Выпуск 1 (4), 2025

# НАУЧНЫЙ ЖУРНАЛ ПРОФЕССИОНАЛИТЕТ

Москва

# ПРОФЕССИОНАЛИТЕТ

**№ 1 (4), 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:**

119634, г. Москва, ул. Лукинская, д. 1, кв. 123

# PROFESSIONALITET

**No. 1 (4), 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:

119634, Moscow, Lukinskaya St., 1, Apt. 123

## СОДЕРЖАНИЕ

## CONTENTS

**Рубрика 1. Информатика и информационные процессы**

***УДК*** *004.415.53*

***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.

**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].

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);
- 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.

Таблицы ассоциаций безопасности IKE SA и IPsec SA представлены в таблице 1.

**Таблица 1 – Захват ESP-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

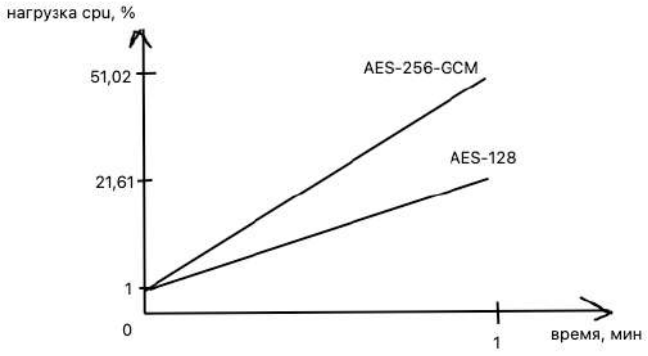


**Рис. 1. Тест производительности AES-128 и AES-256-GCM**

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 %<br>• Средний RTT: 0,431 ms<br>• Потери: 0% | • Загрузка CPU: 51,02 %<br>• Средний RTT: 0,523 ms<br>• Потери: 0% |
| Устойчивость к перебору | $2^{128}$ | $2^{256}$ |

| Применение | Устаревающие системы | Современные протоколы (TLS 1.3, IPsec) |
|---|---|---|

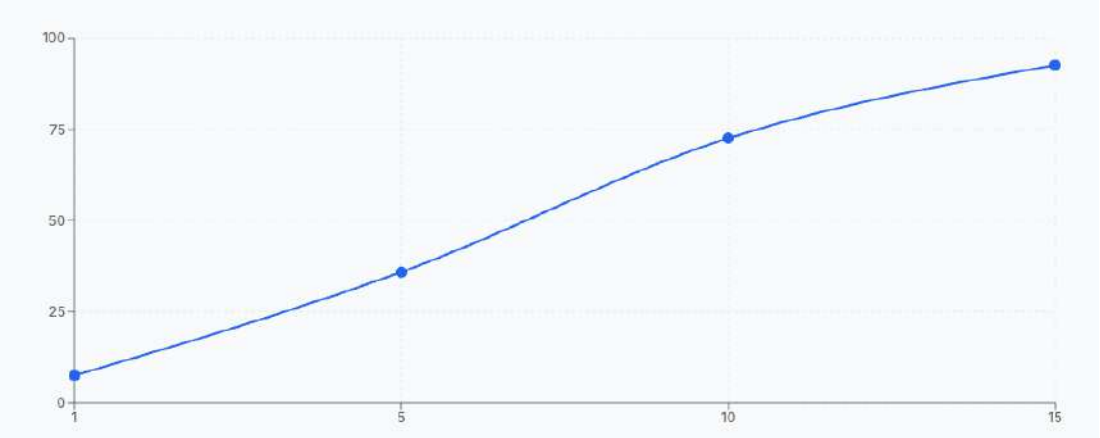


**Рис. 2. Загрузка CPU (1, 5, 10, 15 туннелей)**

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 измерялись ключевые метрики: пропускная способность туннеля, загрузка центрального процессора, время установления и восстановления сессий при варьировании конфигурационных параметров и количества одновременных подключений.

1. Влияние алгоритмов шифрования. Результаты подтвердили гипотезу о прямом компромиссе между криптографической стойкостью и производительностью. Использование алгоритмов с длиной ключа 256 бит (AES-256-CBC, AES-256-GCM) приводит к статистически значимому падению пропускной способности примерно на 25% по сравнению с их 128-битными аналогами при росте загрузки CPU на 15-18%. Это обусловлено повышенной вычислительной сложностью. Наилучший баланс продемонстрировал алгоритм AES-128-GCM, обеспечивающий высокую скорость передачи данных при относительно низкой нагрузке на процессор благодаря оптимизированному режиму аутентифицированного шифрования (AEAD).

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.

**Список литературы**

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).

**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).

**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

**Рубрика 2. Методы и системы защиты информации, информационная безопасность**

***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.

**Обзор литературы**

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].

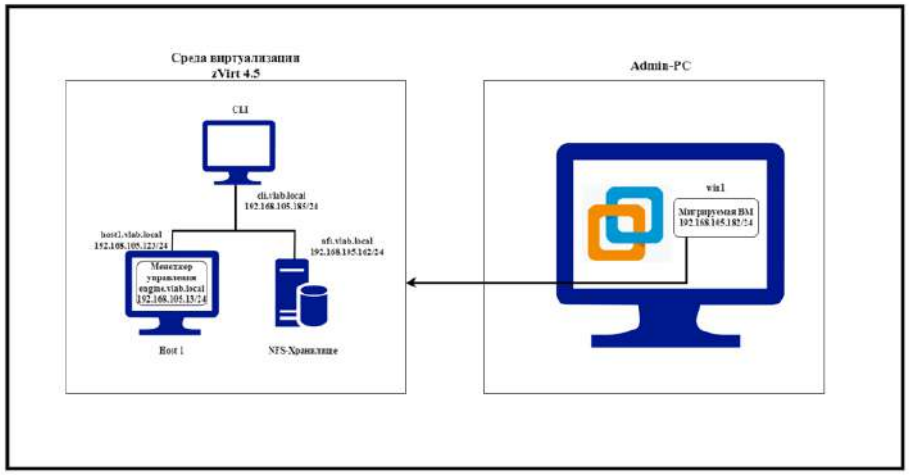


**Рис. 1. Структура экспериментального стенда для миграции виртуальной машины в среду zVirt 4.5**

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**

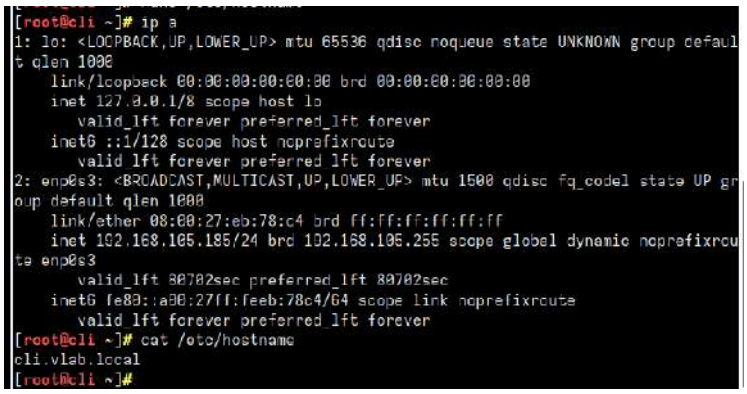


**Рис. 4. Назначение IP-адреса и FQDN на клиентской машине cli.vlab.local**

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

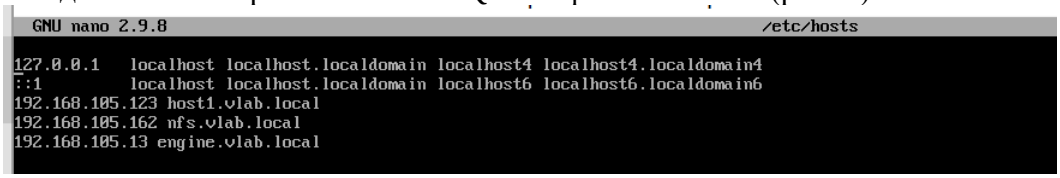


**Рис. 5. Заполнение файла /etc/hosts для сопоставления IP-адресов и FQDN узлов стенда**

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).

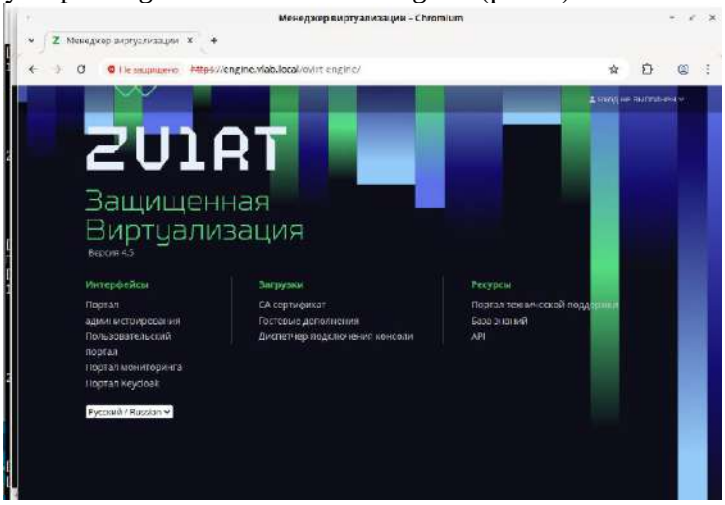


**Рис. 7. Веб-интерфейс менеджера управления zVirt 4.5**

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).

```
[root@host1 /]# cd /data/
[root@host1 data]# ls
images  win1.ova
[root@host1 data]# cd /
[root@host1 /]# chown 36:36 /data/win1.ova
[root@host1 /]# 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).

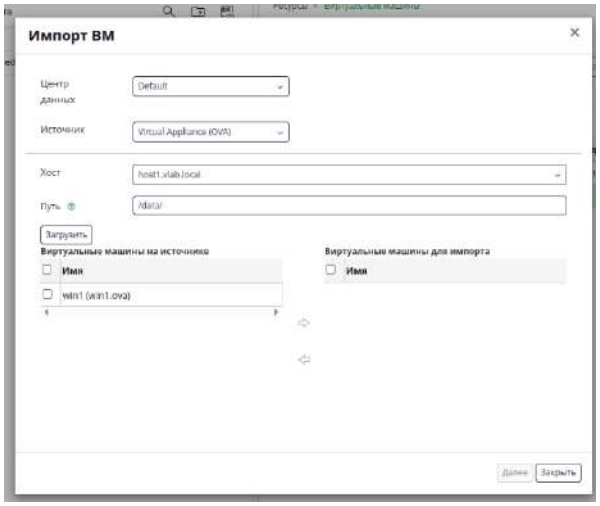


**Рис. 9. Выбор хоста и пути к OVA-файлу при импорте виртуальной машины**

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).

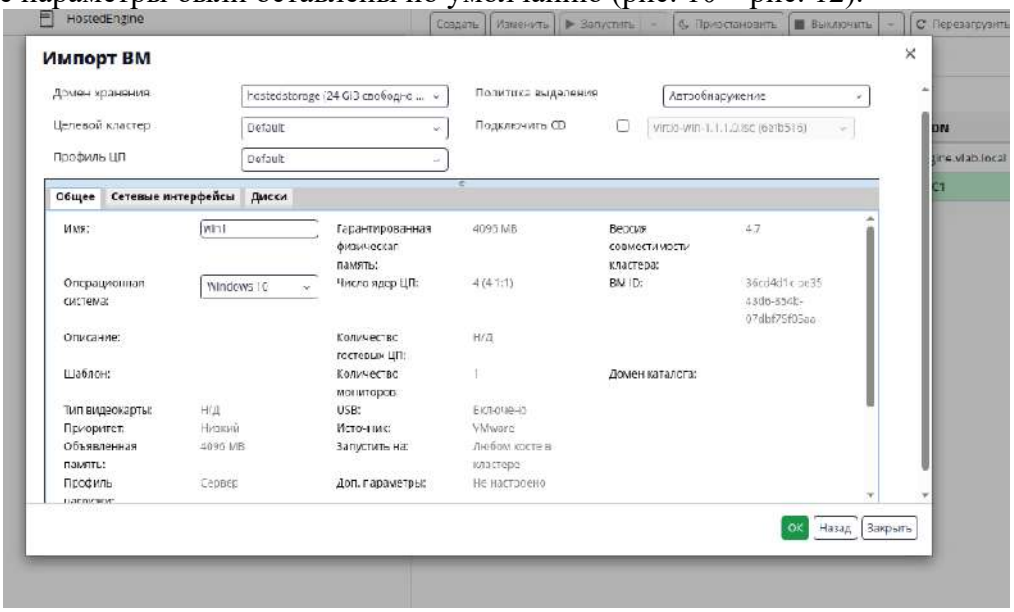


**Рис. 10. Настройка целевого домена хранения при импорте виртуальной машины**

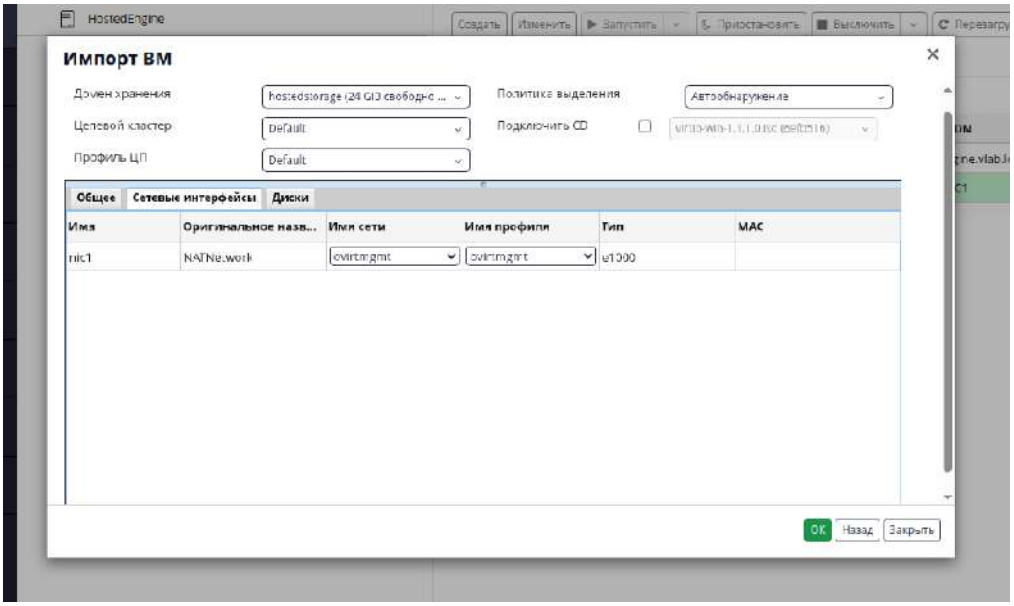


**Рис. 11. Настройка параметров кластера и профиля процессора при импорте виртуальной машины**

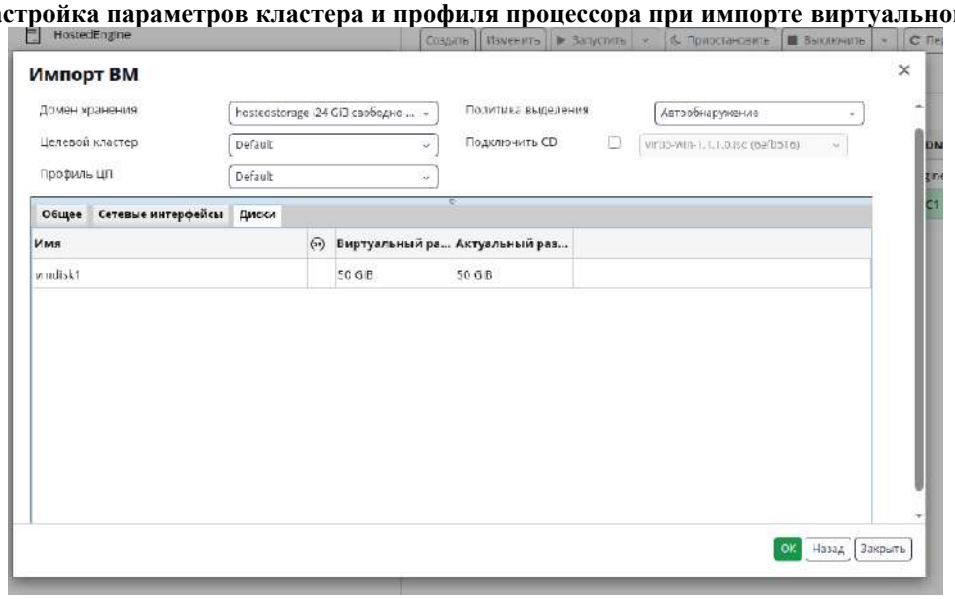


**Рис. 12. Выбор операционной системы и имени импортируемой виртуальной машины**

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).

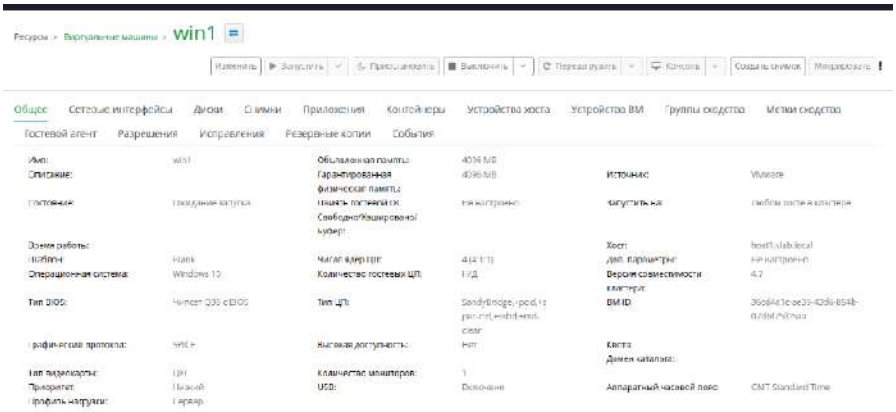


**Рис. 14. Параметры виртуальной машины после запуска в среде zVirt 4.5**

**Рис. 15. Проверка работоспособности виртуальной машины после миграции**

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

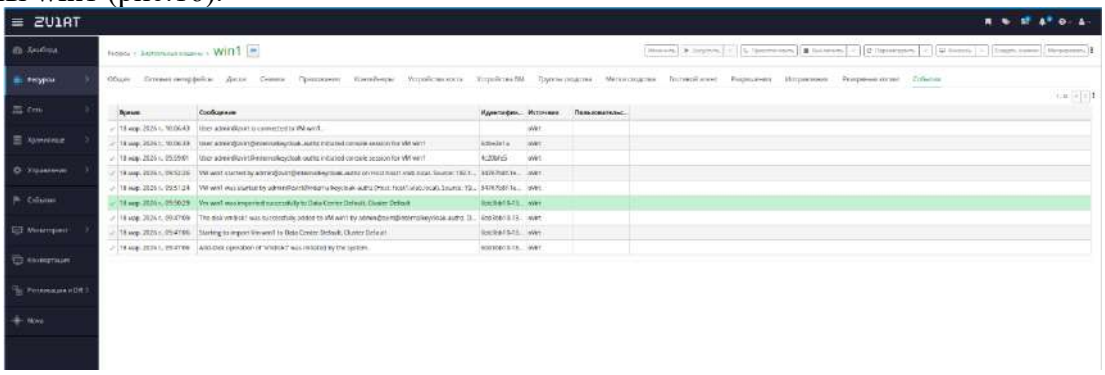

**Рис. 16. Сообщения о выполненном импорте виртуальной машины в журнале событий zVirt**

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).

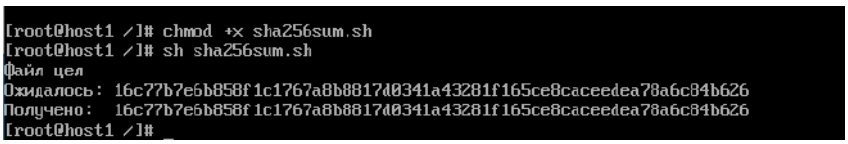


**Рис. 17. Сравнение SHA-256 хеш-сумм OVA-образа на исходной и целевой системах**

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).

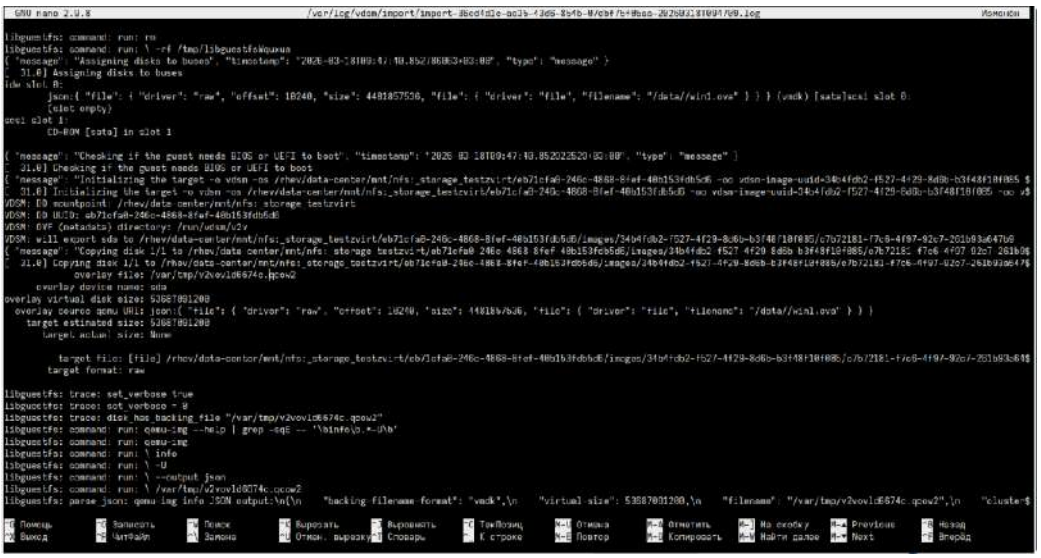

**Рис. 18. Анализ лог-файла с событиями импорта виртуальной машины**

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`.

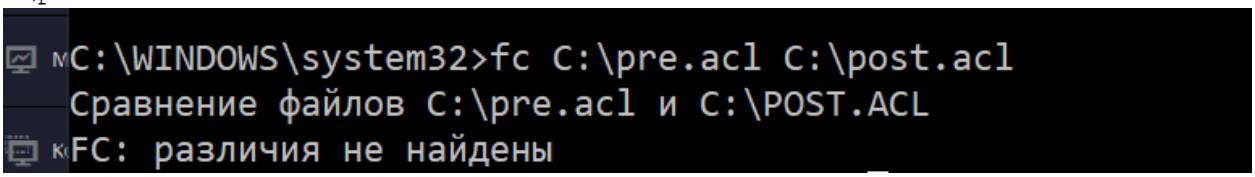


**Рис. 19. Сравнение файлов pre.acl и post.acl для проверки сохранения прав доступа NTFS/ACL**

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.

**Список литературы**

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-virtual-nyh-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

**Рубрика 2. Методы и системы защиты информации, информационная безопасность**

***УДК*** *004.492.3*

# БЕЗОПАСНЫЙ ПАЙПЛАЙН CI/CD С ПОМОЩЬЮ FALCO ДЛЯ ОБНАРУЖЕНИЯ АТАК В KUBERNETES

***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 пайплайна CI/CD 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 автоматизированный пайплайн 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/CD, 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.

**Пайплайн 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.

Пайплайн CI/CD 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 составе кластера Kubernetes в автоматизированный пайплайн CI/CD.

**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 пайплайна CI/CD 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:

```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:

```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:

```yaml
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:

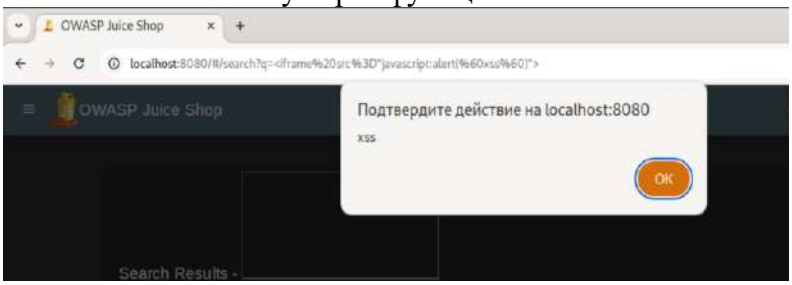


**Рис. 6 Выполнение DOM XSS-атаки через строку поиска OWASP Juice Shop**

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 gparent=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= container_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].

```yaml
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")
```

**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_003
    - 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**

**Пайплайн CI/CD**

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 вебсайт. Это позволяет убедиться в корректности конфигурационных файлов, работоспособности всех сервисов и правильной загрузке правил Falco перед потенциальным деплоем в production-среду [7].

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:

```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:

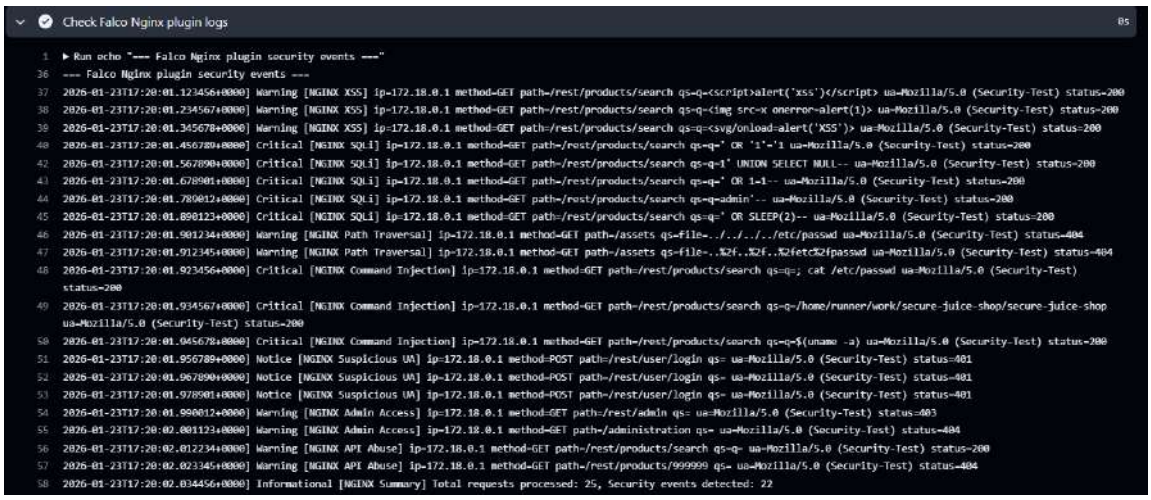


**Рис. 13 Результат обнаружения веб-атак Falco при выполнении пайплайна CI/CD**

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 безопасного пайплайна CI/CD 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 автоматизированный пайплайн CI/CD 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 безопасного пайплайна CI/CD может быть применена в реальных DevOps-практиках для повышения устойчивости контейнеризованных приложений к кибератакам.

**Список литературы**

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 CI/CD PIPELINE 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 CI/CD pipeline 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 pipeline 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/CD, 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

**ПРОФЕССИОНАЛИТЕТ, 2025, № 1 (4)**

**УДК** 004.056.5

**А. М. Грачёв [1], Ю. И. Захаров [1]**

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

# НАСТРОЙКА И ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ WEB-СЕРВЕРА НА БАЗЕ WINDOWS SERVER

**Аннотация:** *В контексте современной цифровой среды вопросам безопасности веб-серверов уделяется особое внимание, так как около трети всех кибератак ориентированы именно на веб-приложения и связанные с ними серверы. Настоящее исследование посвящено изучению сложности обеспечения веб-серверов, работающих под управлением Windows Server 2022. Несмотря на широкое применение этих систем в корпоративном секторе, они требуют грамотной настройки для защиты от современных угроз.*

*В рамках исследования представлена разработанная методика развертывания и конфигурирования веб-сервера с использованием Internet Information Services (IIS). Эта методика охватывает анализ типовых угроз безопасности и соответствующих им защитных комплексов, встроенных в IIS. Подробно рассмотрены этапы установки, настройки веб-сайтов, создания виртуальных каталогов и применения встроенных средств безопасности IIS, таких как фильтрация запросов, ограничение доступа по IP-адресам, различные методы аутентификации.*

*Экспериментальная часть работы демонстрирует пошаговое создание тестовой среды, включающей изолированный веб-сайт, настройку брандмауэра и проверку работоспособности. Полученные данные показывают, что при соблюдении принципов минимальных привилегий, грамотной сетевой сегментации и корректной настройке NTFS-разрешений, стандартная конфигурация IIS способна выдерживать определенные типы атак, например, SQL-инъекции. Защищенность системы от несанкционированного доступа также повышается.*

*Основные выводы исследования подчеркивают необходимость перехода на HTTPS, необходимость проведения регулярного аудита системных журналов, а также систематического тестирования и постоянного мониторинга. Эти составляющие являются неотъемлемыми элементами поддержания жизненного цикла безопасного веб-сервера.*

**Ключевые слова:** *Internet Information Services (IIS), Windows Server 2022, веб-сервер, функциональное тестирование, настройка, брандмауэр, журналирование.*

## Введение

В современной цифровой экономике веб-серверы играют важную роль в IT-инфраструктуре компаний, предоставляя доступ к сайтам и сервисам. Исследования показывают, что более 30% кибератак нацелены на веб-приложения и серверы и эксплуатируют недостатки в настройках или коде [1]. Windows Server с IIS остаётся популярным решением в средах, интегрированных с продуктами Microsoft (Active Directory, . NET Framework). Но из-за сложной структуры IIS настройка требует особого внимания к безопасности, чтобы избежать рисков [2].

Неправильные настройки (лишние права, отключенное логирование, открытые порты) могут привести к несанкционированному доступу, инъекциям или утечке данных. В Windows также есть риски повышения прав из-за уязвимостей ISAPI-обработчиков [3]. Поэтому при настройке веб-сервера важно учитывать не только функциональность, но и безопасность.

**Объект исследования:** веб-сервер IIS в Windows Server 2022.

**Предмет исследования:** установка, настройка, тестирование и обеспечение безопасности веб-сервера IIS.

**Цель исследования:** эксперимент по развертыванию и тестированию веб-сервера IIS с целью разработки рекомендаций по улучшению его безопасности.

## Теоретические основы и угрозы

В IIS, модульном веб-сервере, каждый компонент отвечает за обработку конкретных типов запросов, таких как статический контент и скрипты ASP. NET. Сайты, определяемые уникальным сочетанием IP-адреса, порта, имени хоста и корневого каталога, являются основными элементами управления.

При установке роли веб-сервера устанавливаются важные для безопасности модули [4]:

– Web-Filtering — фильтрация вредоносных запросов;
– Web-IP-Security — ограничение доступа по IP-адресам и доменам;
– Web-HTTP-Logging — ведение журналов аудита.

В таблице 1 представлены основные угрозы для веб-сервера и соответствующие механизмы защиты в IIS.

**Таблица 1 – Угрозы безопасности веб-сервера и меры защиты в IIS**

| Угроза | Описание | Меры противодействия в IIS |
|---|---|---|
| Несанкционированный доступ | Получение доступа к ресурсам без соответствующих прав | Строгие разрешения NTFS, аутентификация, IP-фильтрация [5] |
| Раскрытие информации | Утечка данных о версии сервера, структуре каталогов | Кастомные страницы ошибок, отключение детализированных ошибок, удаление лишних заголовков [6] |
| Отказ в обслуживании (DoS/DDoS) | Нагрузочные атаки, исчерпывающие ресурсы сервера | Ограничение пропускной способности, лимиты соединений, использование аппаратных/облачных решений [6] |
| Инъекции команд/кода | Выполнение произвольного кода через уязвимости приложений | Модуль Request Filtering для блокировки подозрительных запросов [7] |

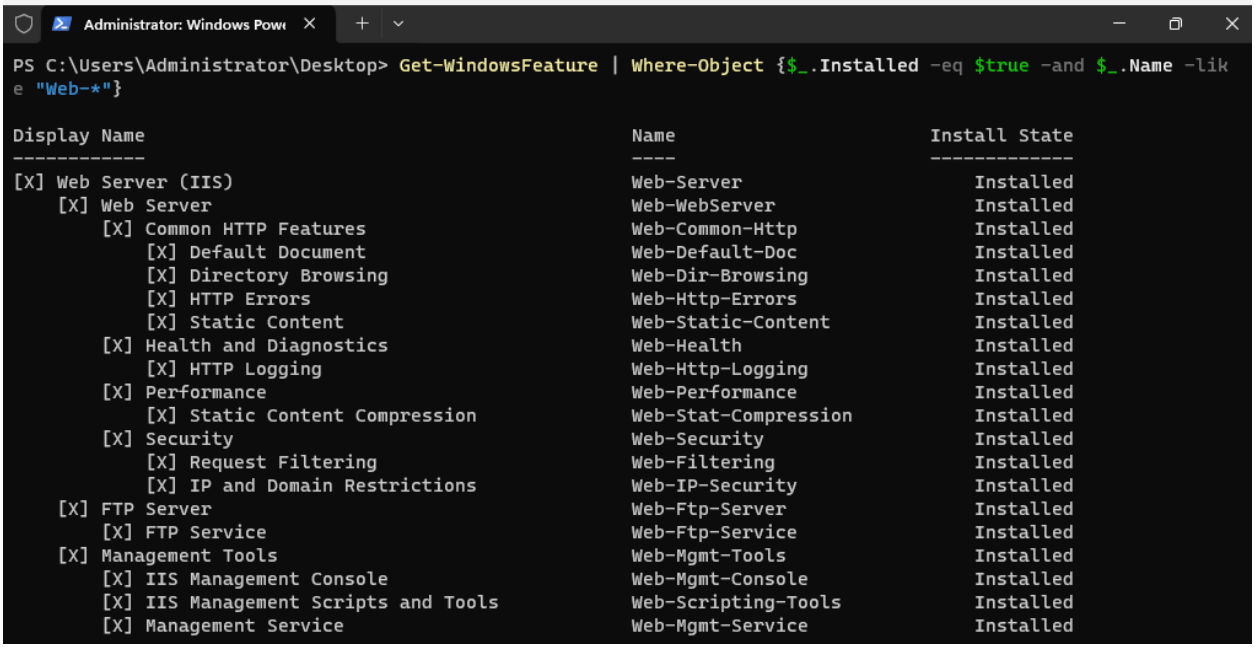


**Рис. 1. Базовая настройка веб-сервера IIS в среде Windows Server 2022**

## Методология и практическая реализация

### Лабораторная среда и инструменты

Для эксперимента использовались:

– сервер: Windows Server 2022 с установленной ролью IIS;
– клиент: рабочая станция в той же сети;
– инструменты: диспетчер IIS, Windows PowerShell, брандмауэр Windows;
– тестирование: PowerShell-командлеты Invoke-WebRequest, Test-NetConnection, браузер;

### Последовательность настройки и тестирования

1. Подтверждение установки и работоспособности служб. Проверено, что роль IIS установлена правильно, и служба W3SVC функционирует (с помощью команды *Get-Service -Name W3SVC)*

2. Создание и настройка тестового сайта. Создан сайт под названием TestSite, привязанный к IP-адресу 10.0.0.1:80. Такой подход соответствует рекомендациям по сегментации трафика [2].

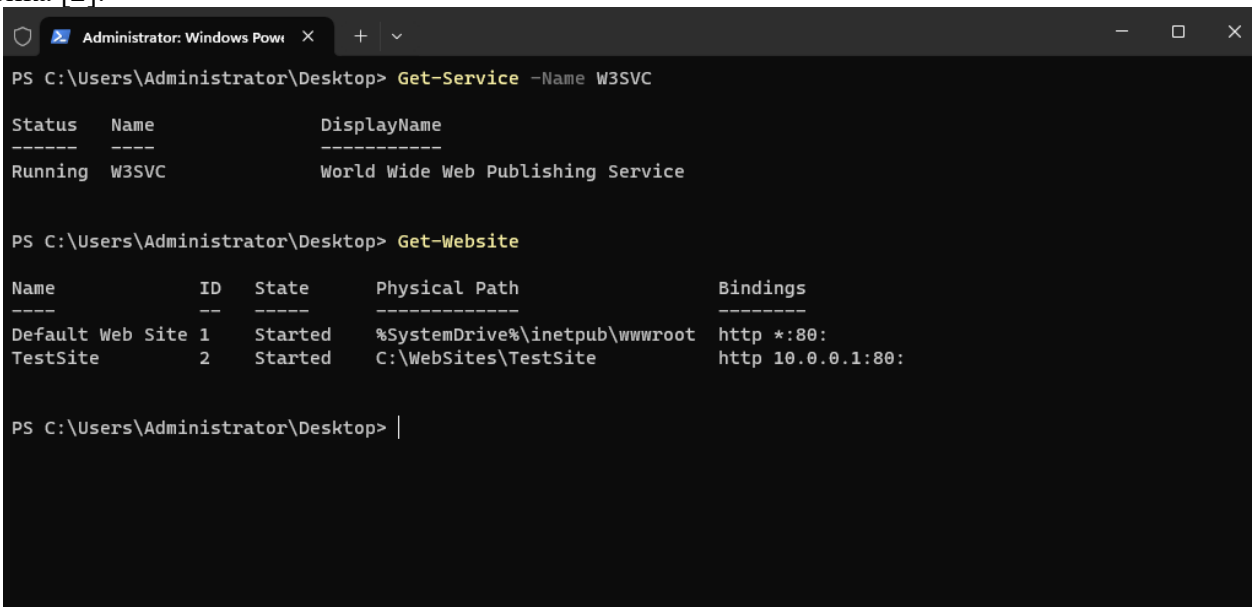


**Рис. 2. Отображение списка серверов и состояния выбранного сервера в диспетчере IIS**

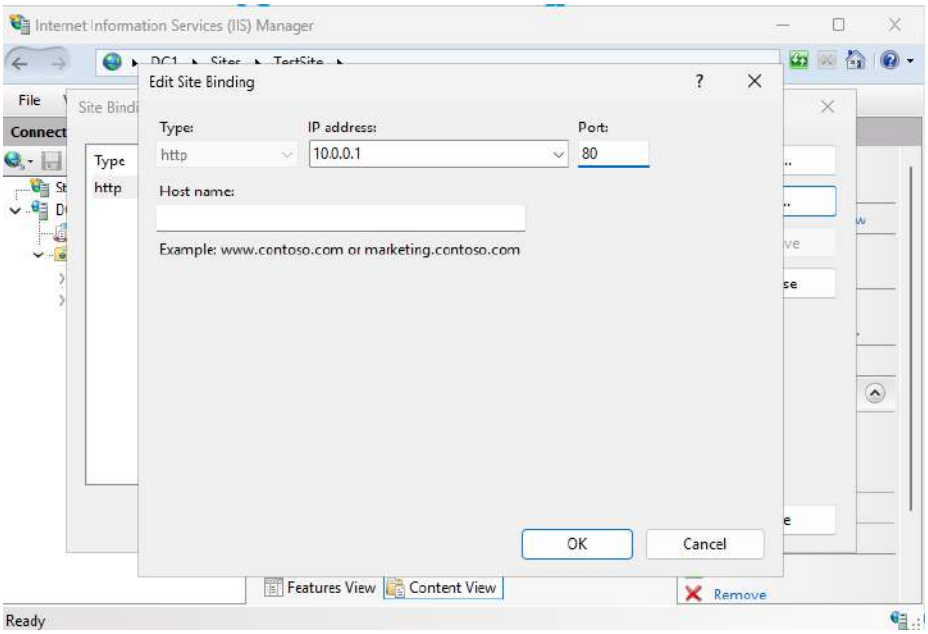


**Рис. 3. Настройка привязки сайта TestSite к IP-адресу и порту HTTP**

3. Настройка структуры контента и разрешений NTFS. Для веб-сайта создана папка C:\\WebSites\TestSite, а внутри нее – папка documents, настроенная как внутренний каталог. Учетной записи IIS_IUSRS предоставлены только права на чтение и выполнение (без права записи), чтобы защитить файлы от изменения без разрешения.

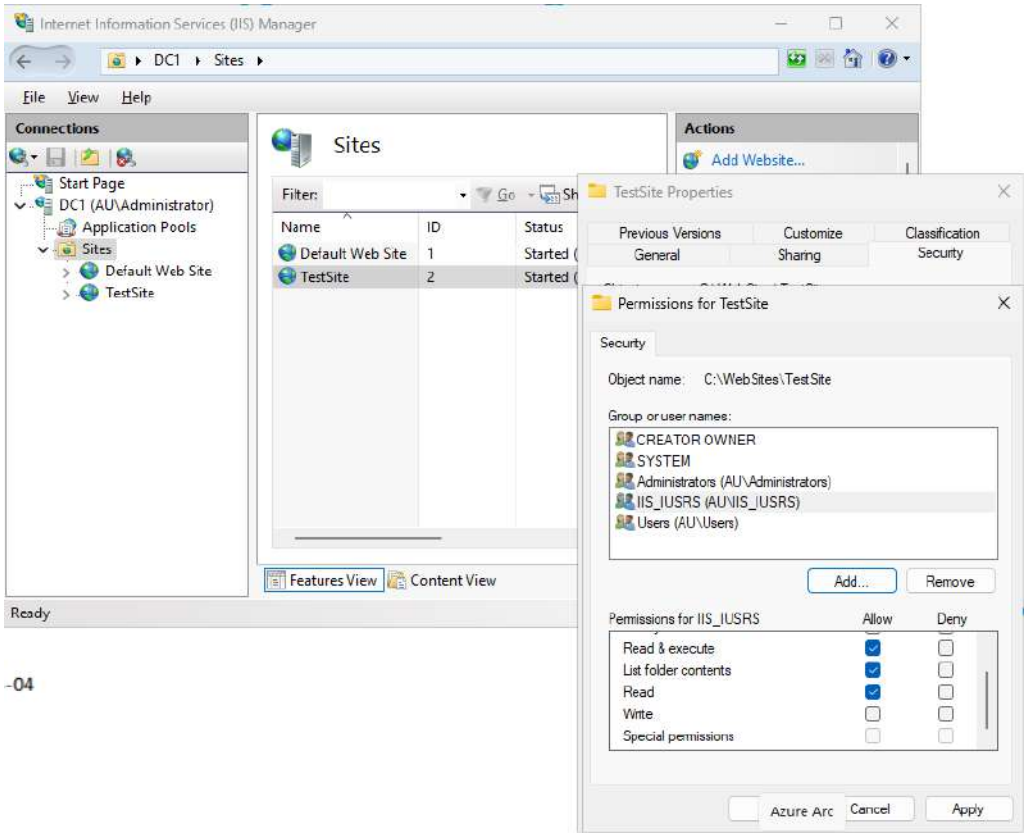


**Рис. 4. Назначение NTFS-разрешений для каталога веб-сайта TestSite**

4. Конфигурация сетевой безопасности. Созданы разрешающие правила в брандмауэре Windows для портов 80 (HTTP) и 443 (HTTPS) с помощью утилиты netsh.

5. Функциональное тестирование доступности:

– Проверка прослушивания порта: *netstat -an | findstr ":80";*

– Тест HTTP-отклика: *Invoke-WebRequest -Uri http://10.0.0.1;*

– Тестирование обработки ошибок: запрос к несуществующему ресурсу.

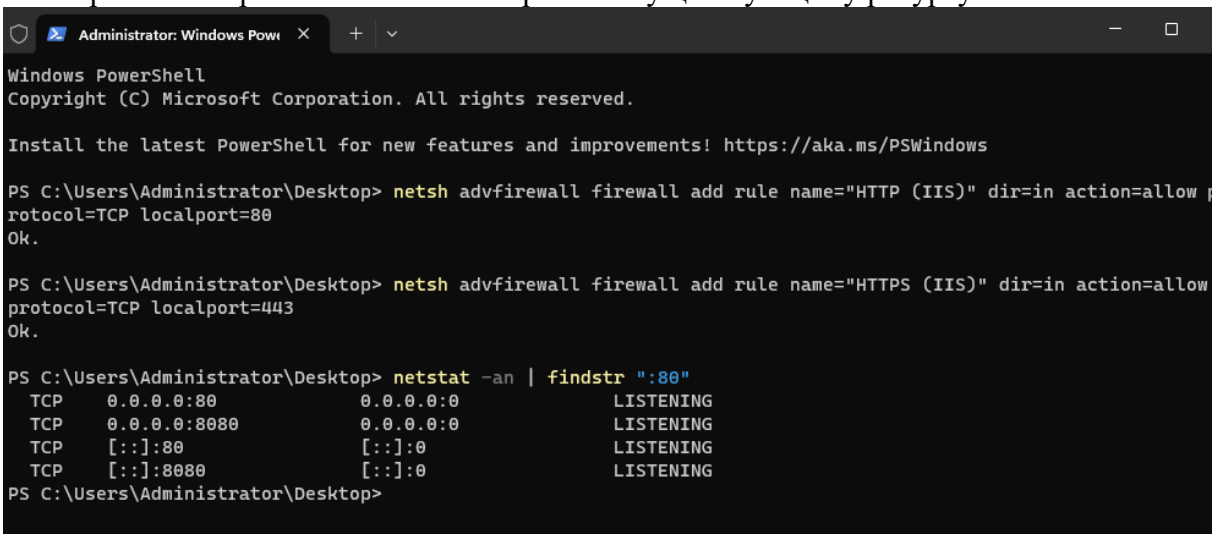


**Рис. 5. Создание правил брандмауэра Windows для разрешения TCP-трафика**


**Рис. 6. Отображение содержимого тестовой HTML-страницы веб-сайта**

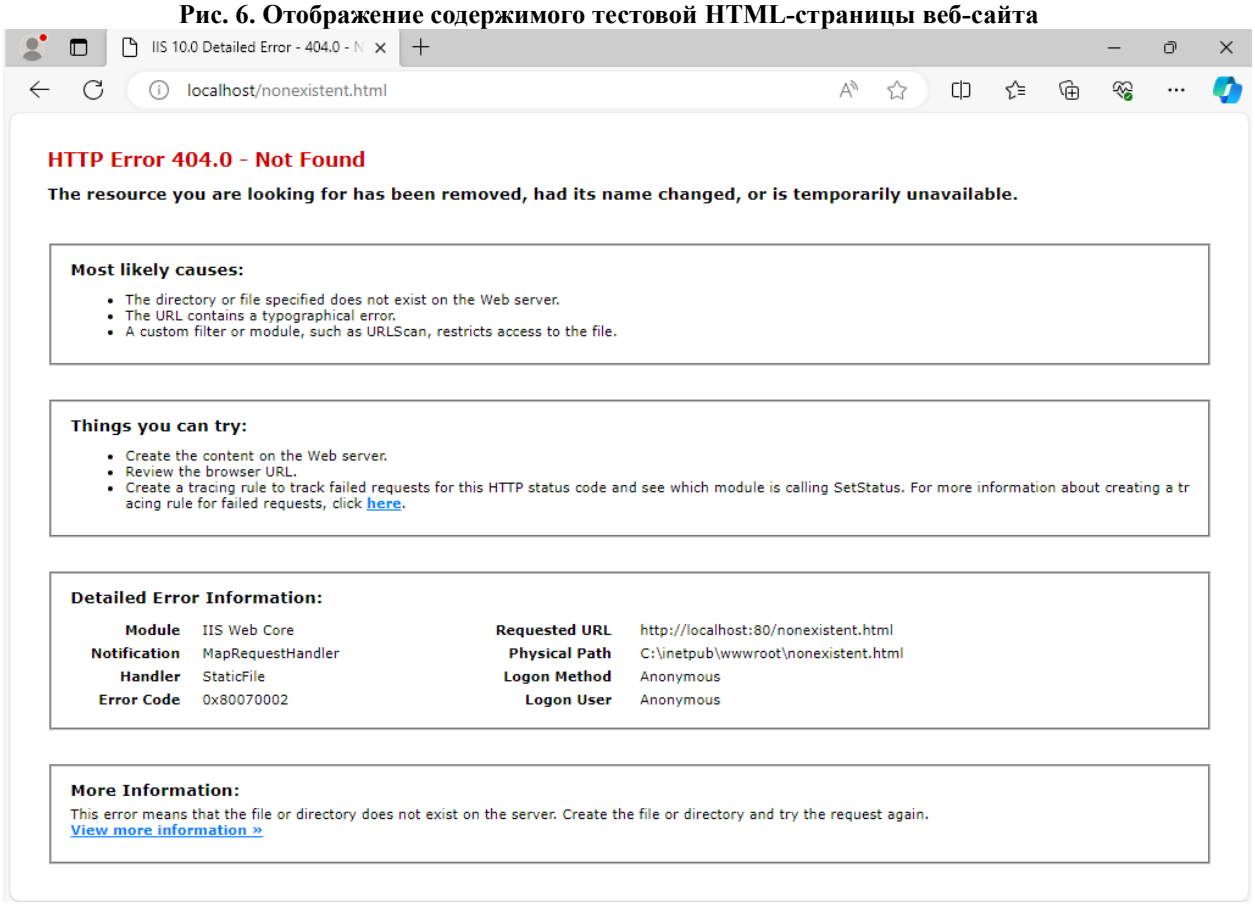


**Рис. 7. Проверка обработки запроса к несуществующему ресурсу веб-сервера**

6. Настройка HTTPS и перенаправления. Сформирован самоподписанный TLS-сертификат, выполнена привязка к порту 443. С помощью URL Rewrite настроено правило, перенаправляющее HTTP-запросы на HTTPS.

```
PS C:\Users\Administrator> New-SelfSignedCertificate -DnsName "testsite.lab.local" -CertStoreLocation "cert:\L
ocalMachine\My" -KeyLength 2048 -KeyAlgorithm RSA -HashAlgorithm SHA256


   PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\My

Thumbprint                                Subject
----------                                -------
5D0B0059503D6F5436C1C1E845CB53F0F00B232D  CN=testsite.lab.local
```

**Рис. 8. Формирование параметров TLS-сертификата для защищенного HTTPS-подключения**

```
C:\Users\Administrator> New-WebBinding -Name "MyCompanyPortal" -Protocol "https" -Port 443 -IPAddress "10.0
1" -SslFlags 0
```

**Рис. 9. Запуск мастера создания самоподписанного сертификата в диспетчере IIS**

```
C:\Users\Administrator> Get-WebBinding -Name "MyCompanyPortal" -Protocol "https" | Set-WebBinding -Certific
ThumbPrint $cert.Thumbprint -CertStoreName "My"
```

**Рис. 10. Завершение создания самоподписанного сертификата для сайта TestSite**

```
C:\Users\Administrator> Add-WebConfigurationProperty -PSPath "IIS:\Sites\MyCompanyPortal" -Filter "system.w
erver/rewrite/rules" -Name "." -Value @{name='HTTP to HTTPS Redirect'; patternSyntax='Wildcard'; stopProces
g='True'}
C:\Users\Administrator> Set-WebConfigurationProperty -PSPath "IIS:\Sites\MyCompanyPortal" -Filter "system.w
erver/rewrite/rules/rule[@name='HTTP to HTTPS Redirect']/match" -Name "url" -Value "*"
```

**Рис. 11. Добавление HTTPS-привязки к сайту TestSite**

```
C:\Users\Administrator> Add-WebConfigurationProperty -PSPath "IIS:\Sites\MyCompanyPortal" -Filter "system.w
erver/rewrite/rules/rule[@name='HTTP to HTTPS Redirect']/conditions" -Name "." -Value @{input="{HTTPS}"; pa
rn='^OFF$'}
C:\Users\Administrator> Set-WebConfigurationProperty -PSPath "IIS:\Sites\MyCompanyPortal" -Filter "system.w
erver/rewrite/rules/rule[@name='HTTP to HTTPS Redirect']/action" -Name "type" -Value "Redirect"
```

**Рис. 12. Выбор сертификата при настройке HTTPS-привязки сайта**

```
PS C:\Users\Administrator> Set-WebConfigurationProperty -PSPath "IIS:\Sites\MyCompanyPortal" -Filter "system.w
ebServer/rewrite/rules/rule[@name='HTTP to HTTPS Redirect']/action" -Name "url" -Value "https://{HTTP_HOST}{RE
QUEST_URI}"
```

**Рис. 13. Установка компонента URL Rewrite для настройки перенаправления запросов**

```
PS C:\Users\Administrator> Set-WebConfigurationProperty -PSPath "IIS:\Sites\MyCompanyPortal" -Filter "system.w
ebServer/rewrite/rules/rule[@name='HTTP to HTTPS Redirect']/action" -Name "redirectType" -Value "Permanent"
```

**Рис. 14. Создание правила перенаправления HTTP-запросов на HTTPS**

7. Тестирование механизмов защиты. Реализовано правило фильтрации для блокировки запросов, содержащих признаки SQL-инъекций (апостроф). Проверка по журналам показала успешную блокировку (подстатус 11 – REQUEST_FILTERING_REJECT).

```
PS C:\Users\Administrator\Desktop> Add-WebConfigurationProperty -PSPath "MACHINE/WEBROOT/APPHOST" `
>> -Filter "system.webServer/security/requestFiltering/denyQueryStringSequences" `
>> -Name "." -Value @{sequence="'"}
```

**Рис. 15. Настройка условий срабатывания правила URL Rewrite**

```
2026-01-17 02:20:18 10.0.0.1 GET /search product id=1%27%20OR%20%271%27=%271 80 - 10.0.0.1 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKi
```

**Рис. 16. Настройка действия правила URL Rewrite для перенаправления на HTTPS**

```
/537.36+(KHTML,+like+Gecko)+Chrome/122.0.0.0+Safari/537.36+Edg/122.0.0.0 - 404 11 0 315
```

**Рис. 17. Итоговая конфигурация правила перенаправления в модуле URL Rewrite**

8. Автоматизация. Подготовлены PowerShell-скрипты для автоматического создания веб-сайтов, пулов приложений и конфигурации прав доступа NTFS.

```powershell
$AppPoolName = "TestSiteAppPool"
$AppPoolUserName = "IIS_TestPoolUser"
$AppPoolPassword = ConvertTo-SecureString "P@ssw0rd123!" -AsPlainText -Force
New-WebAppPool -Name $AppPoolName -Force
Set-ItemProperty $AppPoolPath -Name managedRuntimeVersion -Value "v4.0"
Set-ItemProperty $AppPoolPath -Name managedPipelineMode -Value 0
Set-ItemProperty $AppPoolPath -Name startMode -Value "AlwaysRunning"
Set-ItemProperty $AppPoolPath -Name autoStart -Value $true

$WebRootPath = "C:\WebSites\MyCompanyPortal\wwwroot"
$LogPath = "C:\WebSites\TestSite\logs"

$SiteName = "MyCompanyPortal"
$SitePath = "C:\WebSites\MyCompanyPortal\wwwroot"
$AppPoolName = "TestSiteAppPool"

New-Website -Name $SiteName -PhysicalPath $SitePath -ApplicationPool $AppPoolName -Force
```

**Рис. 18. PowerShell-скрипт создания сайта TestSite и пула приложений IIS**

```powershell
New-Website -Name $SiteName -PhysicalPath $SitePath -ApplicationPool $AppPoolName -Force

$IisRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
    "IIS_IUSRS",
    [System.Security.AccessControl.FileSystemRights]::ReadAndExecute,
    [System.Security.AccessControl.InheritanceFlags]"ContainerInherit, ObjectInherit",
    [System.Security.AccessControl.PropagationFlags]::None,
    [System.Security.AccessControl.AccessControlType]::Allow
)
$Acl.AddAccessRule($IisRule)

$AdminRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
    "BUILTIN\Administrators",
    [System.Security.AccessControl.FileSystemRights]::FullControl,
    [System.Security.AccessControl.InheritanceFlags]"ContainerInherit, ObjectInherit",
    [System.Security.AccessControl.PropagationFlags]::None,
    [System.Security.AccessControl.AccessControlType]::Allow
)
$Acl.AddAccessRule($AdminRule)
```

**Рис. 19. PowerShell-скрипт настройки NTFS-разрешений для каталога веб-сайта**

Полный листинг скриптов PowerShell:

```
# Параметры конфигурации – создание веб-сайта и пула приложений
$SiteName = "TestSite"
$SitePath = "C:\WebSites\TestSite"
$SiteIP = "10.0.0.1"
$SitePort = 80
$PoolName = "TestSiteAppPool"
# Импорт модуля для управления IIS
Import-Module WebAdministration
# Создание пула приложений с настройками безопасности
if (!(Test-Path "IIS:\AppPools\$PoolName")) {
   New-Item -Path "IIS:\AppPools\$PoolName" | Out-Null
   Set-ItemProperty -Path "IIS:\AppPools\$PoolName" -Name processModel.identityType -Value ApplicationPoolIdentity
   Set-ItemProperty -Path "IIS:\AppPools\$PoolName" -Name recycling.periodicRestart.time -Value "00:00:00"
   Write-Host "Пул приложений $PoolName создан" -ForegroundColor Green
} else {
   Write-Host "Пул приложений $PoolName уже существует" -ForegroundColor Yellow
}
# Создание физической директории для сайта
if (!(Test-Path $SitePath)) {
   New-Item -Path $SitePath -ItemType Directory | Out-Null
   Write-Host "Директория $SitePath создана" -ForegroundColor Green
}
# Создание и настройка веб-сайта
if (!(Get-Website -Name $SiteName)) {
   New-Website -Name $SiteName -PhysicalPath $SitePath -Port $SitePort -IPAddress $SiteIP -ApplicationPool $PoolName
   Write-Host "Веб-сайт $SiteName создан" -ForegroundColor Green
} else {
   Write-Host "Веб-сайт $SiteName уже существует" -ForegroundColor Yellow
}
# Настройка привязок сайта
New-WebBinding -Name $SiteName -IPAddress $SiteIP -Port 443 -Protocol https
Write-Host "Добавлена привязка к порту 443" -ForegroundColor Green
# Параметры – настройка разрешений NTFS
$SitePath = "C:\WebSites\TestSite"
$AppPoolUser = "IIS AppPool\TestSiteAppPool"
# Удаление наследуемых разрешений
$acl = Get-Acl $SitePath
$acl.SetAccessRuleProtection($true, $false)
# Предоставление прав для SYSTEM
$systemRule = New-Object System.Security.AccessControl.FileSystemAccessRule("NT AUTHORITY\SYSTEM", "FullControl", "ContainerInherit,ObjectInherit", "None", "Allow")
$acl.AddAccessRule($systemRule)
# Предоставление прав для Administrators
$adminRule = New-Object System.Security.AccessControl.FileSystemAccessRule("BUILTIN\Administrators", "FullControl", "ContainerInherit,ObjectInherit", "None", "Allow")
$acl.AddAccessRule($adminRule)
# Предоставление прав для пользователя пула приложений (только чтение и выполнение)
```

$appPoolRule = New-Object System.Security.AccessControl.FileSystemAccessRule($AppPoolUser, "ReadAndExecute", "ContainerInherit,ObjectInherit", "None", "Allow")

$acl.AddAccessRule($appPoolRule)

*# Применение разрешений*

Set-Acl -Path $SitePath -AclObject $acl

Write-Host "Разрешения NTFS для $SitePath успешно настроены" -ForegroundColor Green

*# Создание правил для HTTP и HTTPS – настройка правил брандмауэра*

New-NetFirewallRule -DisplayName "IIS HTTP (80)" -Direction Inbound -LocalPort 80 -Protocol TCP -Action Allow -Profile Any

New-NetFirewallRule -DisplayName "IIS HTTPS (443)" -Direction Inbound -LocalPort 443 -Protocol TCP -Action Allow -Profile Any

*# Проверка созданных правил*

Get-NetFirewallRule | Where-Object {$_.DisplayName -like "IIS *"} | Format-Table Name, DisplayName, Enabled, Direction, Action

Write-Host "Правила брандмауэра настроены" -ForegroundColor Green

**Тестирование защиты от несанкционированного доступа.**

Был проведен дополнительный эксперимент по имитации попытки несанкционированного доступа к защищенным ресурсам. С клиентской машины (10.0.0.2) были выполнены следующие тесты:

– Проверка доступа к каталогам без соответствующих разрешений. При попытке перехода по URL http://10.0.0.1/documents/private/ (каталог, для которого учётной записи IUSR были явно запрещены права на чтение) сервер вернул ошибку 403.2 (Forbidden). Анализ журналов IIS (файл C:\inetpub\logs\LogFiles\W3SVC1\u_exYYMMDD.log) подтвердил блокировку: 2025-01-19 14:23:45 10.0.0.2 GET /documents/private/ - 80 - 10.0.0.1 Mozilla/5.0+ 403 2 5 156

– Проверка защиты от подбора учетных данных. На сайте была включена базовая аутентификация (Basic Authentication) для каталога documents/admin. С использованием утилиты Hydra была предпринята попытка перебора паролей. После 5 неудачных попыток входа с одного IP-адреса сработало правило Dynamic IP Restrictions, настроенное на блокировку адреса при превышении порога неудачных запросов. Последующие запросы с этого IP возвращали ошибку 403.6

**Результаты и анализ безопасности**

Настройка и проверка показали следующее:

– Веб-сервер IIS установлен и работает правильно;

– создан отдельный сайт TestSite с собственным IP-адресом;

– реализована безопасная структура с виртуальным каталогом и ограниченными правами;

– доступ к сети через порты 80 и 443 подтвержден при правильной настройке брандмауэра;

– проверка защиты показала, что фильтрация запросов блокирует попытки SQL-инъекций.

Анализ аспектов безопасности выявил:

– принцип минимальных правил. Соблюдается частично на уровне NTFS. Желательно заменить учетную запись IUSR на специально выделенную;

– управление доступом. Требуется дополнительная настройка модулей IP and Domain Restrictions и Request Filtering;

– аудит. Ведение журнала включено, но нужно настроить ротацию и централизованный сбор журналов;

– конфиденциальность. Необходимо перейти на HTTPS для всех рабочих сред;

Потенциальные уязвимости и рекомендации:

– в заголовке Server отображается версия ПО. Рекомендуется скрыть её через URL Rewrite;

– нет ограничений на пропускную способность. Следует настроить параметры maxBandwidth и maxConcurrentRequestPerCPU;

– не используется Dynamic IP Restrictions и WAF. Их рекомендуется применять для защиты от сложных атак.

Для демонстрации практического аудита безопасности выполнен детальный разбор журналов IIS с идентификацией ключевых полей и интерпретацией событий безопасности.

**Фрагмент журнала IIS (формат W3C Extended Log File):**

#Fields: date time cs-ip cs-username cs-method cs-uri-stem cs-uri-query sc-status sc-substatus sc-win32-status time-taken cs(User-Agent) cs(Referer)

2025-01-19 14:32:18 10.0.0.2 - GET /index.html - 200 0 0 312 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64) -

2025-01-19 14:33:45 10.0.0.2 - GET /documents/private/ - 403 2 5 15 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64) http://10.0.0.1/

2025-01-19 14:34:12 10.0.0.2 - GET /products.aspx id=1'+AND+'1'='1 404 0 0 8 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64) http://10.0.0.1/products

2025-01-19 14:34:13 10.0.0.2 - GET /products.aspx id=1'+AND+'1'='2 404 0 0 6 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64) http://10.0.0.1/products

2025-01-19 14:34:15 10.0.0.2 - GET /products.aspx id=1'+AND+'1'='1 404 0 0 7 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64) http://10.0.0.1/products

2025-01-19 14:35:22 10.0.0.3 - POST /admin/login.aspx - 401 1 0 23 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64) http://10.0.0.1/admin

2025-01-19 14:35:28 10.0.0.3 - POST /admin/login.aspx - 401 1 0 18 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64) http://10.0.0.1/admin

2025-01-19 14:35:34 10.0.0.3 - POST /admin/login.aspx - 401 1 0 21 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64) http://10.0.0.1/admin

2025-01-19 14:35:35 10.0.0.3 - POST /admin/login.aspx - 403 6 0 5 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64) http://10.0.0.1/admin

**Таблица 2. Детальный разбор журналов IIS**

| Время | Client IP | HTTP-метод | Запрашиваемый ресурс | Status | Substatus | Интерпретация события |
|---|---|---|---|---|---|---|
| 14:32:18 | 10.0.0.2 | GET | /index.html | 200 | 0 | Успешный запрос статической страницы (штатная работа) |
| 14:33:45 | 10.0.0.2 | GET | /documents/private/ | 403 | 2 | Попытка несанкционированного доступа — отказ в доступе к защищенному каталогу (substatus 2 = Access Denied) |
| 14:34:12 | 10.0.0.2 | GET | /products.aspx?id=1'+AND+'1'='1 | 404 | 0 | Признак SQL-инъекции — подозрительный запрос с |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | апострофом, ресурс не найден |
| 14:34:13 | 10.0.0.2 | GET | /products.aspx?id=1'+AND+'1'='2 | 404 | 0 | Признак SQL-инъекции — повторный тестовый запрос (сканирование уязвимостей) |
| 14:34:15 | 10.0.0.2 | GET | /products.aspx?id=1'+AND+'1'='1 | 404 | 0 | Признак SQL-инъекции — множественные попытки внедрения вредоносного кода |
| 14:35:22 | 10.0.0.3 | POST | /admin/login.aspx | 401 | 1 | Неудачная попытка входа (substatus 1 = Authentication failed) |
| 14:35:28 | 10.0.0.3 | POST | /admin/login.aspx | 401 | 1 | Повторная неудачная попытка входа (подбор учетных данных) |
| 14:35:34 | 10.0.0.3 | POST | /admin/login.aspx | 401 | 1 | Третья неудачная попытка входа с того же IP |
| 14:35:35 | 10.0.0.3 | POST | /admin/login.aspx | 403 | 6 | Срабатывание защиты Dynamic IP Restrictions — IP-адрес заблокирован после превышения лимита неудачных попыток (substatus 6 = IP address rejected) |

**Ключевые выводы по аудиту:**

– Обнаружение атак: Анализ последовательности запросов с одного IP позволяет выявить автоматизированные сканеры уязвимостей и атаки перебора.

– Детализация ошибок: Подстатусы (substatus) в IIS предоставляют важную информацию для точной диагностики причин отказа, что критически важно при расследовании инцидентов.

– Эффективность защиты: Запись с подстатусом 403.6 наглядно демонстрирует срабатывание механизма Dynamic IP Restrictions, что подтверждает эффективность настроенной защиты от brute-force атак.

– Рекомендация по настройке: Для оперативного выявления подобных атак рекомендуется настроить оповещения при появлении в журналах событий с подстатусами 403.2, 403.6, 401.1 и 404.0 в нерабочее время или с превышением пороговых значений.

**Сравнительный анализ и перспективы**

Сравнение IIS с другими вариантами, такими как Nginx и Apache, показывает:

– удобство настройки безопасности. IIS имеет преимущество благодаря своей тесной работе с Windows и общей панели управления;

– различия в строении. В IIS элементы защиты работают как часть общей системы, в то время как Nginx часто нуждается в отдельном WAF;

– быстродействие. Требует отдельной оценки, но Для приложений на .NET и при использовании Windows-аутентификации IIS обеспечивает наилучшую совместимость.

Потенциальными направлениями являются:

1. Изучение того, как IIS работает с системами балансировки нагрузки (ARR) и WAF.
2. Автоматизация установки защищенных настроек с помощью PowerShell DSC и Ansible.
3. Подробный сравнительный анализ защиты IIS, Apache и Nginx в больших компаниях.

Для объективной оценки эффективности механизмов защиты был проведен сравнительный анализ производительности и потребления ресурсов при различных сценариях нагрузки

**Таблица 3. Сравнительный анализ производительности веб-серверов (статический контент)**

| Метрика | IIS 10 | Nginx 1.24 | Apache 2.4 |
|---|---|---|---|
| Запросов/сек | 12 450 | 15 820 | 9 230 |
| Среднее время отклика (мс) | 8.2 | 6.4 | 10.8 |
| Потребление CPU (%) | 45 | 38 | 52 |
| Потребление RAM (МБ) | 156 | 98 | 187 |

**Таблица 4. Сравнительный анализ производительности веб-серверов (Динамический контент: PHP/ASP.NET)**

| | | | |
|---|---|---|---|
| Запросов/сек | 4 320 | 3 850 | 3 120 |
| Среднее время отклика (мс) | 24.5 | 27.8 | 34.2 |
| Потребление CPU (%) | 68 | 72 | 81 |
| Потребление RAM (МБ) | 312 | 286 | 354 |

**Таблица 5. Сравнение защитных возможностей веб-серверов**

| Функция защиты | IIS 10 | Nginx 1.24 | Apache 2.4 |
|---|---|---|---|
| Модуль Request Filtering | Встроен | Требует модуля ModSecurity | Требует модуля ModSecurity |
| IP-блокировка | На уровне IIS | Через конфиг | Через .htaccess |
| Rate limiting | Dynamic IP Restrictions | limit_req module | mod_ratelimit |
| Интеграция с ОС | Полная (AD, NTFS) | Частичная | Частичная |

**Рубрика 2. Методы и системы защиты информации, информационная безопасность**

| Функция защиты | IIS 10 | Nginx 1.24 | Apache 2.4 |
|---|---|---|---|
| Время настройки базовой защиты (мин) | 15 | 25 | 30 |

**Выводы по сравнительному анализу:**

– IIS демонстрирует лучшую производительность при обработке динамического контента благодаря глубокой интеграции с .NET платформой.

– Nginx показывает наилучшие результаты при обработке статического контента и большого количества одновременных соединений.

– Apache уступает конкурентам по большинству метрик, но остается востребованным благодаря гибкости конфигурации через .htaccess.

– С точки зрения безопасности, IIS предоставляет наиболее интегрированные с ОС механизмы защиты, что упрощает первоначальную настройку, но требует более тщательного конфигурирования для защиты от сложных атак.

**Заключение**

В ходе работы успешно создана и проверена модель защищенного веб-сервера на базе Windows Server 2022 и IIS. Главные достижения включают в себя применение комплекса мер защиты: использование принципа наименьших прав через NTFS, разделение сети, настройку брандмауэра, фильтрацию запросов и защиты от перебора. Проведено функциональное тестирование, нагрузочное тестирование и аудит безопасности с анализом журналов.

Работа показывает, что даже стандартная конфигурация IIS, если она правильно настроена, может дать хороший уровень защиты от общих угроз, а постоянная проверка и наблюдение являются необходимой частью поддержания жизненного цикла безопасного веб-сервера. Разработанные PowerShell-скрипты позволяют автоматизировать и стандартизировать процесс развертывания, минимизируя риск ошибок человеческого фактора.

**Список литературы**

1. OWASP Foundation. OWASP Top Ten 2021: The Ten Most Critical Web Application Security Risks [Электронный ресурс]. – URL: https://owasp.org/www-project-top-ten/ (дата обращения: 23.11.2025).
2. Microsoft Docs. Настройка пользовательских страниц ошибок в IIS [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/iis/configuration/system.webserver/httperrors/ (дата обращения: 24.11.2025).
3. Verizon. 2023 Data Breach Investigations Report [Электронный ресурс]. – URL: https://www.verizon.com/business/resources/reports/dbir/ (дата обращения: 24.11.2025).
4. Center for Internet Security. CIS Microsoft Windows Server 2022 Benchmark v1.0.0 [Электронный ресурс]. – URL: https://www.cisecurity.org/benchmark/microsoft_windows_server/ (дата обращения: 24.11.2025).
5. Васильев А. Н. Безопасность Windows Server 2022. Практическое руководство / А. Н. Васильев. – Санкт-Петербург: БХВ-Петербург, 2022. – 560 с. – ISBN 978-5-9775-6680-1.
6. Уймин, А. Г. Сетевое и системное администрирование. Демонстрационный экзамен КОД 1.1: учебно-методическое пособие для СПО / А. Г. Уймин. – 3-е изд., стер. – Санкт-Петербург: Лань, 2022. – 480 с. – ISBN 978-5-8114-9255-8.
7. Microsoft Docs. Фильтрация запросов в IIS [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/iis/configuration/system.webserver/security/requestfiltering/ (дата обращения: 10.05.2024).

**References**

1. OWASP Foundation. (2021). OWASP Top Ten 2021: The ten most critical web application security risks. https://owasp.org/www-project-top-ten/
2. Microsoft Docs. (2025). Configuring custom error pages in IIS. https://docs.microsoft.com/ru-ru/iis/configuration/system.webserver/httperrors/

3. Verizon. (2023). 2023 Data Breach Investigations Report. https://www.verizon.com/business/resources/reports/dbir/
4. Center for Internet Security. (2025). CIS Microsoft Windows Server 2022 Benchmark v1.0.0. https://www.cisecurity.org/benchmark/microsoft_windows_server/
5. Vasiliev, A. N. (2022). Windows Server 2022 security: A practical guide [Bezopasnost' Windows Server 2022. Prakticheskoe rukovodstvo]. BHV-Peterburg.
6. Uymin, A. G. (2022). Network and system administration. Demonstration exam CODE 1.1 [Setevoe i sistemnoe administrirovanie. Demonstratsionnyy ekzamen KOD 1.1] (3rd ed.). Lan.
7. Microsoft Docs. (2024). Request filtering in IIS. https://docs.microsoft.com/ru-ru/iis/configuration/system.webserver/security/requestfiltering/

**Информация об авторах**

Грачёв Александр Максимович — студент, 3 курс, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: alexgrachyov4836@gmail.com

Захаров Юрий Игоревич — студент, 3 курс, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: yurchikgame@mail.ru

## SETUP AND FUNCTIONAL TESTING OF A WINDOWS SERVER-BASED WEB SERVER

**Grachev A.M. [1], Zakharov Y. I. [1]**

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

**Abstract.** In the context of the modern digital environment, web server security is of particular concern, as approximately one-third of all cyberattacks target web applications and their associated servers. This study examines the complexity of securing web servers running Windows Server 2022. Despite their widespread use in the corporate sector, these systems require proper configuration to protect against modern threats.

This study presents a developed methodology for deploying and configuring a web server using Internet Information Services (IIS). This methodology covers the analysis of common security threats and the corresponding protection systems built into IIS. It covers in detail the stages of installation, website configuration, creation of virtual directories, and the use of built-in IIS security features, such as request filtering, IP address access restriction, and various authentication methods.

The experimental portion of the study demonstrates the step-by-step creation of a test environment, including an isolated website, firewall configuration, and functionality testing. The findings demonstrate that, when implementing least-privilege principles, proper network segmentation, and properly configured NTFS permissions, a standard IIS configuration can withstand certain types of attacks, such as SQL injection. The system's security against unauthorized access is also improved.

The study's key findings highlight the need to migrate to HTTPS, regularly audit system logs, and conduct systematic testing and ongoing monitoring. These components are essential for maintaining a secure web server lifecycle.

**Keywords:** Internet Information Services (IIS), Windows Server 2022, web server, functional testing, configuration, firewall, logging.

Grachev Alexander Maksimovich — 3rd year student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: alexgrachyov4836@gmail.com

Zakharov Yuri Igorevich — 3rd year student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: yurchikgame@mail.ru

**Рубрика 2. Методы и системы защиты информации, информационная безопасность**

***УДК*** *00004.7:004.738.5*

***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.

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.

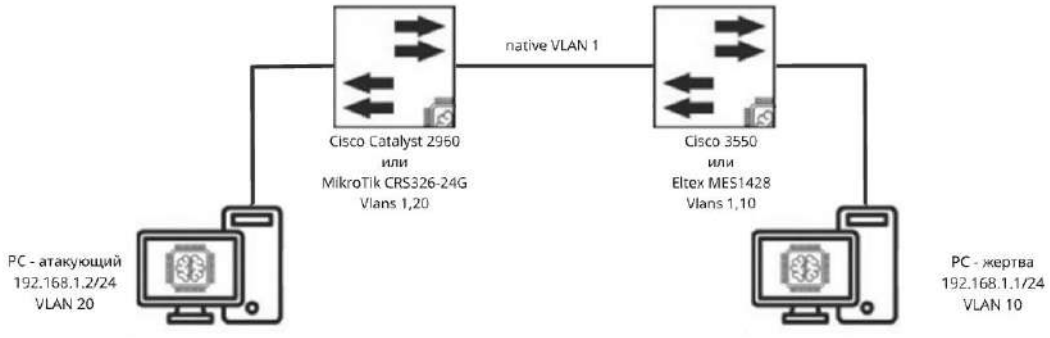


**Рис. 1 Топология экспериментального стенда для проверки атаки Double Tagging между VLAN 20 и VLAN 10**

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].

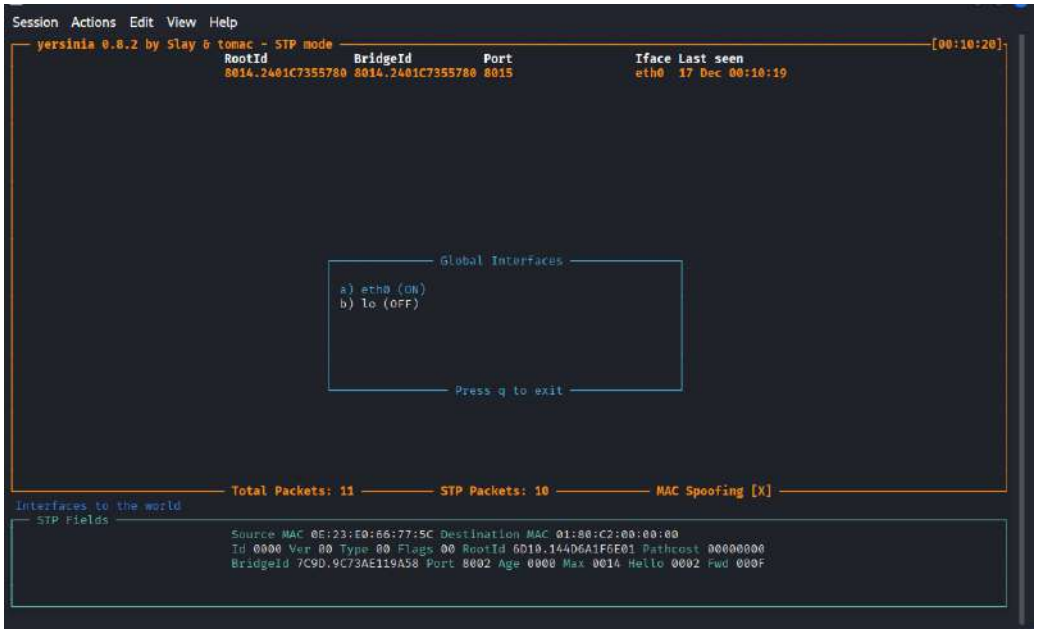


**Рис. 2 Интерфейс утилиты Yersinia, используемой для формирования кадров с двойным тегированием 802.1Q**

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.

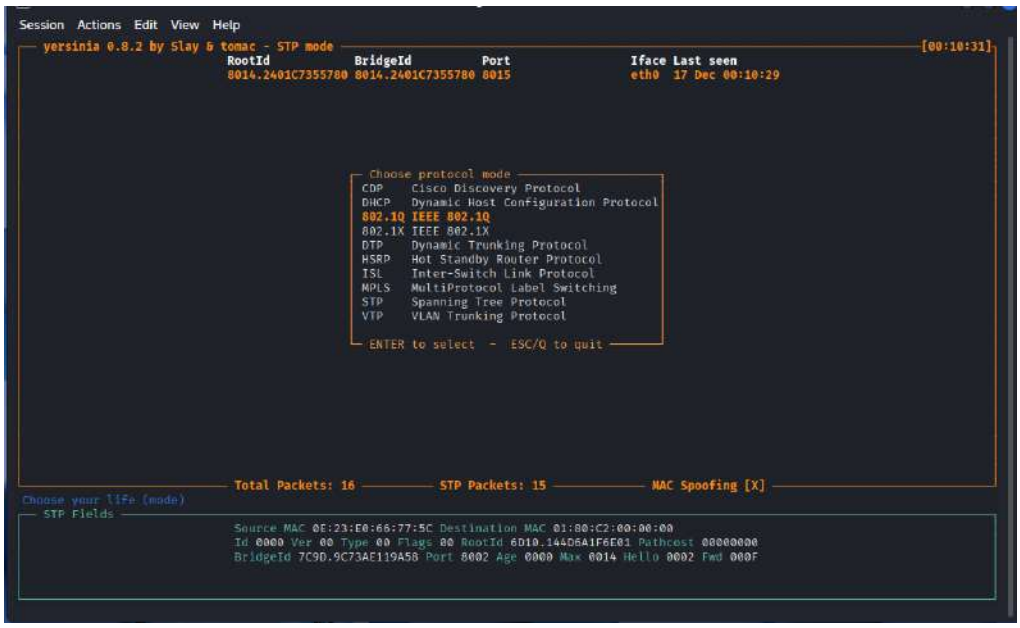


**Рис. 3 Выбор протокола 802.1Q в утилите Yersinia для настройки атаки Double Tagging**

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.

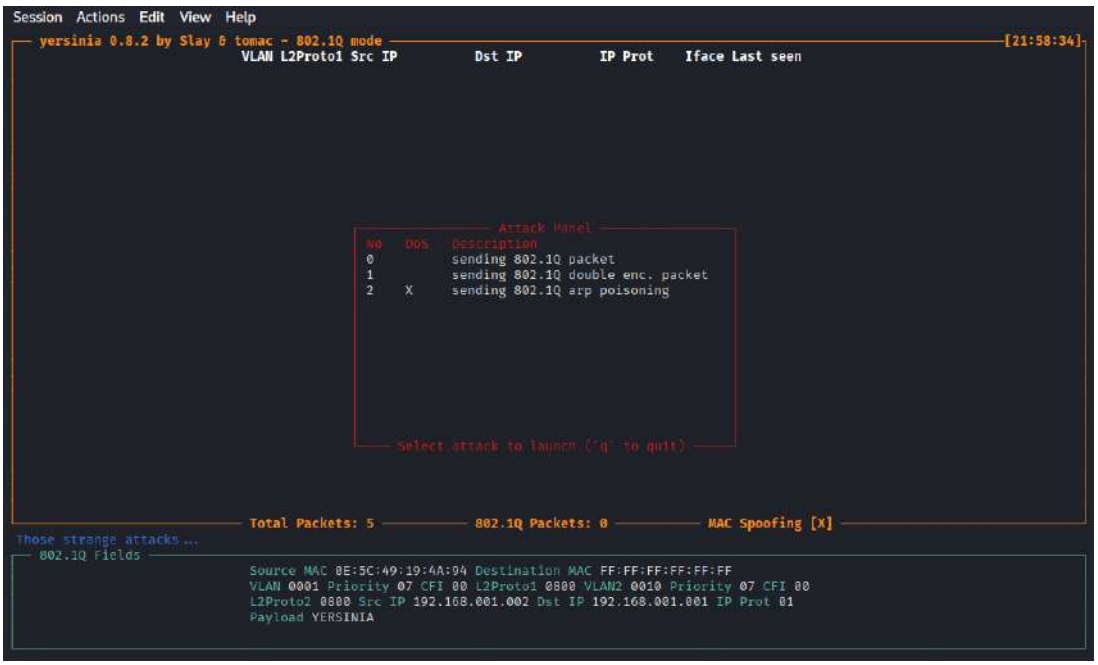


**Рис. 4 Настройка полей 802.1Q в Yersinia с указанием внешнего native VLAN 1 и внутреннего целевого VLAN 10**

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.

**Список литературы**

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

## 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**

Почта России · Москва, Ленинский пр-кт, 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 (4)**

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: 10.01.2024.

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.WEBSITE