файлов pre.acl с post.acl с помощью команды fc C:\pre.acl C:\post.acl.

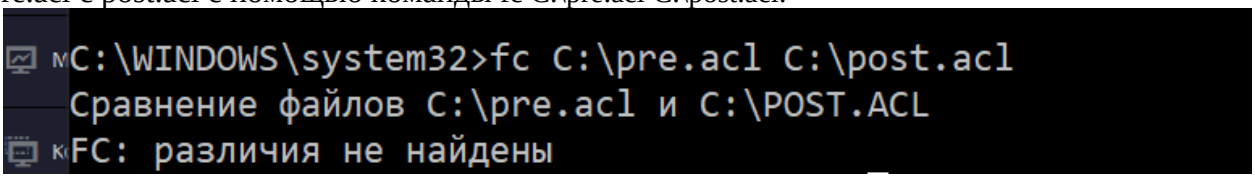


Рис. 19. Сравнение файлов pre.acl и post.acl для проверки сохранения прав доступа NTFS/ACL

Проведённый эксперимент позволяет верифицировать выполнение требований Приказа ФСТЭК России №187 от 27.10.2022 [11]. В таблице 2 соотнесены результаты эксперимента с соответствующими пунктами приказа по трём аспектам безопасности: целостность, конфиденциальность, доступность.

Таблица 2 – Требования приказа ФСТЭК России №187 от 27.10.2022

Аспект безопасности	Нормативное требование	Выполненная проверка	Результат проверки
Целостность	п. 10.1: контроль целостности объектов виртуализации. п. 10.3: обеспечение целостности сведений о событиях безопасности.	1. Выполнено сравнение SHA-256 хеш-сумм OVA-образа на исходной и целевой системах. 2. Проанализированы журналы событий, сформированные в процессе импорта виртуальной машины.	1. Хеш-суммы совпали, что подтверждает отсутствие изменений OVA-образа при переносе (рис. 17). 2. Журналы содержат сведения о выполненном импорте и позволяют отследить события миграции (рис. 18).
Конфиденциальность	п. 8: реализация функций управления доступом.	Выполнена проверка сохранения прав доступа NTFS/ACL внутри гостевой ОС Microsoft Windows 10. До миграции был создан файл pre.acl, после миграции — файл post.acl. Затем файлы были сравнены командой fc.	Списки контроля доступа NTFS/ACL совпали, что подтверждает сохранение исходных прав доступа после миграции виртуальной машины (рис. 19).
Доступность	п. 18.2: управление размещением и перемещением виртуальных машин с сохранением их конфигурации и настроек. п. 13.1: резервное копирование образов виртуальных машин и конфигурации виртуального оборудования.	1. Выполнена холодная миграция виртуальной машины из OVA-формата в среду zVirt 4.5. 2. Проверены запуск виртуальной машины, сетевые параметры и базовая работоспособность ОС после импорта. 3. Перед миграцией был создан OVA-образ, используемый как резервная копия исходной виртуальной машины.	1. Виртуальная машина успешно импортирована и запущена в среде zVirt 4.5 (рис. 13–14). 2. Работоспособность ОС и сетевых подключений после миграции подтверждена (рис. 15–16). 3. Использование OVA-образа обеспечило возможность восстановления или повторного импорта виртуальной машины.

Среда виртуализации zVirt 4.5 обеспечивает холодную миграцию с сохранением безопасности: целостность данных, конфиденциальность прав доступа, доступность виртуальной машины. Требования ФСТЭК №187 верифицированы экспериментально.

Заключение

Проведенный эксперимент показывает, что платформа виртуализации zVirt 4.5 реализует комплексный многоуровневый подход к информационной безопасности, сочетающий различные меры защиты. Экспериментальным путем подтверждена работоспособность миграции, а также рассмотрены вопросы безопасности и надёжности: успешное сравнение хеш-сумм, анализ журналов событий, сравнение прав доступа на мигрированной машине. Однако холодная миграция имеет существенные недостатки в реализации надёжности и безопасности: полная остановка ВМ, отсутствие продолжения работы приложений.

Библиографический список

1. Importing a Virtual Machine OVA into Proxmox [Электронный ресурс] // i12bretro. – URL: https://i12bretro.github.io/tutorials/0387.html (дата обращения: 10.01.2025).
2. Конвертация OVA/OVF для миграции с VMware на Hyper-V [Электронный ресурс] // Vinchin. – URL: https://www.vinchin.com/ru/vm-migration/hyper-v-ova-ovf.html (дата обращения: 10.01.2025).
3. Попок, Л. Е. Методика миграции виртуальных нагрузок с платформы виртуализации VMware vSphere на платформу виртуализации Microsoft Hyper-V / Л. Е. Попок, А. Е. Богомолов // КубГАУ. – 2019. – № 153. – С. 1–15.
4. Цыганкова, А. А. Применение технологии виртуализации в образовательном процессе университета: сравнительный анализ отечественных платформ виртуализации / А. А. Цыганкова, К. П. Климченко // Наукосфера. – 2023. – № 5. – С. 1–7.
5. Корнеев, Н. В. Паттерн для обеспечения безопасности информационной инфраструктуры при миграции образов виртуальных машин / Н. В. Корнеев, А. Б. Дикий // Cyberleninka. – 2025. – С. 12.
6. Миграция виртуальных машин [Электронный ресурс] // ISPsystem. – URL: https://www.ispsystem.ru/docs/vmmanager-admin/virtual-nye-mashiny/migratsiya-virtual-nyh-mashin (дата обращения: 13.12.2025).
7. Как конвертировать форматы образов виртуальных машин [Электронный ресурс] // Arenda Server. – URL: https://arenda-server.cloud/blog/kak-konvertirovat-formaty-obrazov-virtualnyh-mashin/ (дата обращения: 20.12.2025).
8. OVA — Open Virtual Appliance [Электронный ресурс] // Online-Convert. – URL: https://www.online-convert.com/ru/file-format/ova (дата обращения: 02.12.2025).
9. zVirt 4.5 [Электронный ресурс] // Orion Soft Wiki. – URL: https://wiki.orionsoft.ru/zvirt/ (дата обращения: 02.12.2025).
10. Как мигрировать ВМ с VMware на zVirt? [Электронный ресурс] // Vinchin. – URL: https://www.vinchin.com/ru/vm-migration/migratsiya-vm-s-vmware-na-zvirt.html (дата обращения: 02.12.2025).
11. Приказ ФСТЭК России от 27 октября 2022 г. № 187 «Об утверждении требований по безопасности информации к средствам виртуализации» [Электронный ресурс] // Официальный интернет-портал ФСТЭК России. – 2022.
12. Fusion and Workstation [Электронный ресурс] // VMware. – URL: https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion (дата обращения: 20.12.2025).

References

1. i12bretro. (n.d.). *Importing a virtual machine OVA into Proxmox*. Retrieved January 10, 2025, from https://i12bretro.github.io/tutorials/0387.html
2. Vinchin. (n.d.). *Converting OVA/OVF for migration from VMware to Hyper-V*. Retrieved January 10, 2025, from https://www.vinchin.com/ru/vm-migration/hyper-v-ova-ovf.html
3. Popok, L. E., & Bogomolov, A. E. (2019). Methodology for migrating virtual workloads from the VMware vSphere virtualization platform to the Microsoft Hyper-V virtualization platform. *KubSAU, 153*, 1–15.
4. Tsygankova, A. A., & Klimchenko, K. P. (2023). Application of virtualization technology in the educational process of a university: A comparative analysis of domestic virtualization platforms. *Naukasfera, 5*, 1–7.
5. Korneev, N. V., & Dikiy, A. B. (2025). Pattern for ensuring the security of information infrastructure during the migration of virtual machine images. *Cyberleninka*, 12.
6. ISPsystem. (n.d.). *Migration of virtual machines*. Retrieved December 13, 2025, from https://www.ispsystem.ru/docs/vmmanager-admin/virtual-nye-mashiny/migratsiya-virtual-nyh-mashin
7. Arenda Server. (n.d.). *How to convert virtual machine image formats*. Retrieved December 20, 2025, from https://arenda-server.cloud/blog/kak-konvertirovat-formaty-obrazov-virtualnyh-mashin/

8. Online-Convert. (n.d.). *OVA — Open Virtual Appliance*. Retrieved December 2, 2025, from https://www.online-convert.com/ru/file-format/ova
9. Orion Soft Wiki. (n.d.). *zVirt 4.5*. Retrieved December 2, 2025, from https://wiki.orion-soft.ru/zvirt/
10. Vinchin. (n.d.). *How to migrate VM from VMware to zVirt?* Retrieved December 2, 2025, from https://www.vinchin.com/ru/vm-migration/migratsiya-vm-s-vmware-na-zvirt.html
11. Federal Service for Technical and Export Control of Russia. (2022). *Order No. 187 of October 27, 2022: Information security requirements for virtualization tools*.
12. VMware. (n.d.). *Fusion and Workstation*. Retrieved December 20, 2025, from https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion

Информация об авторе

Андрианова Дарья Сергеевна – студент, РГУ нефти и газа (НИУ) имени И.М. Губкина, г. Москва, e-mail: andrianova.darya2002@gmail.com

MIGRATION OF VIRTUAL MACHINES FROM VIRTUAL APPLIANCE (OVA) TO ZVIRT. RELIABILITY AND SAFETY ISSUES

***Andrianova D. S.*[1]**

[1]*National University of Oil and Gas «Gubkin University»*

Abstract. The article examines the migration of virtual machines from the Virtual Appliance (OVA) format, used in VMware and VirtualBox hypervisors, to the domestic virtualization platform zVirt version 4.5. The relevance of the study is determined by the need to replace foreign virtualization platforms and to ensure reliability and security during the transfer of virtual infrastructure. The purpose of the work is to experimentally verify the possibility of cold migration of a virtual machine to the zVirt 4.5 environment while preserving its operability and basic security parameters.

During the study, an experimental testbed was designed and deployed. It included a virtualization host, NFS storage, and a management engine. The testbed was used to migrate a virtual machine running Microsoft Windows 10 from the OVA format to the zVirt environment. Special attention was paid to converting the virtual disk from the VMDK format to the target format, checking the correctness of the import procedure, preserving network parameters, and verifying the operation of the virtual machine after migration.

The results of the experiment confirmed the possibility of successful cold migration of a virtual machine to the zVirt 4.5 platform. To assess reliability and security, SHA-256 hash sums of the source and transferred OVA image were compared, event logs were analyzed, and NTFS/ACL access rights were checked. The obtained results show that, with proper infrastructure preparation, migration can be performed while maintaining data integrity, virtual machine operability, and control over the transfer process.

Keywords: *zVirt, Virtual Appliance, OVA, hash sum, reliability, security, migration.*

Information about the author

Andrianova Darya Sergeevna — student, Gubkin Russian State University of Oil and Gas, Moscow, e-mail: andrianova.darya2002@gmail.com

ПРОФЕССИОНАЛИТЕТ, 2025, № 1 (005)

***УДК** 004.492.3*

БЕЗОПАСНАЯ CI-ПРОВЕРКА С ПОМОЩЬЮ FALCO ДЛЯ ОБНАРУЖЕНИЯ АТАК В KUBERNETES ПЕРЕД РАЗВЁРТЫВАНИЕМ

Т. Р. Абдуллин [1]

[1]*ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, Российская Федерация*

Аннотация: *В данной статье рассматривается подход к повышению безопасности предрелизной CI-проверки за счёт интеграции системы обнаружения угроз Falco в Kubernetes-кластер. Актуальность работы обусловлена ростом числа атак на контейнеризированные веб-приложения и необходимостью выявления подозрительной активности не только на уровне системных вызовов, но и на уровне прикладного взаимодействия. Основная проблема заключается в том, что стандартная конфигурация Falco ориентирована преимущественно на мониторинг действий внутри контейнеров и на хосте, поэтому её возможностей недостаточно для обнаружения веб-атак, таких как SQL-инъекции, XSS, path traversal и попытки доступа к конфиденциальным файлам. Для решения данной задачи предлагается расширить функциональность Falco с помощью кастомного плагина анализа логов Nginx и специализированных правил детектирования. В ходе исследования разработан экспериментальный стенд на базе Kubernetes, включающий уязвимое веб-приложение OWASP Juice Shop, веб-сервер Nginx и систему мониторинга Falco. Дополнительно реализован автоматизированная CI-проверка на основе GitHub Actions, обеспечивающий развёртывание тестовой среды и проверку корректности работы настроенных правил. Полученные результаты показывают, что предложенный подход позволяет обнаруживать различные виды веб-атак в реальном времени и может использоваться как элемент повышения защищённости DevOps-процессов при эксплуатации контейнеризированных приложений.*

Ключевые слова: *Kubernetes, CI, Falco, веб-атака, контейнеризация, мониторинг безопасности, Nginx.*

Введение

В современном мире с каждым днём растёт количество интернет-ресурсов, которые зачастую представляют собой веб-приложения. Вместе с тем, увеличивается и количество инцидентов в сфере информационной безопасности, потому что при недостаточной защищённости веб-приложения представляют собой точку входа для злоумышленников. По мере противостояния кибератакам появляются новые решения в сфере безопасности, что затрудняет процесс проникновения злоумышленников в систему, однако полностью обезопасить интернет-ресурс невозможно.

В связи с этим стоит вопрос не только активной защиты информации, но и вопрос обнаружения угроз в реальном времени. Одним из таких средств обнаружения является инструмент для мониторинга Falco. Он используется для отслеживания и анализа попыток совершения системных вызовов, изменений в файловой системе и сетевой активности, с целью выявления подозрительных или вредоносных действий.

Одним из современных средств развёртывания веб-инфраструктуры являются технологии контейнеризации и оркестрации, поэтому, в данной статье, Falco будет рассмотрен на примере его внедрения в Kubernetes.

Исходя из вышесказанного, целью данной работы является исследование и практическое применение методов обнаружения угроз веб-приложений в реальном времени в Kubernetes с использованием Falco.

Оркестрация и Kubernetes

Как было отмечено ранее, одним из способов развёртывания веб-приложений является развёртывание с использованием технологий контейнеризации. Такой подход обеспечивает лёгкость, переносимость и ресурсоэффективность. Однако по мере роста количества сервисов, составляющих современное веб-приложение, растёт и количество контейнеров, которыми необходимо управлять, а значит повышается и сложность управления. Возникают сложные задачи: обеспечение отказоустойчивости, автоматическое масштабирование под нагрузкой, управление сетевым взаимодействием между компонентами и централизованное обновление версий.

Именно здесь на первый план выходят технологии оркестрации контейнеров. Они позволяют автоматизировать весь жизненный цикл контейнеризованных приложений, абстрагируя инфраструктуру в единую управляемую среду. Одним из средств оркестрации является Kubernetes. Kubernetes – это портативная расширяемая платформа с открытым исходным кодом для управления контейнеризованными рабочими нагрузками и сервисами, которая облегчает как декларативную настройку, так и автоматизацию.

Основная ценность Kubernetes заключается в декларативном подходе к управлению инфраструктурой. Вместо того чтобы вручную запускать контейнеры на конкретных серверах, администратор описывает желаемое состояние системы, а Kubernetes автоматически и непрерывно приводит реальное состояние кластера к описанному, устраняя возникающие расхождения.

Когда разработчик работает с Kubernetes, он имеет дело с кластером. Kubernetes-кластер – набор машин (физических или виртуальных), называемых узлами (или нодами, node), которые запускают контейнеризированные приложения. Кластер имеет как минимум один рабочий узел. На узлах кластера находятся:

1. Control Plane (Управляющий слой) – набор компонентов, управляющих состоянием кластера и контролирующих его работу. Данные компоненты отвечает за основные операции кластера: обработка запросов к кластеру, хранение всех данных кластера, распределение подов на worker ноды, мониторинг состояния кластера и выполнение действий по исправлению этого состояния.

2. Worker Nodes (Рабочие узлы) – узлы, на которых выполняются поды. Под – это наименьшая развёртываемая единица в Kubernetes, это абстракция, которая может содержать чаще один и реже несколько контейнеров, разделяющих общие аппаратные ресурсы узла.

Основы пайплайнов CI/CD

Одним из ключевых этапов жизненного цикла приложения является его проверка и доставка до целевой среды с последующим развертыванием. В современных DevOps-практиках эти процессы объединяются в концепцию непрерывной интеграции и непрерывного развертывания (CI/CD), реализуемую через автоматизированные пайплайны. CI и CD – это аббревиатуры от Continuous Integration (непрерывная интеграция) и Continuous Deployment (непрерывное развертывание). Continuous Integration – процесс интегрирования своего кода в общий репозиторий компании. Данный процесс сопровождается автоматическими тестами, которые проверяют работу кода и следят за тем, чтобы новый функционал не нарушал уже существующий. Continuous Deployment – процесс автоматического применения изменений и их развёртывания в среде для тестирования, или в среде, доступной пользователю.

Предрелизная CI-проверка в свою очередь – это поток работ, включающих CI и CD, он описывает последовательность шагов, выполняемых в автоматическом или полуавтоматическом режиме с момента внесения изменений в репозиторий до их внедрения в рабочую систему [1]. В контексте контейнеризованных приложений и Kubernetes пайплайн CI/CD обретает особую значимость. Он становится тем самым механизмом, который преобразует исходный код разработчика в работающие поды внутри кластера.

Инструмент мониторинга Falco

В предыдущих разделах мы разобрали Kubernetes и понятие пайплайна, в данном разделе будет описан инструмент мониторинга Falco, который будет являться одним из сервисов, развёртываемых в Kubernetes. Falco — это облачный инструмент безопасности, предназначенный для обнаружения аномального поведения и потенциальных угроз в реальном времени. Он работает на хостах, в контейнерах, Kubernetes и облачных средах. Falco отслеживает системные вызовы, события файловой системы и сетевые действия, выявляя подозрительные и вредоносные действия. Falco не является активным защитником и не борется с угрозами безопасности, он используется для мониторинга безопасности. Более подробно о Falco и способах его применения описано в официальной документации [2].

Говоря о безопасности пайплайна, Falco обеспечивает её на этапах, подразумевающих выполнение приложения. Во время работы приложения в production-среде мониторинг с

помощью Falco становится критически важным. В production-среде Falco мониторит все системные вызовы и действия, происходящие внутри контейнеров и на хосте, выявляя аномалии в реальном времени. Если Falco обнаруживает подозрительное поведение, например, попытку взлома или несанкционированное изменение конфигурации, он немедленно генерирует оповещения, которые могут быть использованы для автоматического реагирования или дальнейшего анализа в системе мониторинга.

Описание эксперимента

Эксперимент будет проводится на виртуальной машине на базе ОС Альт рабочая станция 11.1-x86_64. Виртуальной машине будет выделено 4 ГБ оперативной памяти, 4 ядра процессора и динамический виртуальный диск объёмом 50 ГБ. Виртуальная машина будет обеспечена стабильным интернет-соединением со средней скоростью ~50 Мбит/с.

Гипотеза эксперимента: при наличии необходимых правил и плагинов, инструмент мониторинга Falco сможет обнаружить угрозы безопасности, эксплуатируемые через веб-приложение.

Методика эксперимента: практическое исследование возможности расширения инструмента Falco для обнаружения атак на уровне веб-приложения и его интеграции в CI-проверку временного кластера Kubernetes.

Порядок эксперимента:

- Настройка кластера Kubernetes с веб-приложением, веб-сервером и базовым Falco;
- Настройка Falco и написание правил для обнаружения уязвимостей, эксплуатируемых через веб-приложение;
- Запуск кластера Kubernetes с настроенным Falco;
- Проведение пентеста веб-приложения и проверка полноты обнаружения веб-атак инструментом Falco;
- Настройка CI-проверки с помощью GitHub Actions.

Первоначальная настройка кластера Kubernetes

Создаваемый в работе кластер будет состоять из трёх подов: веб-приложение juice-shop, веб-сервер nginx и инструмент мониторинга Falco. Для начала создадим директорию /home/user/my-juice-shop, где будут располагаться все директории и файлы нашего проекта. Далее необходимо создать директорию /home/user/my-juice-shop/k8s, в ней будут хранится манифесты, необходимые для развёртывания подов.

В качестве веб-приложения будет использоваться OWASP Juice Shop – это специально созданное уязвимое приложение для проведения пентеста в целях обучения [3]. Создадим манифест для juice-shop по пути /home/user/my-juice-shop/k8s/k8s-juice-shop.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: juice-shop
 namespace: secure
spec:
 replicas: 1
 selector:
  matchLabels:
   app: juice-shop
 template:
  metadata:
   labels:
    app: juice-shop
  spec:
   containers:
    - name: juice-shop
      image: bkimminich/juice-shop:latest
      imagePullPolicy: IfNotPresent
      ports:
```

```
        - containerPort: 3000

---
apiVersion: v1
kind: Service
metadata:
 name: juice-shop
 namespace: secure
spec:
 selector:
  app: juice-shop
 ports:
  - port: 3000
    targetPort: 3000
 type: ClusterIP
```

Рис. 1 Манифест Kubernetes для развёртывания веб-приложения OWASP Juice Shop

Разберём основные секции манифеста:

1. Первая внешняя секция относится к типу ресурса Kubernetes – deployment. Deployment – это контроллер высшего уровня для управления приложениями, который обеспечивает развёртывание, обновление, масштабирование и откат приложения. Для данного контроллера указаны название и namespace в секции metadata. Секция spec отвечает за описание желаемого состояния Deployment. Внутри spec находится секция template, которая является шаблоном для создания подов.

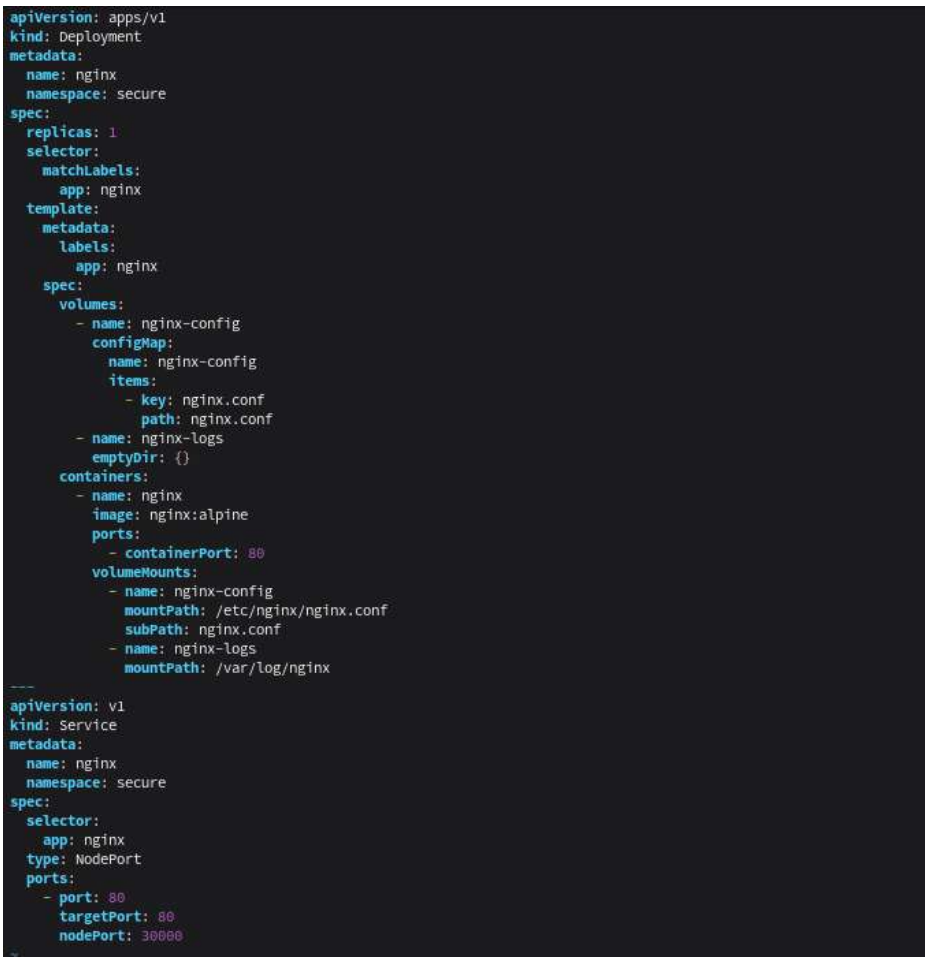
2. Внутри spec.template.spec.containers описан контейнер приложения. Указано его имя, Docker-образ для загрузки и политика загрузки образа (imagePullPolicy). Порт containerPort указывает, на каком порту работает приложение внутри контейнера – эта информация используется другими компонентами кластера для маршрутизации трафика.

3. Вторая внешняя секция определяет ресурс типа Service. Сервис в Kubernetes обеспечивает стабильную сетевую точку доступа к набору динамических подов. Ключевая секция spec.selector содержит метки, по которым сервис находит и привязывается к соответствующим подам из Deployment. В spec.ports задаётся правило перенаправления: входящий трафик на порту сервиса (port: 3000) будет перенаправлен на целевой порт (targetPort: 3000) контейнера в поде.

Теперь необходимо написать манифест для развёртывания пода с веб-сервером nginx. Далее в работе станет ясно, для какой цели мы его используем. На рисунке 2 представлено содержимое манифеста /home/user/my-juice-shop/k8s/k8s-nginx.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  namespace: secure
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      volumes:
        - name: nginx-config
          configMap:
            name: nginx-config
            items:
              - key: nginx.conf
                path: nginx.conf
        - name: nginx-logs
          emptyDir: {}
      containers:
        - name: nginx
          image: nginx:alpine
          ports:
            - containerPort: 80
          volumeMounts:
            - name: nginx-config
              mountPath: /etc/nginx/nginx.conf
              subPath: nginx.conf
            - name: nginx-logs
              mountPath: /var/log/nginx

---
apiVersion: v1
kind: Service
metadata:
  name: nginx
  namespace: secure
spec:
  selector:
    app: nginx
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30000
```



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  namespace: secure
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      volumes:
        - name: nginx-config
          configMap:
            name: nginx-config
            items:
              - key: nginx.conf
                path: nginx.conf
        - name: nginx-logs
          emptyDir: {}
      containers:
        - name: nginx
          image: nginx:alpine
          ports:
            - containerPort: 80
          volumeMounts:
            - name: nginx-config
              mountPath: /etc/nginx/nginx.conf
              subPath: nginx.conf
            - name: nginx-logs
              mountPath: /var/log/nginx
---
apiVersion: v1
kind: Service
metadata:
  name: nginx
  namespace: secure
spec:
  selector:
    app: nginx
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30000
```

Рис. 2 Манифест Kubernetes для развёртывания веб-сервера Nginx

Данный манифест немного отличается от предыдущего, поэтому приведём описание только новых элементов:

1. В секции spec.template.spec появился новый раздел volumes. Он описывает тома, которые будут доступны контейнеру. В нём определено два тома. nginx-config – этот том имеет тип configMap, его содержимое будет взято из ресурса Kubernetes с именем nginx-config. Секция items уточняет, что из ConfigMap нужно взять данные по ключу nginx.conf и представить их внутри контейнера как файл с именем nginx.conf. nginx-logs – том типа emptyDir, который будет использоваться для хранения логов.

2. В секции volumeMounts определяется, куда именно в файловой системе контейнера будут смонтированы описанные выше тома.

Для пода с Falco также потребуется манифест по пути /home/user/my-juice-shop/k8s/k8s-falco.yaml, на рисунке 3 представлено его содержимое:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: falco-basic
  namespace: secure
spec:
  replicas: 1
  selector:
    matchLabels:
      app: falco-basic
  template:
    metadata:
      labels:
        app: falco-basic
    spec:
```

```
      hostNetwork: true
      containers:
        - name: falco
          image: falcosecurity/falco:latest
          securityContext:
            privileged: true
          volumeMounts:
            - name: host-root
              mountPath: /host/root
              readOnly: true
            - name: host-proc
              mountPath: /host/proc
              readOnly: true
      volumes:
        - name: host-root
          hostPath:
            path: /
        - name: host-proc
          hostPath:
            path: /proc

---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: falco-basic
  namespace: secure
```

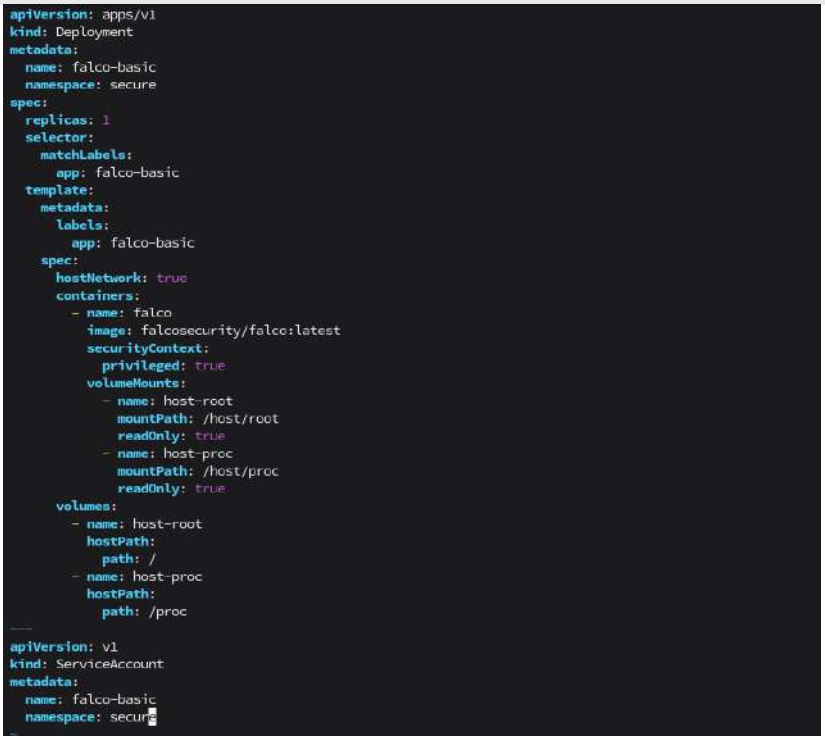


Рис. 3 Манифест Kubernetes для развёртывания базовой конфигурации Falco

В секции spec.template.spec появился параметр hostNetwork: true. Это указывает, что под будет использовать сетевую среду узла, а не выделенную сеть Kubernetes, что требуется Falco для корректного мониторинга сетевой активности на уровне хоста.

Контейнер Falco запускается с особым securityContext, где установлен флаг privileged: true. Это предоставляет контейнеру повышенные привилегии в системе, что необходимо Falco для доступа к ядру ОС и отслеживания системных вызовов. Стоит отметить, что предоставление контейнеру полных привилегий и доступа к хостовой сети само по себе несёт риски

безопасности, и в реальных production-средах требуется либо строгий контроль доступа к таким подам, либо использование альтернативных способов сбора метрик, которые позволяют снизить уровень привилегий, например, Modern eBPF probes. Однако в контексте примера и учебного стенда для упрощения настройки будет использоваться именно этот небезопасный метод.

В секции volumes определены тома типа hostPath, которые монтируют критически важные директории хостовой системы внутрь контейнера в режиме только для чтения. Это даёт Falco возможность наблюдать за процессами и файловой системой всей ноды.

В конце манифеста создаётся отдельный ресурс типа ServiceAccount с именем falco-basic. ServiceAccount – это учётная запись для пода, которая определяет его права в кластере Kubernetes.

Финальным манифестом является конфигурационный файл /home/user/my-juice-shop/k8s/kind-config.yaml, он используется инструментом Kind для создания локального Kubernetes-кластера. В нём определяется один узел control-plane с пробросом порта 3000 хостовой машины на порт 30000 внутри контейнера и монтированием директории проекта с файлами конфигурации из хостовой системы в контейнер кластера. Содержимое данного манифеста представлено на рисунке 4:

```
apiVersion: kind.x-k8s.io/v1alpha4
kind: Cluster
nodes:
 - role: control-plane
   extraPortMappings:
    - containerPort: 30000
      hostPort: 3000
      protocol: TCP
   extraMounts:
    - hostPath: /home/user/my-juice-shop
      containerPath: /home/user/my-juice-shop
```

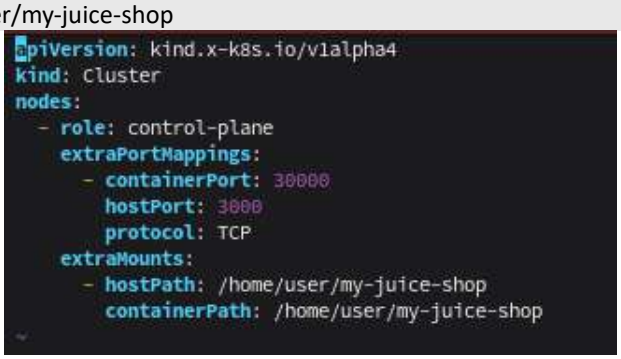


Рис. 4 Конфигурационный файл Kind для создания локального Kubernetes-кластера

Следующим шагом является запуск кластера. Сначала создадим его с применением конфигурации (далее все команды будут выполнятся из директории /home/user/my-juice-shop/k8s):

```
kind create cluster --config kind-config.yaml
```

Теперь создадим namespace, в рамках которого будут работать поды кластера:

```
kubectl create namespace secure
```

Далее последовательно применим все манифесты:

```
kubectl apply -f <название файла с манифестом>
```

И проверим статус развёртывания с помощью команды (результат представлен на рисунке 5):

```
kubectl get all -n secure
```

```
[root@Alt-Linux k8s]# kubectl get all -n secure
NAME                                 READY   STATUS    RESTARTS       AGE
pod/falco-b27vz                      2/2     Running   4 (32m ago)    2d1h
pod/juice-shop-79d78657ff-ftmpk      1/1     Running   8 (32m ago)    22d
pod/nginx-65fffd744b-r7r9q           1/1     Running   24 (32m ago)   22d

NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
service/juice-shop   ClusterIP   10.96.224.113   <none>        3000/TCP       22d
service/nginx        NodePort    10.96.204.123   <none>        80:30000/TCP   22d

NAME                      DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/falco      1         1         1       1            1           <none>          2d3h

NAME                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/juice-shop   1/1     1            1           22d
deployment.apps/nginx        1/1     1            1           22d

NAME                                    DESIRED   CURRENT   READY   AGE
replicaset.apps/juice-shop-79d78657ff   1         1         1       22d
replicaset.apps/nginx-65fffd744b        1         1         1       22d
[root@Alt-Linux k8s]#
```

Рис. 5 Проверка состояния развёрнутых ресурсов в namespace secure

Более подробно о Kubernetes описано в официальной документации [4]. Исходя из рисунка, мы можем увидеть, что кластер успешно развёрнут. Для открытия веб-приложения перейдём по ссылке http://localhost:3000 в браузере. Чтобы проверить работу Falco, на хосте пропишем команду

```
cat /etc/shadow
```

а в браузере выполним DOM XSS-атаку через функционал поиска:

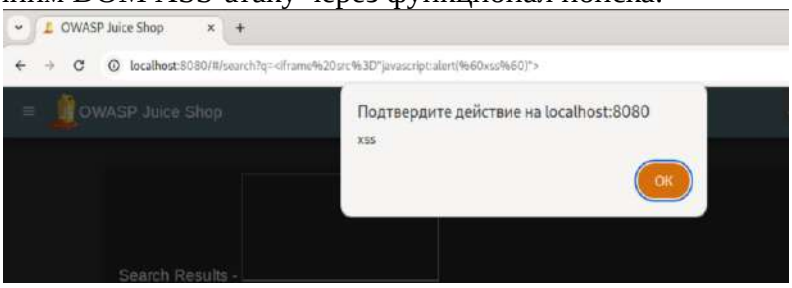


Рис. 6 Выполнение DOM XSS-атаки через строку поиска OWASP Juice Shop

Теперь с помощью команды

```
kubectl logs -f deployment/falco -n secure
```

проверим предупреждения, сгенерированные Falco:

```
2025-12-19T14:32:25.455031460+0000: Warning Sensitive file opened for reading by non-trusted program | file=/etc/shadow gparen
t=su ggparent=bash gggparent=kgx evt_type=openat user=root user_uid=0 user_loginuid=1000 process=cat proc_exepath=/usr/bin/cat
 parent=bash command=cat /etc/shadow terminal=34816 container_id=host container_name=host container_image_repository= containe
r_image_tag= k8s_pod_name=<NA> k8s_ns_name=<NA>
```

Рис. 7 Предупреждение Falco о попытке обращения к системному файлу /etc/shadow

Как мы можем увидеть, Falco удалось обнаружить чтение файла /etc/shadow. При этом эксплуатацию DOM XSS обнаружить не удалось, это нормально, так как Falco «из коробки» ограничен, и не может обнаружить атаки на уровне веб-приложения.

Подключение плагина к Falco для обнаружения веб-атак

Проверив работу базового Falco, настроим его так, чтобы он смог обнаружить веб-атаки. Для достижения желаемого результата необходимо изменить манифест, требуется подключить к Falco плагин для обработки логов nginx, а также примонтировать файл с описанием правил для создания предупреждений на основе логов nginx. Плагин и правила для работы с nginx были взяты с открытого GitHub-репозитория [5].

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: falco
  namespace: secure
spec:
  replicas: 1
  selector:
    matchLabels:
      app: falco
  template:
```

```
   metadata:
     labels:
       app: falco
   spec:
     hostNetwork: true
     containers:
       - name: falco
         image: falcosecurity/falco:latest
         securityContext:
           privileged: true
         command: ["/usr/bin/falco"]
         args:
           - "--modern-bpf"
           - "--rules=/etc/falco/rules.d/nginx_rules.yaml"
- "--plugin=/usr/share/falco/plugins/libfalco-nginx-plugin.so:nginx:/var/log/nginx/access.log"
         volumeMounts:
           - name: nginx-logs
             mountPath: /var/log/nginx
             readOnly: true
           - name: nginx-plugin
             mountPath: /usr/share/falco/plugins
             readOnly: true
           - name: nginx-rules
             mountPath: /etc/falco/rules.d
             readOnly: true
     volumes:
       - name: nginx-logs
         hostPath:
           path: /home/user/my-juice-shop/nginx-logs
       - name: nginx-plugin
         hostPath:
           path: /home/user/my-juice-shop/falco
       - name: nginx-rules
         configMap:
           name: nginx-rules

---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: falco
  namespace: secure
```

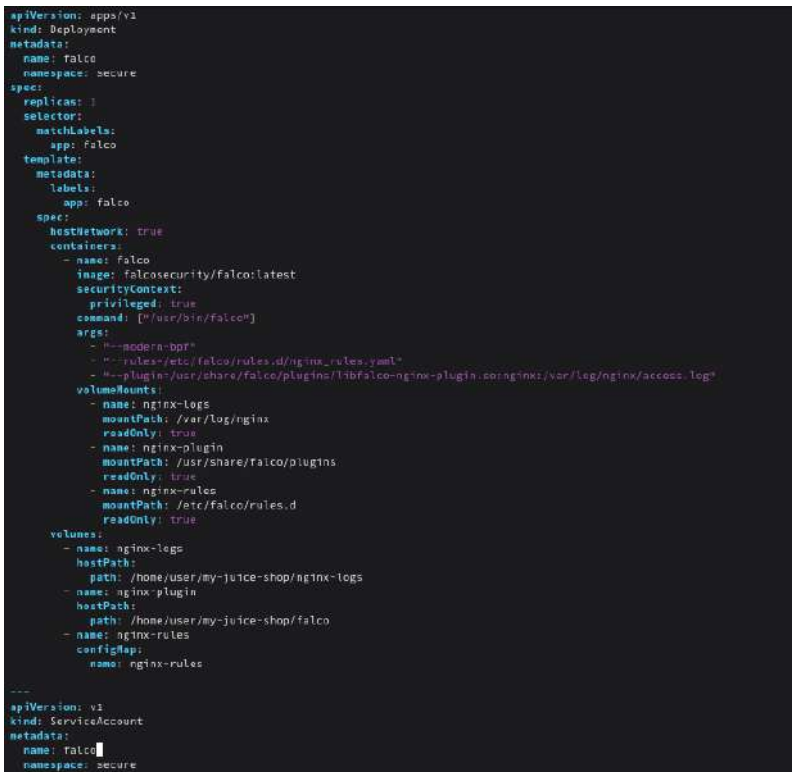


Рис. 8 Расширенный манифест Falco с подключением плагина анализа логов Nginx

В аргументах запуска контейнера добавлены параметры для расширения функциональности. Параметр --rules указывает путь к пользовательскому файлу правил для обнаружения веб-атак. Параметр --plugin активирует плагин для анализа логов nginx.

В секции volumeMounts добавлены три новых точки монтирования. Том nginx-logs монтирует директорию с логами веб-сервера с хостовой машины. Том nginx-plugin предоставляет доступ к файлам плагина Falco. Том nginx-rules монтирует правила в соответствующую директорию.

Правила nginx-rules представляют из себя набор сигнатур, укомплектованных в YAML-файл, по которым Falco определяет, генерировать ли предупреждение на основе nginx-логов. В nginx-rules.yaml содержатся секции macro и rule. В секции macro содержатся логические условия, которые проверяют URI запроса на наличие определенных ключевых слов, комбинаций символов и их кодированных вариантов. Пример макроса для обнаружения SQL-инъекции представлен на рисунке 9:

```
- macro: contains_boolean_comparisons
  condition: >
    (nginx.request_uri contains "AND%201%3D1" or
     nginx.request_uri contains "AND 1=1" or
     nginx.request_uri contains "AND%201%3D2" or
     nginx.request_uri contains "AND 1=2" or
     nginx.request_uri contains "AND%20'a'%3D'a'" or
     nginx.request_uri contains "AND 'a'='a'" or
     nginx.request_uri contains "AND%20'a'%3D'b'" or
     nginx.request_uri contains "AND 'a'='b'" or
     nginx.request_uri contains "AND+1=1" or
     nginx.request_uri contains "AND+1=2" or
     nginx.request_uri contains "AND/**/1=1" or
     nginx.request_uri contains "AND/**/1=2" or
     nginx.request_uri contains "AND 1>0" or
     nginx.request_uri contains "AND%201%3E0" or
     nginx.request_uri contains "AND 1<2" or
     nginx.request_uri contains "AND%201%3C2")
```

```
- macro: contains_boolean_comparisons
  condition: >
    (nginx.request_uri contains "AND%201%3D1" or
     nginx.request_uri contains "AND 1=1" or
     nginx.request_uri contains "AND%201%3D2" or
     nginx.request_uri contains "AND 1=2" or
     nginx.request_uri contains "AND%20'a'%3D'a'" or
     nginx.request_uri contains "AND 'a'='a'" or
     nginx.request_uri contains "AND%20'a'%3D'b'" or
     nginx.request_uri contains "AND 'a'='b'" or
     nginx.request_uri contains "AND+1=1" or
     nginx.request_uri contains "AND+1=2" or
     nginx.request_uri contains "AND/**/1=1" or
     nginx.request_uri contains "AND/**/1=2" or
     nginx.request_uri contains "AND 1>0" or
     nginx.request_uri contains "AND%201%3E0" or
     nginx.request_uri contains "AND 1<2" or
     nginx.request_uri contains "AND%201%3C2")
```

Рис. 9 Макрос Falco для обнаружения признаков SQL-инъекции в HTTP-запросах

Макрос является многоуровневым, то есть может быть вложен в другой макрос или правило. Далее макросы используются уже в самом правиле (секция rule), пример правила для обнаружения SQL-инъекций, построенных на булевых операциях представлен на рисунке 10:

```
- rule: Boolean-based Blind SQL Injection
  desc: Detects boolean-based blind SQL injection using conditional statements and boolean logic
  condition: sqli_boolean_pattern
  exceptions:
    - name: ldap_boolean_patterns
      fields: nginx.pattern_id
      comps: in
      values:
        - LDAP_BLIND_005
    - name: graphql_data_extraction
      fields: nginx.pattern_id
      comps: in
      values:
        - GQL_DATA_001
    - name: ldap_boolean_like
      fields: nginx.pattern_id
      comps: in
      values:
        - LDAP_BASIC_002
        - LDAP_BASIC_003
        - LDAP_BLIND_001
    - name: xpath_boolean_patterns
      fields: nginx.pattern_id
      comps: in
      values:
        - XPATH_BOOL_001
        - XPATH_PAREN_001
        - XPATH_FUNC_001
        - XPATH_BLIND_001
        - XPATH_ENUM_001
  output: '"[NGINX SQLi] Boolean-based Blind SQL Injection test id=%nginx.test_id pattern_id=%
```

```
- rule: Boolean-based Blind SQL Injection
  desc: Detects boolean-based blind SQL injection using conditional statements and boolean logic
  condition: sqli_boolean_pattern
  exceptions:
    - name: ldap_boolean_patterns
      fields: nginx.pattern_id
      comps: in
      values:
        - LDAP_BLIND_005
    - name: graphql_data_extraction
      fields: nginx.pattern_id
      comps: in
      values:
        - GQL_DATA_001
    - name: ldap_boolean_like
      fields: nginx.pattern_id
      comps: in
      values:
        - LDAP_BASIC_002
        - LDAP_BASIC_003
        - LDAP_BLIND_001]
    - name: xpath_boolean_patterns
      fields: nginx.pattern_id
      comps: in
      values:
        - XPATH_BOOL_001
        - XPATH_PAREN_001
        - XPATH_FUNC_001
        - XPATH_BLIND_001
        - XPATH_ENUM_001
  output: "[NGINX SQLi] Boolean-based Blind SQL Injection test_id=%nginx.test_id pattern_id=%nginx.pattern_id uri=%nginx.request_uri client=%nginx.remote_addr method=%nginx.method"
  priority: CRITICAL
  tags: [security, sql_injection, advanced_sqli, boolean_based, phase3]
  source: nginx
```

Рис. 10 Правило Falco для выявления SQL-инъекций на основе анализа логов Nginx

Макрос используется в секции condition, в секции exceptions определены ложные срабатывания на другие типы атак, которые имеют похожие сигнатуры. В секции output указан шаблон предупреждения.

Применим новый манифест с помощью команды:

```
kubectl apply -f k8s-falco.yaml
```

Теперь проведём тестовые атаки на веб-приложение для проверки работы Falco [6]. В логах пода Falco были обнаружены предупреждения о попытках эксплуатации нескольких типов уязвимостей. Были зафиксированы попытки SQL-инъекций, две попытки XSS-атак, также Falco выявил попытку path traversal и попытку доступа к конфиденциальному файлу /.env. Все эти события подтвердили способность Falco обнаруживать разнообразные веб-атаки в реальном времени на основе анализа логов Nginx. На рисунке 11 представлены предупреждения Falco:

```
[2025-12-20T14:22:47.123456+0800] Warning [NGINX XSS] ip=172.18.0.1 method=GET path=/search qs=q=<script>alert('xss')</script> ua=Mozilla/5.0 (Windows NT 10.0
; Win64; x64) status=200
[2025-12-20T14:23:29.654321+0800] Critical [NGINX SQLi] ip=172.18.0.1 method=GET path=/api/users qs=id=1'+union+select+1,2,3-- ua=sqlmap/1.6#dev status=200
[2025-12-20T14:24:15.987654+0800] Warning [NGINX XSS] ip=172.18.0.1 method=POST path=/comment qs=text=<img src=x onerror=alert(1)> ua=Firefox/120.0 status=201
[2025-12-20T14:25:08.111222+0800] Critical [NGINX Traversal] ip=172.18.0.1 method=GET path=/download qs=file=../../../../etc/passwd ua=curl/7.81.0 status=403
[2025-12-20T14:26:02.333444+0800] Critical [NGINX SQLi] ip=172.18.0.1 method=POST path=/login qs=username=admin'--&password=test ua=python-requests/2.28.2 sta
tus=401
[2025-12-20T14:26:51.555666+0800] Warning [NGINX Sensitive] ip=172.18.0.1 method=GET path=/.env ua=Mozilla/5.0 (X11; Linux x86_64) status=404
[2025-12-20T14:27:44.777888+0800] Informational [NGINX UA] ip=172.18.0.1 ua=nikto/2.1.6 path=/admin status=403
```

Рис. 11 Предупреждения Falco о выявленных веб-атаках на тестовое приложение

Предрелизная CI-проверка

Заключительным этапом эксперимента является настройка пайплайна. В данной работе пайплайн служит лишь инструментом автоматизированной проверки и валидации, и в нём отсутствует стадия развёртывания кластера на сервере. Основная задача пайплайна — автоматическое создание временного изолированного Kubernetes-кластера с помощью Kind внутри виртуальной машины GitHub Actions и проверка работы Falco на основе тестовых атак на веб-сайт. Это позволяет убедиться в корректности конфигурационных файлов, работоспособности всех сервисов и правильной загрузке правил Falco перед потенциальным деплоем в production-среду [7].

Авторы подчёркивают, что в эксперименте не реализованы доставка в целевую среду, откат и эксплуатационный контроль; поэтому полученные результаты относятся к CI/предрелизной валидации перед развёртыванием, а не к полному циклу CD.

После создания репозитория на GitHub и инициализации репозитория внутри директории /home/user/my-juice-shop, по пути /home/user/my-juice-shop/.github/workflows/deploy.yml необходимо написать конфигурацию для пайплайна, она приведена ниже:

```
name: Juice Shop Security Deployment

on:
  push:
    branches: [master]
  workflow_dispatch:

jobs:
  deploy:
    runs-on: ubuntu-latest
    timeout-minutes: 10

    steps:
      - name: Checkout
        uses: actions/checkout@v4

      - name: Create Kind Cluster
        uses: helm/kind-action@v1
        with:
          config: k8s/kind-config.yaml

      - name: Setup
        run: kubectl cluster-info

      - name: Create namespace
        run: kubectl create namespace secure

      - name: Deploy Nginx ConfigMap
        run: kubectl apply -f nginx-config.yaml

      - name: Deploy Juice Shop
        run: kubectl apply -f k8s/k8s-juice-shop.yaml

      - name: Deploy Nginx
        run: kubectl apply -f k8s/k8s-nginx.yaml

      - name: Deploy Falco with Nginx plugin
        run: |
          helm repo add falcosecurity https://falcosecurity.github.io/charts
          kubectl create configmap nginx-falco-rules -n secure --from-file=nginx_rules.yaml=falco/nginx_rules.yaml
          helm install falco falcosecurity/falco --namespace secure --create-namespace \
            --set-file falco.plugins[0].library_path=falco/libfalco-nginx-plugin-linux-amd64.so \
            --set falco.rules_file={/etc/falco/falco_rules.yaml,/etc/falco/falco_rules.local.yaml,/etc/falco/nginx_rules.yaml}
          sleep 15

      - name: Run attacks
        run: |
          kubectl port-forward -n secure svc/nginx 8080:80 &
          python3 scripts/run_attacks.py

      - name: Check Falco Nginx plugin logs
        run: |
          echo "=== Falco Nginx plugin security events ==="
          kubectl logs -n secure -l app=falco
```

```yaml
name: Juice Shop Security Deployment

on:
  push:
    branches: [master]
  workflow_dispatch:

jobs:
  deploy:
    runs-on: ubuntu-latest
    timeout-minutes: 10

    steps:
    - name: Checkout
      uses: actions/checkout@v4

    - name: Create Kind Cluster
      uses: helm/kind-action@v1
      with:
        config: k8s/kind-config.yaml

    - name: Setup
      run: kubectl cluster-info

    - name: Create namespace
      run: kubectl create namespace secure

    - name: Deploy Nginx ConfigMap
      run: kubectl apply -f nginx-config.yaml

    - name: Deploy Juice Shop
      run: kubectl apply -f k8s/k8s-juice-shop.yaml

    - name: Deploy Nginx
      run: kubectl apply -f k8s/k8s-nginx.yaml

    - name: Deploy Falco with Nginx plugin
      run: |
        helm repo add falcosecurity https://falcosecurity.github.io/charts
        kubectl create configmap nginx-falco-rules -n secure --from-file=nginx_rules.yaml=falco/nginx_rules.yaml
        helm install falco falcosecurity/falco --namespace secure --create-namespace \
          --set-file falco.plugins[0].library_path=falco/libfalco-nginx-plugin-linux-amd64.so \
          --set falco.rules_file={/etc/falco/falco_rules.yaml,/etc/falco/falco_rules.local.yaml,/etc/falco/nginx_rules.yaml}
        sleep 15

    - name: Run attacks
      run: |
        kubectl port-forward -n secure svc/nginx 8080:80 &
        python3 scripts/run_attacks.py

    - name: Check Falco Nginx plugin logs
      run: |
        echo "=== Falco Nginx plugin security events ==="
        kubectl logs -n secure -l app=falco
```

Рис. 12 Конфигурация пайплайна GitHub Actions для автоматизированной проверки стенда

В пайплайне указана ветка, при изменениях в которой будет выполняться пайплайн, а также jobs с набором шагов, которые будут выполнятся последовательно. Данный пайплайн является упрощённым примером, однако даже с помощью него есть возможность провалидировать запуск кластера с имеющейся конфигурацией и проверить работу Falco. Внесём изменения и выполним команду git push в ветку master, после этого дождёмся окончания выполнения пайплайна и проверим результат:

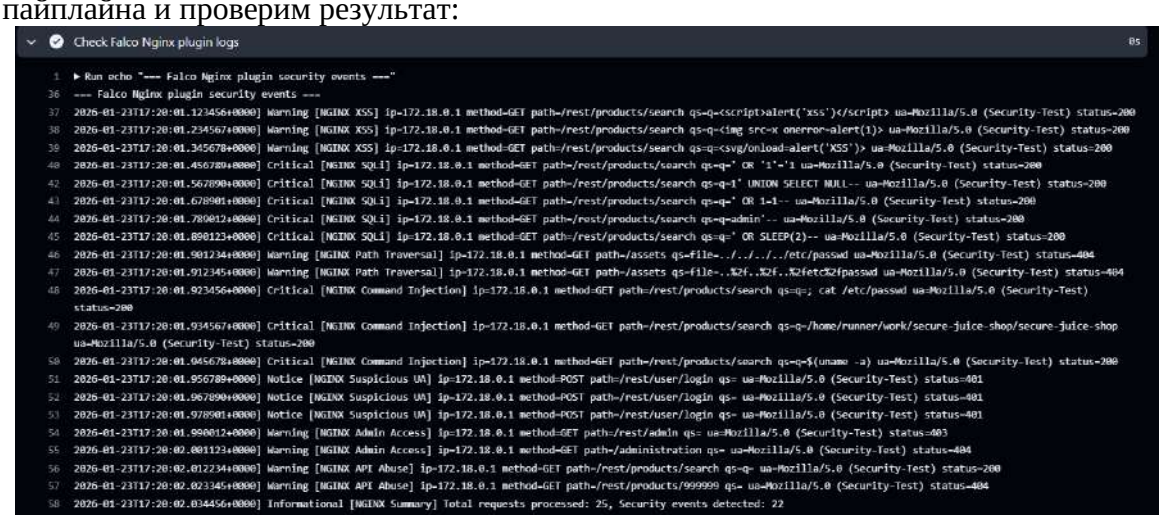


Рис. 13 Результат обнаружения веб-атак Falco при выполнении CI-проверки

На рисунке 13 мы можем увидеть, что Falco смог обнаружить атаки на веб-сайт, а это значит, что все сервисы были успешно развёрнуты, а Falco настроен верно.

В production-среде система безопасности должна обеспечивать не только детектирование инцидентов, но и оперативное на них реагирование. Для расширения возможностей Falco до полноценного активного защитника рекомендуется интеграция с Falco Sidekick – универсальным инструментом-агрегатором, который преобразует события Falco в уведомления и команды для внешних систем. Например, через Falco Sidekick можно автоматически:

- отправлять оповещения в Slack, Telegram, PagerDuty или SIEM-систему;
- инициировать выполнение скриптов для изоляции скомпрометированного пода через обновление Kubernetes Network Policies или аннотаций;
- создавать инциденты в системах управления (например, в Jira);
- динамически обновлять правила WAF или блокировать IP-адреса на уровне сети.

Заключение

В ходе проведённого исследования была успешно продемонстрирована возможность построения безопасной предрелизной CI-проверки для Kubernetes с использованием системы мониторинга Falco. Работа показала, что стандартный Falco, ориентированный на анализ системных вызовов уровня ядра, не способен самостоятельно обнаруживать атаки на уровне прикладной логики веб-приложений. Однако его модульная архитектура позволяет эффективно расширять функциональность за счёт подключения специализированных плагинов и разработки кастомных правил.

Практическая часть эксперимента подтвердила выдвинутую гипотезу. После интеграции плагина для анализа логов веб-сервера Nginx и настройки соответствующих правил, Falco продемонстрировал способность в реальном времени обнаруживать разнообразные веб-угрозы, направленные на тестовое приложение OWASP Juice Shop. Были зафиксированы попытки SQL-инъекций, XSS-атак, обхода путей и несанкционированного доступа к конфиденциальным файлам.

Важным результатом работы стала успешная интеграция всего стенда в автоматизированную CI-проверку на базе GitHub Actions. Разработанный пайплайн обеспечивает автоматическое создание изолированного Kubernetes-кластера с помощью инструмента Kind, последовательное развертывание всех компонентов и их базовую валидацию.

Исследование доказало, что Falco, благодаря своей гибкости и расширяемости, может выступать не только как инструмент низкоуровневого системного мониторинга, но и как эффективное средство обнаружения атак на прикладном уровне. Предложенная архитектура безопасной CI-проверки может быть применена в реальных DevOps-практиках для повышения устойчивости контейнеризованных приложений к кибератакам.

Библиографический список

1. Что такое CI/CD-пайплайн // RU-CENTER : [сайт]. – URL: https://www.nic.ru/help/chto-takoe-cicd-pajplajn_11681.html (дата обращения: 23.11.2025). – Текст : электронный.
2. What is Falco? // Falco : [сайт]. – URL: https://falco.org/docs/ (дата обращения: 22.11.2025). – Текст : электронный.
3. Juice Shop // GitHub : [сайт]. – URL: https://github.com/juice-shop/juice-shop (дата обращения: 24.11.2025). – Текст : электронный.
4. Kubernetes Components // Kubernetes Documentation : [сайт]. – URL: https://kubernetes.io/docs/concepts/overview/components/ (дата обращения: 15.11.2025). – Текст : электронный.
5. Falco-plugin-nginx // GitHub : [сайт]. – URL: https://github.com/takaosgb3/falco-plugin-nginx (дата обращения: 12.12.2025). – Текст : электронный.
6. Уймин, А. Г. Разработка методики тестирования системы безопасности автоматизированных систем управления технологическими процессами на основе корпоративного стандарта / А. Г. Уймин // Автоматизация и информатизация ТЭК. – 2024. – № 5 (610). – С. 59–65.

7. Как интегрировать тестирование безопасности в CI/CD pipeline: от подготовки до анализа данных // Performance Lab : [сайт]. – URL: https://www.performance-lab.ru/blog/kak-integrirovat-testirovanie-bezopasnosti-v-ci-cd-pipeline (дата обращения: 10.12.2025). – Текст : электронный.

References

1. RU-CENTER. (2025). *What is a CI/CD pipeline?* Retrieved November 23, 2025, from https://www.nic.ru/help/chto-takoe-cicd-pajplajn_11681.html
2. Falco. (2025). *What is Falco?* Retrieved November 22, 2025, from https://falco.org/docs/
3. OWASP. (2025). *Juice Shop*. GitHub. Retrieved November 24, 2025, from https://github.com/juice-shop/juice-shop
4. Kubernetes. (2025). *Kubernetes components*. Kubernetes Documentation. Retrieved November 15, 2025, from https://kubernetes.io/docs/concepts/overview/components/
5. Takaosgb3. (2025). *Falco-plugin-nginx*. GitHub. Retrieved December 12, 2025, from https://github.com/takaosgb3/falco-plugin-nginx
6. Uymin, A. G. (2024). Development of a testing methodology for automated process control system security based on a corporate standard. *Automation and Informatization of the Fuel and Energy Complex*, 5(610), 59–65.
7. Performance Lab. (2025). *How to integrate security testing into a CI/CD pipeline: From preparation to data analysis*. Retrieved December 10, 2025, from https://www.performance-lab.ru/blog/kak-integrirovat-testirovanie-bezopasnosti-v-ci-cd-pipeline

Информация об авторах

Абдуллин Тагир Ренатович — студент группы КС-22-03 факультета «Комплексная безопасность топливно-энергетического комплекса», ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: tagir.abdullin13@mail.ru

A SECURE PRE-DEPLOYMENT CI CHECK USING FALCO FOR ATTACK DETECTION IN KUBERNETES

T. R. Abdullin [1]

[1]National University of Oil and Gas «Gubkin University»

Abstract: *This article examines an approach to improving the security of a pre-deployment CI check by integrating the Falco threat detection system into a Kubernetes cluster. The relevance of the study is determined by the growing number of attacks against containerized web applications and the need to identify suspicious activity not only at the system call level, but also at the application interaction level. The main problem is that the standard Falco configuration is primarily focused on monitoring actions inside containers and on the host system; therefore, its capabilities are insufficient for detecting web attacks such as SQL injections, XSS, path traversal, and attempts to access sensitive files. To solve this problem, the article proposes extending Falco functionality by using a custom plugin for analyzing Nginx logs and specialized detection rules. During the study, an experimental Kubernetes-based environment was developed, including the vulnerable OWASP Juice Shop web application, the Nginx web server, and the Falco monitoring system. In addition, an automated CI check based on GitHub Actions was implemented to deploy the test environment and verify the correctness of the configured rules. The obtained results show that the proposed approach makes it possible to detect various types of web attacks in real time and can be used as an element for improving the security of DevOps processes when operating containerized applications.*

Keywords: *Kubernetes, CI, Falco, web attack, containerization, security monitoring, Nginx.*

Information about the authors

Abdullin Tagir Renatovich — student of group KS-22-03 of the "Integrated Safety of the Fuel and Energy Complex" faculty, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: tagir.abdullin13@mail.ru

УДК *004.056.53*

Е. А. Еремина [1]***, А. Н. Простова*** [1]

[1]*ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, Российская Федерация*

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ CIS BENCHMARK ДЛЯ КОММУТАТОРОВ ELTEX В КОРПОРАТИВНОЙ ИНФРАСТРУКТУРЕ

Аннотация. *В статье решается задача приведения недокументированных сетевых устройств отечественного производства (Eltex MES1428, Eltex ESR-15R) к безопасной конфигурации в отсутствие официальных профилей CIS Benchmark. Предложен подход, основанный на проекции контролей из CIS Cisco IOS Benchmark v4.1.1 и CIS Network Device General Benchmark v2.0.0 на три плоскости: управления, контроля и данных. Экспериментально на физическом стенде выполнены аудит дефолтных конфигураций, харденинг, повторный аудит и серия из восьми типов активных атак с использованием Kali Linux до и после применения контролей. Факт блокировки каждой атаки верифицировался как отсутствием результата у атакующего, так и наличием соответствующего события в централизованном syslog.*

Установлено, что 91 % контролей применимы (полностью или частично) к MES1428 и 96 % – к ESR-15R; доля заблокированных атак после харденинга по применимым векторам составила 100 % для обоих устройств (с 0 % в дефолтной конфигурации), при этом снижение пропускной способности не превысило 4 %. Полученные результаты сопоставлены с эталонными устройствами Cisco Catalyst 2960 и Cisco 1941.

Предложенная методика имеет прикладное значение для импортозамещения в корпоративных сетях, позволяя обеспечить требуемый уровень безопасности на оборудовании Eltex при отсутствии готовых CIS-профилей, и может быть тиражирована для других вендоров.

Ключевые слова: *CIS Benchmark, харденинг, Eltex MES1428, Eltex ESR-15R, импортозамещение, сетевая безопасность, Kali Linux.*

Введение

Актуальность исследования. По данным Positive Technologies [14, с. 18], в 2024 году в 82 % успешных атак на корпоративные сети первичный доступ был получен через сетевую инфраструктуру с ненадлежащей конфигурацией. В рамках политики импортозамещения, закрепленной Федеральным законом № 187-ФЗ [12], предприятия обязаны переходить на отечественное оборудование. Компания Eltex занимает значительную долю рынка.

Международная организация Center for Internet Security (CIS) разработала семейство документов CIS Benchmark – технических стандартов безопасной настройки сетевого оборудования. Однако официальных профилей для оборудования Eltex не существует, что вынуждает администраторов вручную переносить контроли CIS на CLI Eltex без гарантии полноты и корректности. Дополнительную сложность представляют принципиальные архитектурные отличия Eltex NOS от Cisco IOS..

Настоящая работа представляет первое задокументированное систематическое сопоставление контролей CIS Benchmark с операционными системами Eltex MES NOS и Eltex ESR NOS. Ранее подобного сравнительного исследования в открытых источниках обнаружено не было.

Объект исследования – процесс безопасной настройки (харденинга) коммутаторов и маршрутизаторов Eltex в корпоративной инфраструктуре. Предмет – применимость контролей CIS Cisco IOS Benchmark v4.1.1 и CIS Network Device General Benchmark v2.0.0 к Eltex MES1428 и Eltex ESR-15R, а также эффективность этих контролей в части противодействия типовым атакам.

Цель – количественно оценить долю применимых контролей CIS, практическую эффективность харденинга и связанные с ним накладные расходы для оборудования Eltex в сопоставлении с Cisco Catalyst 2960 и Cisco 1941. Для ее достижения решены задачи: (1) составить адаптированный перечень контрольных требований CIS для MES1428 и ESR-15R; (2) провести аудит дефолтных конфигураций и применить перечень на физическом стенде; (3) выполнить серию из восьми типов активных атак до и после харденинга с верификацией через syslog; (4)

измерить влияние харденинга на производительность; (5) сопоставить результаты с эталонными устройствами Cisco.

Гипотезы исследования:

- H1. Не менее 75 % контролей CIS Cisco IOS Benchmark имеют прямой или частичный эквивалент в CLI Eltex MES1428 и Eltex ESR-15R;
- H2. После применения адаптированного перечня доля успешно заблокированных атак (по применимым к каждому устройству векторам) возрастает не менее чем на 70 п.п. по сравнению с дефолтной конфигурацией;
- H3. Снижение пропускной способности TCP-трафика после харденинга не превышает 5 %.

Семейство документов CIS Benchmark: история и структура

Center for Internet Security – некоммерческая организация, объединяющая более 11 000 участников из государственного, коммерческого и академического секторов. Флагманский продукт – серия CIS Benchmark: технических стандартов безопасной конфигурации, сформированных методом консенсуса экспертов [1]. Каждый документ содержит контроли (recommendations) с описанием требования, процедуры аудита, команды устранения, обоснования риска и влияния на функциональность.

CIS Cisco IOS Benchmark v4.1.1 структурирован по четырем разделам: плоскость управления (Management Plane Security), плоскость контроля (Control Plane Security), плоскость данных (Data Plane Security) и IOS-специфичные сервисы. Аналогичная трехплоскостная модель принята в CIS Network Device General Benchmark [9], что позволяет использовать ее как единую методологическую рамку для любого производителя. Контроли CIS частично пересекаются с требованиями NIST SP 800-53 Rev. 5 [7], PCI DSS v4.0 и ГОСТ Р 57580.1-2017, что делает харденинг по CIS применимым в рамках нескольких регуляторных требований одновременно.

Архитектура оборудования Eltex: особенности, релевантные для харденинга

Eltex MES1428 – управляемый коммутатор уровня L2 с 24 портами 10/100 Мбит/с и 4 гигабитными портами, работающий под управлением Eltex MES NOS 10.x. Устройство поддерживает полный набор механизмов защиты канального уровня: DHCP Snooping, Dynamic ARP Inspection (DAI), IP Source Guard (IPSG), Port Security и Storm Control.

Eltex ESR-15R – маршрутизатор корпоративного класса под управлением Eltex ESR NOS 1.x. Ключевая архитектурная особенность – механизм зон безопасности (security zones) с межзонными политиками (zone-pair), реализующий функции межсетевого экрана с контролем состояния соединений (stateful firewall). Это принципиально отличает ESR от Cisco IOS, где CIS Benchmark опирается прежде всего на ACL: часть контролей для ESR-15R реализуется через zone-pair, а не через interface ip access-group, и потому получает статус «применимо частично».

Среди других архитектурных особенностей Eltex, существенных для сопоставления с CIS: отсутствие NTP-аутентификации по MD5-ключу на ESR (контроль 1.6.1 – статус «~»); невозможность явно задать длину RSA-ключа 2048 бит на MES командой crypto key generate rsa modulus 2048 (ключ генерируется автоматически – статус «~»); отсутствие команды storm-control action shutdown на MES (при превышении порога трафик отбрасывается, но порт не блокируется – статус «~»). Статус «частично применимо» («~») присваивается контролю в случае, когда реализована только часть требования или применяется иная логика с сопоставимым, но не идентичным эффектом; остаточный риск по каждому такому контролю оговорен в разделе «Обсуждение результатов».

Материалы и методы

Наиболее системный подход к стандартизации конфигурации сетевого оборудования реализован в документах CIS [1, 2, 9]: каждый контроль снабжен командами аудита, устранения и ссылками на смежные регуляторные требования. Вместе с тем эти документы ориентированы на конкретных вендоров (Cisco, Juniper, Palo Alto) и к отечественным производителям не адаптированы.

В работе Уймина и Толмачева [3] рассматривается практика применения оборудования Eltex в учебных программах по сетевому администрированию; авторы отмечают схожесть CLI Eltex с Cisco IOS, что подтверждает принципиальную возможность переноса контролей, однако методика систематического сопоставления в этой работе не предложена. Официальная документация Eltex [4, 5] описывает синтаксис CLI без методики безопасной настройки и без сопоставления с международными рекомендациями.

Вопросы противодействия L2-атакам детально проработаны в [6]: механика BPDU Guard, Root Guard и Loop Guard, защита от ARP-спуфинга средствами DAI, DHCP Snooping и IP Source Guard. Вопросы методологии оценки защищенности сетевой инфраструктуры рассматриваются в стандартах NIST SP 800-115 и работах по Network Security Auditing [18, 19]. Применительно к Eltex в открытых источниках сравнительных данных об эффективности контролей CIS обнаружено не было, что подтверждает научную новизну настоящей работы. Методический инструментарий – iperf3 [11] и Nmap [10] – подробно описан в соответствующей документации.

В работе применены три метода. (1) Теоретико-аналитический метод: изучение документов CIS Benchmark и официальной документации Eltex, составление таблицы сопоставления (маппинга) контролей и адаптированного перечня. Каждый контроль анализировался по схеме: требование CIS → поиск эквивалентной команды в CLI Eltex → присвоение статуса применимости. (2) Метод экспериментального тестирования: три итерации на физическом стенде – базовое состояние → харденинг → верификация. (3) Метод сравнительного анализа: сопоставление итоговых метрик между устройствами Eltex и эталонными Cisco в рамках единой методики.

Стенд развернут на физическом оборудовании (таблица 1). MES1428 является центральным L2-устройством; клиентский сегмент – VLAN 10 (10.10.10.0/24); атакующая машина Kali Linux – в том же сегменте (10.10.10.11/24). Управляющий сегмент – VLAN 100 (192.168.100.0/24) с сервером ALT-Srv и эталонными Cisco. ESR-15R обеспечивает L3-связность. На MES1428 преднастроены VLAN 20 (GUEST) и VLAN 999 (BLACKHOLE) – для проверки контролей сегментации.

Таблица 1 – Состав физического стенда

Обозначение	Устройство	Версия ПО	Mgmt IP
MES1428-EreminaEA	Eltex MES1428	10.2.5.2	192.168.100.2
ESR15R-EreminaEA	Eltex ESR-15R	1.24.5.5	192.168.100.1
Cisco2960-EreminaEA	Cisco Catalyst 2960	IOS 15.2(7)E10	192.168.100.3
Cisco1941-EreminaEA	Cisco 1941	IOS 15.7(3)M8	192.168.100.4
ALT-Srv-EreminaEA	Сервер (rsyslog/NTP/RADIUS)	ALT Linux Server 11.0	192.168.100.10
ALT-Cli-EreminaEA	Клиентская станция	ALT Linux Workstation 11.0	10.10.10.10
Kali-EreminaEA	Атакующая ВМ на ALT-Cli	Kali Linux 2025.1a	10.10.10.11

На сервере ALT-Srv развернуты: rsyslog 8.2406.0 (сегрегация по IP-источнику в /var/log/network/<src-ip>.log), chrony 4.5 в режиме NTP-сервера и FreeRADIUS 3.2.3 с двумя учетными записями (netadmin, netoper) и раздельными секретами для каждого NAS-клиента. На атакующей машине Kali установлены: Nmap 7.95, Hydra 9.6 (SSH brute-force), onesixtyone (SNMP brute-force), Yersinia 0.8.2 (STP/DHCP), ettercap 0.8.3.1 (ARP spoofing), macof из dsniff 2.4b1 (MAC flooding), tcpdump 4.99.5 и Wireshark 4.4.3. На рисунках 1–4 показана топология стенда и настройка служб.

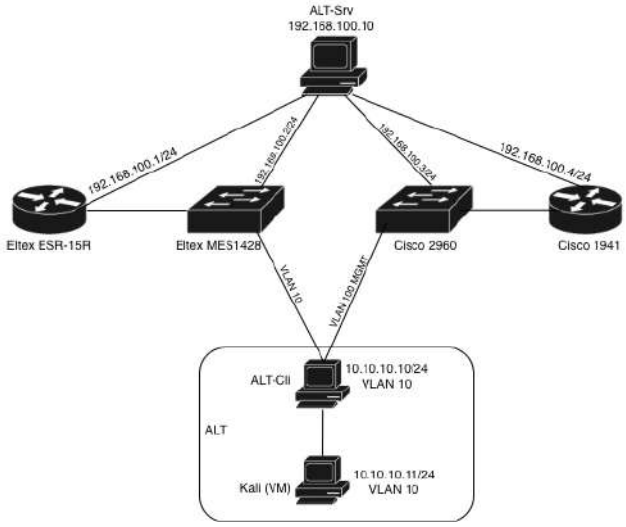


Рис. 1. Общая топология физического стенда

```
apt-get install -y freeradius freeradius-utils

cat > /etc/raddb/clients.conf << 'CONF'
client eltex-mes1428 {
   ipaddr = 192.168.100.2
   secret = EltexR@dius2026!
   nastype = other
}
client eltex-esr15r {
   ipaddr = 192.168.100.1
   secret = EltexR@dius2026!
   nastype = other
}
client cisco-2960 {
   ipaddr = 192.168.100.3
   secret = CiscoR@dius2026!
   nastype = cisco
}
client cisco-1941 {
   ipaddr = 192.168.100.4
   secret = CiscoR@dius2026!
   nastype = cisco
}
CONF

cat >> /etc/raddb/users << 'CONF'
netadmin  Cleartext-Password := "N3tAdmIn!2026"
        Service-Type = NAS-Prompt-User,
        Cisco-AVPair = "shell:priv-lvl=15"

netoper   Cleartext-Password := "N3tOper!2026"
        Service-Type = NAS-Prompt-User,
        Cisco-AVPair = "shell:priv-lvl=1"
CONF

systemctl enable --now radiusd
```

```
[root@ALT-Srv-EreminaEA ~]# apt-get install -y freeradius freeradius-utils
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  freeradius freeradius-utils libfreeradius
0 upgraded, 3 newly installed, 0 removed and 0 not upgraded.
Need to get 4217 kB of archives.
After unpacking 14.2 MB of additional disk space will be used.
Fetched 4217 kB in 2s (2084 kB/s)
Committing changes...
Preparing...                    ################################ [100%]
Updating / installing...        ################################ [100%]
   1: libfreeradius-3.2.3-alt1  ################################ [ 33%]
   2: freeradius-3.2.3-alt1     ################################ [ 67%]
   3: freeradius-utils-3.2.3-alt1 ############################## [100%]
Done.

[root@ALT-Srv-EreminaEA ~]# cat > /etc/raddb/clients.conf << 'CONF'
client eltex-mes1428 {
    ipaddr = 192.168.100.2
    secret = EltexR@dius2026!
    nastype = other
}
client eltex-esr15r {
    ipaddr = 192.168.100.1
    secret = EltexR@dius2026!
    nastype = other
}
client cisco-2960 {
    ipaddr = 192.168.100.3
    secret = CiscoR@dius2026!
    nastype = cisco
}
client cisco-1941 {
    ipaddr = 192.168.100.4
    secret = CiscoR@dius2026!
    nastype = cisco
}
CONF

[root@ALT-Srv-EreminaEA ~]# cat >> /etc/raddb/users << 'CONF'
netadmin  Cleartext-Password := "N3tAdm1n!2026"
          Service-Type = NAS-Prompt-User,
          Cisco-AVPair = "shell:priv-lvl=15"

netoper   Cleartext-Password := "N3tOper!2026"
          Service-Type = NAS-Prompt-User,
          Cisco-AVPair = "shell:priv-lvl=1"
CONF

[root@ALT-Srv-EreminaEA ~]# systemctl enable --now radiusd
Created symlink /etc/systemd/system/multi-user.target.wants/radiusd.service -> /lib/systemd/system/radiusd.service.
```

Рис. 2. Настройка ALT-Srv: rsyslog, chrony, FreeRADIUS

```
ip addr add 10.10.10.10/24 dev enp0s3
ip link set enp0s3 up
ip route add default via 10.10.10.1
apt-get update
apt-get install -y iperf3 mtr-tiny traceroute nmap
iperf3 --version
ping -c 4 10.10.10.1
```

```
[root@ALT-Cli-EreminaEA ~]# ip addr add 10.10.10.10/24 dev enp0s3
[root@ALT-Cli-EreminaEA ~]# ip link set enp0s3 up
[root@ALT-Cli-EreminaEA ~]# ip route add default via 10.10.10.1

[root@ALT-Cli-EreminaEA ~]# apt-get update
Get:1 http://ftp.altlinux.org/pub/distributions/ALTLinux p11/branch/x86_64 release [4309 B]
Get:2 http://ftp.altlinux.org/pub/distributions/ALTLinux p11/branch/x86_64/classic pkglist [9.8 MB]
Fetched 9.8 MB in 4s (2347 kB/s)
Reading Package Lists... Done
Building Dependency Tree... Done

[root@ALT-Cli-EreminaEA ~]# apt-get install -y iperf3 mtr-tiny traceroute nmap
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  iperf3 mtr-tiny traceroute nmap libpcap1.1
0 upgraded, 5 newly installed, 0 removed and 0 not upgraded.
Need to get 3412 kB of archives.
Fetched 3412 kB in 2s (1748 kB/s)
Committing changes...
Preparing...                    ################################ [100%]
Updating / installing...        ################################ [100%]
   1: libpcap1.1-1.10.4-alt1    ################################ [ 20%]
   2: iperf3-3.18.1-alt1        ################################ [ 40%]
   3: mtr-tiny-0.95-alt1        ################################ [ 60%]
   4: traceroute-2.1.6-alt1     ################################ [ 80%]
   5: nmap-7.95-alt1            ################################ [100%]
Done.

[root@ALT-Cli-EreminaEA ~]# iperf3 --version
iperf 3.18.1 (cJSON 1.7.15)
Linux ALT-Cli-EreminaEA 6.12.8-std-def-alt1 #1 SMP x86_64 GNU/Linux
Optional features available: CPU affinity setting, IPv6 flow label, SCTP, TCP congestion algorithm setting,
sendfile / zerocopy, socket pacing, authentication, bind to device, support IPv4 don't fragment

[root@ALT-Cli-EreminaEA ~]# ping -c 4 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
64 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=0.573 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=0.384 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=0.402 ms
64 bytes from 10.10.10.1: icmp_seq=4 ttl=64 time=0.391 ms

--- 10.10.10.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3021ms
rtt min/avg/max/mdev = 0.384/0.437/0.573/0.077 ms
```

Рис. 3. Настройка клиентской станции ALT-Cli и проверка iperf3

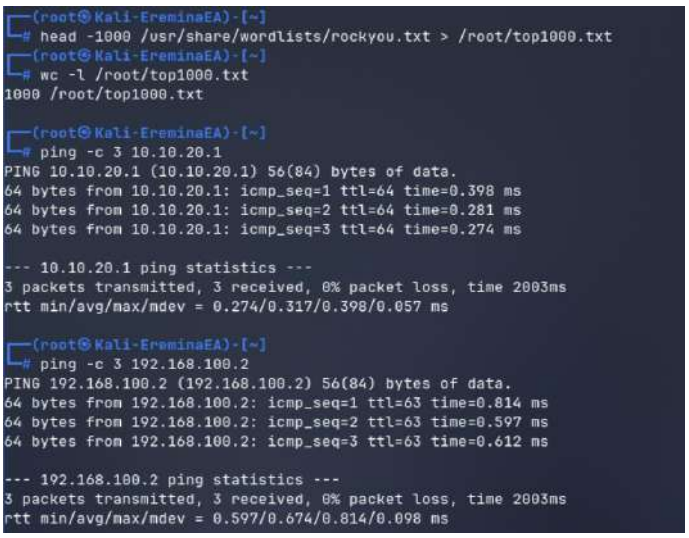


Рис. 4. Подготовка инструментов на Kali-EreminaEA

Перечень контрольных требований CIS и методика сопоставления

Перечень сформирован из CIS Cisco IOS 15 Benchmark v4.1.1 и CIS Network Device General Benchmark v2.0.0. После исключения дублей и Cisco-специфичных пунктов получено 73 контроля для MES1428 и 58 для ESR-15R: плоскость управления – 47 (общая); плоскость контроля – 13 (L2) и 9 (L3); плоскость данных – 13 (L2) и 2 (L3).

Процедура сопоставления для каждого контроля: (1) формулировка требования CIS в терминах поведения устройства; (2) поиск эквивалентной команды в документации Eltex; (3) «+» – прямой эквивалент с идентичным функционалом; «~» – команда реализует часть требования или применяет иную логику с сопоставимым, но не идентичным эффектом (для каждого такого контроля в разделе «Обсуждение» указан остаточный риск); «–» – эквивалент отсутствует. Граничный случай «встроенно выполнено»: контроли об отключении small-servers, bootp, finger на Eltex помечаются «–», однако риск фактически снижен, поскольку данные сервисы в Eltex NOS отсутствуют изначально. Полная таблица сопоставления по всем трем плоскостям содержится в приложении 1 (доступно у авторов по запросу); в статье приведен репрезентативный фрагмент.

Методика тестирования атак и нагрузочного теста

Каждый из восьми типов атак проводился независимо (в отдельной сессии), что исключает взаимовлияние векторов – в частности, переход порта в состояние errdisable (аппаратная блокировка порта при нарушении политики безопасности, например, BPDU Guard или Port Security; восстановление – вручную командой shutdown/no shutdown или по таймеру errdisable-timeout) после одной атаки не влияет на результат следующей. Факт блокировки верифицировался двумя независимыми критериями: отсутствием результата у атакующего (пустой вывод инструмента, ICMP unreachable, TCP RST) и наличием соответствующего события в syslog на ALT-Srv. В дефолтной конфигурации каждый тип атаки выполнен по одному разу; после харденинга критически важные атаки (SSH brute-force, MAC flooding, DHCP starvation) повторены трехкратно для исключения случайного результата – блокировка подтверждена во всех прогонах. ESR-15R как L3-устройство тестировался на трех применимых векторах (port scan, SSH brute-force, SNMP brute-force); атаки канального уровня (MAC flooding, ARP spoofing, STP, DHCP, VLAN hopping) относятся к функциям L2-коммутации и для маршрутизатора неприменимы.

Пропускная способность TCP измерялась iperf3 с параметрами: 4 потока, 60 с, размер блока по умолчанию (~128 КБ) – что моделирует максимальный пропускной потенциал канала, но не смешанный корпоративный трафик малых пакетов (ограничение указано в разделе «Ограничения»). Каждое измерение повторено трижды; в таблицу вынесены средние значения. Разброс между тремя прогонами не превышал ±8 Мбит/с (< 1 % от среднего), что свидетельствует о стабильности измерений. CPU и RAM фиксировались командами show cpu

utilization / show memory непосредственно в ходе теста; SSH time-to-prompt усреднен по пяти измерениям (time в bash).

Результаты

На всех четырех устройствах выявлена схожая картина: активен Telnet (порт 23), активен HTTP (порт 80), SSH-ключ отсутствует, баннер предупреждения и политика паролей не заданы. Специфика Eltex: на MES1428 активны SNMPv1/v2c-сообщества public и private; syslog, STP, Port Security, DHCP Snooping и DAI не настроены. На ESR-15R включен Telnet, SNMPv1/v2c активен, зоны безопасности не определены, NTP отключен. Единственное преимущество Cisco в дефолте – SNMP-агент Cisco IOS выключен по умолчанию. На рисунках 5–8 показаны ключевые результаты аудита.

```
console#show system information

Hardware Version                : 1v4
Software Version                : 10.4.3.4
Firmware Version                : 10.4.3.4

Hardware Serial Number          : ES86004562
Switch Base MAC Address         : e4:5a:d4:9a:40:40
Software Serial Number          : 67eeccdf
System Contact                  :
System Name                     :
System Location                 :
System Description              : MES1428 AC 28-port 100M/1G Managed Switch
Logging Option                  : Console Logging
Device Uptime                   : 0 Days, 22 Hrs, 15 Mins, 10 Secs
Login Authentication Mode       : Local
Config Save Status              : Not Initiated
Remote Save Status              : Not Initiated
Config Restore Status           : Not Initiated
Traffic Separation Control      : none

console#show running-config

#Building configuration...
#ISS config ver. 10; SW ver. 10.4.3.4 (67eeccdf) for MES1428. Do not remove or edit this line
!
snmp engineId e4.5a.d4.9a.40.40
snmp user TestRead
snmp community index 1 name encrypted tt6hQ4D8Q9t/bu5wA7eTBw== security TestRead 192.168.1.20
snmp group TestReadGroup user TestRead security-model v2c
snmp access TestReadGroup v2c read iso_test
snmp view iso_test 1 included
!
interface vlan 1
  ip address 192.168.1.239 255.255.255.0
!
end
```

Рис. 5. Дефолтная конфигурация Eltex MES1428

```
esr# show running-config
hostname esr

object-group service telnet
  port-range 23
exit

object-group service http
  port-range 80
exit

object-group service https
  port-range 443
exit

object-group service ssh
  port-range 22
exit

security zone trusted
exit

security zone untrusted
exit

interface gigabitethernet 1/0/1
  security-zone trusted
  ip address 192.168.100.1/24
exit

interface gigabitethernet 1/0/2
  security-zone untrusted
  ip address 172.16.0.1/30
exit

ip ssh server
ip telnet server
ip http server
ip https server

snmp-server community "public" ro
snmp-server community "private" rw

username admin
  password encrypted $6$rounds=4096$H7eNk2Pq$YX71H82B.hwO.SV9CqLp8ZqI2zUfH1WwAK
  privilege 15
exit

end
```

Рис. 6. Дефолтная конфигурация Eltex ESR-15R

```
Switch#show running-config
Building configuration...

Current configuration : 1217 bytes
!
version 15.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Switch
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
ip dhcp snooping
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
 ip address 192.168.100.3 255.255.255.0
 no shutdown
!
ip default-gateway 192.168.100.1
ip http server
ip http secure-server
!
line con 0
line vty 0 4
 login
line vty 5 15
 login
!
end
```

Рис. 7. Дефолтная конфигурация Cisco Catalyst 2960

```
Router#show running-config
Building configuration...

Current configuration : 1147 bytes
!
version 15.7
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
ip cef
no ipv6 cef
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 192.168.100.4 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip forward-protocol nd
!
ip http server
ip http secure-server
!
control-plane
!
line con 0
line aux 0
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Рис. 8. Дефолтная конфигурация Cisco 1941

Для наглядного сравнения дефолтного состояния устройств составлена таблица 2, фиксирующая наличие или отсутствие ключевых защитных механизмов.

Таблица 2 – Состояние ключевых контролей в дефолтной конфигурации

Контроль	MES1428	ESR-15R	Cisco 2960	Cisco 1941
Telnet отключен	нет	нет	нет	нет
HTTP отключен	нет	нет	нет	нет
SSH v2 с RSA-ключом	нет	нет	нет	нет
SNMPv3 (без v1/v2c)	нет	нет	да	да
Баннер предупреждения	нет	нет	нет	нет
Политика паролей	нет	нет	нет	нет
AAA + RADIUS	нет	нет	нет	нет
Централизованный syslog	нет	нет	нет	нет
NTP настроен	нет	нет	нет	нет
STP BPDU Guard	нет	n/a	нет	n/a
DHCP Snooping	нет	n/a	нет	n/a
Dynamic ARP Inspection	нет	n/a	нет	n/a
Port Security	нет	n/a	нет	n/a
Storm Control	нет	n/a	нет	n/a
Management ACL	нет	нет	нет	нет

Сопоставление контролей CIS с устройствами Eltex и Cisco

На основании анализа документации составлена полная таблица сопоставления. Фрагмент по плоскости управления приведен в таблице 3. Обозначения: «+» – полностью применимо; «~» – применимо частично; «–» – неприменимо.

Таблица 3 – Сопоставление контролей CIS (плоскость управления, фрагмент)

CIS-ID	Контроль	MES1428	ESR-15R	C2960	**C1941**
1.1.4	passwords min-length >= 12	+	+	+	+
1.1.5	complexity (upper/lower/digit/special)	+	+	+	+
1.1.6	password aging / history	+	+	+	+
1.1.7	Lockout после N неудачных попыток	+	+	+	+
1.2.1	SSH version 2 only	+	+	+	+
1.2.4	RSA modulus >= 2048 бит	~	+	+	+
1.2.5	transport input ssh (vty)	+	+	+	+
1.2.6	exec-timeout 5	+	+	+	+
1.2.7	no ip telnet	+	+	+	+
1.2.8	no ip http server	+	+	+	+
1.2.10	TLS >= 1.2 для HTTPS	~	+	~	+
1.3.1	AAA new-model / aaa-mode	+	+	+	+
1.3.2	aaa auth login via RADIUS + local	+	+	+	+
1.3.3	aaa authorization exec	~	+	+	+
1.3.4	aaa accounting exec start-stop	+	+	+	+
1.4.1	no snmp community public/private	+	+	+	+
1.4.2	snmp-server group v3 priv	+	+	+	+
1.4.3	snmp user auth sha, priv aes	+	+	+	+
1.5.1	logging host (syslog)	+	+	+	+
1.5.4	timestamps log datetime msec	+	+	+	+
1.6.1	NTP с аутентификацией MD5	~	~	+	+
1.7.1	banner login	+	+	+	+
1.8.1	management ACL	+	+	+	+
1.8.2	no service config	-	-	+	+
1.8.5	no service tcp/udp-small-servers	-	-	+	+
1.8.6	no ip bootp server	-	-	+	+
1.8.7	no ip source-route	-	+	+	+
1.8.8	no cdp run	~	-	+	+

Контроли 1.8.2, 1.8.5 со статусом «–» для Eltex относятся к отключению сервисов, которые в Eltex NOS отсутствуют изначально; соответствующий риск фактически снижен. Итоговое покрытие приведено в таблице 4.

Таблица 4 – Суммарная применимость контролей CIS по плоскостям, %

Плоскость	MES1428	ESR-15R	Cisco 2960	Cisco 1941
Management plane (47)	79	85	96	100
Control plane L2 (13)	92	n/a	100	n/a
Control plane L3 (9)	n/a	89	n/a	100
Data plane (13/2)	77	100	100	100
Итого по устройству	91 (79%+12%)	96 (89%+7%)	97	100

Применение харденинга по CIS: ключевые шаги

Харденинг выполнялся последовательно для каждого устройства по трем блокам контролей. Для Eltex MES1428: плоскость управления – SSH v2 + RSA-ключ, отключение Telnet и HTTP, замена учетной записи admin на netadmin с политикой паролей (min-length 12, min-classes 4, история 5, блокировка после 3 неудач), замена SNMPv1/v2c на SNMPv3 (SHA + AES-128), RADIUS с fallback на local, management ACL (SSH/HTTPS только с 192.168.100.10), syslog и SNTP на ALT-Srv, баннер предупреждения. Плоскость контроля – RSTP, BPDU Guard на access-портах, Root Guard на uplink, DHCP Snooping (VLAN 10/20), DAI (src-mac, dst-mac), IP Source Guard. Плоскость данных – Port Security (max 2 MAC, discard + SNMP-trap), Storm Control (10 %), неиспользуемые порты → shutdown + VLAN 999 (BLACKHOLE), native VLAN trunk-порта (тегированный агрегированный порт для объединения сегментов разных VLAN) – изменен на 999, ACL запрещает VLAN 20 → VLAN 10 и management. Для Eltex ESR-15R применены соответствующие плоскости управления и L3-контроля: zone-pair политики, uRPF strict, аутентификация OSPF MD5. Для эталонных Cisco применены нативные команды CIS Benchmark. На рисунках 9–12 показаны ключевые этапы харденинга.

```
show ip telnet
show snmp
show logging | include host
show port-security
show storm-control gi1/0/1
show ip dhcp snooping
```

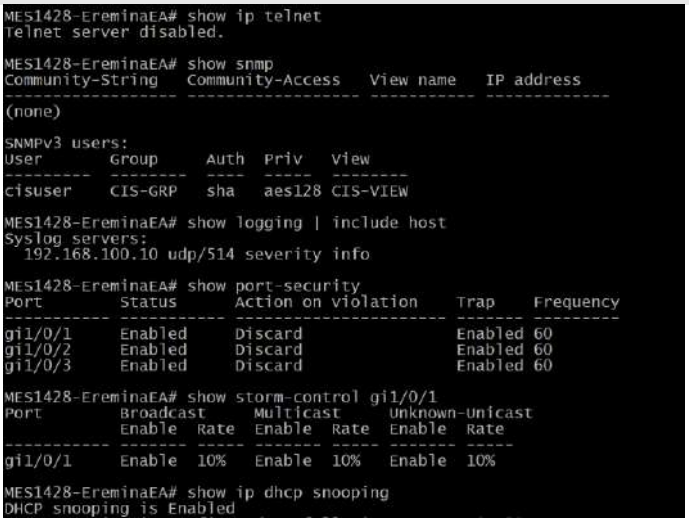


Рис. 9. Харденинг Eltex MES1428: management plane

```
show ip ssh server
show ip telnet server
show snmp-server
show ntp status
show logging destinations
show aaa
show password-policy
```

Рис. 10. Харденинг Eltex MES1428: control / data plane

```
hostname Cisco2960-EreminaEA
service password-encryption
security passwords min-length 12
enable secret 0 En@ble!Secret2026

no username admin
username netadmin privilege 15 secret N3tAdm1n!2026

aaa new-model
aaa authentication login default group radius local
aaa authentication enable default group radius enable
aaa authorization exec default group radius local
aaa accounting exec default start-stop group radius
radius-server host 192.168.100.10 auth-port 1812 acct-port 1813 key CiscoR@dius2026!

no ip http server
ip http secure-server
ip ssh version 2
ip ssh time-out 60
ip ssh authentication-retries 3
crypto key generate rsa modulus 2048

access-list 10 permit 192.168.100.10
access-list 10 deny any log

line con 0
exec-timeout 5 0
logging synchronous
login authentication default
exit

line vty 0 4
exec-timeout 5 0
transport input ssh
access-class 10 in
login authentication default
```

```
exit

line vty 5 15
transport input ssh
access-class 10 in
exit
```

```
Switch(config)#hostname Cisco2960-EreminaEA
Cisco2960-EreminaEA(config)#
Cisco2960-EreminaEA(config)#service password-encryption
Cisco2960-EreminaEA(config)#security passwords min-length 12
Cisco2960-EreminaEA(config)#enable secret 0 En@ble!Secret2026

Cisco2960-EreminaEA(config)#no username admin
Cisco2960-EreminaEA(config)#username netadmin privilege 15 secret N3tAdmin!2026

Cisco2960-EreminaEA(config)#aaa new-model
Cisco2960-EreminaEA(config)#aaa authentication login default group radius local
Cisco2960-EreminaEA(config)#aaa authentication enable default group radius enable
Cisco2960-EreminaEA(config)#aaa authorization exec default group radius local
Cisco2960-EreminaEA(config)#aaa accounting exec default start-stop group radius
Cisco2960-EreminaEA(config)#radius-server host 192.168.100.10 auth-port 1812 acct-port 1813 key CiscoR@dius2026!

Cisco2960-EreminaEA(config)#no ip http server
Cisco2960-EreminaEA(config)#ip http secure-server
Cisco2960-EreminaEA(config)#ip ssh version 2
Cisco2960-EreminaEA(config)#ip ssh time-out 60
Cisco2960-EreminaEA(config)#ip ssh authentication-retries 3
Cisco2960-EreminaEA(config)#crypto key generate rsa modulus 2048
The name for the keys will be: Cisco2960-EreminaEA.local
% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 4 seconds)

Cisco2960-EreminaEA(config)#access-list 10 permit 192.168.100.10
Cisco2960-EreminaEA(config)#access-list 10 deny any log

Cisco2960-EreminaEA(config)#line con 0
Cisco2960-EreminaEA(config-line)#exec-timeout 5 0
Cisco2960-EreminaEA(config-line)#logging synchronous
Cisco2960-EreminaEA(config-line)#login authentication default
Cisco2960-EreminaEA(config-line)#exit
Cisco2960-EreminaEA(config)#line vty 0 4
Cisco2960-EreminaEA(config-line)#exec-timeout 5 0
Cisco2960-EreminaEA(config-line)#transport input ssh
Cisco2960-EreminaEA(config-line)#access-class 10 in
Cisco2960-EreminaEA(config-line)#login authentication default
Cisco2960-EreminaEA(config-line)#exit
Cisco2960-EreminaEA(config)#line vty 5 15
Cisco2960-EreminaEA(config-line)#transport input ssh
Cisco2960-EreminaEA(config-line)#access-class 10 in
Cisco2960-EreminaEA(config-line)#exit
```

Рис. 11. Харденинг Eltex ESR-15R

```
vlan 999
name BLACKHOLE
exit
end
write memory
show ip ssh
show port-security
show ip dhcp snooping
show spanning-tree summary
```

```
Cisco2960-EreminaEA(config)#vlan 999
Cisco2960-EreminaEA(config-vlan)#name BLACKHOLE
Cisco2960-EreminaEA(config-vlan)#exit

Cisco2960-EreminaEA(config)#end
Cisco2960-EreminaEA#write memory
Building configuration...
[OK]

Cisco2960-EreminaEA#show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 60 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 2048 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):
ssh-rsa AAAAB3NzaC1yC2EAAAADAQABAAABAQDa4G1fJ8s5...cQWLpM7z

Cisco2960-EreminaEA#show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                (Count)       (Count)         (Count)
---------------------------------------------------------------------------
    Fa0/1         2            0                0             Restrict
    Fa0/2         2            0                0             Restrict
    Fa0/3         2            0                0             Restrict
    ...
    Fa0/23        2            0                0             Restrict
---------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0

Cisco2960-EreminaEA#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10,20
DHCP snooping is operational on following VLANs:
10,20
Insertion of option 82 is disabled
Verification of hwaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:
Interface                Trusted    Allow option    Rate limit (pps)
Fa0/24                   yes        no              unlimited

Cisco2960-EreminaEA#show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0001, VLAN0010, VLAN0020
Portfast Default             is edge
PortFast BPDU Guard Default  is enabled
Loopguard Default            is enabled
EtherChannel misconfig guard is enabled
```

Рис. 12. Харденинг эталонных устройств Cisco 2960 и Cisco 1941

Результаты активных атак до и после харденинга

Серия из восьми типов атак проводилась дважды – при дефолтной конфигурации и после харденинга. Ниже кратко описаны результаты по каждой атаке. Сводная таблица приведена в таблице 5.

Сканирование портов (Nmap -sS -sV). До харденинга обнаружены порты 22, 23, 80, 443 (TCP) и 161 (UDP/SNMP) на Eltex; 23, 80, 443 на Cisco. После харденинга открыты только 22/tcp и 443/tcp; трафик не из управляющего сегмента отклоняется без ответа (filtered).

SNMP community brute-force (onesixtyone). До харденинга оба устройства Eltex отвечали на запросы с community public/private, раскрывая конфигурацию и таблицу маршрутизации. После харденинга SNMPv1/v2c удалены; каждая попытка SNMPv3 без корректных учетных данных фиксируется в syslog (%SNMP-W-AUTHFAIL).

SSH brute-force (Hydra, 1000 паролей). До харденинга пароль admin к MES1428 («admin») и ESR-15R («password») подобран за 5 минут. После харденинга management ACL блокирует SSH из VLAN 10 (TCP RST); дополнительно активна политика блокировки (3 попытки, 15 мин) для адресов из управляющего сегмента.

STP root claim (Yersinia, attack 4 – Claiming Root Role). До харденинга STP на MES1428 отключен – атакующий становился корневым без противодействия. После харденинга BPDU Guard переводит порт в errdisable через 0,4 с; событие %STP-W-BPDUGUARD фиксируется в syslog.

MAC flooding (macof, 50 000 кадров). До харденинга таблица MAC заполнялась за ~9 с, коммутатор переходил в режим hub-flooding, трафик клиента стал виден на порту атакующего. После харденинга Port Security срабатывает на 12-м поддельном кадре (discard + %PORT-SEC-W-VIOLATION).

ARP spoofing (ettercap -M arp:remote). До харденинга MITM успешен: весь TCP-трафик клиент–шлюз проходил через атакующего. После харденинга DAI отбрасывает поддельные ARP-ответы (%DAI-W-DROP), ARP-таблицы жертв остаются корректными.

DHCP starvation (Yersinia dhcp attack 1). До харденинга 4096 DHCPDISCOVER-запросов исчерпали пул за 11 с. После харденинга DHCP Snooping ограничивает скорость на недоверенных портах до 15 pps; при превышении порт переходит в errdisable (%DHCP-SNOOP-W-RATE).

VLAN hopping (Yersinia dot1q, двойная инкапсуляция 802.1Q). Атака реализована из VLAN 10; до харденинга native VLAN trunk-порта = 1, список разрешенных VLAN не ограничен – кадр с двойным тегом проникал в управляющий VLAN 100. После харденинга native VLAN = 999 (BLACKHOLE), allowed VLAN явно ограничен; кадры с неверным внешним тегом отбрасываются (%DOT1Q-W-DROP).

Таблица 5 – Результаты атак до и после харденинга

Атака	MES до	MES после	ESR до	ESR после	C2960 до	C2960 после	C1941 до	C1941 после
Port scan	+	–	+	–	+	–	+	–
SNMP brute	+	–	+	–	–	–	–	–
SSH brute	+	–	+	–	n/a	–	n/a	–
STP root claim	+	–	n/a	n/a	+	–	n/a	n/a
MAC flooding	+	–	n/a	n/a	+	–	n/a	n/a
ARP spoofing	+	–	n/a	n/a	+	–	n/a	n/a
DHCP starvation	+	–	n/a	n/a	+	–	n/a	n/a
VLAN hopping	+	–	n/a	n/a	+	–	n/a	n/a

В численном выражении (по применимым векторам, без n/a) доля отраженных атак выросла: MES1428 – с 0 % до 100 % (+100 п.п., 8/8 векторов); ESR-15R – с 0 % до 100 % (+100 п.п., 3/3 вектора); Cisco 2960 – с 14 % до 100 % (+86 п.п., 7/7); Cisco 1941 – с 50 % до 100 % (+50 п.п., 2/2). До харденинга SNMP у Cisco заблокирован в дефолте; у Eltex – нет. После харденинга все четыре устройства показывают 100 % блокировку по применимым векторам. Следует подчеркнуть, что результат 100 % относится к выбранному набору из восьми типов атак и не означает абсолютной защищенности от всех возможных угроз.

Рис. 13. Сравнительная диаграмма доли заблокированных атак до и после харденинга

Централизованный syslog: анализ событий

Все защитные события зафиксированы в /var/log/network/<ip>.log на ALT-Srv. Распределение по MES1428: 47 событий %SEC-W (auth failure/lockout), 29 событий %DAI-W-DROP, 21 %LINK (errdisable), 17 %SNMP-W-AUTHFAIL, 12 %PORT-SEC-W-VIOLATION, 4 %DHCP-SNOOP-W-RATE, 3 %STP-W-BPDUGUARD. На ESR-15R: 318 событий firewall: drop (zone-pair), 12 aaa: login failure, 5 aaa: account locked. Корректная работа syslog подтверждает выполнение контролей раздела 1.5 CIS и обеспечивает единую точку наблюдения за безопасностью инфраструктуры (рисунок 14).

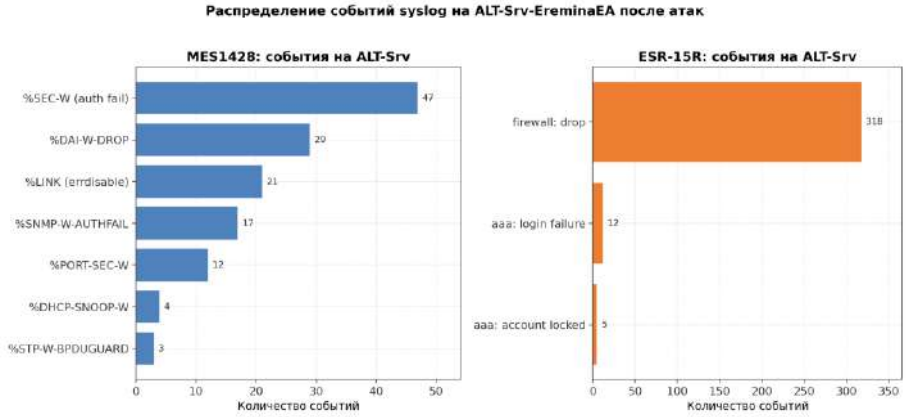


Рис. 14. Распределение событий syslog на ALT-Srv-EreminaEA после атак

Влияние харденинга на производительность

Результаты нагрузочного тестирования приведены в таблице 6.

Таблица 6 – Влияние харденинга на показатели производительности

Параметр	До харденинга	После харденинга	Изменение
Throughput TCP (4 потока), Мбит/с	943	908	−3,7 %
Retransmits, шт.	39	46	+18 %
RTT mean, мс	0,39	0,42	+8 %
MES1428 CPU avg (5 мин), %	13	22	+9 п.п.
MES1428 RAM used, %	33	38	+5 п.п.
ESR-15R CPU avg (5 мин), %	7	18	+11 п.п.
ESR-15R RAM used, %	20	24	+4 п.п.
SSH time-to-prompt (среднее), с	0,91	1,34	+0,43 с

Снижение пропускной способности на 3,7 % объясняется инспекцией кадров. Прирост числа ретрансмиссий не выходит за пределы нормального разброса. Рост CPU на MES1428 обусловлен обработкой ACL и SNMP-trap; на ESR-15R – активацией stateful firewall и RADIUS-аутентификацией. SSH time-to-prompt увеличился на 0,43 с из-за добавления RADIUS-запроса и смены KEX-алгоритма на diffie-hellman-group14-sha256 (рисунки 15–17).

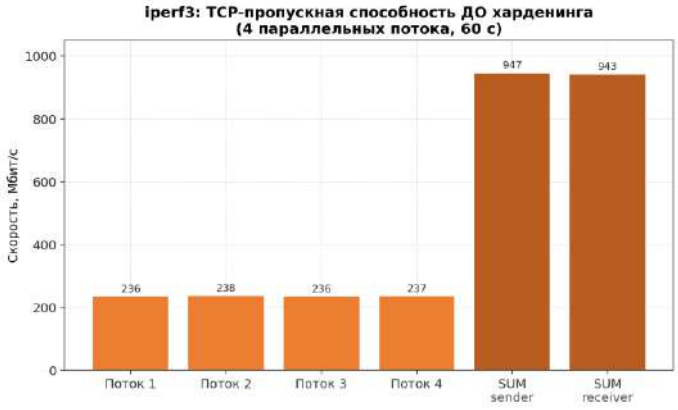


Рис. 15. Iperf3: TCP-пропускная способность до харденинга

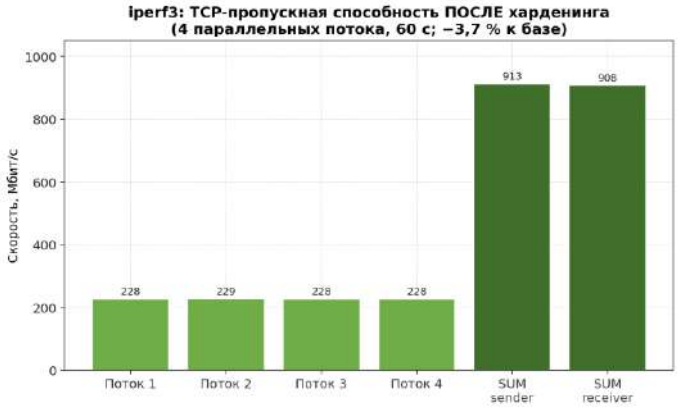


Рис. 16. Iperf3: TCP-пропускная способность после харденинга

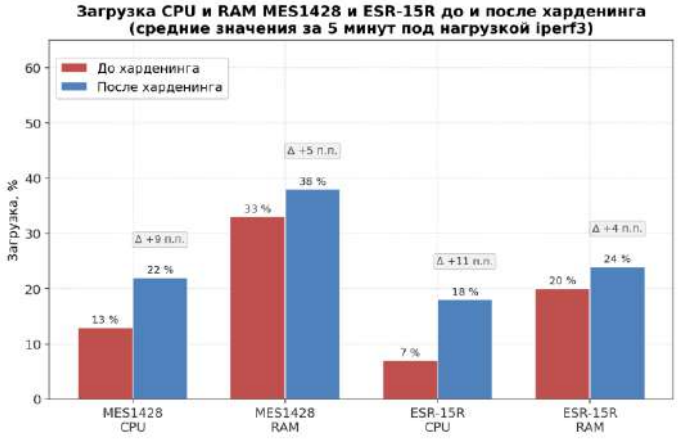


Рис. 17. Загрузка CPU и RAM MES1428 и ESR-15R до и после харденинга

Проверка гипотез

- H1 подтверждена: 91 % контролей (полностью + частично) применимы к MES1428 и 96 % – к ESR-15R, что превышает заявленный порог в 75 %. Из них полностью применимых («+») – 79 % и 89 % соответственно;
- H2 подтверждена: прирост доли заблокированных атак составил 100 п.п. для MES1428 (8/8 векторов) и 100 п.п. для ESR-15R (3/3 вектора), что существенно превышает порог 70 п.п. Для Cisco: +86 п.п. (C2960, 7/7) и +50 п.п. (C1941, 2/2). Меньшее значение для C1941 объясняется тем, что для маршрутизатора применимы только два вектора, один из которых (SNMP) блокируется еще в дефолте;
- H3 подтверждена: снижение пропускной способности TCP на 3,7 % не превышает установленного порога 5 %.

Сравнение Eltex и Cisco в контексте CIS Benchmark

Разрыв в покрытии контролей (~18 п.п. для MES1428 относительно Cisco 2960) в значительной мере объясняется не принципиальными пробелами в защите, а архитектурными

особенностями Eltex NOS. Контроли CIS об отключении small-servers, finger и bootp неприменимы к Eltex, поскольку данные сервисы никогда не реализовывались – соответствующий риск изначально отсутствует.

Реальные отличия сосредоточены в трех областях. Первая – NTP-аутентификация: ESR не проверяет источник NTP по MD5-ключу; остаточный риск (подмена источника времени) частично снижается ограничением NTP на zone-pair. Вторая – длина RSA-ключа: отсутствие явной верификации modulus 2048 на MES означает, что администратор не может подтвердить длину ключа командой show; при этом де-факто ключ генерируется с достаточной длиной. Третья – storm-control action: отсутствие режима shutdown требует ручного вмешательства или настройки SNMP-trap с реакцией NMS.

По применимым векторам атак все четыре устройства после харденинга демонстрируют 100 % блокировку, что свидетельствует о сопоставимой практической эффективности адаптированного перечня и нативного CIS Benchmark.

Ограничения исследования

- Стенд построен на одном экземпляре каждого устройства; результаты могут незначительно отличаться на иных ревизиях аппаратного обеспечения или версиях прошивки;
- Рассмотрено восемь типов атак L2/L3; за рамками остались векторы приложений (BGP hijacking, OSPF injection, DNS spoofing);
- Тест iperf3 измеряет максимальную пропускную способность при однородном TCP-трафике с крупными пакетами и не отражает реальную корпоративную нагрузку (мелкие пакеты, UDP, смешанный профиль). Дополнительный тест со смешанным профилем не проводился;
- MES1428 тестировался с тремя активными портами; на производственных стендах с 24–28 активными портами накладные расходы CPU при DAI и Port Security могут быть выше;
- Сценарий восстановления после errdisable не тестировался в рамках данной работы; рекомендуется верифицировать алгоритм восстановления (shutdown/no shutdown или errdisable-timeout) в production перед внедрением.

Практические рекомендации

На основании результатов исследования сформулированы ключевые рекомендации для специалистов по информационной безопасности, развертывающих оборудование Eltex в корпоративной инфраструктуре.

Плоскость управления (приоритет 1). Отключить Telnet и HTTP, удалить SNMPv1/v2c-сообщества, активировать SNMPv3 (SHA + AES-128), включить RADIUS с fallback на local, задать management ACL (только IP административных станций). Для ESR-15R: ограничить источник NTP-запросов на zone-pair, компенсируя отсутствие MD5-аутентификации NTP.

Плоскость контроля. Включить RSTP с BPDU Guard на всех access-портах и Root Guard на uplink; DHCP Snooping настраивать до DAI. На ESR-15R включить uRPF strict и аутентификацию OSPF MD5 на всех внешних интерфейсах.

Плоскость данных. Port Security: max 1–2 MAC (рабочие станции), действие discard. Неиспользуемые порты: shutdown + BLACKHOLE VLAN (например, VLAN 999). Native VLAN trunk-порта – отличный от VLAN 1 и рабочих VLAN. Межсегментный ACL на ESR-15R обязателен: маршрутизация между VLAN по умолчанию разрешена, что нивелирует сегментацию L2.

Организационные меры. Оформить адаптированный перечень контрольных требований как корпоративный стандарт, проверять соответствие не реже одного раза в квартал (Ansible playbook). Верифицировать перечень при переходе на новую мажорную версию NOS Eltex. Хранить syslog-логи не менее 12 месяцев в соответствии с требованиями регулятора.

При активации механизмов BPDU Guard и Port Security необходимо предусмотреть процедуру восстановления портов из состояния errdisable (например, вручную командой shutdown/no shutdown или автоматически по таймеру errdisable-timeout с помощью команды errdisable recovery cause all interval 300), так как легитимный пользователь может временно

потерять доступ к сети при срабатывании защиты. Рекомендуется протестировать выбранный алгоритм восстановления на тестовом стенде перед внедрением в production.

Заключение

В настоящей работе выполнено систематическое исследование применимости методологии CIS Benchmark к отечественным сетевым устройствам Eltex MES1428 и Eltex ESR-15R. Составлен адаптированный перечень контрольных требований (73 контроля для L2, 58 для L3) по трем плоскостям управления.

Установлено, что 91 % контролей применимы (полностью или частично) к MES1428 и 96 % – к ESR-15R. Из них полностью применимых («+») – 79 % и 89 % соответственно, частично применимых («~») – 12 % и 7 %.

После применения адаптированного перечня доля нейтрализованных атак возросла с 0 % до 100 % для MES1428 (8/8 векторов) и для ESR-15R (3/3 вектора). Важно подчеркнуть, что 100 % относится к выбранному набору из восьми типов атак и не свидетельствует об абсолютной защищенности. Все события безопасности корректно зарегистрированы в централизованном syslog. Пропускная способность TCP снизилась на 3,7 %; загрузка CPU увеличилась на 9–11 п.п.

Все три гипотезы подтверждены, что доказывает практическую применимость методологии CIS Benchmark к отечественному сетевому оборудованию Eltex.

С научной точки зрения показано, что отсутствие официального профиля CIS для конкретного вендора не является препятствием при условии систематического сопоставления контролей с CLI устройства. С практической точки зрения предложенный перечень может служить основой корпоративного стандарта конфигурации Eltex и быть расширен автоматизированными средствами (Ansible, OpenSCAP).

Перспективные направления: верификация перечня на Eltex MES2000/MES3000/ESR-100; расширение набора атак за счет L4/L7-векторов; интеграция с Ansible и OpenSCAP; сравнительный анализ в виртуализированной среде GNS3.

Библиографический список

1. CIS Benchmarks. Center for Internet Security [Электронный ресурс]. – URL: https://www.cisecurity.org/cis-benchmarks (дата обращения: 31.03.2025).
2. CIS Cisco IOS 15 Benchmark v4.1.1 / Center for Internet Security. – East Greenbush, NY : CIS, 2024. – 184 p.
3. Уймин, А. Г. Применение отечественного сетевого оборудования Eltex и EcoRouter в рамках специальности 09.02.06 «Сетевое и системное администрирование» / А. Г. Уймин, И. М. Толмачев // Автоматизация и информатизация ТЭК. – 2025. – № 11(628). – С. 58–62. [Работа обосновывает принципиальную совместимость CLI Eltex и Cisco IOS, что является предпосылкой для переноса контролей CIS.]
4. Документация Eltex MES-series. Версия 10.2 [Электронный ресурс]. – URL: https://docs.eltex-co.ru/display/MES10/ (дата обращения: 31.03.2025).
5. Документация Eltex ESR-series. Версия 1.24 [Электронный ресурс]. – URL: https://docs.eltex-co.ru/display/ED124/ (дата обращения: 31.03.2025).
6. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы : учебник / В. Г. Олифер, Н. А. Олифер. – 5-е изд. – Санкт-Петербург : Питер, 2020. – 1008 с.
7. NIST SP 800-53 Rev. 5. Security and Privacy Controls for Information Systems and Organizations / National Institute of Standards and Technology. – Gaithersburg : NIST, 2020. – 492 p.
8. Уймин, А. Г. Сетевое и системное администрирование : учебно-методическое пособие / А. Г. Уймин. – 3-е изд. – Санкт-Петербург : Лань, 2022. – 480 с.
9. CIS Network Device General Benchmark v2.0.0 / Center for Internet Security. – East Greenbush, NY : CIS, 2024. – 96 p.
10. Nmap: the Network Mapper [Электронный ресурс] / Gordon Lyon. – URL: https://nmap.org/book/man.html (дата обращения: 31.03.2025).

11. Документация iPerf3 [Электронный ресурс]. – URL: https://iperf.fr/ (дата обращения: 31.03.2025).
12. Федеральный закон от 26.07.2017 № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации». – URL: https://www.consultant.ru/document/cons_doc_LAW_220885/ (дата обращения: 31.03.2025).
13. ГОСТ Р 57580.1-2017. Безопасность финансовых (банковских) операций. Защита информации финансовых организаций. – М. : Стандартинформ, 2018.
14. Positive Technologies. Актуальные киберугрозы: итоги 2024 года. – URL: https://www.ptsecurity.com/ru-ru/research/analytics/ (дата обращения: 31.03.2025).
15. RFC 5424. The Syslog Protocol / R. Gerhards. – IETF, March 2009.
16. RFC 4330. Simple Network Time Protocol (SNTP) Version 4 / D. Mills. – IETF, January 2006.
17. FreeRADIUS Project. FreeRADIUS documentation [Электронный ресурс]. – URL: https://freeradius.org/documentation/ (дата обращения: 31.03.2025).
18. Bejtlich, R. The Practice of Network Security Monitoring / R. Bejtlich. – San Francisco : No Starch Press, 2013. – 376 p.
19. Wool, A. A Quantitative Study of Firewall Configuration Errors / A. Wool // IEEE Computer. – 2004. – Vol. 37, No. 6. – P. 62–67.

References

1. CIS Benchmarks. (2025). *CIS Benchmarks*. Center for Internet Security. Retrieved March 31, 2025, from https://www.cisecurity.org/cis-benchmarks (accessed: 31.03.2025).
2. Center for Internet Security. (2024). *CIS Cisco IOS 15 Benchmark v4.1.1*. East Greenbush, NY: CIS. (accessed: 31.03.2025).
3. Uymin, A. G., & Tolmachev, I. M. (2025). Application of domestic network equipment Eltex and EcoRouter within the specialty 09.02.06 "Network and System Administration". *Automation and Informatization of the Fuel and Energy Complex*, (11), 58–62. (accessed: 31.03.2025).
4. Eltex. (2024). *Eltex MES-series Documentation. Version 10.2*. Retrieved March 31, 2025, from https://docs.eltex-co.ru/display/MES10/ (accessed: 31.03.2025).
5. Eltex. (2024). *Eltex ESR-series Documentation. Version 1.24*. Retrieved March 31, 2025, from https://docs.eltex-co.ru/display/ED124/ (accessed: 31.03.2025).
6. Olifer, V. G., & Olifer, N. A. (2020). *Computer Networks. Principles, Technologies, Protocols* (5th ed.). St. Petersburg: Piter. (accessed: 31.03.2025).
7. National Institute of Standards and Technology. (2020). *NIST SP 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations*. Gaithersburg: NIST. (accessed: 31.03.2025).
8. Uymin, A. G. (2022). *Network and System Administration: A Teaching Manual* (3rd ed.). St. Petersburg: Lan. (accessed: 31.03.2025).
9. Center for Internet Security. (2024). *CIS Network Device General Benchmark v2.0.0*. East Greenbush, NY: CIS. (accessed: 31.03.2025).
10. Lyon, G. (2025). *Nmap: the Network Mapper*. Retrieved March 31, 2025, from https://nmap.org/book/man.html (accessed: 31.03.2025).
11. iPerf3. (2025). *iPerf3 Documentation*. Retrieved March 31, 2025, from https://iperf.fr/ (accessed: 31.03.2025).
12. Russian Federation. (2017). *Federal Law No. 187-FZ of July 26, 2017 "On the Security of Critical Information Infrastructure of the Russian Federation"*. Retrieved March 31, 2025, from https://www.consultant.ru/document/cons_doc_LAW_220885/ (accessed: 31.03.2025).
13. GOST R 57580.1-2017. (2018). *Security of financial (banking) operations. Information protection of financial organizations*. Moscow: Standartinform. (accessed: 31.03.2025).
14. Positive Technologies. (2025). *Current Cyber Threats: Results of 2024*. Retrieved March 31, 2025, from https://www.ptsecurity.com/ru-ru/research/analytics/ (accessed: 31.03.2025).
15. Gerhards, R. (2009, March). *RFC 5424: The Syslog Protocol*. IETF. (accessed: 31.03.2025).

16. Mills, D. (2006, January). *RFC 4330: Simple Network Time Protocol (SNTP) Version 4*. IETF. (accessed: 31.03.2025).
17. FreeRADIUS Project. (2025). *FreeRADIUS Documentation*. Retrieved March 31, 2025, from https://freeradius.org/documentation/ (accessed: 31.03.2025).
18. Bejtlich, R. (2013). *The Practice of Network Security Monitoring*. San Francisco: No Starch Press. (accessed: 31.03.2025).
19. Wool, A. (2004). A Quantitative Study of Firewall Configuration Errors. *IEEE Computer*, 37(6), 62–67. (accessed: 31.03.2025).

Информация об авторах

Еремина Елизавета Алексеевна – студент ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: elizavetaeremina2@gmail.com

Простова Алиса Никитична – студент ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: prostrova2005@bk.ru

COMPARATIVE ANALYSIS OF THE EFFECTIVENESS OF APPLYING CIS BENCHMARK FOR ELTEX SWITCHES IN CORPORATE INFRASTRUCTURE

Eremina E. A.[1], Prostova A. N.[1]

[1]National University of Oil and Gas «Gubkin University»

Abstract. The paper addresses the problem of securing undocumented domestic network devices (Eltex MES1428, Eltex ESR-15R) in the absence of official CIS Benchmark profiles. An approach based on projecting controls from CIS Cisco IOS Benchmark v4.1.1 and CIS Network Device General Benchmark v2.0.0 onto three planes — management, control, and data — is proposed. Experimentally, on a physical testbed, default configuration audits, hardening, re-audits, and a series of eight types of active attacks using Kali Linux were performed before and after applying the controls. Attack blocking was verified both by the absence of results for the attacker and by the presence of a corresponding event in the centralized syslog.

It was found that 91 % of the controls are applicable (fully or partially) to the MES1428 and 96 % to the ESR-15R; the share of blocked attacks after hardening over applicable vectors reached 100 % for both devices (up from 0 % in the default configuration), while throughput degradation did not exceed 4 %. The results obtained are compared with the reference Cisco Catalyst 2960 and Cisco 1941 devices.

The proposed methodology has practical significance for import substitution in corporate networks, making it possible to ensure the required level of security on Eltex equipment in the absence of ready-made CIS profiles, and can be replicated for other vendors.

Keywords: CIS Benchmark, hardening, Eltex MES1428, Eltex ESR-15R, import substitution, network security, Kali Linux.

Information about the authors

Eremina Elizaveta Alekseevna – student Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: elizavetaeremina2@gmail.com

Prostova Alisa Nikitichna – student Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: prostrova2005@bk.ru

ПРОФЕССИОНАЛИТЕТ, 2025, № 1 (005)

УДК *004.732:004.056.53*

Л.Г. Хорошилов [1], Н.А. Костин [1]

[1]ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, Российская Федерация

АНАЛИЗ ЭФФЕКТИВНОСТИ ЗАЩИТЫ ОТ АТАКИ DOUBLE TAGGING В ГЕТЕРОГЕННЫХ СЕТЯХ (CISCO, MIKROTIK, ELTEX)

Аннотация: *В статье представлен практический анализ атаки класса VLAN Hopping, реализуемой методом Double Tagging, которая позволяет обходить логическую изоляцию виртуальных локальных сетей, построенных на основе стандарта IEEE 802.1Q. Рассматриваются архитектурные особенности технологии VLAN, связанные с механизмом обработки native VLAN на транковых портах коммутаторов, которые создают предпосылки для реализации данной атаки на канальном уровне модели OSI. Подробно описан принцип формирования Ethernet-кадров с двойным тегированием 802.1Q и механизм их обработки сетевыми устройствами, приводящий к несанкционированной передаче трафика между логически изолированными сегментами сети.*

Экспериментальная часть исследования выполнена на специализированном тестовом стенде с использованием реального сетевого оборудования различных производителей, включая Cisco, MikroTik и Eltex, что позволяет оценить особенности реализации атаки в гетерогенной сетевой среде. Для генерации кадров с двойным тегированием применялся инструмент Yersinia, а фиксация результатов атаки осуществлялась с использованием анализатора сетевого трафика tcpdump.

В рамках работы проведено сравнение поведения оборудования при стандартных (дефолтных) конфигурациях и после применения защитных мер. Показано, что использование native VLAN по умолчанию на транковых портах делает сетевую инфраструктуру уязвимой к атаке Double Tagging независимо от производителя оборудования. На основе полученных результатов сформулированы практические рекомендации по защите, включающие изменение идентификатора native VLAN на неиспользуемый, явную настройку режимов портов и отказ от автоматического согласования транковых соединений. Результаты исследования подтверждают критическую важность корректной конфигурации сетевого оборудования для обеспечения безопасности сегментированных корпоративных сетей.

Ключевые слова: *VLAN Hopping, Double Tagging, сетевая безопасность, тестирование на проникновение, native VLAN, IEEE 802.1Q, сегментация сети.*

Введение

Логическая сегментация с помощью виртуальных локальных сетей (VLAN) на основе стандарта IEEE 802.1Q является фундаментальным механизмом для построения безопасных, производительных и управляемых корпоративных сетей [2]. Она позволяет создавать изолированные домены в рамках единой физической инфраструктуры. Однако сама архитектура VLAN содержит фундаментальные уязвимости, которые позволяют злоумышленнику обходить изоляцию сегментов. Наиболее критичной из таких угроз является атака VLAN Hopping, и особенно ее метод Double Tagging [4], который эксплуатирует особенности обработки native VLAN в сетевых устройствах.

Несмотря на то, что фундаментальные принципы атаки VLAN Hopping методом Double Tagging были описаны ещё в начале 2000-х годов, практическая реализация данной угрозы в современных сетях остается актуальной. Это связано с широким распространением гетерогенных сетевых инфраструктур, в которых одновременно используется оборудование различных вендоров, включая зарубежных (Cisco) и отечественных производителей (Eltex), а также популярные решения класса SMB и SOHO (MikroTik).

В условиях импортозамещения и активного внедрения отечественного сетевого оборудования особую значимость приобретает вопрос корректности типовых (дефолтных) конфигураций и их устойчивости к классическим атакам канального уровня. На практике сетевые администраторы часто воспроизводят шаблоны конфигураций, ориентированные на оборудование Cisco, без учета особенностей реализации VLAN-механизмов у других производителей.

Таким образом, научная новизна данного исследования заключается не в повторном описании механики атаки Double Tagging, а в сравнительном практическом анализе её реализации на современном оборудовании различных вендоров (Cisco, MikroTik, Eltex), а также в оценке универсальности защитных мер в гетерогенной сетевой среде.

Рубрика 2. Методы и системы защиты информации, информационная безопасность

Объект исследования: безопасность использования технологии виртуальных локальных сетей (VLAN) для логической сегментации сетевой инфраструктуры.

Предмет исследования: механизм атаки VLAN Hopping, реализуемый методом Double Tagging, средства тестирования соответствующей уязвимости, а также оценка эффективности мер защиты от подобных угроз.

Цель исследования: выполнить практический анализ атаки Double Tagging и на основе полученных результатов сформировать комплексный подход к защите сетевой инфраструктуры, предусматривающий методику проверки на уязвимость и практические рекомендации по конфигурации для предотвращения данной угрозы.

Литературный обзор.

Для начала необходимо определить ключевые термины и их значения.

Виртуальная локальная сеть (VLAN) – это логическая группа устройств, объединённых в отдельный широковещательный домен, независимо от их физического расположения. VLAN позволяют сегментировать сеть для повышения безопасности, производительности и управляемости.

Стандарт IEEE 802.1Q – основной протокол, определяющий механизм тегирования трафика для передачи информации о принадлежности кадра к определённому VLAN по магистральным (транковым) соединениям. Тег 802.1Q добавляется в заголовок кадра Ethernet и содержит, среди прочего, 12-битный идентификатор VLAN (VID) [1].

Порт доступа (Access Port) – порт коммутатора, предназначенный для подключения конечных узлов сети, таких как компьютеры или конечные устройства. Эти порты образуют нетегированный поток данных. Каждому порту доступа назначается определённая виртуальная локальная сеть (PVID), и все приходящие кадры с этого порта получают метку этого VLAN. Исходящие кадры покидают порт без метки, так как их направление уже определено VLAN [3].

Транковый порт (Trunk Port) – порт коммутатора, предназначенный для создания линии связи (транка) между двумя коммутаторами или между коммутатором и маршрутизатором. Эта линия транспортирует поток данных от нескольких VLAN. На транковых портах происходит маркировка (тегирование) кадров с метками VLAN, что позволяет принимающей стороне различать кадры для разных VLAN [3].

Native VLAN – специально назначенная VLAN на транковом порту, трафик которой передаётся без тега 802.1Q. По умолчанию на многих устройствах в этой роли выступает VLAN 1.

Атака VLAN Hopping – класс атак, целью которых является обход логической изоляции VLAN для получения несанкционированного доступа к трафику другого сегмента сети. Одним из методов реализации является Double Tagging (двойное тегирование) [4].

Следует отметить, что атаки класса VLAN Hopping включают в себя несколько различных векторов, наиболее известными из которых являются Double Tagging и Switch Spoofing (основанный на эксплуатации протокола DTP). В рамках данного исследования рассматривается исключительно атака Double Tagging. Это обусловлено тем, что на большинстве современных коммутаторов, включая исследуемые модели Cisco, MikroTik и Eltex, протокол динамического согласования транков (DTP) либо отключён по умолчанию, либо не реализован вовсе, что делает атаку Switch Spoofing нерелевантной для рассматриваемых конфигураций.

В то же время механизм обработки native VLAN и тегов IEEE 802.1Q сохраняет свою актуальность независимо от вендора и модели оборудования, что обосновывает фокус исследования именно на методе Double Tagging как наиболее универсальном и воспроизводимом в современных сетях.

Атака Double Tagging – техника, при которой злоумышленник формирует кадр Ethernet с двумя вложенными тегами 802.1Q. Первый (внешний) тег соответствует native VLAN на целевом транковом порту, что приводит к его удалению при обработке. Второй (внутренний) тег указывает на VLAN-жертву, в который кадр в итоге и перенаправляется, что позволяет преодолеть границу сегментации [5].

На основе анализа литературы и понимания указанных терминов можно сформулировать следующие гипотезы данного исследования:

1. H1: Типовая конфигурация коммутаторов различных вендоров является уязвимой к атаке Double Tagging, что позволяет осуществить несанкционированную передачу трафика между логически изолированными VLAN.
2. H2: Практическая методика демонстрации атаки Double Tagging на реальном сетевом оборудовании является эффективным инструментом для наглядного представления данной угрозы и может быть использована в учебном процессе для подготовки специалистов в области информационной безопасности.

Методы исследования

Исследование является экспериментальным и будет проводиться на специализированном тестовом стенде.

Характеристика среды исследования: сеть включает физические коммутаторы различных вендоров.

Узел 1 (жертва) – компьютер с операционной системой Kali Linux.

Узел 2 (атакующий) – компьютер с операционной системой Kali Linux.

Методы сбора данных:

1. Анализ и настройка конфигураций сетевого оборудования (уязвимой и защищённой).
2. Активное тестирование уязвимостей с использованием специализированного инструментария. В ходе эксперимента применялась утилита Yersinia, позволяющая генерировать сетевые кадры с двумя вложенными 802.1Q тегами (Double Tagging).
3. Пассивный мониторинг сетевого трафика с помощью анализатора tcpdump для фиксации этапов атаки и подтверждения её успешности.

Описание процедуры проведения исследования: на узле-жертве, находящемся в целевом VLAN (VLAN 10), запускается инструмент захвата трафика (tcpdump) для мониторинга входящих кадров, в то время как на узле-атакующем, физически подключенном к другому VLAN (VLAN 20), с помощью инструмента Yersinia реализуется атака Double Tagging. Данная атака заключается в отправке сформированных кадров Ethernet с двумя вложенными тегами 802.1Q (внешний — native VLAN, внутренний — целевой VLAN), что приводит к несанкционированной доставке трафика в изолированный VLAN-сегмент.

Эксперимент проводится в несколько этапов. Подготовительный этап направлен на настройку необходимого сетевого оборудования (коммутаторов различных вендоров) с созданием уязвимой конфигурации, при которой native VLAN на транковых портах оставлена равной значению по умолчанию (VLAN 1). Основным этапом является непосредственное проведение атаки Double Tagging и захват её результатов на узле-жертве. Заключительным этапом становится анализ полученных данных, проверка эффективности мер защиты (изменения native VLAN на неиспользуемый идентификатор) и подведение итогов эксперимента.

Методы обработки данных: Сравнительный и качественный анализ, направленный на сопоставление поведения оборудования разных вендоров, интерпретация данных сетевого трафика, обобщение результатов для подтверждения выдвинутых гипотез.

Ход исследования

Для начала необходимо настроить наши коммутаторы. Отметим, что настройка производилась отдельно для каждого этапа эксперимента с участием конкретного коммутатора определенного вендора в рамках топологии, состоящей из двух компьютеров и двух коммутаторов.

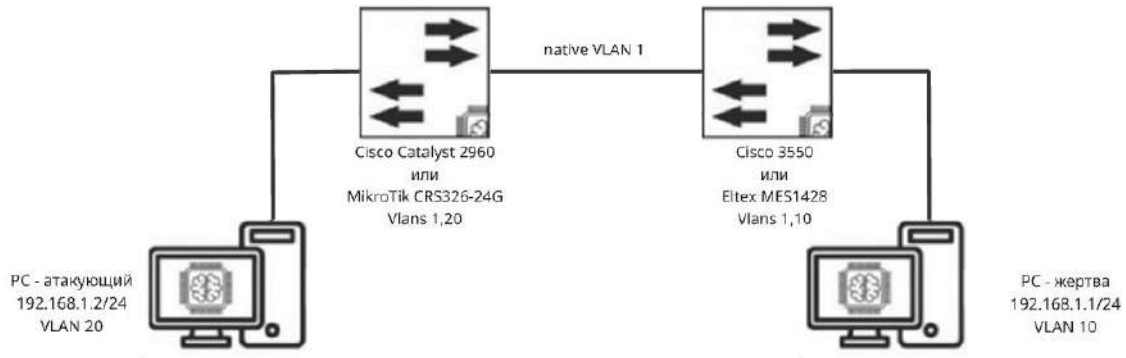


Рис. 1 Топология экспериментального стенда для проверки атаки Double Tagging между VLAN 20 и VLAN 10

Для начала рассмотрим настройку Cisco Catalyst 2960 и Cisco 3550.

Начнём с объяснения настройки коммутатора Cisco Catalyst 2960. Создадим VLAN 20, настроим два порта, один будет в режиме access, так как ведет к конечному узлу атакующего, другой в режиме trunk. Также у транкового порта явно укажем, что в качестве native VLAN берём VLAN 1. Конфигурация приведена в листинге 1.

Листинг 1 – Конфигурация коммутатора Cisco Catalyst 2960

```
conf t
vlan 20
 name VLAN20_V
 exit
int fa0/21
 switchport mode access
 switchport access vlan 20
 no shutdown
 exit
int fa0/14
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport trunk native vlan 1
 switchport trunk allowed vlan 1,20
 no shutdown
 exit
```

Теперь перейдем к настройке коммутатора Cisco 3550. Аналогично, создадим VLAN 10, настроим два порта, один будет в режиме access, так как ведет к конечному узлу-жертве, другой в режиме trunk. Также, у транкового порта явно укажем, что в качестве native VLAN берём VLAN 1. Конфигурация приведена в листинге 2.

Листинг 2 – Конфигурация коммутатора Cisco Catalyst 3550

```
conf t
vlan 10
 name VLAN10_V
 exit
int fa0/13
 switchport mode access
 switchport access vlan 10
 no shutdown
 exit
int fa0/21
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport trunk native vlan 1
 switchport trunk allowed vlan 1,10
 no shutdown
```

```
exit
```

Теперь после настройки коммутаторов, перейдём к настройке PC. На атакующем PC настроим IP адрес (access порт VLAN 20). Настройка представлена на листинге 3.

Листинг 3 – Настройка IP-адреса на атакующем ПК

```
ip addr flush dev eth0
ip addr add dev eth0 192.168.1.2/24
```

На ПК-жертве настройка IP (VLAN 10) аналогичная, только IP здесь: 192.168.1.1/24.

Для запуска самой атаки потребуется специальная утилита Yersinia (интерфейс представлен на рисунке 2), широко применяемая в практиках тестирования безопасности сетей канального уровня [6].

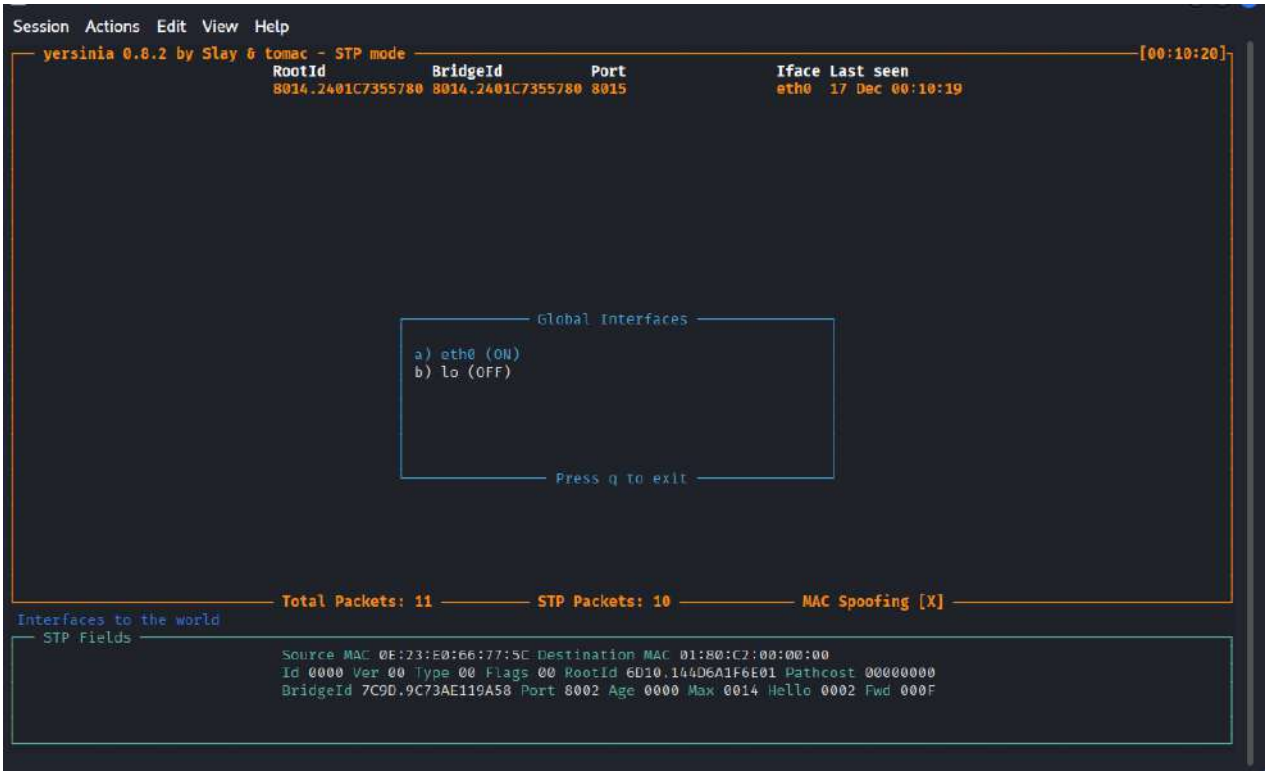


Рис. 2 Интерфейс утилиты Yersinia, используемой для формирования кадров с двойным тегированием 802.1Q

Для её запуска в консоли прописать команду, которая представлена на листинге 4.

Листинг 4 – Запуск приложения Yersinia

```
yersinia -I
```

Прежде чем запускать саму атаку, нужно убедиться, что в качестве интерфейса выбран нужный нам (в данном случае это eth0). Далее, нужно выбрать, какую именно атаку мы хотим совершить. Для этого нажимаем клавишу G и появляется иконка выбора протокола (Рисунок 3). Выбираем 802.1Q и нас перебросит в меню атаки.

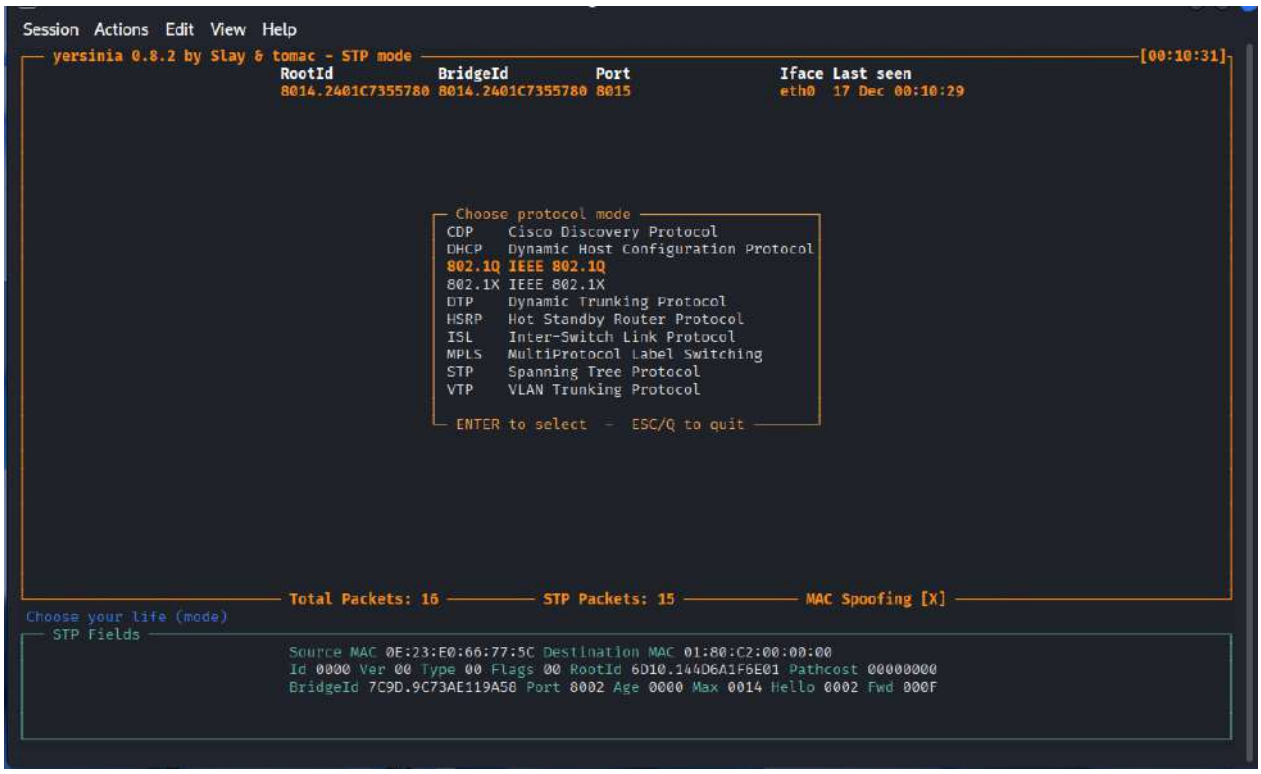


Рис. 3 Выбор протокола 802.1Q в утилите Yersinia для настройки атаки Double Tagging

Теперь, нажимаем клавишу E, чтобы отредактировать поля протокола 802.1Q. В качестве VLAN (внешнего) выбираем наш native VLAN 1, а внутренний VLAN указываем 10. Также важно указать Src IP – 192.168.1.2 (атакующий) и Dst IP – 192.168.1.1 (жертва). После этого выходим из меню настройки нажатием Esc и производим саму атаку. Для этого, нажимаем клавишу x и выбираем нужную нам Double tagging, нажав на 1. Настройка представлена на рисунке 4.

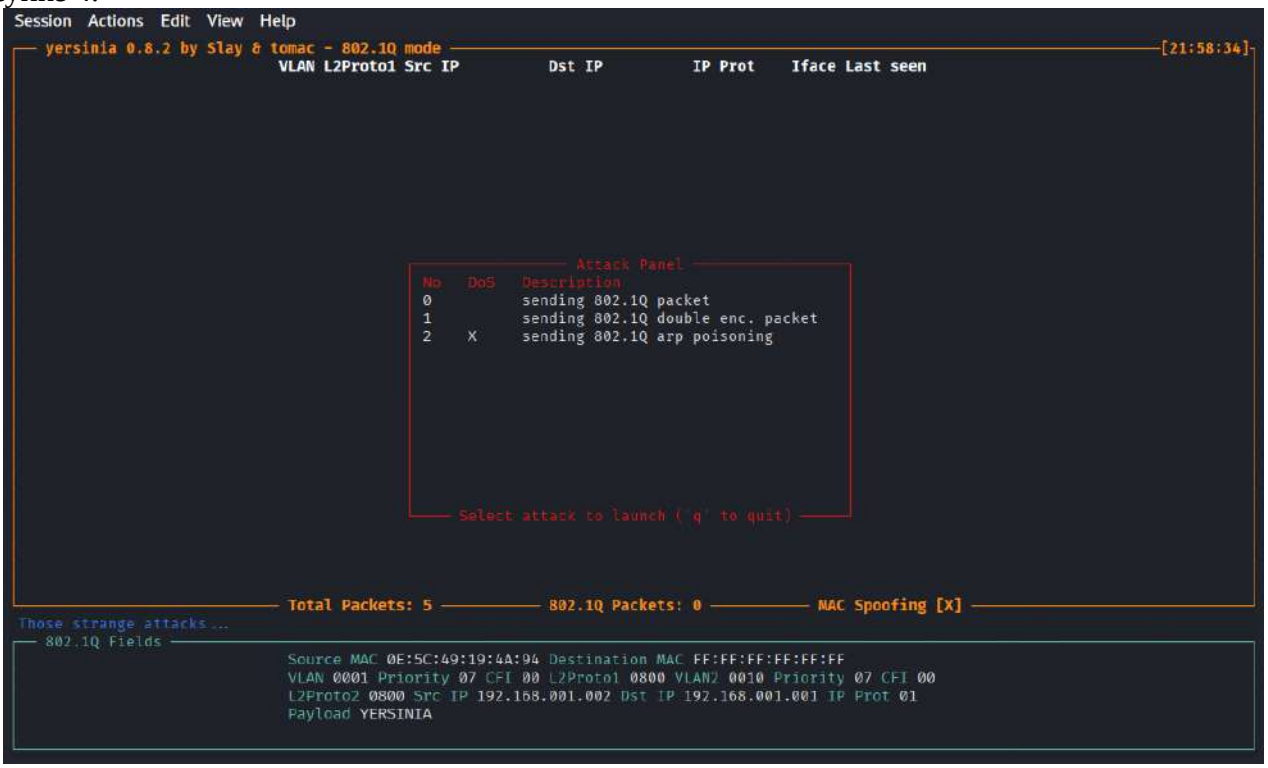


Рис. 4 Настройка полей 802.1Q в Yersinia с указанием внешнего native VLAN 1 и внутреннего целевого VLAN 10

После выполнения предыдущих шагов, мы успешно отправили один пакет, для более детального анализа отправим еще несколько пакетов. Для того, чтобы посмотреть, отправляются ли у нас пакеты, в консоли воспользуемся утилитой tcpdump (Листинг 5).

Листинг 5 – Фиксация отправки кадров с двойным тегированием на атакующем узле

```
tcpdump -i eth0 -e -v vlan
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
192.168.1.2 > 192.168.1.1: ICMP echo request, id 66, seq 66, length 16
12:41:03.112233 0e:5c:49:19:4a:94 (oui Unknown) > Broadcast, ethertype 802.1Q (0x8100), length 58: vlan 1, p 7, ethertype 802.1Q (0x8100), vlan 10, p 7, ethertype IPv4 (0x8000), (tos 0x0, ttl 64, id 66, offset 0, flags [none], proto ICMP (1), length 36)
```

Как видно из представленных данных, пакеты с двойным тегом, где внутренний — VLAN 10, а внешний VLAN 1 успешно отправляются. Теперь, воспользуемся этой же утилитой, чтобы посмотреть доходят ли пакеты до узла-жертвы.

Листинг 6 – Фиксация ICMP-трафика на узле-жертве при успешной атаке

```
tcpdump -i eth0 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
12:41:03.115678 IP 192.168.1.2 > 192.168.1.1: ICMP echo request, id 66, seq 66, length 16
```

Как видно из представленных данных, пакеты успешно достигают узла-жертвы, что говорит об успешном проведении атаки. Теперь, перейдем к защите. Для этого вернемся к настройкам коммутаторов Cisco и изменим ID native VLAN на неиспользуемый 999 (Листинг 7).

Листинг 7 – Изменение native VLAN на транковом порту Cisco Catalyst 3550

```
conf t
int fa0/21
 switchport trunk native vlan 999
 exit
```

На Cisco Catalyst 2960 аналогичным образом изменим ID native VLAN на неиспользуемый 999. Теперь попробуем снова запустить атаку. Как видно на листинге 8 пакеты не доходят до жертвы, что говорит об успешной защите нашей сети от данного вида атаки.

Листинг 8 – Результат повторной атаки после изменения native VLAN

```
tcpdump -i eth0 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C
0 packets captured
```

Теперь, проделаем данный эксперимент с коммутаторами MikroTik CRS326-24G и Eltex MES1428.

Начнём с объяснения настройки коммутатора MikroTik CRS326-24G. Создадим bridge – виртуальный коммутатор внутри MikroTik, который объединяет физические порты. И добавим к нему два порта, один из которых находится в VLAN 20 режиме access и другой в режиме trunk. Последнему мы указываем untagged для vlan-ids 1, что делает его native VLAN для этого порта. Конфигурация приведена в листинге 9.

Листинг 9 – Конфигурация VLAN на коммутаторе MikroTik CRS326

```
/int br port add interface=ether16 bridge=test
/int br port add interface=ether21 bridge=test
/int br vlan add bridge=test untagged=ether16,ether21
vlan-ids:1
/int bridge port set [find interface=ether16] pvid=1
/int bridge port set [find interface=ether21] pvid=20
/int bridge set test vlan-filtering=yes
```

Теперь перейдем к настройке коммутатора Eltex MES1428. Аналогично, создадим VLAN 10, настроим два порта, один будет в режиме access, так как ведет к конечному узлу-жертве, другой в режиме trunk. Также, у транкового порта явно укажем, что в качестве native VLAN берём VLAN 1. Конфигурация приведена в листинге 10.

Листинг 10 – Конфигурация VLAN на коммутаторе Eltex MES1428

```
conf t
```

```
vlan 1
 exit
vlan 10
 exit
int fa0/21
 switchport mode access
 switchport access vlan 10
 no shutdown
 exit
int fa0/13
 switchport mode general
 switchport general allowed vlan add 1 untagged
 switchport general pvid 1
 no shutdown
 exit
```

После настройки коммутаторов, перейдем на атакующий PC и проведем с него атаку Double Tagging с помощью утилиты Yersinia. Настройки аналогичные, что были на этапе атаки коммутаторов Cisco. Для того, чтобы посмотреть отправляются ли у нас пакеты, в консоли воспользуемся утилитой tcpdump.

Листинг 11 – Фиксация отправки кадров с двойным тегированием на атакующем узле

```
tcpdump -i eth0 -e -v vlan
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
192.168.1.2 > 192.168.1.1: ICMP echo request, id 66, seq 66, length 16
12:57:54.250734 0e:5c:49:19:4a:94 (oui Unknown) > Broadcast, ethertype 802.1Q (0x8100), length 58: vlan 1, p 7, ethertype 802.1Q (0x8100), vlan 10, p 7, ethertype IPv4 (0x8000), (tos 0x0, ttl 64, id 66, offset 0, flags [none], proto ICMP (1), length 36)
```

Как видно из представленных данных, пакеты с двойным тегом успешно отправляются. Также на ПК-жертве посмотрим доходят ли до него пакеты.

Листинг 12 – Фиксация ICMP-трафика на узле-жертве при успешной атаке

```
tcpdump -i eth0 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
12:57:54.251167 IP 192.168.1.2 > 192.168.1.1: ICMP echo request, id 66, seq 66, length 16
```

Как видно из представленных данных, пакеты успешно достигают узла-жертвы, что говорит об успешном проведении атаки. Теперь, перейдем к защите. Для этого вернемся к настройкам коммутаторов и изменим ID native VLAN на неиспользуемый 999. Настройка MikroTik CRS326-24G приведена на листинге 13.

Листинг 13 – Изменение native VLAN на коммутаторе MikroTik CRS326

```
/interface bridge vlan set [find vlan-ids 1] vlan-ids=999 untagged=ether16,ether21
```

Таким образом, в конфигурации коммутатора MikroTik идентификатор нативной VLAN был изменён со стандартного значения «1» на неиспользуемый «999» для минимизации рисков, связанных с использованием стандартного нативного VLAN. Порт ether16 сконфигурирован как нетегированный (untagged) для VLAN 999, выполняя роль транкового порта (trunk), который передаёт трафик с соответствующими тегами VLAN. Аналогичная процедура была применена к коммутатору Eltex MES1428, где ID нативной VLAN также был заменён на неиспользуемый — 999. Соответствующие настройки для устройства Eltex представлены на листинге 14.

Листинг 14 – Изменение native VLAN на коммутаторе Eltex MES1428

```
int fa0/13
 switchport general pvid 999
 switchport general allowed vlan add 999 untagged
 exit
```

Теперь попробуем снова запустить атаку, но теперь как видно из листинга 15 пакеты не доходят до жертвы, что говорит об успешной защите нашей сети от данного вида атаки.

Листинг 15 – Результат повторной атаки после изменения native VLAN

```
tcpdump -i eth0 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

```
^C
0 packets captured
```

Результаты исследования

Необходимо понять, как именно работает атака Double Tagging: атакующий узел, находящийся в VLAN 20, с помощью инструмента Yersinia отправляет сформированные кадры Ethernet с двойным тегированием 802.1Q. При этом первый (внешний) тег соответствует native VLAN на транковом порту коммутатора (по умолчанию VLAN 1), а второй (внутренний) тег указывает на целевой VLAN (VLAN 10). Первый коммутатор, получив такой кадр с порта доступа, пересылает его на транк, где удаляет внешний тег, так как он совпадает с native VLAN. Следующее устройство получает кадр с единственным тегом VLAN 10 и, считая его легитимным, перенаправляет в целевой сегмент сети.

В процессе тестирования на всех коммутаторах при конфигурации по умолчанию (native VLAN = 1 на транковых портах) атака Double Tagging была успешно реализована. На узле-жертве, расположенном в изолированном VLAN 10, с помощью анализатора трафика tcpdump был зафиксирован ICMP-трафик, отправленный атакующим узлом из VLAN 20. Это прямое доказательство того, что граница логической сегментации была преодолена.

Наиболее эффективной мерой защиты от атаки Double Tagging, проверенной в ходе эксперимента, является изменение идентификатора native VLAN на транковых портах с используемого по умолчанию (VLAN 1) на неиспользуемый (например, VLAN 999). После применения этой настройки на всех тестируемых коммутаторах повторная попытка проведения атаки завершилась неудачей: сформированные кадры с двойным тегом более не достигали узла-жертвы, так как первый коммутатор больше не удалял внешний тег при отправке по транку.

Таким образом, результаты наглядно демонстрируют, что типовая конфигурация оборудования различных производителей не обеспечивает защиту от атаки Double Tagging, однако простая корректировка настройки native VLAN является универсальным и высокоэффективным методом защиты.

Заключение

В рамках исследования был проведён эксперимент, направленный на изучение уязвимости стандартных конфигураций коммутаторов различных вендоров к атаке типа VLAN Hopping (метод Double Tagging). На специализированном тестовом стенде с использованием реального оборудования (Cisco, Eltex, MikroTik) была успешно смоделирована атака с применением инструмента Yersinia. В ходе эксперимента осуществлялся мониторинг сетевого трафика и анализ поведения оборудования, что позволило на практике подтвердить возможность обхода логической изоляции VLAN.

Результат проверки гипотез:

1. Гипотеза H1 подтвердилась полностью. Экспериментально установлено, что типовая конфигурация протестированных коммутаторов (Cisco Catalyst 2960, Eltex MES1428, MikroTik CRS326) является уязвимой к атаке Double Tagging. При стандартных настройках, когда на транковых портах используется native VLAN по умолчанию (VLAN 1), оборудование не препятствует отправке кадров с двойным тегированием, что позволяет злоумышленнику успешно передавать трафик между логически изолированными VLAN.
2. Гипотеза H2 подтвердилась полностью. Разработанная и применённая методика практической демонстрации атаки Double Tagging на реальном сетевом оборудовании показала высокую наглядность и эффективность. Простота воспроизведения атаки в сочетании с убедительностью полученных результатов (прямой перехват трафика между VLAN) делает данную методику ценным инструментом для наглядного представления внутренних угроз информационной безопасности. Это открывает перспективы её использования в учебном процессе для подготовки специалистов.

Направления дальнейшего исследования: разработка автоматизированных систем обнаружения атак VLAN Hopping, анализ устойчивости современных протоколов сетевой виртуализации и программно-определяемых сетей (SDN) к атакам обхода сегментации на канальном

уровне, изучение новых векторов атак, связанных с IPv6 и современными протоколами сетевой виртуализации.

Проведённое исследование подтвердило, что атака VLAN Hopping, реализуемая методом Double Tagging, остается актуальной угрозой, корень которой лежит в ошибках базовой конфигурации, а не в недостатках оборудования. Полученные результаты имеют практическую ценность для системных администраторов и специалистов по безопасности, демонстрируя критическую важность даже простых, но правильных настроек для обеспечения надёжной защиты сетевой инфраструктуры.

Библиографический список

1. IEEE Standard 802.1Q-2018 - Bridges and Bridged Networks [Электронный ресурс] // IEEE Standards Association. – 2018. – URL: https://standards.ieee.org/standard/802_1Q-2018.html (дата обращения: 15.12.2025).
2. Cisco Systems, Inc. Understanding and Configuring VLAN Trunk Protocol (VTP) // Cisco Technical Documentation. – 2023. – URL: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960/software/release/12-2_55_se/configuration/guide/scg_2960/swvtp.html (дата обращения: 15.12.2025).
3. Уймин, А. Г. Компьютерные сети. L2-технологии : практикум для СПО / А. Г. Уймин. — Саратов ; Москва : Профобразование, Ай Пи Ар Медиа, 2024. — 190 с. — ISBN 978-5-4497-2559-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: https://www.iprbookshop.ru/135231.html (дата обращения: 13.12.2025).
4. Convery, S. Hacking Layer 2: Fun with Ethernet Switches // Black Hat USA 2002. – 2002. – URL: https://blackhat.com/presentations/bh-usa-02/bh-us-02-convery-switches.pdf (дата обращения: 11.12.2025).
5. Knobbe, F. VLAN Security // SANS Institute InfoSec Reading Room. – 2002. – URL: https://www.sans.org/reading-room/whitepapers/protocols/vlan-security-853 (дата обращения: 12.12.2025).
6. SANS Institute. Network Penetration Testing Survey 2023 [Электронный ресурс] // SANS Survey Report. - 2023. - URL: https://www.sans.org/reading-room/whitepapers/survey/network-penetration-testing-survey-2023-39845 (дата обращения: 12.12.2025)
7. MikroTik Documentation. VLAN Security and Bridge Filtering // MikroTik Official Documentation. – 2024. – URL: https://help.mikrotik.com/docs/display/ROS/VLAN (дата обращения: 14.12.2025).

References

1. IEEE Standards Association. *IEEE Standard 802.1Q-2018 — Bridges and Bridged Networks* [Electronic resource]. IEEE Standards Association, 2018. URL: https://standards.ieee.org/standard/802_1Q-2018.html (access date: 15.12.2025).
2. Cisco Systems, Inc. Understanding and Configuring VLAN Trunk Protocol (VTP) // *Cisco Technical Documentation*. 2023. URL: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960/software/release/12-2_55_se/configuration/guide/scg_2960/swvtp.html (access date: 15.12.2025).
3. Uymin, A. G. *Computer Networks. Layer 2 Technologies: Practical Guide for Secondary Vocational Education*. Saratov; Moscow: Profobrazovanie, IPR Media, 2024. 190 p. ISBN 978-5-4497-2559-2. Electronic resource. URL: https://www.iprbookshop.ru/135231.html (access date: 13.12.2025).
4. Convery, S. Hacking Layer 2: Fun with Ethernet Switches // *Proceedings of Black Hat USA 2002*. 2002. URL: https://blackhat.com/presentations/bh-usa-02/bh-us-02-convery-switches.pdf (access date: 11.12.2025).
5. Knobbe, F. VLAN Security // *SANS Institute InfoSec Reading Room*. 2002. URL: https://www.sans.org/reading-room/whitepapers/protocols/vlan-security-853 (access date: 12.12.2025).

6. SANS Institute. Network Penetration Testing Survey 2023 [Electronic resource] // *SANS Survey Report*. 2023. URL: https://www.sans.org/reading-room/whitepapers/survey/network-penetration-testing-survey-2023-39845 (access date: 12.12.2025).
7. MikroTik. VLAN Security and Bridge Filtering // *MikroTik Official Documentation*. 2024. URL: https://help.mikrotik.com/docs/display/ROS/VLAN (access date: 14.12.2025).

Информация об авторах

Хорошилов Лев Геннадьевич — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: levyahorosh@yandex.ru

Костин Никита Андреевич — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: knikita200558@mail.ru

ANALYSIS OF PROTECTION EFFECTIVENESS AGAINST DOUBLE TAGGING ATTACKS IN HETEROGENEOUS NETWORK ENVIRONMENTS (CISCO, MIKROTIK, ELTEX)

Khoroshilov L. G.[1], Kostin N. A.[1]

[1]National University of Oil and Gas «Gubkin University»

Abstract. This paper presents a practical analysis of a VLAN Hopping attack implemented using the Double Tagging technique, which allows attackers to bypass the logical isolation of virtual local area networks based on the IEEE 802.1Q standard. The study examines architectural characteristics of VLAN technology related to native VLAN processing on trunk ports of Ethernet switches that create conditions for exploiting Layer 2 vulnerabilities. The mechanism of forming Ethernet frames with double 802.1Q tagging is described in detail, along with the frame forwarding logic that allows such packets to be delivered into a target VLAN segment without authorization.

The experimental part of the research was conducted on a dedicated testbed using real network equipment from multiple vendors, including Cisco, MikroTik, and Eltex. This heterogeneous environment allows evaluation of device behavior under identical attack conditions. The Yersinia utility was used to generate Ethernet frames with double tagging, while network traffic was captured and analyzed using the tcpdump tool to confirm successful delivery of traffic to a protected VLAN segment.

The study compares device behavior under default configurations and after applying standard security countermeasures. The results demonstrate that the default native VLAN configuration on trunk ports makes network infrastructures vulnerable to Double Tagging attacks regardless of the equipment vendor. Based on the experimental findings, practical recommendations for improving VLAN-based network segmentation security are proposed, including assigning a non-default native VLAN, explicitly configuring port modes, and disabling automatic trunk negotiation mechanisms. The research highlights the critical importance of correct Layer 2 configuration practices for ensuring secure segmentation in modern heterogeneous network environments.

Keywords: VLAN Hopping, Double Tagging, Network Security, Penetration Testing, native VLAN, IEEE 802.1Q, Network Segmentation.

Information about the authors

Khoroshilov Lev Gennadievich — student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: levyahorosh@yandex.ru

Kostin Nikita Andreevich — student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: knikita200558@mail.ru

Рубрика 2. Методы и системы защиты информации, информационная безопасность

УДК 004.056.5:004.93

А. Г. Уймин[1]

[1]ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, Российская Федерация

ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В СИСТЕМАХ БИОМЕТРИЧЕСКОЙ АУТЕНТИФИКАЦИИ: ВОЗМОЖНОСТИ И УЯЗВИМОСТИ

Аннотация. *Статья посвящена системному анализу возможностей и уязвимостей, возникающих при применении алгоритмов искусственного интеллекта в современных системах биометрической аутентификации. Актуальность исследования обусловлена качественным сдвигом в архитектуре цифровой идентичности к середине 2020-х годов: внедрением стандартов WebAuthn Level 3 и passkeys, ужесточением нормативно-методических требований EU AI Act и NIST SP 800-63B-4, а также стремительным развитием атак на основе морфинга и дипфейков. В статье рассматриваются ключевые биометрические модальности — распознавание лиц, отпечатков пальцев (включая бесконтактные технологии), радужки и сетчатки глаза — в контексте применения нейросетевых архитектур (CNN, ResNet/DRN). Анализируются метрики качества биометрических систем (FPIR, FNIR, APCER, BPCER, IAPMR) и международные бенчмарки FRVT и IREX. Отдельное внимание уделяется угрозам: атакам предъявления (PA), морфингу, дипфейкам, бесконтактному спуфингу, а также методам противодействия (PAD, MAD). Рассматривается архитектура passkeys/WebAuthn L3 как модель разделения биометрической верификации и криптографической аутентификации. Предложена методология оценки и снижения рисков, совместимая с требованиями EU AI Act, ISO/IEC 30107-3 и NIST SP 800-63B-4. Уточнены архитектурные и организационные меры, позволяющие связать технические метрики с требованиями управления рисками. Сделан вывод о том, что дальнейшее развитие биометрических систем определяется не ростом вычислительных мощностей, а способностью объединить интеллектуальную обработку признаков, этические ограничения и инженерную воспроизводимость в рамках подотчётных и верифицируемых архитектур.*

Ключевые слова: *биометрическая аутентификация, искусственный интеллект, атака предъявления, морфинг, дипфейк, WebAuthn, EU AI Act.*

Введение

Биометрическая аутентификация как процедура верификации личности на основе уникальных физиологических или поведенческих характеристик человека к середине 2020-х годов претерпевает качественную трансформацию. В предшествующее десятилетие основным вектором развития отрасли являлось повышение точности распознавания за счёт наращивания вычислительных мощностей и объёмов обучающих данных. К 2025 году актуализированы принципиально иные исследовательские и инженерные задачи: обеспечение регуляторной совместимости биометрических систем, их управляемости и подотчётности в условиях применения алгоритмов искусственного интеллекта (ИИ).

Указанный сдвиг обусловлен совокупностью взаимосвязанных факторов. Первый фактор: принятие Европейским союзом Акта об искусственном интеллекте (EU AI Act) [1] и обновление руководства NIST по цифровой идентичности (SP 800-63B-4) [2] сформировали новые нормативно-методические рамки для биометрических систем, применяемых в критически важных сценариях идентификации и аутентификации. Второй фактор: интенсивное развитие генеративных нейросетевых моделей породило новый класс угроз — морфинг и дипфейки. Указанные типы атак принципиально отличаются от традиционных атак предъявления по векторам реализации и требуют специализированных методов противодействия [3, 4]. Третий фактор: принятие стандарта WebAuthn Level 3 консорциумом W3C и широкое распространение технологии passkeys существенно изменили архитектуру цифровой идентичности. В рамках указанных архитектурных сценариев биометрические данные функционально переориентированы: вместо самостоятельного сетевого фактора аутентификации они приобретают статус локального механизма разблокировки криптографического ключа [5].

Совокупность этих факторов определяет актуальность системного анализа возможностей и уязвимостей ИИ-алгоритмов в биометрических системах аутентификации. Существующие обзорные работы, как правило, сосредоточены либо на технических аспектах

распознавания [6, 7], либо на отдельных видах атак [8, 9], либо на регуляторных вопросах [10]. Комплексного рассмотрения, охватывающего одновременно технические возможности, актуальные угрозы, метрики оценки и регуляторный контекст, в отечественной литературе явно недостаточно.

Цель настоящей статьи — сформулировать и обосновать системное понимание возможностей и уязвимостей, возникающих при применении алгоритмов ИИ в биометрических системах аутентификации, и предложить практически применимую методологию оценки, снижения рисков и обеспечения соответствия регуляторным требованиям.

Для достижения поставленной цели решаются следующие задачи:

- охарактеризовать современное состояние биометрической аутентификации с учётом роли ИИ-алгоритмов;
- систематизировать ключевые биометрические модальности и применяемые нейросетевые архитектуры;
- проанализировать метрики качества и международные бенчмарки оценки биометрических систем;
- классифицировать актуальные угрозы и методы противодействия;
- рассмотреть архитектуру passkeys/WebAuthn L3 как модель разделения биометрической и криптографической составляющих аутентификации;
- предложить методологию оценки рисков, совместимую с действующими регуляторными требованиями.

Материалы и методы

Исследование выполнено методами системного анализа, сравнительного анализа технических стандартов и регуляторных документов, а также обзора актуальной научно-технической литературы. В качестве основных источников использованы: международные стандарты ISO/IEC 30107-3 [11], ANSI/NIST ITL [12], руководство NIST SP 800-63B-4 [2], регламент EU AI Act [1], технические спецификации W3C WebAuthn Level 3 [5], а также результаты бенчмарков FRVT (Face Recognition Vendor Test) и IREX (Iris Exchange), публикуемых Национальным институтом стандартов и технологий США (NIST) [13, 14].

Анализ угроз проводился в соответствии с таксономией атак предъявления, закреплённой в ISO/IEC 30107-3, с учётом расширений для морфинга и дипфейков, предложенных в работах [3, 4, 8]. Архитектурный анализ passkeys/WebAuthn L3 основан на спецификациях FIDO Alliance [15] и W3C [5]. Регуляторный анализ выполнен путём сопоставления требований EU AI Act, NIST SP 800-63B-4 и ISO/IEC 30107-3 применительно к биометрическим системам высокого риска.

Результаты

1. Биометрическая аутентификация в 2025 году: качественный сдвиг

К 2025 году биометрическая аутентификация прошла путь от экспериментальных систем с высоким уровнем ошибок до промышленных решений, обеспечивающих точность распознавания лиц на уровне, превышающем возможности человека-эксперта в контролируемых условиях [6]. Этот прогресс стал возможным благодаря переходу от традиционных алгоритмов обработки изображений к глубоким нейронным сетям — прежде всего свёрточным нейронным сетям (CNN) и архитектурам с остаточными связями (ResNet, DRN) [7].

Вместе с тем качественный сдвиг 2024–2025 годов состоит не в очередном улучшении метрик точности, а в изменении самой парадигмы применения биометрии. Можно выделить пять ключевых тенденций.

Сводная схема взаимосвязи биометрических модальностей, требований и инструментов оценки представлена на рис. 1.

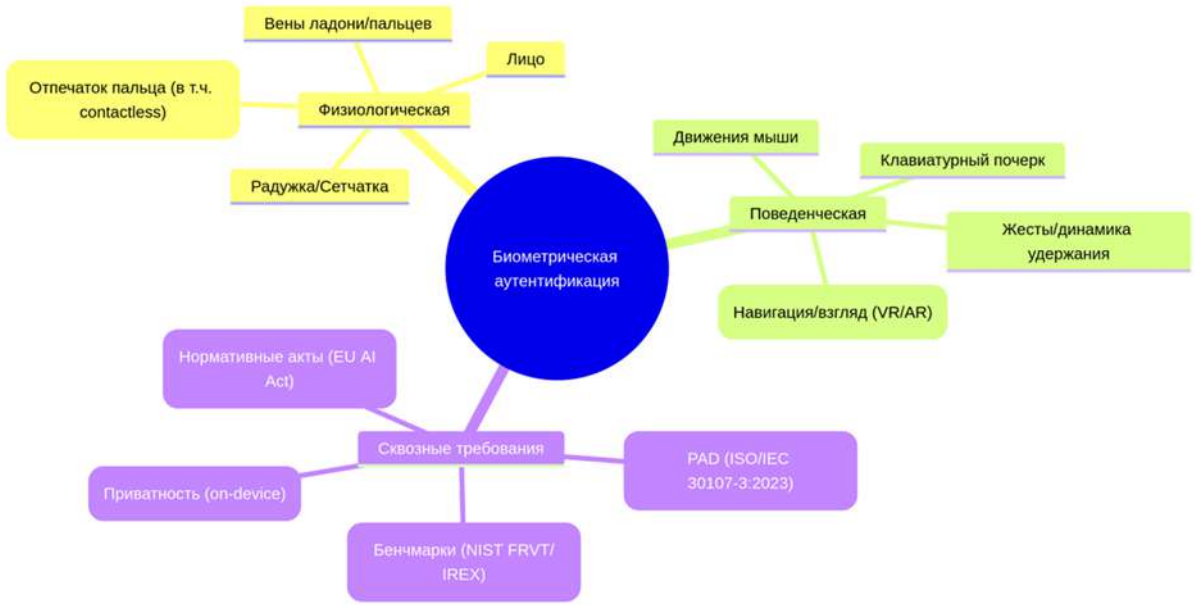


Рис. 1. Биометрическая аутентификация в 2025 году: модальности, требования и инструменты оценки

Тенденция 1: переход к passkeys и WebAuthn L3. Широкое внедрение технологии passkeys, основанной на стандарте WebAuthn Level 3 консорциума W3C, принципиально изменило роль биометрии в архитектуре аутентификации. В модели passkeys биометрическая верификация (например, сканирование отпечатка пальца или лица) выполняется локально на устройстве пользователя и служит исключительно для разблокировки криптографического ключа, хранящегося в защищённом хранилище устройства (platform authenticator). Сервер (Relying Party) не получает биометрические данные: при входе он проверяет криптографическое подтверждение владения приватным ключом, а сведения о свойствах аутентификатора могут дополнительно подтверждаться механизмом аттестации [5, 15]. Это принципиально отличается от традиционных схем, в которых биометрический шаблон передавался на сервер для сопоставления.

Тенденция 2: ужесточение регуляторных и нормативно-методических требований. Вступление в силу EU AI Act [1] и обновление NIST SP 800-63B-4 [2] сформировали новые требования к биометрическим системам, применяемым в сценариях высокого риска. EU AI Act относит отдельные сценарии удалённой биометрической идентификации и биометрической категоризации к запрещённым или строго ограниченным практикам, а биометрическую верификацию в критически важных инфраструктурах — к высокорисковым сценариям, требующим оценки соответствия, ведения документации и обеспечения человеческого надзора [1].

Хронология ключевых нормативных изменений и стандартов, влияющих на проектирование биометрических систем, приведена на рис. 2.

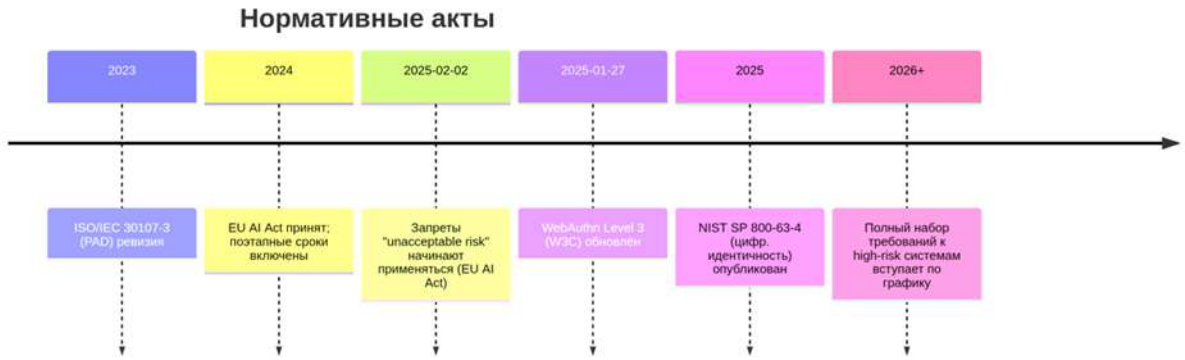


Рис. 2. Нормативный контекст развития биометрических систем и ИИ-компонентов

Тенденция 3: развитие технологий PAD/MAD. Стремительное развитие генеративных нейросетей сделало создание высококачественных синтетических биометрических данных (морфинг, дипфейки) доступным для широкого круга злоумышленников. В ответ активно развиваются технологии обнаружения атак предъявления (PAD — Presentation Attack Detection) и обнаружения морфинга (MAD — Morphing Attack Detection) [3, 8].

Тенденция 4: распространение бесконтактных и мультимодальных решений. Пандемия COVID-19 ускорила переход к бесконтактным технологиям захвата биометрических данных, прежде всего к бесконтактному сканированию отпечатков пальцев с помощью обычных камер. Мультимодальные системы, объединяющие несколько биометрических модальностей, обеспечивают более высокую устойчивость к атакам и более низкий уровень ошибок [6].

Тенденция 5: воспроизводимость и сопоставимость через бенчмарки. Критически важной становится возможность объективного сравнения биометрических систем разных производителей. Бенчмарки FRVT (для систем распознавания лиц) и IREX (для систем распознавания радужки), проводимые NIST, стали де-факто стандартом оценки производительности [13, 14].

2. Биометрические модальности и нейросетевые архитектуры

2.1. Распознавание лиц

Распознавание лиц является наиболее широко применяемой биометрической модальностью. Современные системы распознавания лиц основаны на глубоких нейронных сетях, обученных на больших наборах изображений. Архитектуры ResNet и их производные (DRN — Deep Residual Networks) позволяют извлекать высокоуровневые признаки лица, относительно устойчивые к изменению освещения, ракурса и возрастным изменениям [7].

Процесс работы системы распознавания лиц включает несколько этапов: обнаружение лица на изображении, нормализацию (выравнивание по ключевым точкам), извлечение признаков (embedding) с помощью нейронной сети и сопоставление полученного вектора признаков с шаблоном (template), хранящимся в базе данных. В режиме верификации (1:1) система сравнивает два вектора признаков и принимает решение о совпадении на основе порогового значения. В режиме идентификации (1:N) система ищет наиболее близкое совпадение в базе данных шаблонов.

Бенчмарк FRVT, проводимый NIST, позволяет объективно сравнивать системы распознавания лиц разных производителей по стандартизированным метрикам на стандартизированных наборах данных [13]. Результаты FRVT показывают, что лучшие современные системы достигают показателя FNIR (False Negative Identification Rate — доля пропущенных совпадений) менее 0,3% при FPIR (False Positive Identification Rate — доля ложных совпадений) 0,01% на отдельных наборах данных высокого качества.

Обобщённая матрица практических преимуществ и ограничений систем распознавания лиц представлена на рис. 3.

Рис. 3. Матрица преимуществ и ограничений распознавания лиц в условиях актуальных атак

2.2. Бесконтактное сканирование отпечатков пальцев

Традиционное контактное сканирование отпечатков пальцев (с использованием ёмкостных или оптических сенсоров) дополняется бесконтактными технологиями (contactless fingerprint), основанными на захвате изображения пальца обычной камерой. Это открывает новые возможности для применения биометрии в гигиенически чувствительных средах и при удалённой аутентификации.

Основная техническая сложность бесконтактного сканирования состоит в необходимости компенсации перспективных искажений, неравномерного освещения и деформаций кожи. Современные нейросетевые подходы позволяют решить эти задачи, однако интероперабельность (совместимость) шаблонов, полученных бесконтактным способом, с шаблонами, полученными контактным способом, остаётся открытой проблемой [6].

Типовая последовательность обработки при бесконтактном захвате отпечатка пальца показана на рис. 4.

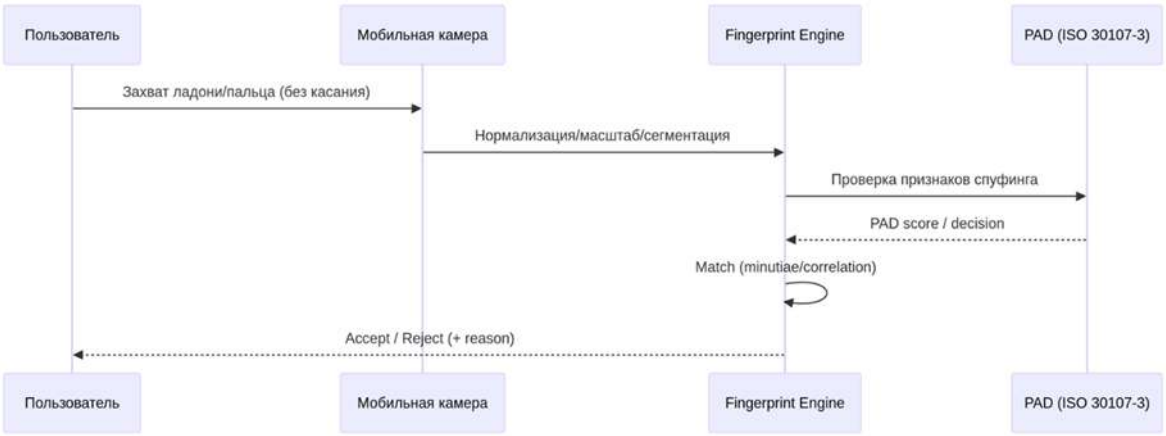


Рис. 4. Переход к бесконтактному сканированию отпечатка пальца: захват, нормализация и PAD-проверка

2.3. Распознавание радужки и сетчатки

Радужная оболочка глаза уникальна для каждого человека и сохраняет свою структуру на протяжении всей жизни — эти свойства позволяют относить её к числу наиболее надёжных биометрических модальностей. Современные системы распознавания радужки работают в ближнем инфракрасном диапазоне (NIR), который обеспечивает чёткое изображение текстуры вне зависимости от цвета глаз и условий освещения [18].

Бенчмарк IREX, проводимый NIST, оценивает производительность систем распознавания радужки по метрикам FNIR и FPIR [14]. Распознавание сетчатки может обеспечивать очень высокую точность, но требует специализированного оборудования и тесного контакта с ним. Из-за этого сетчатку используют преимущественно на высокозащищённых объектах.

Укрупнённая схема обработки данных радужки и сетчатки в биометрической системе приведена на рис. 5.

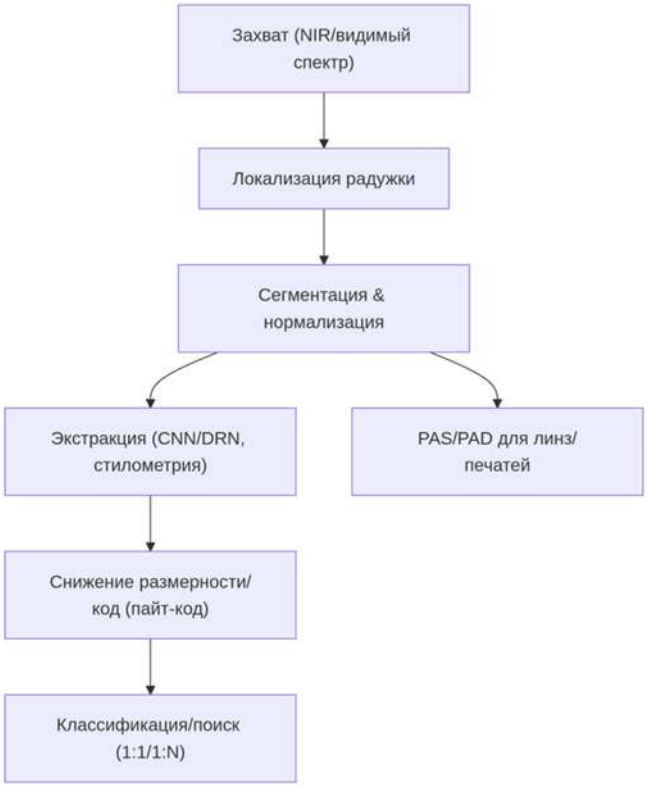


Рис. 5. Обработка данных радужки и сетчатки: захват, локализация, сегментация и сопоставление

3. Метрики качества биометрических систем

Оценка качества биометрических систем требует применения специализированных метрик, учитывающих специфику задачи верификации или идентификации личности. Стандартизация метрик, закреплённая в ISO/IEC 30107-3 [11] и стандартах ANSI/NIST ITL [12], обеспечивает возможность объективного сравнения систем разных производителей.

3.1. Метрики верификации и идентификации

В задаче верификации (1:1) используются три основные метрики:

- FAR (False Acceptance Rate) — доля случаев, когда система принимает самозванца за легитимного пользователя;
- FRR (False Rejection Rate) — доля случаев, когда система отвергает легитимного пользователя;
- EER (Equal Error Rate) — точка, в которой FAR = FRR; служит интегральной характеристикой системы.
- В задаче идентификации (1:N) применяются:
- FPIR (False Positive Identification Rate) — доля случаев, когда система ошибочно идентифицирует самозванца как одного из зарегистрированных пользователей;

– FNIR (False Negative Identification Rate) — доля случаев, когда система не находит совпадения для легитимного пользователя.

3.2. Метрики оценки устойчивости к атакам

В рамках стандарта ISO/IEC 30107-3 [11] установлена номенклатура количественных метрик для оценки устойчивости биометрических систем к атакам предъявления. Базовый набор показателей включает следующие величины.

APCER (Attack Presentation Classification Error Rate) определяется как доля атак предъявления, ошибочно классифицированных биометрической системой в качестве подлинных предъявлений (bona fide).

BPCER (Bona Fide Presentation Classification Error Rate) определяется как доля подлинных предъявлений, ошибочно классифицированных биометрической системой в качестве атак предъявления.

IAPMR (Impostor Attack Presentation Match Rate) определяется как доля атак предъявления, при реализации которых биометрическая система одновременно принимает предъявление в качестве подлинного и осуществляет успешное сопоставление полученного биометрического образца с шаблоном целевого пользователя.Среди указанных метрик

IAPMR обладает наибольшей значимостью с позиции обеспечения информационной безопасности биометрической системы. Численное значение IAPMR количественно характеризует вероятность того, что предпринятая атака предъявления приведёт к успешному преодолению процедуры биометрической аутентификации.

3.3. Компромисс между безопасностью и удобством использования

При настройке биометрических систем неизбежно возникает противоречие: снижение APCER обычно достигается за счёт роста BPCER. Чем выше чувствительность системы PAD, тем больше легитимных пользователей она может ошибочно отвергать. Оптимальный порог определяется сценарием применения, моделью угроз и требованиями регулятора [11].

4. Угрозы и атаки на биометрические системы

4.1. Классификация атак предъявления

Атака предъявления (Presentation Attack, PA) — это попытка злоумышленника обмануть биометрическую систему путём предъявления артефакта (Presentation Attack Instrument, PAI), имитирующего биометрические характеристики легитимного пользователя. Стандарт ISO/IEC 30107-3 [11] классифицирует PAI по типу: фотографии, видеозаписи, маски (2D и 3D), силиконовые накладки, контактные линзы с нанесённым рисунком радужки и т.д.

Традиционные атаки предъявления (распечатанные фотографии, воспроизведение видео) относительно легко обнаруживаются современными системами PAD. Значительно более сложную угрозу представляют атаки с использованием высококачественных 3D-масок и, особенно, атаки на основе синтетических данных, генерируемых нейросетями.

4.2. Морфинг

Морфинг (morphing) — это синтез изображения, сочетающего биометрические черты нескольких людей таким образом, что полученное изображение успешно сопоставляется с шаблонами каждого из них. Морфинг-атаки представляют особую угрозу для систем верификации документов, удостоверяющих личность: злоумышленник может получить паспорт с морфированным изображением, которое будет успешно верифицировано как его лицо, так и лицо сообщника [3, 19].

Современные алгоритмы морфинга основаны на генеративно-состязательных сетях (GAN) и диффузионных моделях, что позволяет создавать морфированные изображения высокого качества, практически неотличимые от подлинных фотографий. Обнаружение морфинга (MAD — Morphing Attack Detection) является активной областью исследований; существующие методы MAD делятся на одноизображенческие (single-image MAD, применяемые при пограничном контроле) и дифференциальные (differential MAD, требующие сравнения с доверенным изображением) [3].

Процессная логика выявления морфинга и дипфейков при регистрации и проверке документов показана на рис. 6.

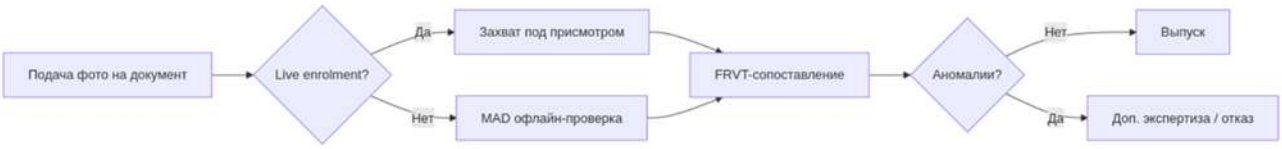


Рис. 6. Процессная схема противодействия морфингу и дипфейкам при регистрации и проверке

4.3. Дипфейки

Дипфейк (deepfake) определяется как синтетическое изображение, видеозапись или аудиосигнал, сгенерированные нейросетевой моделью с целью имитации визуальных или голосовых характеристик конкретного человека. В контексте задач биометрической аутентификации дипфейки формируют угрозу преимущественно для систем удалённой верификации лица, функционирующих в режиме реального времени с применением механизмов обнаружения признаков жизни (liveness detection) [4].

Современные системы liveness detection реализуют различные методы обнаружения синтетических данных: анализ микродвижений лицевой мускулатуры, реакция на случайно генерируемые запросы (challenge-response), спектральный анализ текстуры кожного покрова в нескольких диапазонах излучения, оценка физиологических микропризнаков. Гонка вооружений между системами генерации синтетических данных и системами их обнаружения сохраняется в активной фазе: совершенствование детекторов сопровождается соответствующим совершенствованием генеративных моделей [4, 8].

4.4. Бесконтактный спуфинг

Бесконтактный спуфинг (contactless spoof) определяется как атака на системы бесконтактного сканирования отпечатков пальцев, реализуемая с применением печатных, экранных или синтетически сгенерированных изображений папиллярного узора. Переход индустрии биометрических технологий к бесконтактным методам захвата отпечатков пальцев сопровождается формированием принципиально новых векторов атак. Традиционные методы обнаружения атак предъявления (PAD), разработанные для контактных сенсорных систем, не во всех случаях применимы к бесконтактным архитектурам захвата изображений [6].

4.5. Атаки на процесс регистрации

Самостоятельную категорию угроз биометрическим системам составляют атаки на процедуру первичной регистрации (enrollment). При успешной регистрации злоумышленником поддельного биометрического шаблона в системе — например, посредством компрометации процедуры верификации клиента KYC (Know Your Customer) — последующие операции аутентификации будут устойчиво проходить для атакующего субъекта. Повышенную опасность формирует регистрация морфированного шаблона: указанный тип атаки обеспечивает возможность использования единой учётной записи несколькими физическими лицами одновременно [3, 19].

5. Методы противодействия угрозам

5.1. Обнаружение атак предъявления (PAD)

Технологии обнаружения атак предъявления PAD (Presentation Attack Detection) ориентированы на выявление попыток компрометации биометрической системы посредством использования синтетических артефактов. Современные методы PAD группируются по нескольким категориям в зависимости от информативного признака, лежащего в основе детекции.

Текстурный анализ. Метод базируется на выявлении различий в микротекстуре живого кожного покрова и поверхности артефакта (силиконовых масок, фотографических отпечатков). Нейросетевые алгоритмы анализа обеспечивают обнаружение тонких текстурных различий, недоступных для регистрации традиционными алгоритмическими методами на основе ручной экстракции признаков.

Мультиспектральный анализ. Метод реализует одновременную регистрацию изображений в нескольких спектральных диапазонах: видимом свете, ближнем инфракрасном,

тепловом инфракрасном. Живой кожный покров обладает характерным спектральным профилем отражения и поглощения излучения, который трудно воспроизвести в материалах артефактов.

Анализ микродвижений. Метод обеспечивает обнаружение характерных для живого человека физиологических микродвижений: моргание, пульсация кожи под действием сердечного цикла, спонтанные микродвижения головы, саккадические движения глазных яблок. Системы типа challenge-response запрашивают у пользователя выполнение случайно генерируемых действий — поворот головы, мимическая реакция, — что существенно затрудняет применение статических артефактов в атаках предъявления.

Анализ пространственной геометрии. Методы структурированного освещения и стереоскопической съёмки обеспечивают получение трёхмерной информации о геометрии лица. Реализация указанного подхода делает плоские артефакты (фотографические изображения, экраны мобильных устройств) детектируемыми по геометрическому признаку, поскольку объёмная модель живого лица принципиально отличается от плоского изображения по топологии поверхности.

Стандарт ISO/IEC 30107-3 [11] устанавливает методологические основания тестирования систем PAD и требования к формату представления результатов. Унифицированная методология испытаний обеспечивает возможность объективного сравнительного анализа эффективности систем разных производителей.

5.2. Обнаружение морфинга (MAD)

Методы MAD (Morphing Attack Detection) направлены на выявление морфированных изображений в документах, удостоверяющих личность.

Одноизображенческие методы анализируют артефакты морфинга непосредственно в предъявляемом изображении: следы обработки, аномалии текстуры, несоответствия в деталях лица. Дифференциальные методы сравнивают предъявляемое изображение с доверенным, например с фотографией, сделанной при пограничном контроле, и выявляют признаки морфинга по расхождениям между ними [3].

Нейросетевые методы MAD, как правило, точнее традиционных алгоритмов, однако их устойчивость к новым алгоритмам морфинга, особенно основанным на диффузионных моделях, остаётся предметом активных исследований.

5.3. Непрерывная аутентификация и оценка риска

Традиционная модель аутентификации реализует однократную верификацию личности пользователя в момент входа в информационную систему. Технология непрерывной аутентификации (continuous authentication) обеспечивает мониторинг биометрических и поведенческих признаков субъекта на протяжении всего пользовательского сеанса и направлена на выявление попыток несанкционированного перехвата сессии.

Контекстно-зависимая аутентификация на основе динамической оценки рисков (risk-based authentication) обеспечивает адаптацию требований к процедуре аутентификации в зависимости от текущего контекста доступа. При фиксации аномальных характеристик поведенческого профиля — нетипичных географических координат сессии, отклонений временно́го паттерна входа, нехарактерной последовательности пользовательских действий — система инициирует процедуру запроса дополнительного фактора аутентификации [2].

5.4. Red teaming и тестирование на проникновение

Red teaming — имитация атак на биометрическую систему силами специально подготовленной команды — позволяет выявлять уязвимости, которые стандартное тестирование может не обнаружить. Регулярное проведение таких учений даёт возможность оценить реальную устойчивость системы к актуальным угрозам [9].

6. Архитектура passkeys/WebAuthn L3

6.1. Принципы архитектуры

Технология passkeys, основанная на стандарте WebAuthn Level 3 [5] и спецификациях FIDO Alliance [15], реализует принципиально иную модель аутентификации по сравнению с

традиционными парольными схемами и классическими серверными биометрическими системами.

В архитектуре passkeys каждый пользователь имеет пару криптографических ключей: приватный ключ (private key), хранящийся в защищённом хранилище устройства (platform authenticator), и публичный ключ (public key), зарегистрированный на сервере (Relying Party). При аутентификации сервер отправляет случайный запрос (challenge), который устройство подписывает приватным ключом. Биометрическая верификация (сканирование отпечатка пальца, лица и т.д.) выполняется локально на устройстве и служит исключительно для разблокировки приватного ключа — биометрические данные никогда не покидают устройство [5, 15].

Укрупнённая архитектура разделения локальной биометрической проверки и криптографической аутентификации представлена на рис. 7.

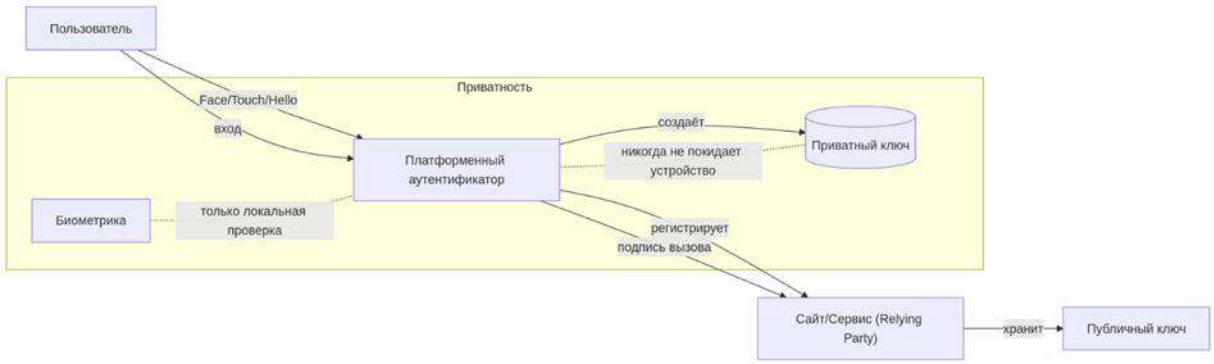


Рис. 7. Архитектура passkeys/WebAuthn L3: локальная биометрия и серверная криптографическая проверка

Такая архитектура обеспечивает несколько ключевых преимуществ с точки зрения безопасности:

– устойчивость к фишингу: приватный ключ привязан к конкретному домену (Relying Party), что делает невозможным его использование на поддельном сайте;
– защита биометрических данных: биометрические данные не передаются на сервер и не хранятся централизованно, что исключает их компрометацию при взломе сервера;
– устойчивость к перехвату: даже при перехвате трафика злоумышленник получает лишь подписанный запрос, который не может быть использован повторно.

6.2. Аттестация и доверие к аутентификатору

Механизм аттестации (attestation) в WebAuthn при регистрации или привязке аутентификатора даёт серверу криптографическое подтверждение его свойств: производителя, модели, уровня сертификации и поддерживаемых функций безопасности. За счёт этого сервер принимает решение о доверии к аутентификатору на основе объективных данных, а не только заявлений пользователя [5]. В корпоративных сценариях аттестация позволяет допускать к работе только сертифицированные аутентификаторы, что снижает риск появления в инфраструктуре скомпрометированных или ненадёжных устройств.

6.3. Ограничения и резервные сценарии

Архитектура passkeys имеет ряд ограничений. Первое — требование совместимого устройства с поддержкой platform authenticator: устаревшие устройства или корпоративные среды с жёсткими требованиями к управлению устройствами могут оказаться несовместимы. Второе — потеря или кража устройства требует механизма восстановления доступа (fallback), который сам по себе становится потенциальным вектором атаки.

Стандарт WebAuthn L3 предусматривает синхронизацию passkeys между устройствами одного пользователя через облачные хранилища производителей. Проблема потери устройства таким образом решается, но возникают новые вопросы — в части безопасности и конфиденциальности [5].

7. Регуляторный контекст и методология оценки рисков

7.1. EU AI Act и биометрические системы

EU AI Act [1] вводит дифференцированный подход к регулированию систем ИИ в зависимости от уровня риска. Применительно к биометрическим системам можно выделить следующие категории и ограничения.

Запрещённые и строго ограниченные практики: биометрическая категоризация по чувствительным признакам, а также отдельные сценарии дистанционной биометрической идентификации в публичных пространствах, прежде всего в режиме реального времени. Для правоохранительных целей предусмотрены специальные исключения и процедурные ограничения.

Системы высокого риска: биометрические системы, применяемые в критически важных инфраструктурах, при принятии решений о доступе к образованию, занятости, кредитованию и т.д., а также ряд систем дистанционной биометрической идентификации, не подпадающих под прямой запрет. Для таких систем EU AI Act требует проведения оценки соответствия, ведения технической документации, обеспечения прозрачности, реализации механизмов человеческого надзора и регистрации в базе данных ЕС.

Системы ограниченного риска: системы, взаимодействующие с людьми (например, чат-боты с биометрической верификацией), обязаны информировать пользователей о том, что они взаимодействуют с системой ИИ.

7.2. NIST SP 800-63B-4 и уровни доверия к аутентификации

Руководство NIST SP 800-63B-4 [2] рассматривает биометрию в контексте многофакторной аутентификации и уровней доверия к аутентификатору (Authenticator Assurance Levels, AAL). Биометрическая характеристика не признаётся самостоятельным аутентификатором: она может использоваться вместе с физическим аутентификатором либо как локальный фактор активации многофакторного аутентификатора.

NIST SP 800-63B-4 устанавливает требования к применению биометрии в государственных сценариях: система должна обеспечивать FMR (False Match Rate) не хуже 1:10000 для всех релевантных демографических групп и, как правило, демонстрировать FNMR (False Non-Match Rate) менее 5%. Для распознавания лиц PAD является обязательным, а для радужки и отпечатков пальцев — рекомендованным; также задаются требования к ограничению числа неудачных попыток и контролю инъекционных атак [2].

7.3. ISO/IEC 30107-3 и тестирование PAD

Стандарт ISO/IEC 30107-3 [11] определяет методологию тестирования систем PAD, включая классификацию PAI по типу и уровню сложности, порядок проведения тестов, метрики оценки (APCER, BPCER, IAPMR) и требования к отчётности. Соответствие ISO/IEC 30107-3 может использоваться как один из элементов доказательной базы при сертификации и оценке соответствия биометрических систем.

7.4. Предлагаемая методология оценки рисков

На основе анализа регуляторных требований и актуальных угроз предлагается следующая методология оценки рисков для биометрических систем, основанных на алгоритмах ИИ.

Шаг 1: классификация системы по EU AI Act. Определить, к какой категории риска относится система (запрещённая, высокого риска, ограниченного риска, минимального риска). Для систем высокого риска последующие шаги становятся обязательной частью управления соответствием и безопасностью.

Шаг 2: оценка угроз. Провести систематическую оценку актуальных угроз с учётом типа биометрической модальности, сценария применения (верификация/идентификация, контролируемая/неконтролируемая среда, удалённая/локальная аутентификация) и доступности инструментов атаки для потенциальных злоумышленников.

Шаг 3: тестирование PAD. Провести тестирование системы PAD в соответствии с ISO/IEC 30107-3 с использованием актуального набора PAI, включая синтетические данные (морфинг, дипфейки). Зафиксировать значения APCER, BPCER, IAPMR.

Шаг 4: верификация на бенчмарках. Для систем распознавания лиц — сопоставить результаты с FRVT или принять участие в тестировании; для систем распознавания радужки — использовать IREX. Результаты бенчмарков обеспечивают независимую верификацию заявленных производителем характеристик.

Шаг 5: оценка архитектурных рисков. Проанализировать архитектуру системы с точки зрения защиты биометрических шаблонов (шифрование, хранение в защищённом хранилище), защиты канала передачи данных, устойчивости процедуры регистрации к атакам, а также наличия и безопасности механизмов fallback.

Шаг 6: red teaming. Провести имитацию атак с использованием актуальных инструментов (морфинг, дипфейки, 3D-маски, бесконтактный спуфинг). Зафиксировать выявленные уязвимости и разработать меры по их устранению.

Шаг 7: документирование и мониторинг. Для систем высокого риска по EU AI Act — обеспечить ведение технической документации, журналов аудита и механизмов человеческого надзора. Установить процедуры регулярного пересмотра оценки рисков с учётом появления новых угроз.

8. Ключевые риски и меры противодействия в 2025 году

На основе проведённого анализа можно выделить ключевые риски для биометрических систем аутентификации в 2025 году и соответствующие меры противодействия.

Итоговая приоритизация рисков представлена на рис. 8.

Рис. 8. Приоритизация ключевых рисков биометрической аутентификации в 2025 году

Риск 1. Морфинг-атаки на документы, удостоверяющие личность. Уровень угрозы оценивается как высокий, поскольку инструменты морфинга широко доступны, а качество морфированных изображений продолжает расти. Основные меры противодействия включают внедрение дифференциальных методов MAD при пограничном контроле, использование цифровых подписей для защиты фотографий в документах и обязательную live enrolment для документов высокого уровня доверия.

Риск 2. Дипфейк-атаки на системы удалённой верификации. Уровень угрозы оценивается как высокий, поскольку удалённая верификация широко применяется в KYC-процедурах и онлайн-банкинге. Мерами противодействия выступают многоуровневые системы liveness detection, challenge-response-протоколы и интеграция с аттестованными platform authenticators на базе passkeys/WebAuthn.

Риск 3. Атаки на процедуру регистрации. Уровень угрозы оценивается как средний, так как успешная атака обычно требует физического присутствия или компрометации оператора.

Снижение риска обеспечивается обязательной live enrolment для систем высокого уровня доверия, многоуровневой верификацией при регистрации и аудитом процедур регистрации.

Риск 4. Компрометация централизованных баз биометрических шаблонов. Уровень угрозы является высоким, поскольку утечка биометрических данных необратима: в отличие от пароля, биометрический признак нельзя полноценно заменить. Мерами защиты являются переход к on-device-хранению, применение биометрических криптосистем и шифрование шаблонов, включая перспективные схемы гомоморфного шифрования.

Риск 5. Несоответствие регуляторным требованиям. Уровень угрозы оценивается как высокий из-за возможных санкций и репутационных последствий. Для смягчения риска необходимы оценка соответствия EU AI Act, внедрение процессов управления жизненным циклом ИИ-систем и регулярный аудит соответствия.

Риск 6. Деградация точности при изменении условий эксплуатации. Уровень угрозы оценивается как средний, поскольку нейросетевые модели могут снижать точность при изменении освещения, ракурса, качества сенсора или возрастных характеристик пользователя. Основные меры — регулярное re-enrolment, мониторинг метрик качества в промышленной эксплуатации и использование адаптивных моделей с контролируемыми механизмами обновления.

Риск 7. Предвзятость алгоритмов (algorithmic bias). Уровень угрозы является высоким с точки зрения регуляторного и репутационного риска, поскольку EU AI Act прямо требует оценки и устранения предвзятости. Необходимы тестирование на репрезентативных наборах данных, мониторинг качества по демографическим группам и регулярный аудит моделей.

Обсуждение

Проведённый анализ позволяет сформулировать принципиальные выводы о природе современных вызовов в области биометрической аутентификации.

Первое положение касается фундаментального противоречия между показателями точности и обеспечения безопасности. Нейросетевые алгоритмы обеспечили достижение рекордных показателей точности биометрического распознавания. Одновременно сформировался новый класс уязвимостей: те же нейросетевые технологии, обеспечивающие высокую точность распознавания, применяются злоумышленниками для генерации высококачественных синтетических атак (морфинг, дипфейки). Дальнейшее повышение точности распознавания само по себе не обеспечивает решения проблемы безопасности — необходимо параллельное развитие методов обнаружения синтетических атак [8, 17].

Второе положение касается роли архитектурных решений в построении биометрических систем. Переход к архитектуре passkeys/WebAuthn L3 демонстрирует возможность устранения значительной части рисков биометрической аутентификации на уровне системной архитектуры, а не на уровне совершенствования алгоритмов распознавания. Разделение биометрической верификации, локально выполняемой на пользовательском устройстве, и криптографической аутентификации, реализуемой на серверной стороне, устраняет целый класс угроз, связанных с компрометацией централизованных хранилищ биометрических данных [5, 15].

Третье положение касается роли нормативно-правового регулирования. Принятие EU AI Act и обновление NIST SP 800-63B-4 формируют отраслевой переход от экспериментальных подходов к разработке ИИ-систем к подотчётным и верифицируемым архитектурам. Соответствие нормативным требованиям приобретает характер не только юридического обязательства, но и инженерного требования, определяющего архитектурные решения на этапе проектирования [1, 2].

Четвёртое положение касается воспроизводимости и сопоставимости результатов испытаний биометрических систем. Бенчмарки FRVT и IREX, проводимые NIST, выполняют системообразующую функцию в обеспечении объективной оценки эффективности биометрических систем. Участие в указанных бенчмарках и публикация полученных результатов приобретают статус необходимого условия формирования доверия к биометрической системе со стороны регуляторов и заказчиков [13, 14].

Ограничения исследования. Применительно к настоящему обзору необходимо учитывать ряд ограничений. Первое ограничение определяется высокой динамикой развития предметной области: конкретные количественные метрики и технические характеристики систем, приведённые в статье, могут утратить актуальность в течение нескольких лет. Второе ограничение связано с источниковой базой: анализ опирается на публично доступные данные, тогда как реальные характеристики коммерческих систем могут существенно отличаться от публикуемых. Третье ограничение определяется незавершённостью формирования регуляторного ландшафта: правоприменительная практика EU AI Act находится на начальной стадии формирования.

Заключение

Проведённый анализ позволяет сформулировать следующие выводы.

К 2025 году алгоритмы искусственного интеллекта приобрели статус определяющего фактора точности, адаптивности и масштабируемости биометрических систем аутентификации. Одновременно указанные алгоритмы стали источником формирования новых классов рисков. Современные нейросетевые модели демонстрируют рекордные показатели на бенчмарках FRVT/IREX. Применение этих моделей формирует уязвимость биометрических систем перед атаками морфинга, дипфейками и синтетическими спуфами. Проблематика доверия и управляемости подобных решений приобретает первоочередную значимость в исследовательской и инженерной повестке.

Международные стандарты и регуляторные рамки — EU AI Act, ISO/IEC 30107-3, NIST SP 800-63B-4, ISO/IEC 24745 — формируют отраслевой переход от экспериментальных подходов к разработке ИИ-систем к подотчётным и верифицируемым архитектурам. Применительно к сценариям использования высокого уровня риска соответствие нормативным требованиям приобретает обязательный характер вместо рекомендательного.

Архитектурное решение passkeys/WebAuthn L3 обеспечивает функциональное разделение процедуры биометрической верификации и процедуры криптографической аутентификации. Указанное разделение устраняет целый класс угроз информационной безопасности, формируемых централизованным хранением биометрических данных.

Предложенная в работе методология оценки рисков обеспечивает интеграцию системы классификации по EU AI Act, процедуры тестирования PAD по ISO/IEC 30107-3, верификации на бенчмарках FRVT/IREX и регулярного red teaming. Совокупность указанных компонентов формирует практически применимый инструментарий для разработчиков и операторов биометрических систем.

Дальнейшее развитие биометрических технологий определяется не количественным ростом вычислительных мощностей. Решающим фактором становится способность к интеграции интеллектуальной обработки признаков, этических ограничений и принципов инженерной воспроизводимости в рамках подотчётных системных архитектур.

Библиографический список

1. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) // Official Journal of the European Union. — 2024. — L 1689. — URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1689 (дата обращения: 31.03.2025).
2. Temoshok D., Fenton J., Lefkovitz N., Regenscheid A., Richer J. Digital Identity Guidelines: Authentication and Authenticator Management: NIST Special Publication 800-63B. — Gaithersburg, MD: National Institute of Standards and Technology, 2025. — DOI: 10.6028/NIST.SP.800-63B-4.
3. Scherhag U., Rathgeb C., Merkle J., Breithaupt R., Busch C. Face Recognition Systems Under Morphing Attacks: A Survey // IEEE Access. — 2019. — Vol. 7. — P. 23012–23026. — DOI: 10.1109/ACCESS.2019.2899367.
4. Tolosana R., Vera-Rodriguez R., Fierrez J., Morales A., Ortega-Garcia J. Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection // Information Fusion. — 2020. — Vol. 64. — P. 131–148. — DOI: 10.1016/j.inffus.2020.06.014.

5. Balfanz D., Czeskis A., Hodges J., Jones J. C., Jones M. B., Kumar A., Lindemann R., Powers A., Verrept J. Web Authentication: An API for accessing Public Key Credentials Level 3: W3C Recommendation. — 2023. — URL: https://www.w3.org/TR/webauthn-3/ (дата обращения: 31.03.2025).
6. Maltoni D., Maio D., Jain A. K., Feng J. Handbook of Fingerprint Recognition. — 3rd ed. — Cham: Springer, 2022. — 512 p. — DOI: 10.1007/978-3-030-83624-5.
7. Guo Y., Zhang L., Hu Y., He X., Gao J. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition // Lecture Notes in Computer Science. — 2016. — Vol. 9907. — P. 87–102. — DOI: 10.1007/978-3-319-46487-9_6.
8. Ramachandra R., Busch C. Presentation Attack Detection Methods for Face Recognition Systems: A Comprehensive Survey // ACM Computing Surveys. — 2017. — Vol. 50, No. 1. — Art. 8. — DOI: 10.1145/3038924.
9. Biggio B., Roli F. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning // Pattern Recognition. — 2018. — Vol. 84. — P. 317–331. — DOI: 10.1016/j.patcog.2018.07.023.
10. Regan P. M., Jesse J. Ethical Challenges of Edtech, Big Data and Personalized Learning: Twenty-First Century Student Sorting and Tracking // Ethics and Information Technology. — 2019. — Vol. 21, No. 3. — P. 167–179. — DOI: 10.1007/s10676-018-9492-2.
11. ISO/IEC 30107-3:2023. Information technology — Biometric presentation attack detection — Part 3: Testing and reporting. — Geneva: International Organization for Standardization, 2023.
12. NIST Special Publication 500-290. Biometric Specifications for Personal Identity Verification. — Gaithersburg, MD: National Institute of Standards and Technology, 2020.
13. Grother P., Ngan M., Hanaoka K. Face Recognition Vendor Test (FRVT). Part 3: Demographic Effects: NIST Interagency Report 8280. — Gaithersburg, MD: National Institute of Standards and Technology, 2019. — DOI: 10.6028/NIST.IR.8280.
14. Grother P., Ngan M. Iris Exchange (IREX) 10: Performance of Iris Recognition Algorithms: NIST Interagency Report 8247. — Gaithersburg, MD: National Institute of Standards and Technology, 2018. — DOI: 10.6028/NIST.IR.8247.
15. FIDO Alliance. FIDO2: WebAuthn & CTAP. Technical Overview. — 2023. — URL: https://fidoalliance.org/fido2/ (дата обращения: 31.03.2025).
16. Jain A. K., Nandakumar K., Ross A. 50 Years of Biometric Research: Accomplishments, Challenges, and Opportunities // Pattern Recognition Letters. — 2016. — Vol. 79. — P. 80–105. — DOI: 10.1016/j.patrec.2015.12.013.
17. Marcel S., Nixon M. S., Fierrez J., Evans N. (Eds.) Handbook of Biometric Anti-Spoofing: Presentation Attack Detection and Liveness Detection. — 3rd ed. — Cham: Springer, 2023. — 650 p. — DOI: 10.1007/978-3-031-48349-6.
18. Daugman J. How Iris Recognition Works // IEEE Transactions on Circuits and Systems for Video Technology. — 2004. — Vol. 14, No. 1. — P. 21–30. — DOI: 10.1109/TCSVT.2003.818350.
19. Ferrara M., Franco A., Maltoni D. The Magic Passport // Proceedings of the IEEE International Joint Conference on Biometrics (IJCB). — 2014. — P. 1–7. — DOI: 10.1109/BTAS.2014.6996240.
20. Уймин А. Г. Применение технологий цифрового двойника в подготовке специалистов по сетевому и системному администрированию // Современные наукоёмкие технологии. — 2023. — № 5. — С. 112–118.

References

1. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union, 2024, L 1689. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1689 (accessed 31.03.2025).

2. Temoshok D., Fenton J., Lefkovitz N., Regenscheid A., Richer J. Digital Identity Guidelines: Authentication and Authenticator Management. NIST Special Publication 800-63B. Gaithersburg, MD: National Institute of Standards and Technology, 2025. DOI: 10.6028/NIST.SP.800-63B-4.
3. Scherhag U., Rathgeb C., Merkle J., Breithaupt R., Busch C. Face Recognition Systems Under Morphing Attacks: A Survey. IEEE Access, 2019, vol. 7, pp. 23012–23026. DOI: 10.1109/ACCESS.2019.2899367.
4. Tolosana R., Vera-Rodriguez R., Fierrez J., Morales A., Ortega-Garcia J. Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection. Information Fusion, 2020, vol. 64, pp. 131–148. DOI: 10.1016/j.inffus.2020.06.014.
5. Balfanz D., Czeskis A., Hodges J., Jones J. C., Jones M. B., Kumar A., Lindemann R., Powers A., Verrept J. Web Authentication: An API for accessing Public Key Credentials Level 3. W3C Recommendation, 2023. Available at: https://www.w3.org/TR/webauthn-3/ (accessed 31.03.2025).
6. Maltoni D., Maio D., Jain A. K., Feng J. Handbook of Fingerprint Recognition. 3rd ed. Cham: Springer, 2022. 512 p. DOI: 10.1007/978-3-030-83624-5.
7. Guo Y., Zhang L., Hu Y., He X., Gao J. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. Lecture Notes in Computer Science, 2016, vol. 9907, pp. 87–102. DOI: 10.1007/978-3-319-46487-9_6.
8. Ramachandra R., Busch C. Presentation Attack Detection Methods for Face Recognition Systems: A Comprehensive Survey. ACM Computing Surveys, 2017, vol. 50, no. 1, art. 8. DOI: 10.1145/3038924.
9. Biggio B., Roli F. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. Pattern Recognition, 2018, vol. 84, pp. 317–331. DOI: 10.1016/j.patcog.2018.07.023.
10. Regan P. M., Jesse J. Ethical Challenges of Edtech, Big Data and Personalized Learning: Twenty-First Century Student Sorting and Tracking. Ethics and Information Technology, 2019, vol. 21, no. 3, pp. 167–179. DOI: 10.1007/s10676-018-9492-2.
11. ISO/IEC 30107-3:2023. Information technology — Biometric presentation attack detection — Part 3: Testing and reporting. Geneva: International Organization for Standardization, 2023.
12. NIST Special Publication 500-290. Biometric Specifications for Personal Identity Verification. Gaithersburg, MD: National Institute of Standards and Technology, 2020.
13. Grother P., Ngan M., Hanaoka K. Face Recognition Vendor Test (FRVT). Part 3: Demographic Effects. NIST Interagency Report 8280. Gaithersburg, MD: National Institute of Standards and Technology, 2019. DOI: 10.6028/NIST.IR.8280.
14. Grother P., Ngan M. Iris Exchange (IREX) 10: Performance of Iris Recognition Algorithms. NIST Interagency Report 8247. Gaithersburg, MD: National Institute of Standards and Technology, 2018. DOI: 10.6028/NIST.IR.8247.
15. FIDO Alliance. FIDO2: WebAuthn & CTAP. Technical Overview, 2023. Available at: https://fidoalliance.org/fido2/ (accessed 31.03.2025).
16. Jain A. K., Nandakumar K., Ross A. 50 Years of Biometric Research: Accomplishments, Challenges, and Opportunities. Pattern Recognition Letters, 2016, vol. 79, pp. 80–105. DOI: 10.1016/j.patrec.2015.12.013.
17. Marcel S., Nixon M. S., Fierrez J., Evans N. (Eds.) Handbook of Biometric Anti-Spoofing: Presentation Attack Detection and Liveness Detection. 3rd ed. Cham: Springer, 2023. 650 p. DOI: 10.1007/978-3-031-48349-6.
18. Daugman J. How Iris Recognition Works. IEEE Transactions on Circuits and Systems for Video Technology, 2004, vol. 14, no. 1, pp. 21–30. DOI: 10.1109/TCSVT.2003.818350.
19. Ferrara M., Franco A., Maltoni D. The Magic Passport. Proceedings of the IEEE International Joint Conference on Biometrics (IJCB), 2014, pp. 1–7. DOI: 10.1109/BTAS.2014.6996240.
20. Uymin A. G. Primenenie tekhnologiy tsifrovogo dvoynika v podgotovke spetsialistov po setevomu i sistemnomu administrirovaniyu [Application of digital twin technologies in

training specialists in network and system administration]. Sovremennye naukoemkie tekhnologii [Modern High Technologies], 2023, no. 5, pp. 112–118. (In Russian).

Информация об авторах

Уймин Антон Григорьевич — аспирант 2 курса, кафедра безопасности информационных технологий, факультет комплексной безопасности ТЭК, Российский государственный университет нефти и газа (НИУ) имени И.М. Губкина, Москва, Россия. E-mail: uymin.ag@gubkin.ru

THE USE OF ARTIFICIAL INTELLIGENCE ALGORITHMS IN BIOMETRIC AUTHENTICATION SYSTEMS: CAPABILITIES AND VULNERABILITIES

***Uymin A. G.*[1]**

[1]National University of Oil and Gas «Gubkin University», Moscow, Russian Federation

Abstract. *The article presents a systematic analysis of the capabilities and vulnerabilities arising from the application of artificial intelligence algorithms in modern biometric authentication systems. The relevance of the study is driven by a qualitative shift in digital identity architecture by the mid-2020s: the adoption of WebAuthn Level 3 and passkeys standards, tightening regulatory and methodological requirements of the EU AI Act and NIST SP 800-63B-4, and the rapid development of morphing and deepfake attacks. The article examines key biometric modalities — face recognition, fingerprint recognition (including contactless technologies), iris and retina — in the context of neural network architectures (CNN, ResNet/DRN). Quality metrics of biometric systems (FPIR, FNIR, APCER, BPCER, IAPMR) and international benchmarks FRVT and IREX are analyzed. Special attention is paid to threats: presentation attacks (PA), morphing, deepfakes, and contactless spoofing, as well as countermeasures (PAD, MAD). The passkeys/WebAuthn L3 architecture is examined as a model for separating biometric verification and cryptographic authentication. A risk assessment and mitigation methodology compatible with EU AI Act, ISO/IEC 30107-3 and NIST SP 800-63B-4 requirements is proposed. Architectural and organizational measures linking technical metrics with risk management requirements are specified. The conclusion is drawn that the further development of biometric systems is determined not by the growth of computing power, but by the ability to combine intelligent feature processing, ethical constraints and engineering reproducibility within accountable and verifiable architectures.*

Keywords: *biometric authentication, artificial intelligence, presentation attack, morphing, deepfake, WebAuthn, EU AI Act.*

Information about the authors

Anton G. Uymin — 2nd year postgraduate student, Department of Information Technology Security, Faculty of Integrated Security of the Fuel and Energy Complex, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, Russia. E-mail: uymin.ag@gubkin.ru

ПРОФЕССИОНАЛИТЕТ, 2025, № 1 (005)

УДК *378.147.88:004.7*

И. М. Морозов[1]

[1]*ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И. М. Губкина», г. Москва, Российская Федерация*

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО МОНИТОРИНГУ КОМПЬЮТЕРНЫХ СЕТЕЙ НА ОСНОВЕ ОТЕЧЕСТВЕННОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ И СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: МЕТОДИКА ПОСТРОЕНИЯ И АВТОМАТИЗИРОВАННАЯ ВЕРИФИКАЦИЯ РЕЗУЛЬТАТОВ

Аннотация. *Рассматривается проблема организации лабораторного практикума по дисциплинам, связанным с администрированием и мониторингом компьютерных сетей, в условиях перехода на отечественные операционные системы и импортонезависимые инструменты администрирования. Обосновывается необходимость двухэтапной проверки результатов выполнения лабораторных работ: автоматизированной верификации технического задания и экспертной оценки архитектурных решений. Описывается методика построения воспроизводимой учебной среды на основе стека GNS3 + VirtualBox + Vagrant + ALT Linux + Zabbix, позволяющей моделировать инфраструктуру малого предприятия. Представлены две лабораторные работы: по основам протокола SNMP и по настройке мониторинга сетевых устройств средствами Zabbix с построением карты сети. Показано, что применение механизма Vagrant box обеспечивает портируемость учебной среды, ускоряет подготовку аудитории к занятию и упрощает дистанционную работу студентов. Педагогическая эффективность предложенного подхода обосновывается через концепцию практико-ориентированного обучения, деятельностный подход и принцип профессиональной направленности содержания образования. Методика апробирована в рамках курса «Управление и мониторинг компьютерных сетей» магистерской программы «Компьютерные системы и сети» НИУ ВШЭ и адаптирована для уровня среднего профессионального образования по специальности 09.02.06 «Сетевое и системное администрирование». Результаты апробации подтверждают, что предложенная среда формирует у обучающихся устойчивые профессиональные умения и навыки работы с реальными инструментами мониторинга, применяемыми в ИТ-отрасли. Отдельное внимание уделяется связи лабораторных заданий с профессиональными компетенциями сетевого и системного администратора.*

Ключевые слова: *лабораторный практикум, мониторинг сетей, Zabbix, ALT Linux, SNMP, Vagrant, автоматизированная верификация.*

Введение

Современная система профессионального образования в области информационных технологий переживает период глубокой трансформации, обусловленной двумя взаимосвязанными процессами: стремительным развитием цифровой экономики и курсом на технологический суверенитет Российской Федерации. Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.06 «Сетевое и системное администрирование» (приказ Минпросвещения от 10.07.2023 № 519) прямо указывает на необходимость формирования у выпускников умений работать с отечественными программными продуктами и обеспечивать функционирование информационно-коммуникационных систем в условиях импортозамещения. Аналогичные требования предъявляет профессиональный стандарт 06.026 «Системный администратор информационно-коммуникационных систем» (приказ Минтруда от 29.09.2020 № 680н), закрепляющий в качестве трудовых функций администрирование операционных систем, настройку сетевого оборудования и организацию мониторинга инфраструктуры.

Вместе с тем практика показывает, что учебные лаборатории большинства образовательных организаций по-прежнему ориентированы на проприетарные зарубежные решения — симуляторы Cisco Packet Tracer, операционные системы семейства Windows Server, коммерческие системы мониторинга. Переход на импортонезависимый стек с отечественной операционной системой требует не только замены программного обеспечения, но и пересмотра методики организации лабораторного практикума: способов развёртывания учебной среды, методов проверки результатов, форм оценивания компетенций.

Как справедливо отмечает А. М. Новиков, «практическое обучение — это не просто выполнение практических заданий, это особая форма организации учебно-познавательной деятельности, в которой теоретические знания становятся инструментом решения

профессиональных задач» [1, с. 214]. Именно эта идея лежит в основе предлагаемой методики: лабораторный практикум строится не как иллюстрация теоретического материала, а как моделирование реальной профессиональной деятельности системного администратора в условиях, максимально приближенных к производственным.

Цель настоящей статьи — описать методику построения воспроизводимого лабораторного практикума по мониторингу компьютерных сетей на основе отечественной операционной системы и свободных инструментов администрирования, обосновать педагогическую эффективность предложенного подхода и представить конкретные примеры лабораторных работ, апробированных в учебном процессе.

1. Теоретико-методологические основания

1.1. Практико-ориентированный подход как методологическая база

Теоретическим фундаментом предлагаемой методики служит практико-ориентированный подход к профессиональному образованию, разработанный в трудах С. Я. Батышева, А. М. Новикова, Т. Ю. Ломакиной и других представителей отечественной профессиональной педагогики.

С. Я. Батышев, основоположник теории производственной педагогики, подчёркивал: «Подготовка квалифицированного рабочего и специалиста должна строиться на принципе единства теории и практики, причём практика должна не следовать за теорией, а предшествовать ей или сопровождать её, создавая мотивационную основу для усвоения теоретических знаний» [2, с. 187]. Применительно к подготовке системных администраторов это означает, что студент должен сначала столкнуться с реальной задачей — настроить мониторинг сети, диагностировать инцидент, восстановить работоспособность сервиса — и лишь затем получать теоретическое объяснение механизмов, лежащих в основе этих действий.

А. М. Новиков развивает эту идею в концепции учебно-профессиональной деятельности: «Учебная деятельность в профессиональном образовании должна быть организована таким образом, чтобы её предмет, средства, продукт и условия максимально совпадали с предметом, средствами, продуктом и условиями реальной профессиональной деятельности» [1, с. 231]. Именно этот принцип реализуется в предлагаемом практикуме: студент работает с теми же инструментами (Zabbix, GNS3, ALT Linux), теми же протоколами (SNMP, ICMP, SSH) и теми же задачами (настройка мониторинга, построение карты сети, диагностика инцидентов), с которыми он столкнётся на производстве.

Т. Ю. Ломакина, исследуя тенденции развития непрерывного профессионального образования, указывает на необходимость «опережающей подготовки специалистов, ориентированной не на вчерашние, а на завтрашние требования производства» [3, с. 94]. В контексте импортозамещения это требование приобретает особую остроту: специалист, подготовленный исключительно на западном программном обеспечении, оказывается профессионально беспомощным в условиях, когда отечественная ИТ-отрасль переходит на российские решения.

1.2. Деятельностный подход и формирование профессиональных компетенций

Деятельностный подход, восходящий к трудам А. Н. Леонтьева и получивший развитие в работах В. В. Давыдова, А. А. Вербицкого и других исследователей, рассматривает компетенцию не как совокупность знаний, а как способность к осуществлению деятельности в определённых условиях.

А. А. Вербицкий, разрабатывая концепцию контекстного обучения, утверждает: «Знания усваиваются не в логике учебного предмета, а в логике деятельности, имеющей для обучающегося личностный смысл. Это обеспечивает трансформацию учебной деятельности в профессиональную» [4, с. 56]. Лабораторный практикум, построенный на реальных профессиональных задачах, создаёт именно такой контекст: студент не «изучает Zabbix», а «настраивает мониторинг инфраструктуры предприятия» — и это принципиальное различие в мотивации и глубине усвоения.

Э. Ф. Зеер, исследуя профессиональное становление личности, выделяет ключевую роль «профессионально ориентированных технологий обучения, создающих условия для формирования профессиональной идентичности и субъектной позиции обучающегося» [5, с. 143].

Работа в среде, воспроизводящей реальную производственную инфраструктуру, формирует именно такую идентичность: студент начинает воспринимать себя не как учащегося, выполняющего задание, а как специалиста, решающего профессиональную задачу.

1.3. Проблема верификации результатов в практическом обучении

Одной из ключевых педагогических проблем лабораторного практикума по сетевым технологиям является объективная оценка результатов. В отличие от программирования, где существуют развитые системы автоматической проверки (ejudge, Codeforces, Яндекс.Контест), в области системного администрирования и сетевых технологий подобные инструменты практически отсутствуют.

Н. Г. Хмызова и Е. В. Кузнецова, исследуя практико-ориентированные модули формирования общепрофессиональных компетенций, подчёркивают: «Образовательная среда должна обеспечивать такой режим работы, при котором формируются, проверяются и оцениваются результаты выполнения конкретных практических заданий в условиях, максимально приближенных к реальным производственным» [6, с. 45]. Это требование предполагает не только наличие соответствующего программного обеспечения, но и разработку методики автоматизированной проверки, позволяющей объективно оценить как факт выполнения задания, так и качество архитектурного решения.

В. К. Разгоняев, анализируя опыт использования виртуальных компьютеров при преподавании дисциплин, связанных с информационными технологиями, указывает на то, что «виртуализация учебной среды позволяет не только снизить затраты на оборудование, но и обеспечить воспроизводимость условий выполнения заданий, что является необходимым условием объективного оценивания» [7, с. 38]. Именно воспроизводимость — ключевое свойство предлагаемой методики: каждый студент работает в идентичной среде, что исключает влияние случайных факторов на результат.

2. Архитектура учебной среды

2.1. Обоснование выбора программного стека

Выбор программного стека для лабораторного практикума определялся тремя критериями: соответствием требованиям импортозамещения, функциональной полнотой для решения учебных задач и возможностью автоматизации развёртывания и проверки.

Анализ доступных решений для импортонезависимого сетевого администрирования позволил выделить следующие ключевые компоненты:

- ALT Linux — отечественная операционная система на базе Linux, включённая в реестр российского программного обеспечения Минцифры России. Она может использоваться в качестве серверной и рабочей ОС в учебных и корпоративных инфраструктурах;
- Zabbix — система мониторинга с открытым исходным кодом, разработанная латвийской компанией Zabbix SIA, де-факто являющаяся стандартом мониторинга в российской ИТ-отрасли и широко применяемая в качестве замены проприетарным решениям (SolarWinds, PRTG, Nagios XI);
- GNS3 (Graphical Network Simulator-3) — эмулятор сетевой инфраструктуры с открытым исходным кодом, позволяющий запускать образы сетевых устройств в виртуальной среде;
- VirtualBox — гипервизор с открытым исходным кодом, обеспечивающий виртуализацию серверных компонентов учебной среды;
- Vagrant — инструмент автоматизации развёртывания виртуальных машин, позволяющий описать всю инфраструктуру в виде кода (Infrastructure as Code) и воспроизвести её на любом рабочем месте.

Принципиально важно, что выбранный стек не является «учебным упрощением»: он объединяет отечественную операционную систему и открытые инструменты, применимые в практико-ориентированной подготовке сетевых и системных администраторов. Это обеспечивает прямую связь формируемых компетенций с профессиональной деятельностью.

2.2. Концепция Vagrant box как основа воспроизводимой учебной среды

Центральным элементом предлагаемой методики является использование механизма Vagrant box для создания воспроизводимой учебной среды. Vagrant box представляет собой упакованный образ виртуальной машины (или набора машин), содержащий предустановленное и предварительно сконфигурированное программное обеспечение, готовое к немедленному развёртыванию.

В контексте лабораторного практикума по мониторингу сетей Vagrant box включает следующие компоненты:

- эмулятор сети GNS3 с предустановленными образами сетевых устройств (маршрутизаторы и коммутаторы);
- сервер ALT Linux с установленным и предварительно сконфигурированным Zabbix Server;
- набор скриптов автоматизированной проверки результатов выполнения лабораторных работ;
- документацию и методические указания в электронном виде.

Развёртывание всей учебной среды на рабочем месте студента или в учебной аудитории выполняется единственной командой:

vagrant up

После выполнения этой команды студент получает полностью готовую к работе инфраструктуру, идентичную той, что используется на всех остальных рабочих местах. Это решает сразу несколько педагогических проблем:

Во-первых, устраняется «проблема первого занятия» — значительная часть учебного времени традиционно тратится на установку и настройку программного обеспечения, что не является целью лабораторной работы. В предлагаемой методике студент с первых минут занятия работает непосредственно с учебным заданием.

Во-вторых, обеспечивается равенство стартовых условий для всех студентов группы, что является необходимым условием объективного оценивания.

В-третьих, решается проблема дистанционного обучения: студент может развернуть идентичную учебную среду на домашнем компьютере и выполнить лабораторную работу самостоятельно.

В-четвёртых, преподаватель получает возможность быстро «сбросить» среду в исходное состояние после занятия командой vagrant destroy && vagrant up, что исключает накопление ошибок конфигурации от группы к группе.

Готовые Vagrant box для различных учебных сценариев публикуются на платформе HashiCorp Vagrant Cloud (app.vagrantup.com), что обеспечивает их доступность для образовательных организаций по всей стране.

2.3. Топология учебной сети

Для наглядной связи теоретического описания с лабораторной методикой в статью включены пример учебной топологии и параметры стенда, используемые при выполнении практической работы по построению карты сети в Zabbix.

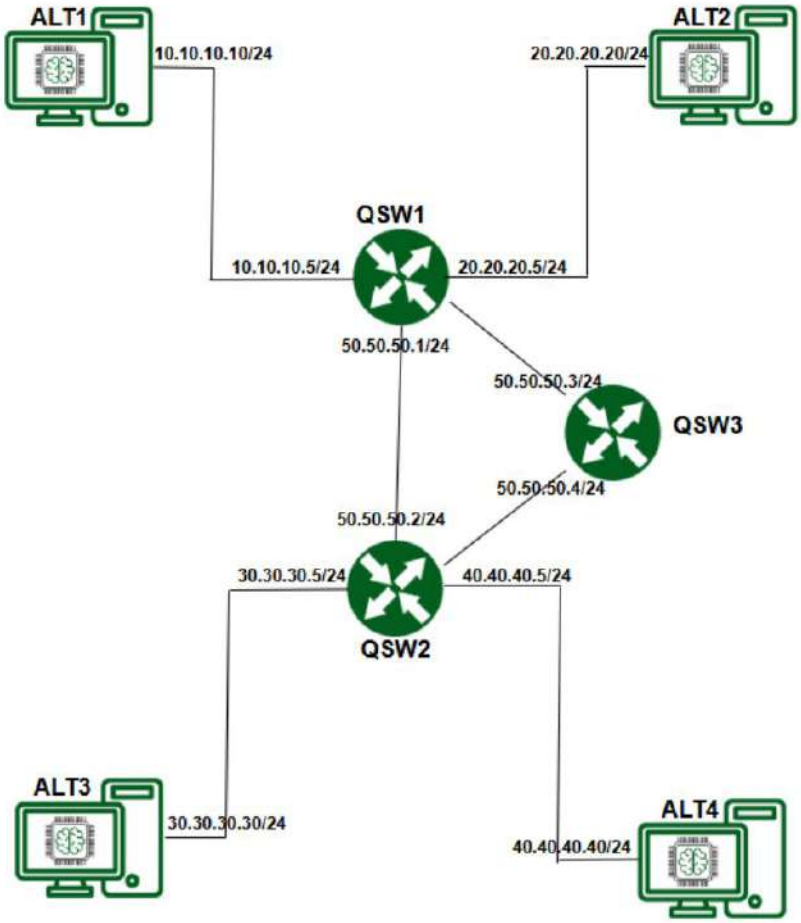


Рис. 1. Топология учебной сети для лабораторного практикума

Таблица 1 – Характеристики виртуальных машин учебного стенда

Устройство	HDD/SSD, ГБ	CPU	RAM, ГБ	ОС
ALT1	30	2	2	ОС Альт Сервер 11.0
ALT2	30	2	2	Альт Рабочая станция 11.1
ALT3	30	2	2	ОС Альт Сервер 11.0
ALT4	30	2	2	Альт Рабочая станция 11.1
vESR1	8	2	4	vesr-1.18.9-build3
vESR2	8	2	4	vesr-1.18.9-build3
QSW3	8	2	4	EcoRouterOS 3.2.1

Таблица 2 – Сетевые параметры узлов учебной топологии

Устройство	Интерфейс	IP-адрес / маска	Шлюз
ALT1	enp0s3	10.10.10.10/24	10.10.10.5
ALT2	enp0s3	20.20.20.20/24	20.20.20.5
ALT3	enp0s3	30.30.30.30/24	30.30.30.5
ALT4	enp0s3	40.40.40.40/24	40.40.40.5
vESR1	enp0s3	10.10.10.5/24	10.10.10.10
	enp0s8	20.20.20.5/24	20.20.20.20
	enp0s9	50.50.50.1/24	50.50.50.2
vESR2	enp0s3	50.50.50.2/24	50.50.50.1
	enp0s8	30.30.30.5/24	30.30.30.30
	enp0s9	40.40.40.5/24	40.40.40.40
QSW3	enp0s3	50.50.50.3/24	-
	enp0s8	50.50.50.4/24	-

В рассматриваемом варианте учебная топология моделирует сеть малого предприятия и включает следующие элементы:

– сервер ALT1 с Zabbix Server, выполняющий функции центрального узла мониторинга;
– рабочие станции и серверы ALT2–ALT4, выступающие в роли контролируемых узлов;

- маршрутизаторы vESR1 и vESR2, обеспечивающие маршрутизацию между сетевыми сегментами;
- маршрутизатор QSW3 на базе EcoRouterOS, используемый для расширения топологии и настройки дополнительного SNMP-мониторинга.

Данная топология намеренно упрощена по сравнению с производственными сетями, однако содержит ключевые элементы, необходимые для формирования базовых профессиональных компетенций: маршрутизируемую сеть, серверную инфраструктуру, рабочие станции, сетевые устройства и систему мониторинга.

3. Методика двухэтапной верификации результатов

3.1. Концептуальные основания двухэтапной проверки

Предлагаемая методика верификации результатов основана на разграничении двух принципиально различных аспектов профессиональной компетентности: операциональной (умение выполнить конкретное техническое действие) и архитектурной (умение принять обоснованное проектное решение).

Это разграничение соответствует классификации уровней профессиональной компетентности, предложенной в профессиональной педагогике. Как указывает В. А. Федоров, «профессиональная компетентность специалиста включает не только операциональный компонент — умение выполнять конкретные трудовые действия, — но и проектировочный компонент — умение планировать, обосновывать и оценивать профессиональную деятельность» [8, с. 67]. Именно эти два компонента проверяются на двух этапах верификации.

Первый этап — автоматизированная проверка технического задания — направлен на оценку операционального компонента: выполнил ли студент конкретные технические требования задания. Проверка осуществляется набором скриптов, которые анализируют конфигурационные файлы, состояние сервисов, параметры сетевых устройств и сравнивают их с эталонными значениями.

Второй этап — экспертная оценка архитектурного решения — направлен на оценку проектировочного компонента: насколько обоснованным, эффективным и масштабируемым является принятое студентом решение. Этот этап не поддаётся полной автоматизации и требует участия преподавателя-эксперта.

3.2. Автоматизированная верификация: принципы и реализация

Автоматизированная верификация строится на принципе сравнения фактического состояния системы с эталонным. Для каждой лабораторной работы разрабатывается набор проверочных скриптов, реализующих следующие типы проверок:

Структурные проверки — верификация наличия и содержимого конфигурационных файлов. Например, для лабораторной работы по настройке SNMP проверяется наличие файла /etc/snmp/snmpd.conf, корректность указанных в нём community-строк, разрешённых IP-адресов и версии протокола.

Функциональные проверки — верификация работоспособности настроенных сервисов. Скрипт выполняет реальный SNMP-запрос к настроенному устройству и проверяет корректность ответа:

```
snmpwalk -v2c -c public 192.168.1.1 1.3.6.1.2.1.1
```

Если устройство отвечает корректно — проверка пройдена; если нет — скрипт возвращает диагностическое сообщение с указанием возможных причин ошибки.

Интеграционные проверки — верификация взаимодействия компонентов системы. Для лабораторной работы по Zabbix проверяется, что сервер Zabbix успешно получает данные от настроенных хостов, триггеры срабатывают корректно, карта сети отображает актуальную топологию.

Проверки на наличие альтернативных решений — важный элемент, предотвращающий «формальное» выполнение задания. Скрипты проверяют не только наличие «правильного» решения, но и отсутствие некорректных альтернатив. Например, если студент настроил SNMP community-строку «public» вместо требуемого по заданию уникального значения — это фиксируется как нарушение требований безопасности.

Результаты автоматизированной проверки формируются в виде структурированного отчёта, который доступен как студенту (для самодиагностики), так и преподавателю (для итогового оценивания).

3.3. Экспертная оценка архитектурных решений

Второй этап верификации — экспертная оценка — проводится преподавателем на основе заранее разработанных критериев. Для каждой лабораторной работы формируется экспертный лист, включающий следующие критерии:

- обоснованность выбора параметров конфигурации (почему выбраны именно эти значения, а не другие);
- соответствие решения требованиям безопасности (использование безопасных версий протоколов, ограничение прав доступа, шифрование);
- масштабируемость решения (возможность расширения на большее количество устройств без переработки архитектуры);
- документированность решения (наличие комментариев в конфигурационных файлах, описание принятых решений).

Важно подчеркнуть, что для одного и того же технического задания может существовать несколько корректных архитектурных решений. Задача экспертной оценки — не проверить соответствие единственному «правильному» ответу, а оценить обоснованность и профессиональную зрелость принятого решения. Это соответствует принципу, сформулированному А. М. Новиковым: «В профессиональном образовании оценивается не только результат, но и процесс его достижения, не только правильность ответа, но и качество профессионального мышления» [1, с. 267].

4. Лабораторная работа № 1: Основы протокола SNMP

4.1. Место в учебном плане и формируемые компетенции

Лабораторная работа «Основы протокола SNMP» является первой в цикле работ по мониторингу сетей и предшествует работе по настройке Zabbix. Её цель — сформировать у студентов понимание механизмов, лежащих в основе систем мониторинга сетевого оборудования, прежде чем они начнут работать с высокоуровневыми инструментами.

В соответствии с ФГОС СПО 09.02.06 данная работа направлена на формирование следующих профессиональных компетенций:

- ПК 1.1 «Устанавливать и настраивать операционные системы»;
- ПК 1.3 «Настраивать сетевые протоколы и сервисы»;
- ПК 2.2 «Осуществлять мониторинг и анализ работы сети».

Профессиональный стандарт 06.026 связывает данные компетенции с трудовой функцией B/02.5 «Администрирование сетевых устройств», предполагающей умение «применять методы статической и динамической конфигурации параметров сетевых устройств» и «идентифицировать инциденты в работе сети».

4.2. Теоретические основы: протокол SNMP

Simple Network Management Protocol (SNMP) — протокол прикладного уровня модели OSI, предназначенный для управления и мониторинга сетевых устройств. Протокол определён в серии RFC: RFC 1157 (SNMPv1), RFC 1901–1908 (SNMPv2c), RFC 3411–3418 (SNMPv3).

Архитектура SNMP включает три ключевых компонента:

SNMP Manager (менеджер) — программный компонент, осуществляющий опрос управляемых устройств и обработку полученных данных. В учебной среде роль менеджера выполняет Zabbix Server.

SNMP Agent (агент) — программный компонент, запущенный на управляемом устройстве и отвечающий на запросы менеджера. На сетевых устройствах vESR и EcoRouter SNMP-агент настраивается средствами сетевой ОС; на серверах ALT Linux используется пакет net-snmp.

MIB (Management Information Base) — иерархическая база данных, описывающая структуру информации, доступной через SNMP. Каждый объект MIB идентифицируется

уникальным OID (Object Identifier) — последовательностью чисел, разделённых точками. Например, OID 1.3.6.1.2.1.1.1.0 соответствует системному описанию устройства (sysDescr).

В учебной среде студенты работают с тремя версиями протокола:

– SNMPv1 — базовая версия, использующая community-строку для аутентификации (не рекомендуется для производственного использования из-за отсутствия шифрования);
– SNMPv2c — расширенная версия с улучшенной обработкой ошибок и поддержкой 64-битных счётчиков;
– SNMPv3 — современная версия с поддержкой аутентификации и шифрования (рекомендуется для производственного использования).

4.3. Описание лабораторной работы

Цель работы: изучить принципы работы протокола SNMP, научиться настраивать SNMP-агент на сетевых устройствах и серверах под управлением ALT Linux, освоить инструменты командной строки для работы с SNMP.

Задание 1. Настройка SNMP-агента на маршрутизаторе vESR.

Студент подключается к маршрутизатору vESR через консольный интерфейс и выполняет базовую конфигурацию SNMP:

snmp-server
snmp-server community [community_string] ro
snmp-server host [zabbix_server_ip]
do commit

Параметры community-строки и адреса сервера мониторинга задаются индивидуально для каждого студента или учебной группы, что снижает вероятность механического копирования конфигурации и позволяет проверять корректность настройки по фактическому состоянию стенда.

Задание 2. Настройка SNMP-агента на сервере ALT Linux.

На сервере ALT Linux студент устанавливает и настраивает пакет net-snmp:
apt-get install net-snmp net-snmp-utils

Конфигурационный файл /etc/snmp/snmpd.conf редактируется в соответствии с требованиями задания: указывается community-строка, разрешённые IP-адреса для опроса, контактная информация и местоположение.

Задание 3. Работа с SNMP-утилитами командной строки.

Практическая фиксация работы SNMP выполняется через командный запрос к устройству. Такой скриншот используется в статье как подтверждение работоспособности базового канала мониторинга.

```
[root@ALT1 ~]# snmpwalk -v2c -c public 50.50.50.1 | grep "OID"
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
SNMPv2-MIB::sysORID.1 = OID: SNMP-FRAMEWORK-MIB::snmpFrameworkMIBCompliance
SNMPv2-MIB::sysORID.2 = OID: SNMP-MPD-MIB::snmpMPDCompliance
SNMPv2-MIB::sysORID.3 = OID: SNMP-USER-BASED-SM-MIB::usmMIBCompliance
SNMPv2-MIB::sysORID.4 = OID: SNMPv2-MIB::snmpMIB
SNMPv2-MIB::sysORID.5 = OID: SNMP-VIEW-BASED-ACM-MIB::vacmBasicGroup
SNMPv2-MIB::sysORID.6 = OID: TCP-MIB::tcpMIB
SNMPv2-MIB::sysORID.7 = OID: UDP-MIB::udpMIB
SNMPv2-MIB::sysORID.8 = OID: IP-MIB::ip
```

Рис. 2. Проверка доступности SNMP-агента средствами snmpwalk

Студент осваивает базовые утилиты пакета net-snmp:

– snmpget — получение значения одного OID:
snmpget -v2c -c [community] [ip_address] sysDescr.0
– snmpwalk — обход поддерева MIB:
snmpwalk -v2c -c [community] [ip_address] interfaces
– snmptrap — отправка SNMP-трапа:
snmptrap -v2c -c [community] [manager_ip] '' linkDown

Задание 4. Анализ трафика SNMP.

С помощью утилиты tcpdump студент захватывает SNMP-трафик и анализирует структуру PDU (Protocol Data Unit) в различных версиях протокола. Это задание формирует

понимание механизмов безопасности (точнее, их отсутствия в SNMPv1/v2c) и обосновывает необходимость использования SNMPv3 в производственных средах.

Задание 5. Настройка SNMPv3.

Студент настраивает SNMPv3 с аутентификацией (протокол SHA) и шифрованием (протокол AES):

```
snmpusm localhost create [username]
snmpusm localhost passwd "" [auth_password] [username]
snmpusm localhost passwd -x "" [priv_password] [username]
```

4.4. Автоматизированная проверка результатов

По завершении работы студент запускает скрипт проверки:

./check_lab1.sh

Скрипт выполняет следующие проверки:

1. Наличие и корректность конфигурационного файла /etc/snmp/snmpd.conf.
2. Работоспособность SNMP-агента на сервере ALT Linux (реальный SNMP-запрос).
3. Работоспособность SNMP-агента на маршрутизаторе vESR или EcoRouter.
4. Корректность параметров SNMPv3 (аутентификация и шифрование).
5. Отсутствие небезопасных конфигураций.

Результат проверки выводится в виде таблицы с указанием статуса каждой проверки (PASS/FAIL) и диагностических сообщений для неуспешных проверок.

5. Лабораторная работа № 2: Настройка мониторинга сетевых устройств средствами Zabbix с построением карты сети

5.1. Место в учебном плане и формируемые компетенции

Лабораторная работа «Настройка мониторинга сетевых устройств средствами Zabbix» является второй в цикле и строится на знаниях и навыках, полученных в первой работе. Её цель — сформировать у студентов практические умения работы с профессиональной системой мониторинга, применяемой в реальных производственных средах.

Данная работа направлена на формирование компетенций:

– ПК 2.2 «Осуществлять мониторинг и анализ работы сети»;
– ПК 2.3 «Обеспечивать защиту информации в сети»;
– ПК 3.1 «Устанавливать и настраивать серверное программное обеспечение».

5.2. Zabbix как инструмент профессионального мониторинга

– Zabbix — система мониторинга с открытым исходным кодом, разработанная в 2001 году Алексеем Владышевым и ставшая одним из распространённых решений для мониторинга ИТ-инфраструктуры. В учебном практикуме Zabbix используется как профессиональный инструмент для сбора метрик, настройки триггеров, визуализации состояния узлов и построения карты сети.

Архитектура Zabbix включает следующие компоненты:

– Zabbix Server — центральный компонент, осуществляющий сбор данных, обработку триггеров и хранение исторических данных. В учебной среде развёртывается на сервере ALT Linux.
– Zabbix Agent — лёгкий агент, устанавливаемый на контролируемые хосты для сбора локальных метрик (загрузка CPU, использование памяти, состояние дисков, сетевой трафик). В учебной среде агент устанавливается на серверы и рабочие станции ALT Linux.
– Zabbix Proxy — промежуточный компонент для мониторинга удалённых сегментов сети. В учебной среде не используется, однако его назначение объясняется в теоретической части.
– Zabbix Web Interface — веб-интерфейс для управления системой мониторинга, просмотра данных и настройки оповещений. Реализован на PHP и работает в связке с веб-сервером Apache или Nginx.

5.3. Описание лабораторной работы

Рубрика 3. Методология и технология профессионального образования

Цель работы: освоить процесс установки и настройки системы мониторинга Zabbix на сервере ALT Linux, научиться добавлять хосты для мониторинга, настраивать SNMP-мониторинг сетевых устройств, создавать триггеры и оповещения, строить карту сети.

Задание 1. Установка и первоначальная настройка Zabbix Server.

В учебной среде Zabbix Server предустановлен в Vagrant box, однако студент выполняет его первоначальную настройку: изменяет пароль администратора, настраивает параметры SMTP для отправки оповещений, проверяет работоспособность всех компонентов.

Для проверки работоспособности студент выполняет:

```
systemctl status zabbix-server
systemctl status zabbix-agent
systemctl status apache2
```

Задание 2. Добавление хостов для мониторинга.

В методических материалах добавление узлов в Zabbix раскрывается через последовательную настройку хоста и элемента данных. Эти материалы иллюстрируют переход от сетевой связности к регулярному сбору метрик.

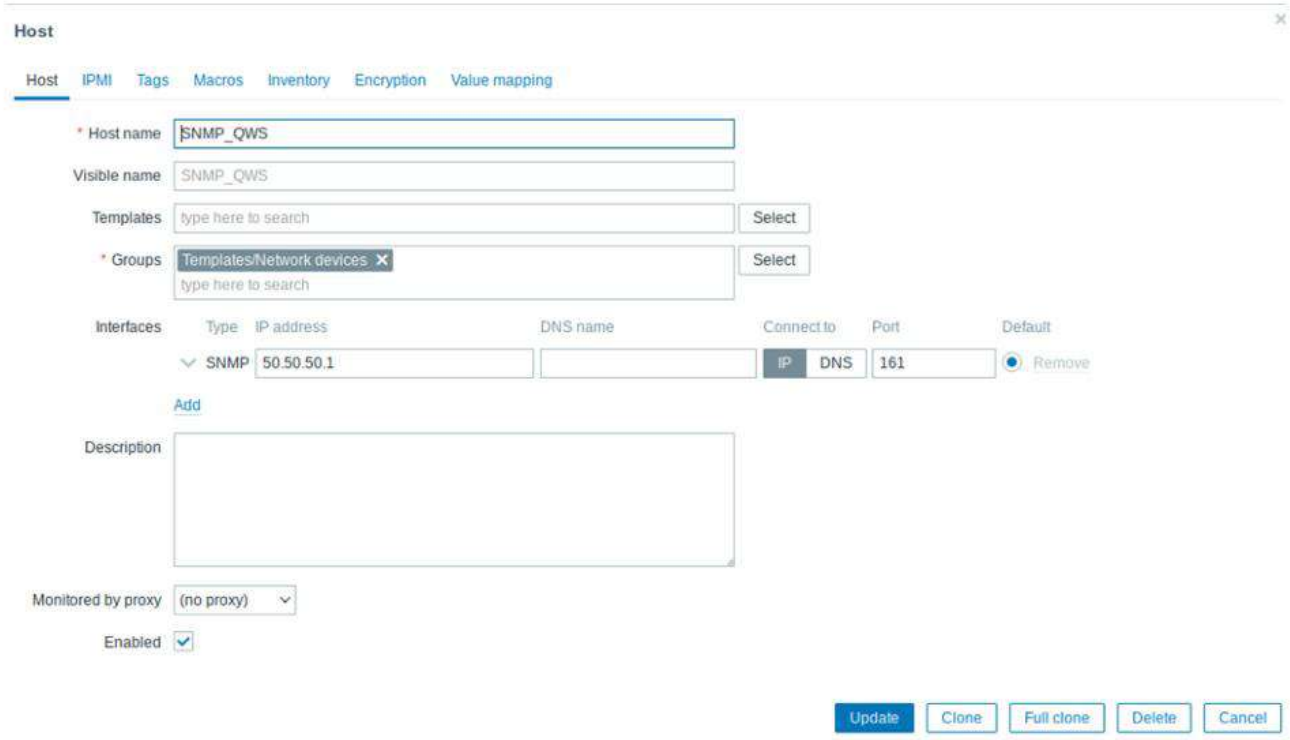


Рис. 3. Создание хоста в Zabbix для SNMP-мониторинга

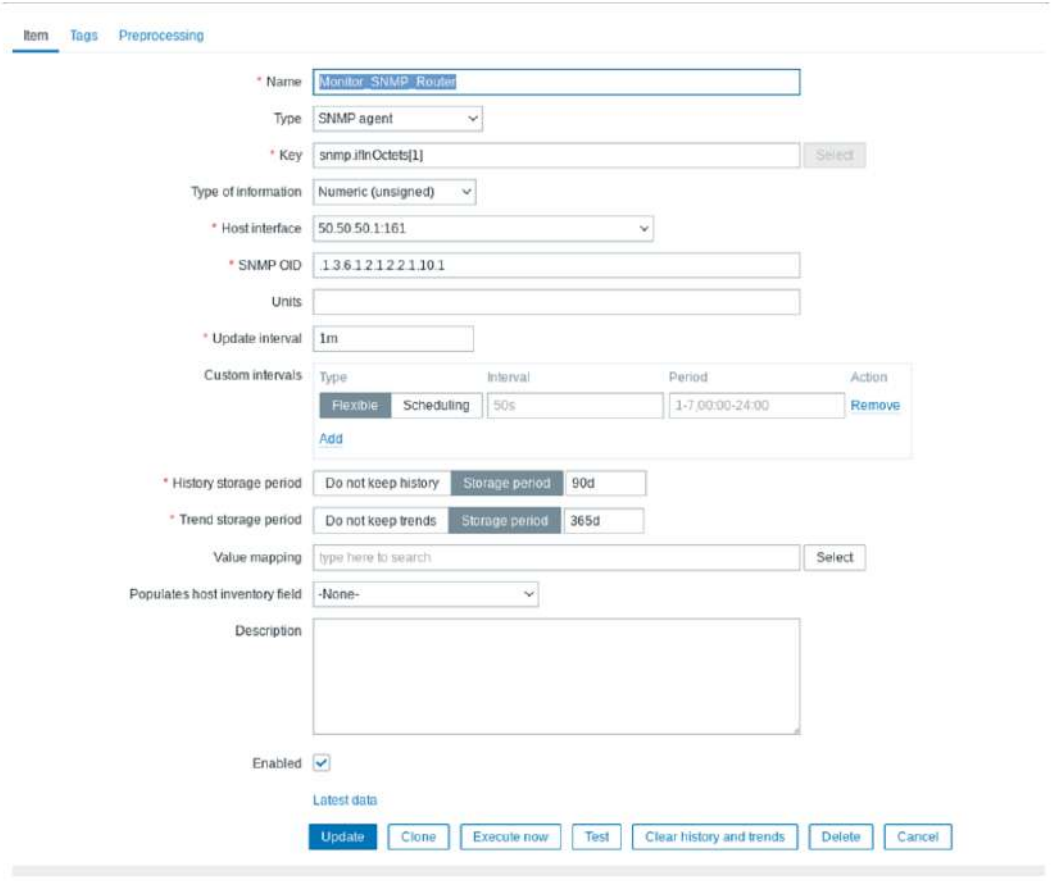


Рис. 4. Настройка элемента данных SNMP в Zabbix

Студент добавляет в Zabbix следующие хосты:

– сервер ALT Linux (мониторинг через Zabbix Agent);
– маршрутизатор vESR или EcoRouter (мониторинг через SNMP).

Для каждого хоста настраиваются:

– интерфейс подключения (IP-адрес, порт);
– метод мониторинга (Zabbix Agent или SNMP);
– шаблон мониторинга (например, Template OS Linux для сервера и типовой SNMP-шаблон сетевого устройства для маршрутизатора).

Задание 3. Настройка SNMP-мониторинга маршрутизатора.

Студент настраивает SNMP-мониторинг маршрутизатора, используя знания, полученные в первой лабораторной работе. Для устройства настраиваются следующие метрики:

– загрузка CPU (OID уточняется по MIB конкретной сетевой ОС);
– использование памяти (OID уточняется по MIB конкретной сетевой ОС);
– состояние интерфейсов (например, OID 1.3.6.1.2.1.2.2.1.8);
– входящий и исходящий трафик на интерфейсах (например, OID 1.3.6.1.2.1.2.2.1.10 и 1.3.6.1.2.1.2.2.1.16).

Задание 4. Создание триггеров и оповещений.

Студент создаёт триггеры для следующих событий:

– загрузка CPU маршрутизатора превышает 80% в течение 5 минут;
интерфейс маршрутизатора перешёл в состояние Down;
свободное место на диске сервера ALT Linux менее 10%.

Для каждого триггера настраивается действие: отправка оповещения на электронную почту администратора.

Задание 5. Построение карты сети.

Ключевым заданием лабораторной работы является построение карты сети в Zabbix. Карта сети выступает итоговым визуальным представлением лабораторного стенда: на ней объединяются хосты, маршрутизаторы, связи и индикаторы состояния.

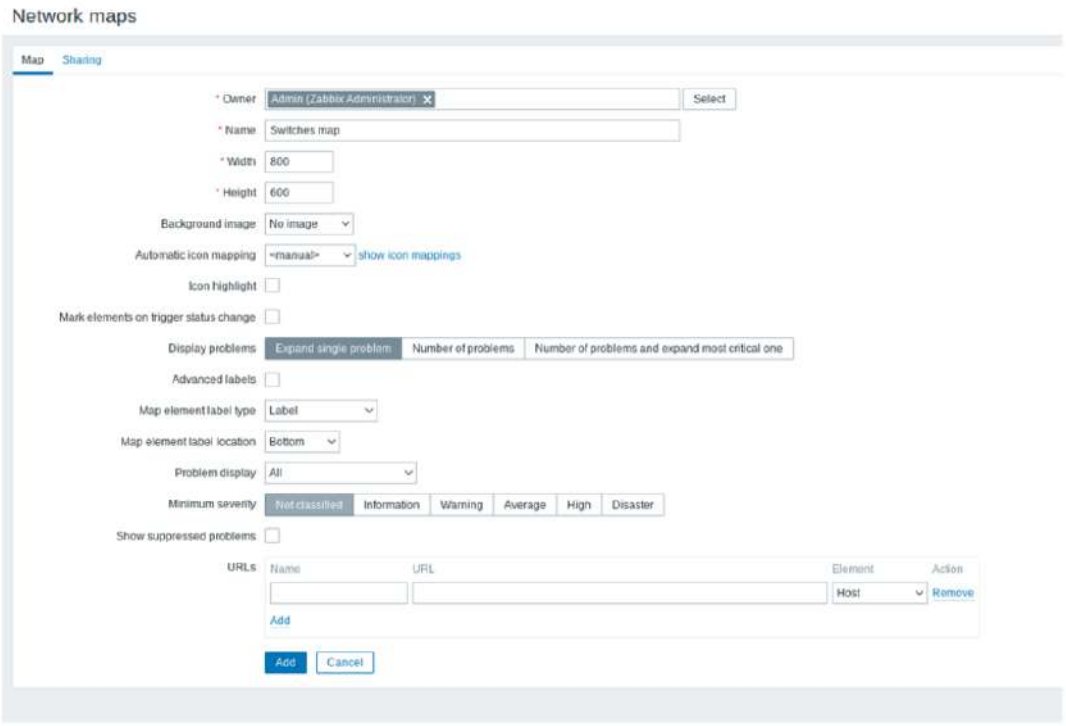


Рис. 5. Создание карты сети в веб-интерфейсе Zabbix

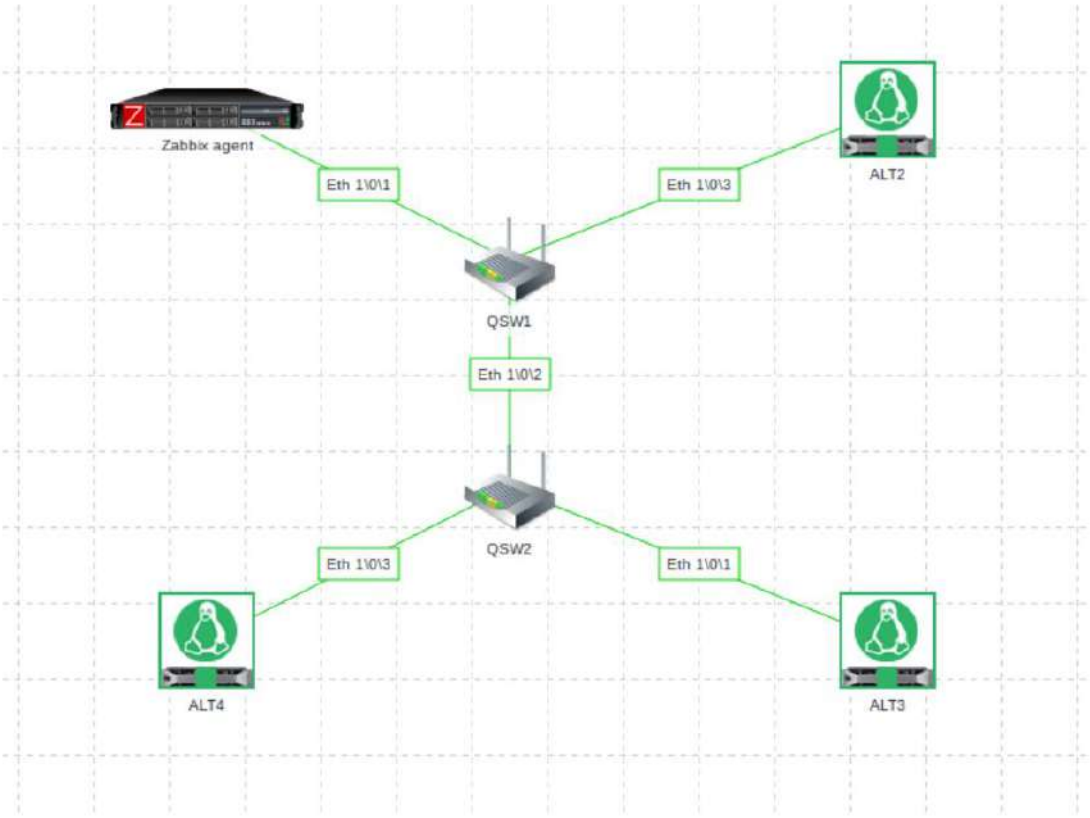


Рис. 6. Итоговая карта сети с настроенными связями между узлами

Дополнительно методика может включать пользовательские сценарии, позволяющие переходить от элемента карты к подключению по SSH или VNC. Это расширяет карту сети от статической схемы до интерактивного рабочего инструмента администратора.

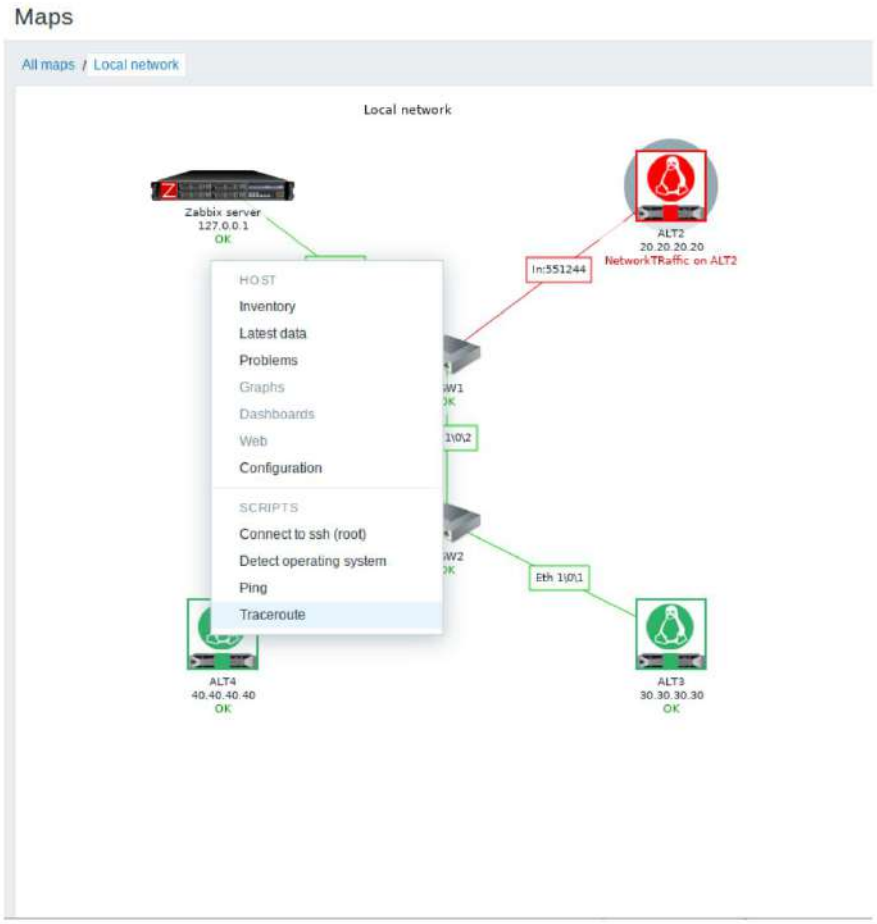


Рис. 7. Запуск пользовательского сценария подключения с карты сети

После создания карты студент размещает на ней элементы инфраструктуры, настраивает связи между ними и задаёт правила визуального отображения состояния устройств и каналов связи.

Студент создаёт карту сети, включающую:

– иконки всех контролируемых устройств (сервер ALT Linux, маршрутизаторы vESR/Eco-Router, рабочие станции);
– связи между устройствами с отображением состояния интерфейсов;
– цветовую индикацию состояния устройств (зелёный — норма, жёлтый — предупреждение, красный — критическая ошибка);
– подписи с ключевыми метриками (загрузка CPU, использование памяти).

Построение карты сети выполняется через веб-интерфейс Zabbix: Monitoring → Maps → Create map. Студент добавляет элементы карты (Map elements), настраивает их свойства и связи (Links), задаёт условия изменения цвета в зависимости от состояния триггеров.

Задание 6. Имитация инцидента и проверка системы мониторинга.

Для проверки корректности мониторинга целесообразно использовать не только факт добавления узлов, но и реакцию системы на изменение нагрузки или срабатывание триггеров.

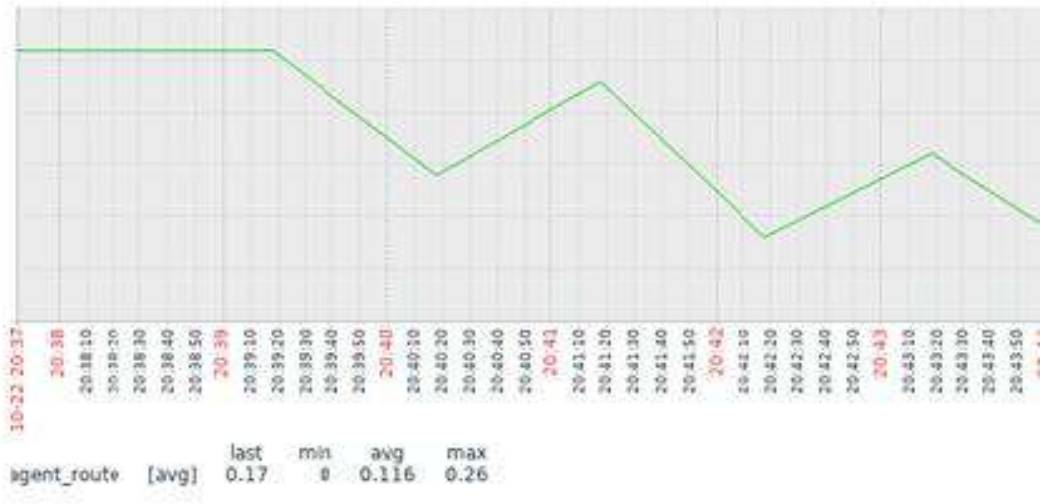


Рис. 8. Отображение изменения сетевого трафика на графике Zabbix

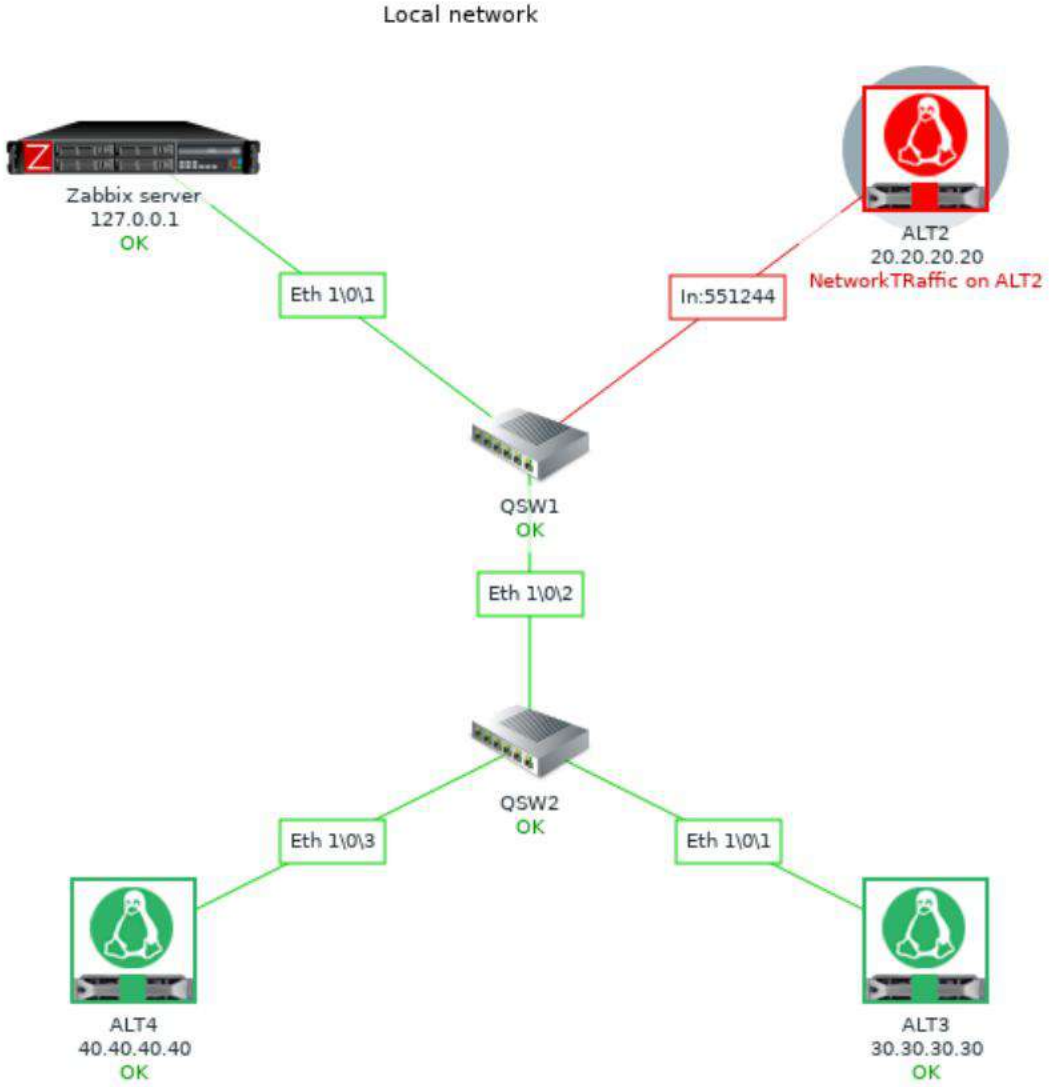


Рис. 9. Визуальное отображение срабатывания триггера на карте сети

Для проверки работоспособности настроенной системы мониторинга студент имитирует инцидент: переводит один из интерфейсов маршрутизатора в неактивное состояние, например командой shutdown в режиме настройки интерфейса:

shutdown

Студент наблюдает, как система мониторинга фиксирует изменение состояния интерфейса, срабатывает соответствующий триггер, на карте сети изменяется цвет связи, а ответственному специалисту отправляется оповещение. После восстановления интерфейса студент наблюдает автоматическое возвращение системы мониторинга в нормальное состояние.

Это задание формирует у студентов понимание полного цикла работы системы мониторинга: от возникновения инцидента до его обнаружения и оповещения ответственного специалиста.

5.4. Автоматизированная проверка результатов

По завершении работы студент запускает скрипт проверки:
./check_lab2.sh

Скрипт выполняет следующие проверки через Zabbix API:

1. Наличие всех требуемых хостов в системе мониторинга.
2. Корректность настройки SNMP-интерфейса для маршрутизатора.
3. Наличие и корректность настройки триггеров.
4. Наличие карты сети с требуемыми элементами.
5. Получение данных от всех хостов (проверка актуальности данных).
6. Корректность настройки оповещений.

Дополнительно скрипт проверяет историю событий Zabbix: был ли зафиксирован инцидент с отключением интерфейса и последующим его восстановлением (задание 6).

6. Педагогическая эффективность предложенного подхода

6.1. Формирование профессиональных компетенций

Предложенная методика обеспечивает формирование профессиональных компетенций в соответствии с пятикомпонентной моделью, принятой в профессиональной педагогике (знания, умения, навыки, практический опыт, профессионально значимые личностные качества).

Знания формируются через изучение теоретических основ протокола SNMP, архитектуры систем мониторинга, принципов работы Zabbix. Студент усваивает понятийный аппарат: OID, MIB, community-строка, триггер, шаблон мониторинга, карта сети.

Умения формируются через выполнение конкретных профессиональных задач: настройка SNMP-агента, добавление хостов в систему мониторинга, создание триггеров, построение карты сети. Каждое умение отрабатывается в условиях, максимально приближенных к производственным.

Навыки формируются через многократное повторение типовых операций: подключение к сетевому устройству через консоль, редактирование конфигурационных файлов, работа с командной строкой Linux, навигация в веб-интерфейсе Zabbix. К концу курса эти операции выполняются студентами автоматически, без обращения к инструкциям.

Практический опыт формируется через задание 6 (имитация инцидента): студент сталкивается с нестандартной ситуацией — отказом сетевого устройства — и должен не только зафиксировать факт инцидента, но и восстановить нормальное состояние системы. Это задание моделирует реальную профессиональную ситуацию, с которой системный администратор сталкивается в повседневной работе.

Профессионально значимые личностные качества — внимательность, аккуратность, способность работать в условиях неопределённости — формируются через требования к качеству выполнения заданий: некорректная конфигурация SNMP-агента, небезопасные параметры, отсутствие документирования фиксируются скриптом проверки и снижают оценку.

6.2. Роль импортозамещения в профессиональной подготовке

Использование отечественной операционной системы и импортонезависимых инструментов в лабораторном практикуме имеет не только организационное, но и педагогическое значение. Как отмечает Т. Ю. Ломакина, «содержание профессионального образования должно опережать текущее состояние производства и ориентироваться на перспективные требования рынка труда» [3, с. 112]. В условиях масштабного перехода российских предприятий на отечественные и открытые решения специалист, не имеющий опыта работы с ALT Linux,

Zabbix и аналогичными инструментами, оказывается менее конкурентоспособным на рынке труда.

Вместе с тем важно подчеркнуть, что предлагаемая методика не сводится к формальной замене одного программного продукта другим. Zabbix — функциональная и гибкая система мониторинга; ALT Linux — полноценная серверная операционная система, соответствующая требованиям учебного и производственного использования; GNS3 — профессиональный инструмент сетевого эмулирования. Использование такого стека не снижает качества подготовки, а переориентирует её на актуальные требования рынка труда.

6.3. Результаты апробации

Предложенная методика апробирована в рамках курса «Управление и мониторинг компьютерных сетей» магистерской программы «Компьютерные системы и сети» Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ). В апробации приняли участие студенты двух учебных групп (общее количество — 34 человека).

Апробация показала следующие результаты:

Сокращение времени на развёртывание учебной среды: в традиционном подходе (ручная установка и настройка программного обеспечения) подготовка рабочего места занимала 40–60 минут учебного времени. При использовании Vagrant box это время сократилось до 5–7 минут (время выполнения команды vagrant up).

Повышение однородности результатов: в традиционном подходе значительная часть ошибок студентов была обусловлена различиями в конфигурации рабочих мест (разные версии программного обеспечения, различные настройки операционной системы). При использовании Vagrant box все студенты работали в идентичной среде, что позволило сосредоточиться на содержательных аспектах задания.

Улучшение качества самодиагностики: наличие автоматизированного скрипта проверки позволило студентам самостоятельно выявлять и исправлять ошибки до сдачи работы преподавателю. По данным опроса, 78% студентов отметили, что скрипт проверки помог им обнаружить ошибки, которые они не заметили самостоятельно.

Повышение мотивации: работа в среде, воспроизводящей реальную производственную инфраструктуру, повысила мотивацию студентов. По данным опроса, 85% студентов отметили, что лабораторные работы «ощущались как реальная профессиональная задача», а не как учебное упражнение.

Авторы рассматривают опрос 34 участников как пилотное описательное исследование: поскольку анкета, абсолютные значения n/N по каждому вопросу и правила округления не приведены, указанные проценты не интерпретируются как статистически доказанный эффект.

Методика также адаптирована для уровня среднего профессионального образования по специальности 09.02.06 «Сетевое и системное администрирование». Адаптация включала упрощение теоретической части, добавление более подробных пошаговых инструкций и снижение требований к архитектурной части экспертной оценки. При этом технический стек и принципы автоматизированной проверки остались неизменными.

7. Обсуждение

7.1. Ограничения предложенного подхода

Предложенная методика имеет ряд ограничений, которые необходимо учитывать при её внедрении.

Аппаратные требования: развёртывание Vagrant box с GNS3 и несколькими виртуальными машинами требует значительных вычислительных ресурсов. Минимальные требования к рабочему месту студента: процессор с поддержкой аппаратной виртуализации (Intel VT-x или AMD-V), 8 ГБ оперативной памяти, 50 ГБ свободного места на диске. Это может быть ограничением для образовательных организаций с устаревшим парком компьютеров.

Кривая обучения для преподавателей: разработка и поддержка Vagrant box требует от преподавателя навыков работы с инструментами DevOps (Vagrant, VirtualBox, bash-скриптинг). Это может быть барьером для преподавателей, не имеющих соответствующего опыта.

Ограниченность эмуляции: несмотря на высокую степень реализма, эмулированная среда GNS3 не полностью воспроизводит поведение реального оборудования. Некоторые функции сетевых устройств (например, аппаратное ускорение, специфические ASIC-функции) недоступны в эмуляторе.

7.2. Перспективы развития

Перспективным направлением развития предложенной методики является интеграция с системами управления обучением (LMS). Автоматизированный скрипт проверки может быть адаптирован для передачи результатов непосредственно в LMS (Moodle, 1С:Образование), что позволит автоматизировать выставление оценок и формирование отчётности.

Другим перспективным направлением является расширение стека программного обеспечения, применимого в импортонезависимой инфраструктуре. В частности, представляет интерес включение в учебную среду систем автоматизации конфигураций, отечественных систем виртуализации (Р-Виртуализация, zVirt) и российских сетевых операционных систем (EcoRouter).

Наконец, перспективным является развитие методики в направлении геймификации: введение элементов соревновательности (рейтинг студентов по результатам автоматизированной проверки), сценарных заданий (студент выступает в роли системного администратора, получающего задачи от «руководства»), командных проектов (несколько студентов совместно администрируют общую инфраструктуру).

Заключение

В статье представлена методика организации лабораторного практикума по мониторингу компьютерных сетей на основе отечественной операционной системы и свободных инструментов администрирования. Ключевыми элементами методики являются:

- использование стека GNS3 + VirtualBox + Vagrant + ALT Linux + Zabbix, обеспечивающего моделирование учебной инфраструктуры, близкой к производственной;
- механизм Vagrant box, обеспечивающий воспроизводимость учебной среды и упрощающий её развёртывание;
- двухэтапная верификация результатов: автоматизированная проверка технического задания и экспертная оценка архитектурных решений;
- две лабораторные работы, последовательно формирующие компетенции в области SNMP-мониторинга и работы с системой Zabbix.

Педагогическая эффективность методики обоснована через концепцию практико-ориентированного обучения (С. Я. Батышев, А. М. Новиков), деятельностный подход (А. А. Вербицкий, Э. Ф. Зеер) и принцип опережающей профессиональной подготовки (Т. Ю. Ломакина). Результаты апробации в НИУ ВШЭ подтверждают, что предложенная методика сокращает непроизводительные затраты учебного времени, повышает однородность условий выполнения заданий, улучшает качество самодиагностики студентов и повышает их мотивацию.

Методика применима как на уровне высшего образования (магистерские программы по направлению «Информатика и вычислительная техника»), так и на уровне среднего профессионального образования (специальность 09.02.06 «Сетевое и системное администрирование»). Переход на такой программный стек не снижает качества подготовки, а переориентирует её на актуальные требования российского рынка труда в условиях цифрового суверенитета.

Библиографический список

1. Новиков, А. М. Методология учебной деятельности / А. М. Новиков. — Москва : Эгвес, 2005. — 176 с. — Текст : непосредственный.
2. Профессиональная педагогика : учебник / под ред. С. Я. Батышева, А. М. Новикова. — 3-е изд., перераб. — Москва : Эгвес, 2010. — 456 с. — Текст : непосредственный.
3. Ломакина, Т. Ю. Диверсификация профессионального образования / Т. Ю. Ломакина. — Москва : ЦПНО ИТОП РАО, 2000. — 145 с. — Текст : непосредственный.

Рубрика 3. Методология и технология профессионального образования

4. Вербицкий, А. А. Активное обучение в высшей школе: контекстный подход : методическое пособие / А. А. Вербицкий. — Москва : Высшая школа, 1991. — 207 с. — ISBN 5-06-002079-7. — Текст : непосредственный.
5. Зеер, Э. Ф. Психология профессионального образования : учебное пособие для студентов высших учебных заведений / Э. Ф. Зеер. — Москва : Издательский центр «Академия», 2009. — 378 с. — Текст : непосредственный.
6. Хмызова, Н. Г. Практико-ориентированные модули формирования общепрофессиональных компетенций у бакалавров профессионального обучения (по отраслям) / Н. Г. Хмызова, Е. В. Кузнецова. — Текст : электронный // Ученые записки Орловского государственного университета. Серия: Гуманитарные и социальные науки. — 2022. — № 1 (94). — С. 43–49. — URL: https://cyberleninka.ru/article/n/praktiko-orientirovannye-moduli-formirovaniya-obscheprofessionalnyh-kompetentsiy-u-bakalavrov-professionalnogo-obucheniya-po (дата обращения: 31.03.2025).
7. Разгоняев, В. К. Использование виртуальных компьютеров при преподавании дисциплин, связанных с информационными технологиями / В. К. Разгоняев. — Текст : электронный // Вестник СИБИТа. — 2020. — № 2 (34). — С. 35–42. — URL: https://sano.ru/magazine/N34/N34.pdf (дата обращения: 31.03.2025).
8. Федоров, В. А. Профессионально-педагогическое образование: теория, эмпирика, практика : монография / В. А. Федоров. — Екатеринбург : Изд-во Урал. гос. проф.-пед. ун-та, 2001. — 330 с. — ISBN 5-8050-0089-X. — Текст : непосредственный.
9. Уймин, А. Г. Инструментальные средства обучения компьютерным сетям. Развёртывание на базе российского программного обеспечения / А. Г. Уймин, Г. И. Токарев // Системы управления и информационные технологии. — 2022. — № 4 (90). — С. 88–92. — DOI 10.36622/VSTU.2022.90.4.019. — EDN KAMNET. — Текст : непосредственный.
10. Уймин, А. Г. Инструменты верификации обучаемого и оценки его эмоционально-психологического состояния при дистанционной работе / А. Г. Уймин // Педагогическое образование: история становления и векторы развития : материалы международной научно-практической конференции. — Москва : МПГУ, 2022. — С. 769–773. — Текст : непосредственный.
11. Гибадуллин, А. А. Методология использования интеллектуальных компьютерных игр в обучении информационным технологиям / А. А. Гибадуллин. — Текст : электронный // International Journal of Advanced Studies. — 2019. — Т. 9, № 4. — С. 11–14. — URL: https://cyberleninka.ru/article/n/metodologiya-ispolzovaniya-intellektualnyh-kompyuternyh-igr-v-obuchenii-informatsionnym-tehnologiyam (дата обращения: 31.03.2025).
12. Пирогов, В. Ю. О возможностях дистанционного преподавания программирования для студентов технических специальностей / В. Ю. Пирогов. — Текст : электронный // Мир науки. Педагогика и психология. — 2021. — Т. 9, № 6. — URL: https://mir-nauki.com/PDF/27PDMN621.pdf (дата обращения: 31.03.2025).
13. Matkurbanov, D. Analysis of network emulation and simulation software / D. Matkurbanov, K. Rakhimjanov // Sciences of Europe. — 2021. — No. 79-1. — P. 38–46. — DOI 10.24412/3162-2364-2021-79-1-38-46. — Текст : непосредственный.
14. Zenk, V. A comparative study of Docker and Vagrant regarding performance on machine level provisioning / V. Zenk, M. Malmström. — Malmö : Malmö University, 2020. — 58 p. — URL: https://www.diva-portal.org/smash/record.jsf?pid=diva2:1444178 (дата обращения: 31.03.2025).
15. Арипова, Г. И. Использование категорий информационно-коммуникационных технологий при обучении студентов информатике / Г. И. Арипова. — Текст : электронный // Наука и образование сегодня. — 2022. — № 2 (71). — С. 28–34. — URL: https://cyberleninka.ru/article/n/ispolzovanie-kategoriy-informatsionno-kommunikatsionnyh-tehnologiy-pri-obuchenii-studentov-informatike (дата обращения: 31.03.2025).

References

1. Novikov, A. M. (2005). *Metodologiya uchebnoy deyatelnosti* [Methodology of educational activity]. Egves.
2. Batyshev, S. Ya., & Novikov, A. M. (Eds.). (2010). *Professionalnaya pedagogika* [Professional pedagogy] (3rd rev. ed.). Egves.
3. Lomakina, T. Yu. (2000). *Diversifikatsiya professionalnogo obrazovaniya* [Diversification of vocational education]. TsPNO ITOP RAO.
4. Verbitsky, A. A. (1991). *Aktivnoye obucheniye v vysshey shkole: kontekstnyy podkhod* [Active learning in higher education: A contextual approach]. Vysshaya shkola.
5. Zeer, E. F. (2009). *Psikhologiya professionalnogo obrazovaniya* [Psychology of vocational education]. Akademiya.
6. Khmyzova, N. G., & Kuznetsova, E. V. (2022). Praktiko-oriyentirovannyye moduli formirovaniya obshcheprofessionalnykh kompetentsiy u bakalavrov professionalnogo obucheniya (po otraslyam) [Practice-oriented modules for the formation of general professional competencies in bachelors of vocational training (by industry)]. *Uchenyye zapiski Orlovskogo gosudarstvennogo universiteta. Seriya: Gumanitarnyye i sotsialnyye nauki*, 1(94), 43–49. https://cyberleninka.ru/article/n/praktiko-orientirovannye-moduli-formirovaniya-obscheprofessionalnyh-kompetentsiy-u-bakalavrov-professionalnogo-obucheniya-po
7. Razgonyaev, V. K. (2020). Ispolzovaniye virtualnykh kompyuterov pri prepodavanii distsiplin, svyazannykh s informatsionnymi tekhnologiyami [Using virtual computers in teaching disciplines related to information technology]. *Vestnik SIBITa*, 2(34), 35–42. https://sano.ru/magazine/N34/N34.pdf
8. Fedorov, V. A. (2001). *Professionalno-pedagogicheskoye obrazovaniye: teoriya, empirika, praktika* [Vocational and pedagogical education: Theory, empirics, practice]. Ural State Vocational Pedagogical University Press.
9. Uymin, A. G., & Tokarev, G. I. (2022). Instrumentalnyye sredstva obucheniya kompyuternym setyam. Razvyortyvaniye na baze rossiyskogo programmnogo obespecheniya [Instrumental tools for teaching computer networks. Deployment based on Russian software]. *Sistemy upravleniya i informatsionnyye tekhnologii*, 4(90), 88–92. https://doi.org/10.36622/VSTU.2022.90.4.019
10. Uymin, A. G. (2022). Instrumenty verifikatsii obuchayemogo i otsenki yego emotsionalno-psikhologicheskogo sostoyaniya pri distantsionnoy rabote [Tools for learner verification and assessment of emotional-psychological state in remote work]. In *Pedagogicheskoye obrazovaniye: Istoriya stanovleniya i vektory razvitiya: Materialy mezhdunarodnoy nauchno-prakticheskoy konferentsii* (pp. 769–773). MPGU.
11. Gibadullin, A. A. (2019). Metodologiya ispolzovaniya intellektualnykh kompyuternykh igr v obuchenii informatsionnym tekhnologiyam [Methodology of using intelligent computer games in teaching information technologies]. *International Journal of Advanced Studies*, 9(4), 11–14. https://cyberleninka.ru/article/n/metodologiya-ispolzovaniya-intellektualnyh-kompyuternyh-igr-v-obuchenii-informatsionnym-tehnologiyam
12. Pirogov, V. Yu. (2021). O vozmozhnostyakh distantsionnogo prepodavaniya programmirovaniya dlya studentov tekhnicheskikh spetsialnostey [On the possibilities of distance teaching of programming for students of technical specialties]. *Mir nauki. Pedagogika i psikhologiya*, 9(6), Article 27PDMN621. https://mir-nauki.com/PDF/27PDMN621.pdf
13. Matkurbanov, D., & Rakhimjanov, K. (2021). Analysis of network emulation and simulation software. *Sciences of Europe*, 79(1), 38–46. https://doi.org/10.24412/3162-2364-2021-79-1-38-46
14. Zenk, V., & Malmström, M. (2020). *A comparative study of Docker and Vagrant regarding performance on machine level provisioning*. Malmö University. https://www.diva-portal.org/smash/record.jsf?pid=diva2:1444178
15. Aripova, G. I. (2022). Ispolzovaniye kategoriy informatsionno-kommunikatsionnykh tekhnologiy pri obuchenii studentov informatike [Using categories of information and communication technologies in teaching students informatics]. *Nauka i obrazovaniye segodnya*,

2(71), 28–34. https://cyberleninka.ru/article/n/ispolzovanie-kategoriy-informatsionno-kommunikatsionnyh-tehnologiy-pri-obuchenii-studentov-informatike

Информация об авторах

Морозов Илья Михайлович — ассистент, РГУ нефти и газа (НИУ) имени И. М. Губкина, г. Москва, Российская Федерация; e-mail: au-mail@ya.ru

LABORATORY WORKSHOP ON COMPUTER NETWORK MONITORING BASED ON A DOMESTIC OPERATING SYSTEM AND OPEN-SOURCE SOFTWARE: METHODOLOGY AND AUTOMATED VERIFICATION OF RESULTS

***Morozov I. M.*[1]**

[1] *National University of Oil and Gas "Gubkin University", Moscow, Russian Federation*

Abstract. *The article addresses the problem of organizing a laboratory workshop on computer network administration and monitoring disciplines in the context of transition to domestic operating systems and import-independent administration tools. The necessity of a two-stage verification of laboratory work results is substantiated: automated verification of technical specifications and expert evaluation of architectural solutions. A methodology for building a reproducible educational environment based on the GNS3 + VirtualBox + Vagrant + ALT Linux + Zabbix stack is described, enabling simulation of a small enterprise infrastructure. Two laboratory works are presented: on SNMP protocol fundamentals and on configuring network device monitoring using Zabbix with network map construction. It is shown that the use of the Vagrant box mechanism ensures portability of the educational environment, accelerates classroom preparation, and simplifies remote student work. The pedagogical effectiveness of the proposed approach is substantiated through the concept of practice-oriented learning, the activity-based approach, and the principle of professional orientation of educational content. The methodology was tested within the course "Computer Network Management and Monitoring" of the master's program "Computer Systems and Networks" at HSE University and adapted for the level of secondary vocational education in specialty 09.02.06 "Network and System Administration". The results confirm that the proposed environment develops students' stable professional skills in working with real monitoring tools used in the IT industry.*

Keywords: *laboratory workshop, network monitoring, Zabbix, ALT Linux, SNMP, Vagrant, automated verification.*

Information about the authors

Morozov Ilya Mikhailovich — Assistant, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, Russian Federation; e-mail: au-mail@ya.ru

УСЛОВИЯ И ПОРЯДОК НАПРАВЛЕНИЯ МАТЕРИАЛОВ

Редакция научного журнала «ПРОФЕССИОНАЛИТЕТ» принимает к рассмотрению статьи, исправленные версии рукописей, ответы на замечания редакции и рецензентов, а также сопроводительную переписку по конкретной статье. Направление материалов осуществляется в электронной форме в соответствии с внутренним регламентом редакции.

Каждое письмо должно содержать информативную тему и заполненный сопроводительный текст. Переписка ведется в официально-деловом стиле. В теме письма указываются тип обращения, фамилия автора, краткое название статьи и, при необходимости, номер итерации исправления.

В тексте письма обязательно приводятся полное название статьи, сведения обо всех авторах, дисциплина (если применимо), номер итерации исправления и контактные данные ответственного автора. Для исправленных версий дополнительно рекомендуется указывать, на какое письмо редакции или рецензии дается ответ, и кратко обозначать внесенные исправления.

Файл рукописи направляется в формате .docx, на кириллице, по установленной форме наименования. Исправленные версии направляются исключительно ответом на письмо редакции с обязательным указанием номера итерации в теме письма, тексте письма и имени файла.

Материалы, оформленные с нарушением установленных требований, могут быть возвращены без рассмотрения до устранения технических замечаний.

Контактные данные редакционной коллегии

Почтовый адрес для корреспонденции: Москва, Ленинский пр-кт, д. 65/1, отделение почтовой связи № 119296.

До востребования.

Получатель: Павловский Владимир Владимирович.

Тел. +7(950) 632-04-38

e-mail: organizers@au-team.ru

главный редактор – Уймин Антон Григорьевич;

ответственный секретарь - Уймина Ольга Ивановна;

технический редактор - Козлов Глеб Васильевич.

ПРОФЕССИОНАЛИТЕТ, 2025, № 1 (005)

Научное электронное издание.

Сведения о программном обеспечении, использованном для создания электронного издания:
LibreOffice — набор, вёрстка текста, генерация PDF
https://ru.libreoffice.org

Техническая обработка и подготовка материалов выполнены авторами.

Подписано к использованию: 31.03.2025.

Объём издания: 76,6 Мб.

Комплектация издания: pdf.

Запись на физический носитель: Уймин А. Г., тел. +7 (950) 632-04-38.

Издатель — редакция научного журнала «ПРОФЕССИОНАЛИТЕТ».

Место издания: Москва.

Электронная версия подготовлена редакцией журнала для распространения в локальной и сетевой форме.

Носитель электронного издания: URALOLIMP.WEBSIT