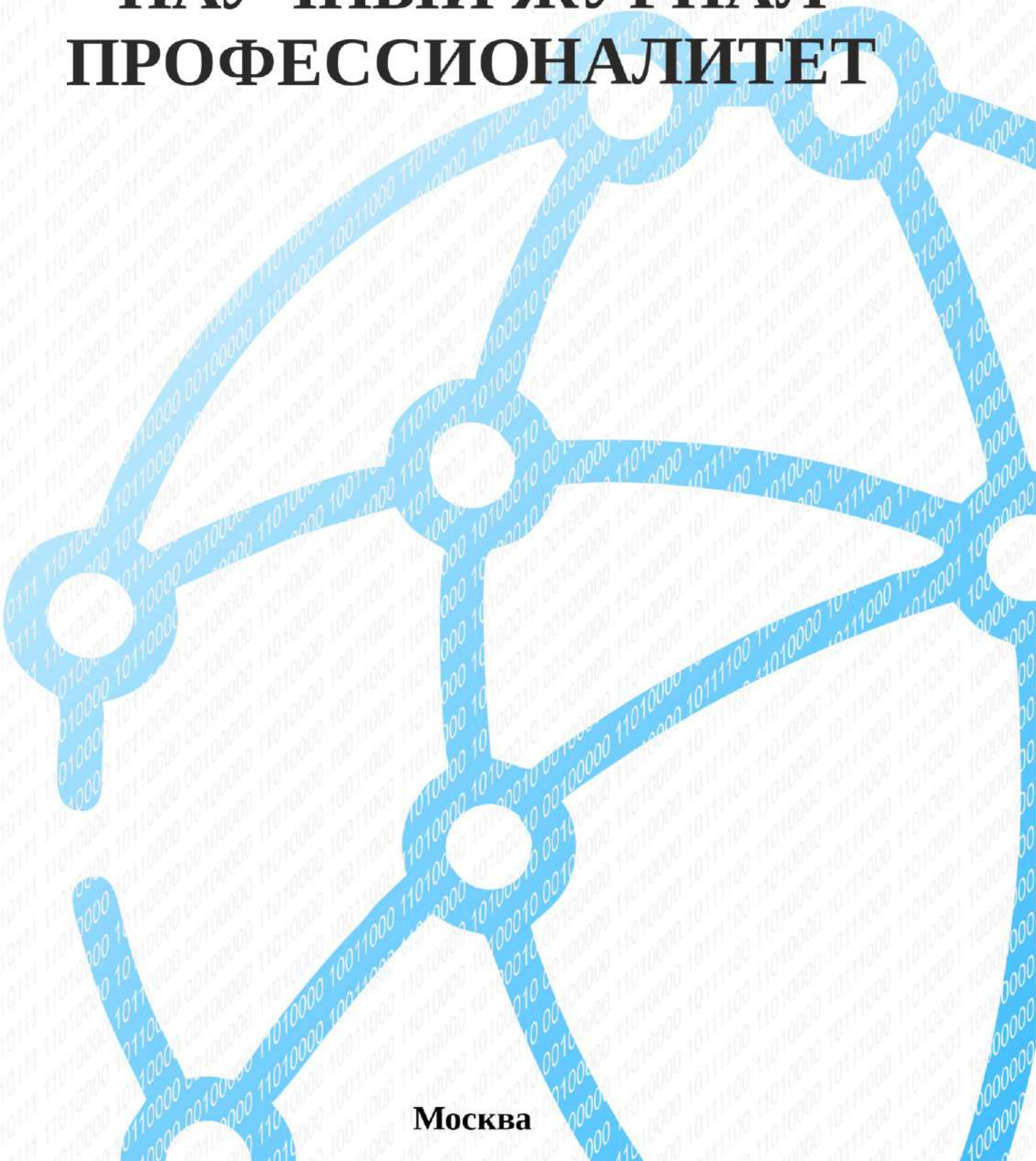
ISSN 3033-9359
Выпуск 4 (008), 2025

НАУЧНЫЙ ЖУРНАЛ ПРОФЕССИОНАЛИТЕТ



Москва

ПРОФЕССИОНАЛИТЕТ

№ 4 (008), 2025

Научный журнал

Основан в 2023 году

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

Номер свидетельства ЭЛ № ФС 77-84640

Дата регистрации 01.02.2023

Учредитель:

Уймин А.Г.

Адрес редакции:

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

ПРОФЕССИОНАЛИТЕТ, 2025, № 4 (008)

PROFESSIONALITET

No.4 (008), 2025

Scientific Journal

Founded in 2023

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

Certificate of Registration: EL No. FS 77-84640

Registration Date: 01.02.2023

Founder:

Uymin A.G.

Editorial Office:

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

СОДЕРЖАНИЕ

PROFESSIONALITET, 2025, No. 4 (008)

CONTENTS

Рубрика 1. Информатика и информационные процессы

***УДК** 004.63*

Д. Д. Новикова[1], С. С. Чурсина[1]

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

АНАЛИЗ СТРУКТУР HFS+ И МЕТОДИКА ВОССТАНОВЛЕНИЯ ДАННЫХ

Аннотация. В статье представлен структурный анализ файловой системы HFS+ и описана методика ручного восстановления данных на основе прямого изучения служебных областей тома. Актуальность работы обусловлена тем, что HFS+ продолжает использоваться на внешних накопителях, резервных копиях и устаревших устройствах Apple, а автоматические средства восстановления не всегда корректно обрабатывают повреждённые метаданные. Рассматриваются назначение Volume Header, параметры Catalog File, устройство B-дерева каталогов, структура листовых узлов, массивы смещений записей, ключи HFSPlusCatalogKey и основные поля записей файлов и папок. На практическом примере показано, как определить размер блока, вычислить физическое смещение каталогового файла, найти записи корневого каталога, установить идентификаторы объектов и извлечь содержимое файла по данным экстентов. Методика ориентирована на работу в hex-редакторе и не требует применения дорогостоящего специализированного программного обеспечения. Особое внимание уделено ограничениям ручного анализа: высокой трудоёмкости, необходимости точного расчёта смещений и риску ошибок при интерпретации записей. Поэтому предложенный подход рассматривается не как массовая замена автоматизированным утилитам, а как экспертный инструмент для проверки результатов и восстановления отдельных важных объектов. Полученные результаты демонстрируют, что ручной низкоуровневый анализ позволяет восстанавливать отдельные критически важные файлы, проверять корректность автоматизированных утилит и использовать HFS+ как учебную модель для освоения принципов организации файловых систем. Предложенный подход полезен специалистам по восстановлению данных, цифровой криминалистике и системному администрированию.

Ключевые слова: HFS+, восстановление данных, структурный анализ, B-дерево, каталоговый файл, экстенты, файловая система Apple.

Введение

HFS+ (Hierarchical File System Plus), также известная под названиями Mac OS Extended и HFS Extended, представляет собой основную журналируемую файловую систему, применяемую в операционных системах семейства Mac OS X с момента выхода Mac 8.1 в 1998 году. В современных версиях macOS данная файловая система стандартно идентифицируется как Mac OS Extended (Journaled) [1]. Хотя в 2017 году Apple перешла на оптимизированную для SSD систему APFS (Apple File System), заменившую HFS+ в macOS High Sierra [2], последняя сохраняет популярность как формат дисков для Mac в силу своей высокой совместимости. Несмотря на переход на APFS, файловая система HFS+ продолжает встречаться на практике — в первую очередь на внешних накопителях, используемых для резервного копирования (например, Time Machine), а также на устаревших устройствах Apple. В задачах цифровой криминалистики и аварийного восстановления данных часто требуется работа с такими legacy-томами, особенно когда повреждены метаданные или автоматические инструменты не могут корректно интерпретировать структуру раздела. В этих случаях глубокое понимание внутреннего устройства HFS+ становится критически важным. Архитектура HFS+ предполагает хранение числовых параметров в формате big endian, что необходимо учитывать при обработке служебных структур. Архитектура HFS+ подробно описана в официальной технической документации Apple (Technical Note TN1150), а её особенности в контексте цифровой криминалистики — в работах Брайана Кэрриэра и других исследователей.

Эта файловая система стала более усовершенствованной по сравнению со своей предшественницей, HFS. Ключевые улучшения включают переход на 32-битную адресацию блоков, поддержку Unicode-имён файлов, работу с томами значительно большего размера, а также внедрение эффективных механизмов хранения данных и ведения журнала. Благодаря новой архитектуре HFS+ обеспечивает работу с томами и файлами размером до 8 эксабайт (2^{63} байт при двоичном понимании единиц) и может содержать до 2,1 млрд файлов в рамках одного тома, обеспечивая необходимую масштабируемость для современных задач.

Основу функционирования HFS+ составляют шесть структур метаданных. Заголовок тома (Volume Header), расположенный в начале раздела, содержит основополагающую

информацию о томе, такую как дата создания, размеры блоков и указатели на другие служебные файлы. Статус блоков распределения (свободен/занят) отслеживается с помощью файла распределения (Allocation File), который является простой битовой картой. Сердцем файловой системы является файл каталога (Catalog File) – большое B-дерево, содержащее записи для всех файлов и папок и образующее иерархическую структуру тома. Именно в этих записях хранятся первые восемь экстентов, описывающих расположение данных файлов. Если файл сильно фрагментирован, дополнительные экстенты сохраняются в файле переполнения экстентов (Extents Overflow File). Дополнительные метаданные могут храниться в файле атрибутов (Attribute File), а загрузочный файл (Startup File) содержит код, необходимый для загрузки системы с устаревших компьютеров. Ниже, на рис. 1 приведена схема тома HFS+ [3].

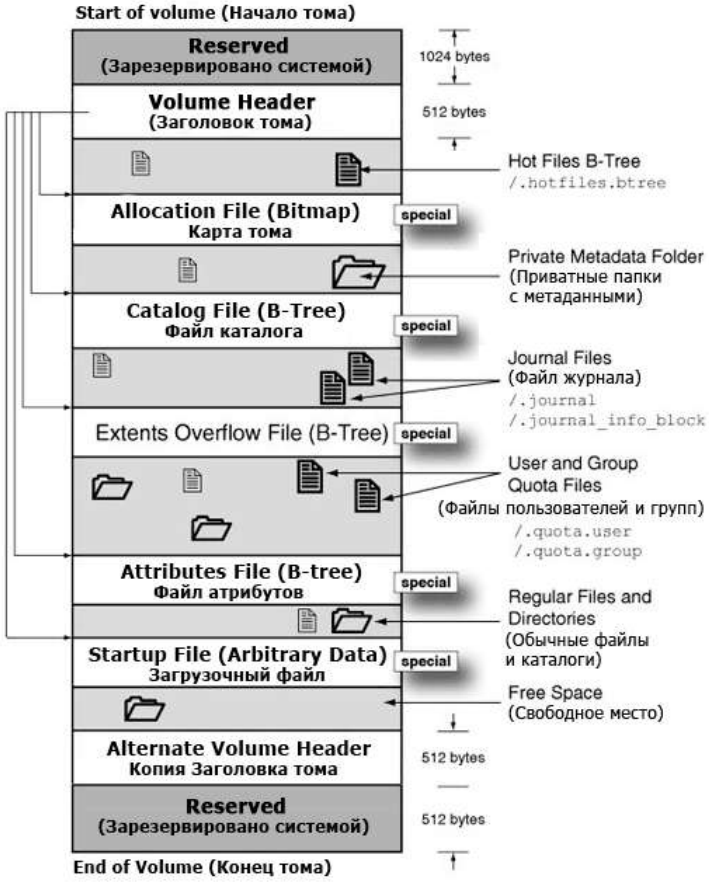


Рис. 1. Структура тома с HFS+

Помимо этих структур, в файловой системе существует множество вспомогательных, но именно перечисленные выше имеют наибольшее значение при анализе и восстановлении данных.

Заголовок Тома (Volume Header)

Volume Header размещается во втором секторе тома (offset – 0x400) и содержит все ключевые параметры файловой системы. Резервная копия заголовка располагается симметрично – во втором секторе от конца тома, обеспечивая восстановление системы при повреждениях. Ниже (табл. 1) приведены наиболее важные параметры Volume Header для анализа и восстановления данных [3].

Таблица 1 – Основные атрибуты Volume Header

Смещение	Размер (в байтах)	Название	Описание
0x000	2	signature	Подтверждение, что это HFS+ (H+ = 0x482B)
0x00C	4	journalInfoBlock	Если ≠ 0 - том журналируемый
0x010	4	createDate	Когда создан том. В секундах с 1 января 1904 года по Гринвичу (GMT). Для московского времени прибавляем 3 часа.

0x020	4	fileCount	Количество файлов на томе
0x024	4	folderCount	Количество папок на томе
0x028	4	blockSize	Размер блока
0x02C	4	totalBlocks	Количество блоков на томе
0x110	80	catalogFile	Данные Catalog B-tree

Поле catalogFile в Volume Header описывает расположение каталогового файла на диске. Это структура HFSPlusForkData размером 80 байт:

1. 0x00 (8 байт) – logicalSize (логический размер)
2. 0x08 (4 байта) – clumpSize (размер кластера)
3. 0x0C (4 байта) – totalBlocks (всего блоков)
4. 0x10 (64 байта) – extents (массив из восьми экстентов)

Массив экстентов в структуре catalogFile состоит из восьми элементов размером 8 байт каждый. Каждый элемент содержит два параметра: номер начального блока startBlock (4 байта) и количество последовательных блоков blockCount (4 байта). Данные Catalog File физически расположены в блоках, описываемых этими экстентами. Для определения физического смещения Catalog File используется формула:

$$catalogFileOffs = startBlock \times blockSize \quad (1)$$

где значение blockSize берётся из заголовка тома (Volume Header).

На рис. 2 продемонстрирован фрагмент тома в среде шестнадцатеричного редактора HxD, на котором обозначены параметры Volume Header из таблицы 1.

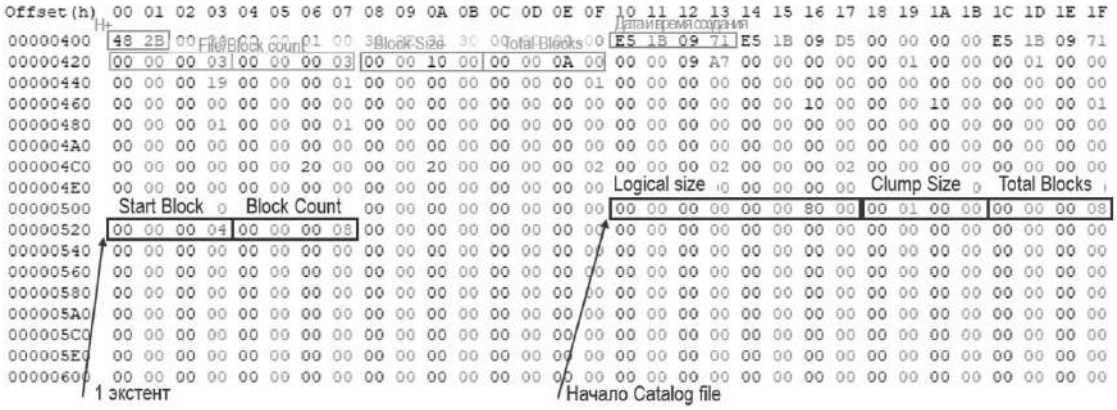


Рис. 2. Разбор основных атрибутов Volume Header

Анализ структуры catalogFile начинается со смещения 0x110 от начала Volume Header. Параметр logicalSize указывает на размер 32 768 байт (0x8000). Первый экстент содержит значения Start block = 4 и Block count = 8, что определяет расположение данных каталогового файла в виде последовательности из восьми блоков, начиная с блока №4. Физическое смещение рассчитывается по формуле:

$$catalogFileOffs = 4\ (startBlock) \times 4096\ (blockSize) = 16384\ байт = 0x4000$$

Заголовок B-дерева (B-Tree Header Node)

B-Tree Header Node – это служебная структура, расположенная в начале файла B-дерева (Catalog, Extents Overflow или Attributes File). Он содержит метаинформацию о всей структуре B-дерева, необходимую для навигации по ней. В таблице 2 выборочно представлены параметры заголовка узла B-дерева.

Таблица 2 – Атрибуты B-Tree Header Node

Смещение	Размер (в байтах)	Название	Описание
0x08	1	kind	Тип узла: –1 = leaf, 0 = index, 1 = header, 2 = map
0x0A	2	numRecords	Количество записей в узле

0x10	4	rootNode	Номер корневого узла
0x14	4	leafRecords	Количество записей в листовых узлах
0x18	4	firstLeafNode	Первый листовой узел
0x20	2	nodeSize	Размер узла

Формула для расчёта смещения листового узла:

$$leafNodeOffs = catalogFileOffs + firstLeafNode \times nodeSize \quad (2)$$

На рис. 3 визуализированы параметры заголовка B-дерева файла, найденного на предыдущем шаге. Видно, что rootNode и firstLeafNode = 1, nodeSize = 0x1000 (4096 байт).

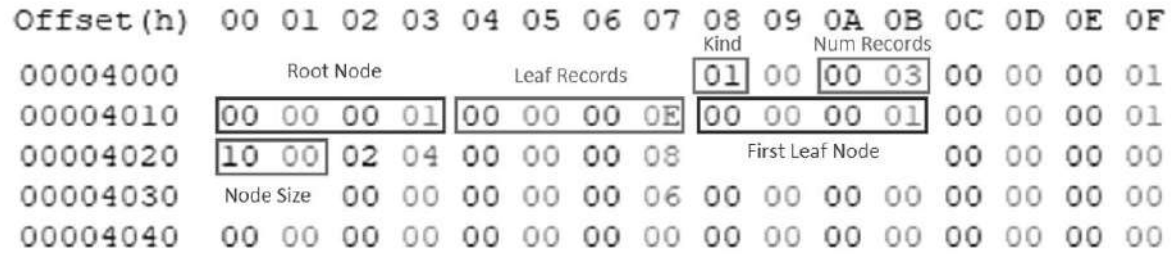


Рис. 3. Разбор основных атрибутов B-Tree Header Node

Это означает, что корневой узел (rootNode) одновременно является и первым листовым узлом (firstLeafNode). В B-деревьях это возможно, когда дерево очень маленькое (например, содержит всего несколько файлов и папок) и вся информация помещается в один-единственный узел. В данном случае структура B-дерева не требует навигации через индексные узлы – все элементы метаданных содержатся в узле №1. Смещение листового узла определяется по формуле:

$$leafNodeOffs = 0x4000 + (1 \times 0x1000) = 0x5000$$

Листовый узел (Leaf Node)

При анализе листового узла (рис. 4) проверяется значение типа узла (kind) по смещению 0x08, которое должно составлять 0xFF (Leaf Node). Затем по смещению 0x0A извлекается значение numRecords, указывающее количество записей в узле – в данном случае 0xE [4].

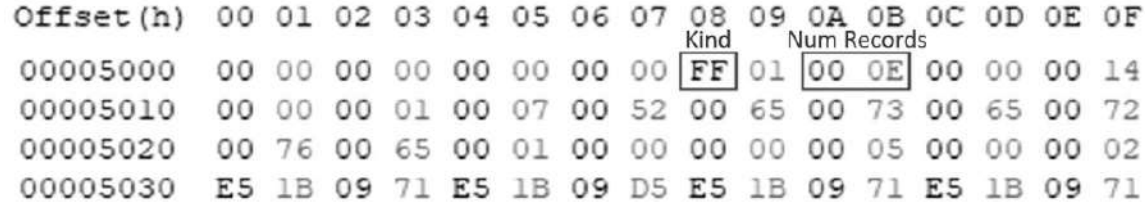


Рис. 4. Основные атрибуты в листовом узле для поиска смещений

В конце узла есть массив смещений на каждую запись. Каждое смещение – число размером 2 байта (само смещение во втором байте пары), которое показывает, насколько нужно отступить от начала узла, чтобы найти запись. Смещения идут в обратном порядке, то есть последние 2 байта узла соответствуют смещению первой записи, предпоследние 2 байта — смещению второй записи и т. д. Первая запись всегда начинается со смещения 14, поэтому последние 2 байта узла обычно содержат число 0x0E.

Для определения массива смещений записей используется следующее вычисление: при количестве записей N = 14 (0xE) и размере узла 0x1000 байт, расчёт конечного смещения массива выполняется по формуле:

$$recOffs = leafNodeOffs + nodeSize - N \times 2 \quad (3)$$

Расчёт:

$$recOffs = 0x5000 + 0x1000 - (0xE \times 2) = 0x5FE4$$

По полученному адресу (рис. 5) располагается массив двухбайтовых смещений, указывающих начала записей в узле.

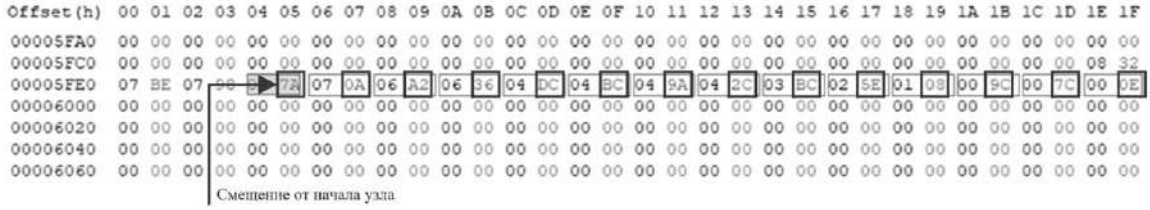


Рис. 5. Все записи в массиве смещений

Алгоритм обработки записей листового узла предполагает последовательное чтение значений из массива смещений recOffs. Каждый элемент recOffs[N] содержит относительное смещение N-ой записи от начала узла. Для доступа к N-ой записи используется формула:

$$startRecOffs = leafNodeOffs + recOffs[N] \quad (4)$$

HFSPlusCatalogKey

Ключ каталога используется для идентификации и поиска записей в B-дереве каталогов (табл. 3):

Таблица 3 – Атрибуты HFSPlusCatalogKey

Смещение	Размер (в байтах)	Название	Описание
0x00	2	keyLength	Длина данных ключа (parentID + nodeName)
0x02	4	parentID	Идентификатор родительской папки (CNID)
0x06	Переменный	nodeName	Имя файла/папки в Unicode

Структура HFSUniStr255, которая поясняет поле nodeName (табл. 4):

Таблица 4 – Атрибуты HFSUniStr255

Смещение	Размер (в байтах)	Название	Описание
0x00	2	length	Количество символов в имени (N)
0x02	N × 2	unicode	Символы UTF-16BE

Файловая система HFS+ использует систему численных идентификаторов (CNID) для служебных и пользовательских объектов. Зарезервированные диапазоны включают идентификаторы с 1 по 8 для важных компонентов (например, CNID 2 для корневой папки, CNID 4 для каталогового файла. Тогда как значения от 16 и выше закрепляются за пользовательскими файлами и каталогами. Данное разделение позволяет однозначно определять тип объекта в процессе восстановления данных.

Для восстановления структуры каталогов необходимо найти все записи, содержащие в поле parentID значение 0x00000002, что соответствует корневому каталогу тома (рис. 6).

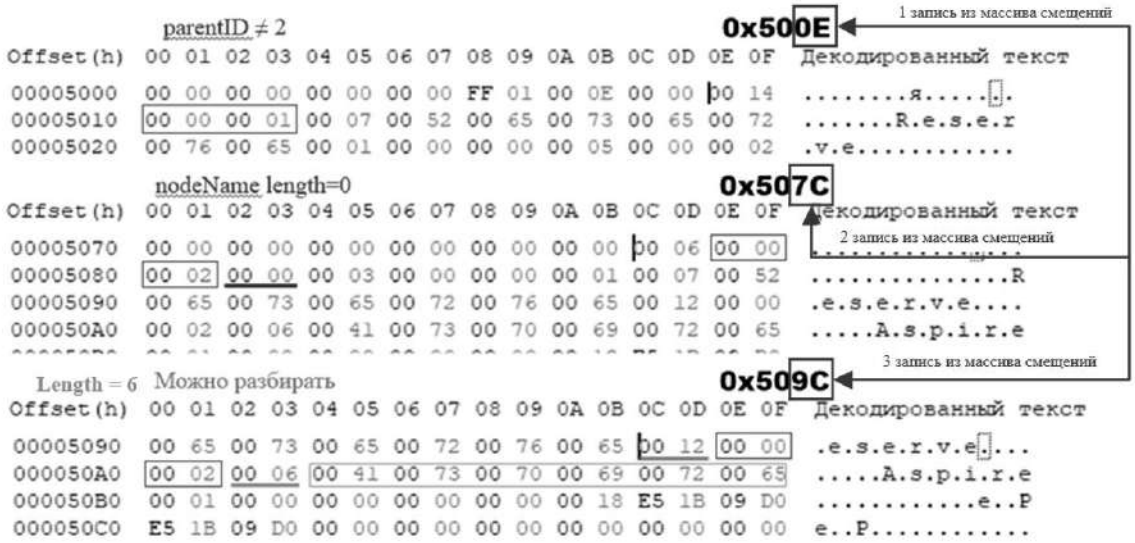


Рис. 6. Разбор записей в листовом узле

Первая запись в узле располагается по смещению 0xE, следовательно, адрес начала записи равен 0x5000 + 0xE = 0x500E.

Поле parentID имеет значение 0x00000001, что соответствует идентификатору родителя корневой папки. Таким образом, данная запись описывает корневую папку тома, имя которой является меткой тома. На данном этапе нашли метку тома (nodeName) – Reserve.

Примечание. Значения не противоречат друг другу: parentID = 0x00000001 относится к записи самого корневого каталога, тогда как parentID = 0x00000002 используется у дочерних записей, непосредственно размещённых в корневом каталоге.

Поле parentID (CNID) этой записи равно 0x00000002, что указывает на корневой каталог (kHFSRootFolderID), структура которого далее будет восстанавливаться [3].

Переходим ко второй записи 0x5000 + 7C = 0x507C. ParentID = 2, но длина имени nodeName = 0. Пустая запись пропускается. Адрес 3 записи: 0x5000 + 9C = 0x509C. Длина имени ненулевая, parentID = 0x2 (корень) – разбираем.

keyLength – 00 12 (18_{10})

parentID – 00 00 00 02

nodeName:

length – 00 06

Само имя 6 * 2 = 12 байт – 00 41 00 73 00 70 00 69 00 72 00 65 (Aspire)

Catalog File/Folder Records

После ключа (HFSPlusCatalogKey) в листовом узле располагается тело записи – область, содержащая информацию о файле или папке [5].

Тип записи определяется полем recordType (2 байта) и описывается одной из структур: HFSPlusCatalogFile – для файлов; HFSPlusCatalogFolder – для папок. Эти структуры содержат основные атрибуты объектов файловой системы: уникальный идентификатор (CNID), даты создания и изменения, права доступа, а также указатели на данные (экстенты для файлов).

Типы записей (recordType):

1. 0x0001 – HFSPlusFolderRecord (запись о папке)
2. 0x0002 – HFSPlusFileRecord (запись о файле)
3. 0x0003 – HFSPlusFolderThread («поток» папки, то есть служебная ссылка на родителя)
4. 0x0004 – HFSPlusFileThread («поток» файла, то есть служебная ссылка на родителя)

В таблице 5 смещения представлены относительно начала тела записи.

Таблица 5 – Структура записи папки (HFSPlusCatalogFolder)

Смещение	Размер (в байтах)	Поле	Описание
0x0000	2	recordType	Тип записи (0x0001)
0x0004	4	valence	Количество элементов в папке
0x0008	4	folderID	CNID папки
0x000C	4	createDate	Дата создания

Поле folderID используется для построения иерархии – на него ссылаются записи, где parentID = folderID. Поле valence позволяет сверить, сколько элементов должно содержаться в папке (рис.7).

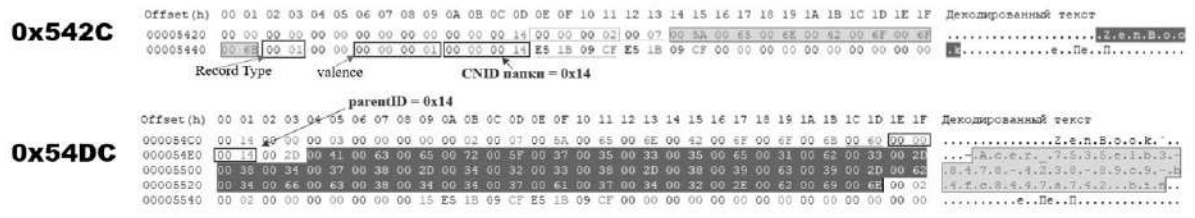


Рис. 7. Разбор записи папки

По адресу 0x5000 + 0x042C = 0x542C видим, что у нас папка (recordType = 1), смотрим ее CNID = 0x14. Теперь мы перебираем оставшиеся смещения и записи. Наша задача – найти значение parentID = 0x14. Нашли одну запись по смещению 0x54DC, которая является файлом. Аналогичным образом ищется содержимое остальных каталогов.

Восстановление файлов

Для извлечения содержимого файлов требуется анализ соответствующих записей в каталоговом B-дереве. Структура HFSPlusCatalogFile содержит всю необходимую метаинформацию (табл. 6), включая указатель на данные файла.

Таблица 6 – Структура записи файла (HFSPlusCatalogFile)

Смещение	Размер (в байтах)	Поле	Описание
0x0000	2	recordType	Тип записи (0x0002)
0x0008	4	fileID	CNID файла
0x000C	4	createDate	Дата создания
0x0058	80	dataFork	Данные data fork (содержимое файла) logicalSize (8 байт) содержит фактический размер файла в байтах

Поле fileID используется для идентификации записи, а dataFork – для поиска реальных данных файла на диске. Экстенты (HFSPlusExtentDescriptor) в dataFork указывают начальный блок и длину данных (рис.8).

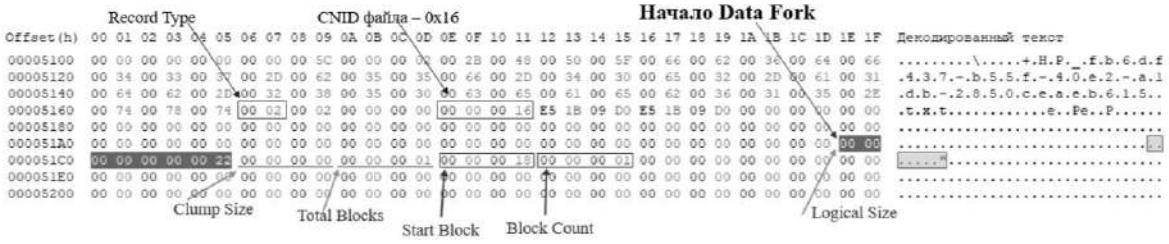


Рис. 8. Разбор записи файла

В данном случае по смещению 0x5000 + 0x0108 = 0x5108 находится HP_fb6df437-b55f-40e2-a1db-2850ceaeb615.txt. Так как recordType = 2, мы понимаем, что это – файл. Продолжаем разбор атрибутов и находим его startBlock = 0x18 и blockCount = 1. Содержимое файла находится по адресу 0x1000 (blockSize) * 0x18 = 0x18000. Значение logicalSize = 0x22 (34 байта) указывает, что реальная длина содержимого файла – 34 байта несмотря на то, что ему выделен один блок размером 4096 байт (blockCount = 1).

Файловая система HFS+ не обнуляет содержимое блоков при записи, особенно в случаях, когда файл был удалён или перезаписан. Поэтому только первые 0x22 байта блока содержат актуальные данные, соответствующие значению logicalSize (34 байта), а оставшееся пространство может включать остаточные данные от ранее записанных файлов, не имеющие отношения к текущему содержимому.

Переходим к рассчитанному смещению 0x18000, копируем 0x22 байта информации и сохраняем в новом файле с соответствующим расширением .txt (рис.9).

Рис. 9. Восстановление содержимого файла

Проведённое исследование показало, что структурный анализ файловой системы HFS+ может быть эффективно применён для решения практических задач по восстановлению данных. Предложенная методика основана на пошаговом изучении ключевых метаданных – от Volume Header до листовых узлов B-дерева каталогов – и позволяет извлекать пользовательские файлы даже без использования специализированного программного обеспечения. Важно подчеркнуть, что предложенная методика не предназначена для массового восстановления больших объёмов данных. Ручной анализ в hex-редакторе — трудоёмкий и подверженный ошибкам процесс. Однако он оказывается незаменимым в ряде узких сценариев: при восстановлении отдельных критически важных файлов, при обучении принципам низкоуровневого анализа файловых систем, а также для верификации результатов, полученных автоматизированными утилитами. Таким образом, методика представляет собой скорее инструмент эксперта, чем повседневное решение для системного администратора.

Существуют открытые инструменты, такие как TestDisk, The Sleuth Kit и утилита hfsinspect, которые автоматизируют анализ и восстановление HFS+-томов [6, 7]. Однако при серьёзных повреждениях — например, полной потере заголовка тома, некорректной таблицы разделов или сбое журналирования — автоматика часто «сдаётся», не находя даже базовых метаданных. В таких ситуациях ручной анализ, основанный на прямом чтении структур в hex-редакторе, позволяет восстановить данные там, где программные решения оказываются бессильны.

Несмотря на переход Apple на файловую систему APFS, HFS+ остаётся актуальной благодаря широкому распространению носителей, использующих данную структуру. Разработанный алгоритм представляет интерес для специалистов в области компьютерной криминалистики и восстановления данных, поскольку помогает глубже понять принципы организации HFS+ и может послужить основой для разработки автоматизированных инструментов анализа. В дальнейшем планируется расширить методику для работы со сложными случаями восстановления, включая обработку фрагментированных файлов и нестандартных структур каталогов.

Библиографический список

1. HFS+: Structures and Features, exFAT vs. HFS+ vs. NTFS [Электронный ресурс] // iBoysoft. – 2025. – URL: https://iboysoft.com/wiki/hfs-plus.html (дата обращения: 05.11.2025).

Рубрика 1. Информатика и информационные процессы

2. Apple File System Reference [Электронный ресурс] // Apple Developer Documentation. – Apple Inc., 2017. – URL: https://developer.apple.com/support/downloads/Apple-File-System-Reference.pdf (дата обращения: 15.10.2025).
3. Apple Inc. Technical Note TN1150: HFS Plus Volume Format [Электронный ресурс] // Apple Developer Documentation. – 2004. – URL: https://developer.apple.com/library/archive/technotes/tn/tn1150.html (дата обращения: 15.10.2025).
4. Порядок восстановления данных в файловой системе HFS+ [Электронный ресурс] // Хабр. – 2021. – URL: https://habr.com/ru/companies/hetmansoftware/articles/547716/ (дата обращения: 06.11.2025).
5. Carrier, B. File System Forensic Analysis / B. Carrier. – Boston : Addison-Wesley Professional, 2005. – 640 p.
6. The Sleuth Kit [Электронный ресурс] // The Sleuth Kit. – URL: https://www.sleuthkit.org/sleuthkit/ (дата обращения: 06.11.2025).
7. Grenier, C. TestDisk Documentation [Электронный ресурс] / C. Grenier // CGSecurity. – URL: https://www.cgsecurity.org/wiki/TestDisk (дата обращения: 06.11.2025).

References

1. iBoysoft. (2025). HFS+: Structures and features, exFAT vs. HFS+ vs. NTFS. iBoysoft. https://iboysoft.com/wiki/hfs-plus.html
2. Apple Inc. (2017). Apple File System Reference. Apple Developer Documentation. https://developer.apple.com/support/downloads/Apple-File-System-Reference.pdf
3. Apple Inc. (2004). Technical Note TN1150: HFS Plus Volume Format. Apple Developer Documentation. https://developer.apple.com/library/archive/technotes/tn/tn1150.html
4. Hetman Software. (2021). Procedure of data recovery in HFS+ file system. Habr. https://habr.com/ru/companies/hetmansoftware/articles/547716/
5. Carrier, B. (2005). File system forensic analysis. Addison-Wesley Professional.
6. The Sleuth Kit. (n.d.). The Sleuth Kit. https://www.sleuthkit.org/sleuthkit/
7. Grenier, C. (n.d.). TestDisk documentation. CGSecurity. https://www.cgsecurity.org/wiki/TestDisk

Информация об авторах

Чурсина Серафима Сергеевна – студент кафедры «Математических методов обеспечения безопасности систем», ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: chursinass@mail.ru.

Новикова Дарья Дмитриевна – студент кафедры «Математических методов обеспечения безопасности систем», ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: novikova.d@mail.ru.

ANALYSIS OF HFS+ STRUCTURES AND DATA RECOVERY TECHNIQUE

D. D. Novikova1, S. S. Chursina1
1National University of Oil and Gas «Gubkin University»

Abstract. This article presents a structural analysis of the HFS+ file system and describes a manual data recovery technique based on direct examination of service areas of a volume. The relevance of the study is determined by the fact that HFS+ is still encountered on external drives, backup media and legacy Apple devices, while automated recovery tools do not always correctly interpret damaged metadata. The paper examines the role of the Volume Header, Catalog File parameters, the organization of catalog B-trees, leaf node layout, record offset arrays, HFSPlusCatalogKey fields and the main structures of file and folder records. A practical example demonstrates how to determine the allocation block size, calculate the physical offset of the catalog file, identify root directory records, establish object identifiers and extract file contents using extent descriptors. The proposed method is intended for work in a hex editor and does not require expensive specialized software. The results show that manual low-level analysis can be used to recover individual critical files, verify automated tool output and study the internal principles of file system organization. The technique is useful for data recovery specialists, digital forensics practitioners and system administrators working with Apple storage media.

Keywords: HFS+, data recovery, structural analysis, B-tree, catalog file, extents, Apple file system.

Information about the authors

Chursina Serafima Sergeevna – student of the Department of Mathematical Methods of System Safety Assurance, Gubkin Russian State University of Oil and Gas (National University), Moscow, e-mail: chursinass@mail.ru.

Novikova Daria Dmitrievna – student of the Department of Mathematical Methods of System Safety Assurance, Gubkin Russian State University of Oil and Gas (National University), Moscow, e-mail: novikova.d@mail.ru.

Рубрика 1. Информатика и информационные процессы

УДК *004.77*

К. С. Тюхтин [1]

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

АРХИТЕКТУРА NETFILTER. ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ В ОС «АЛЬТ»

Аннотация. Статья посвящена практическому исследованию реализации подсистемы netfilter в российской операционной системе «Альт». Актуальность работы обусловлена необходимостью верификации корректности и полноты базовых сетевых механизмов безопасности в отечественном программном обеспечении. Решается задача комплексного функционального тестирования архитектуры netfilter через её современный интерфейс — фреймворк nftables.

На изолированном лабораторном стенде, моделирующем корпоративный сетевой сегмент, проведена серия экспериментов. Тестирование охватывает ключевые функции: пакетную фильтрацию (ICMP, TCP, UDP), трансляцию сетевых адресов (NAT), манипуляцию заголовками (mangle), а также управление трафиком с использованием механизмов ограничения скорости (limit) и бессостоятельной (stateless) фильтрации. Отдельное внимание уделено сравнению поведенческих аспектов действий REJECT и DROP при блокировке доступа.

Результаты подтвердили корректную и предсказуемую работу протестированных механизмов в указанной версии ОС «Альт», конфигурации стенда и наборе сценариев. Нагрузочное тестирование позволило наблюдать влияние различных правил фильтрации на загрузку центрального процессора маршрутизатора. Практическим результатом является подтверждение работоспособности реализации netfilter в ОС «Альт» в пределах проведённых функциональных и нагрузочных проверок; вывод о применимости в критической инфраструктуре требует отдельной оценки соответствия. Полученные данные и методика тестирования могут быть использованы для сертификации и аудита безопасности отечественных дистрибутивов Linux.

Выводы работы свидетельствуют о том, что проверенные функции netfilter/nftables в ОС «Альт» ведут себя сопоставимо с ожидаемой моделью Linux в рамках исследованной версии и конфигурации. Представленные результаты демонстрируют готовность данной платформы к использованию в качестве основы для построения надёжных и эффективных корпоративных межсетевых экранов.

Ключевые слова: Netfilter, nftables, ОС Альт, функциональное тестирование, межсетевой экран, фильтрация пакетов, сетевая безопасность.

Введение

Стабильность и безопасность сетевой инфраструктуры напрямую зависят от корректности работы базовых механизмов операционной системы. В Linux таким фундаментальным механизмом является подсистема netfilter, реализующая функционал межсетевого экрана, трансляции адресов (NAT) и управления трафиком [1]. Её современная реализация — фреймворк nftables — стала стандартом в новых дистрибутивах, включая российскую операционную систему «Альт» [2].

С архитектурной точки зрения netfilter представляет собой набор точек перехвата (hook) в стеке IPv4/IPv6, через которые проходит каждый пакет при приёме, маршрутизации и отправке: PREROUTING, INPUT, FORWARD, OUTPUT и POSTROUTING. На этих хуках ядро вызывает зарегистрированные обработчики, к которым относится фреймворк nftables: он организует правила в таблицы и цепочки, размещённые на соответствующих хуках, и по результатам сопоставления пакета с правилами принимает решения о его пропуске, блокировке, модификации или трансляции адресов.

В дистрибутиве «Альт» nftables используется как единый интерфейс ко всем возможностям netfilter: фильтрации транзитного трафика в цепочке forward, реализации NAT в таблице nat и модификации заголовков в таблице mangle. Настраивая базовые цепочки семейства inet на нужных хуках (прежде всего forward и postrouting), можно гибко управлять прохождением ICMP-, TCP- и UDP-пакетов через маршрутизатор и анализировать влияние политики фильтрации на загрузку системы. Далее приведена структурная схема архитектуры.

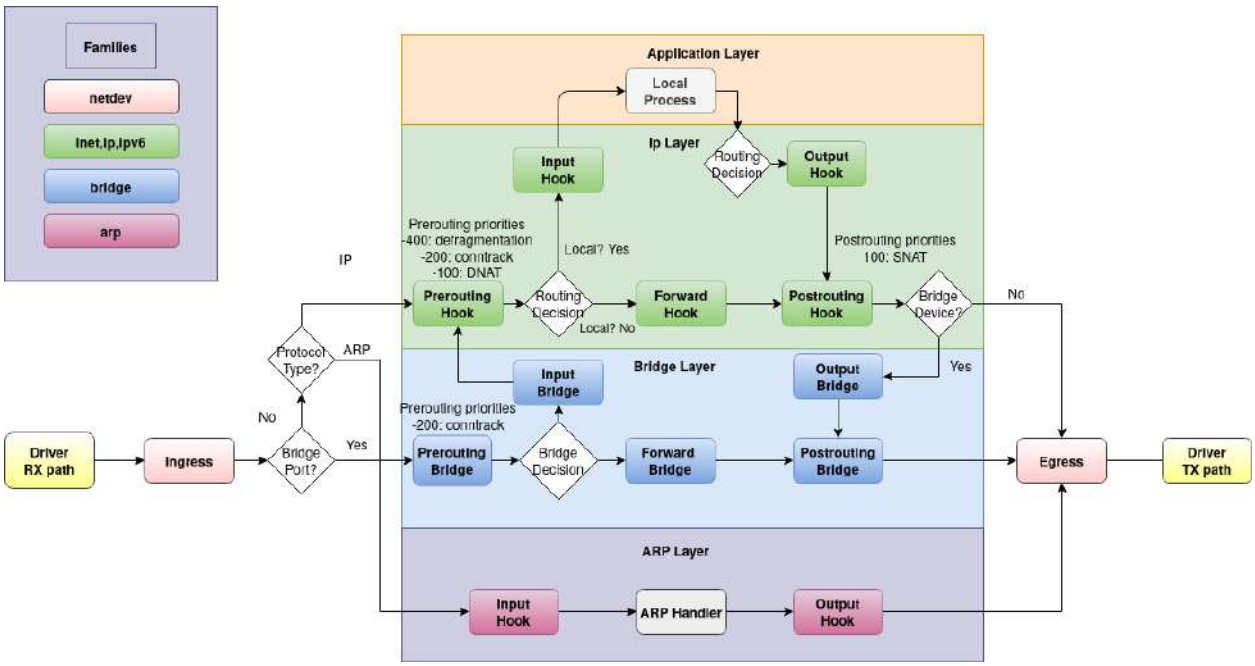


Рис.1. Структурная схема прохождения пакета через хуки подсистемы netfilter [1]

На структурной схеме (рис. 1) показан путь пакета от приёма драйвером сетевого устройства (ingress) до передачи на выход (egress) через уровни ARP, IP и мостового взаимодействия ядра Linux, а также соответствующие им хуки подсистемы netfilter. Пакеты по типу протокола (ARP или IP) и наличию мостового интерфейса попадают в соответствующие цепочки PREROUTING, INPUT, FORWARD, OUTPUT и POSTROUTING как на IP-уровне, так и на уровне bridge, где последовательно могут обрабатываться таблицами raw, mangle, nat и модулями отслеживания соединений (conntrack). Эти цепочки, настраиваемые через nftables для семейств inet, bridge и arp, регистрируются на соответствующих хуках и определяют, будет ли пакет доставлен локальному процессу, переадресован через мост или маршрутизирован далее, а также подлежит ли он модификации, трансляции адресов либо отбрасыванию.

Введение: объект, предмет и цель

Объект исследования — сетевая подсистема операционной системы «Альт», обеспечивающая фильтрацию и управление сетевым трафиком.

Предмет исследования — архитектура и функциональные возможности подсистемы netfilter, реализуемые через фреймворк nftables в ОС «Альт».

Цель исследования — провести комплексное функциональное тестирование архитектуры netfilter на основе nftables в ОС «Альт» для оценки её корректности и возможностей в типовых сценариях.

Задачи исследования:

1. Развернуть тестовый стенд с клиентом, сервером и маршрутизатором под управлением ОС «Альт» для моделирования реального сетевого сегмента.

2. Настроить набор правил nftables, реализующий основные функции netfilter: фильтрацию ICMP/HTTP/SSH-трафика, манипуляцию пакетами (mangle), stateless-фильтрацию, ограничение трафика (limit, iperf3) и действия REJECT/DROP.

3. Экспериментально проверить корректность этих функций с использованием утилит ping, iperf3 и ssh и представить ключевые результаты в наглядной форме.

4. Проанализировать результаты и сформулировать выводы о готовности и специфике реализации архитектуры netfilter в ОС «Альт» для практических задач сетевой безопасности.

Обзор источников

Подсистема netfilter и фреймворк nftables подробно описаны в официальной документации проекта Netfilter, где приводятся архитектура хуков ядра, таблицы raw, mangle, nat, filter и их роль в обработке пакетов в IPv4/IPv6 и мостовых конфигурациях. Данные

материалы служат основой для практических руководств по построению брандмауэров на базе Linux и определяют рекомендуемые подходы к организации правил фильтрации и трансляции адресов [3].

Вопросы производительности и моделирования работы межсетевых экранов рассматриваются в ряде научных работ. В статье Salah K., El-Badawi K., El-Badawi A. предложена аналитическая модель файрвола на основе теории массового обслуживания, позволяющая оценивать задержки, длину очереди и пропускную способность в зависимости от интенсивности трафика и числа правил. Аналогичные подходы используются и в более поздних исследованиях, посвящённых оценке метрик производительности при ранжировании наборов правил и оптимизации их структуры [4].

Практическая производительность подсистемы netfilter была детально исследована Kadlecsik J., который измерял пропускную способность Linux-файрвола при чистой маршрутизации, отслеживании соединений, фильтрации и NAT, а также влияние числа правил iptables/ipset/nf-hipac на достижимый трафик. Дополнительные работы фокусируются на сравнении netfilter с альтернативными механизмами фильтрации (например, eBPF/XDP) и оценке их эффективности под высоконагруженным трафиком [5].

Сравнение существующих исследований показывает, что большинство работ сосредоточено либо на теоретическом моделировании производительности обобщённых файрволов, либо на экспериментальной оценке netfilter в контексте типовых серверных дистрибутивов. В отличие от них, данная статья ориентирована на практическое функциональное тестирование реализации netfilter/nftables именно в российской ОС «Альт» с акцентом на проверку корректности базовых механизмов фильтрации, управления нагрузкой и модификации трафика в условиях лабораторного стенда.

Методы тестирования

В рамках проведения данного исследования для достижения поставленной цели были применены следующие методы:

Теоретико-аналитический метод. Была изучена документация и научная литература по архитектуре netfilter и фреймворку nftables, а также по особенностям сетевого стека ОС «Альт». Это позволило составить план функционального тестирования и разработать соответствующие правила фильтрации [6].

Метод экспериментального тестирования. Была развернута изолированная тестовая сеть, в которой последовательно настраивались и проверялись различные функции netfilter: фильтрация трафика (HTTP, ICMP, SSH), управление соединениями (stateful), ограничение скорости (limit) и другие. Для генерации контролируемого сетевого трафика и проверки корректности работы использовались утилиты iperf3, curl и ssh.

Методика проведения тестирования

Методика функционального тестирования строилась на сериях экспериментов с фиксированными сценариями генерации трафика и заранее заданными наборами правил nftables. Для ICMP-трафика применялись одиночные запросы и режим высокочастотного ping-флуда, для SSH и HTTP — интерактивные сессии и нагрузка ApacheBench, для UDP — устойчивый легитимный поток iperf3 в сочетании с дополнительно генерируемым «атакующим» трафиком. Во всех сценариях оценивались факт установления сетевого доступа, доля успешно доставленных и отброшенных пакетов, а также задержки и наблюдаемая пропускная способность соответствующих соединений [7].

Для анализа влияния правил netfilter на производительность регистрировалась динамика загрузки центрального процессора маршрутизатора: каждую серию начинали с базового прогона без фильтрации, после чего последовательно добавлялись правила, реализующие пропуск легитимного трафика и отбрасывание вредоносного, а также сценарии ограничения частоты ICMP-флуда (limit), модификации полей заголовка (mangle) и использования действий REJECT/DROP для блокировки доступа к сервисам. Это позволило сравнивать изменение системной нагрузки при неизменных параметрах генерируемого трафика и фиксированной структуре легитимных потоков.

Ожидаемое поведение формулировалось для каждой группы сценариев отдельно: для базовой фильтрации — гарантированный проход разрешённого ICMP, SSH, HTTP и UDP-трафика при полной блокировке запрещённых потоков; для режимов DROP/REJECT — различия во времени реакции клиентских приложений при сопоставимом уровне изоляции сервиса; для механизма limit — ограничение числа обрабатываемых ICMP-запросов до заданного порога на фоне флуда и снижение нагрузки по сравнению с неограниченным режимом; для mangle — предсказуемое изменение выбранных полей IP-заголовка (например, TTL) без влияния на достижимость узлов. Критерием успешности считалось соответствие наблюдаемого поведения этим ожиданиям и отсутствие заметной деградации производительности для типовых наборов фильтрующих правил и легитимных сетевых сценариев.

Результаты исследования

Настроили устройства так, чтобы получилась топология, соответствующая рисунку 2.

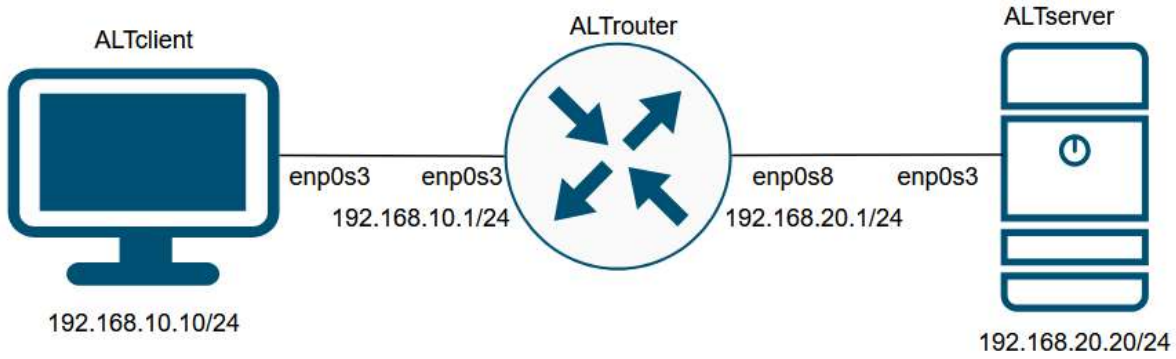


Рис. 2. Топология

Далее приступили непосредственно к проведению исследования. Во всех экспериментах использовалась единая базовая конфигурация: создавалась таблица inet filter и цепочка forward с политикой запрета по умолчанию, через которую проходил транзитный трафик между клиентом и сервером. Для обеспечения работы легитимных ICMP, SSH, HTTP и UDP-соединений последовательно добавлялись минимально необходимые разрешающие правила, после чего поверх них включались сценарии фильтрации, ограничения скорости и модификации пакетов. В таблице 1 представлено краткое обобщение экспериментальных сценариев, в которых использовались запрещающие и ограничивающие правила фильтрации трафика в подсистеме netfilter. Каждый сценарий описывает тип трафика, ключевые параметры правил nftables и целевую задачу эксперимента, что позволяет далее сопоставлять полученные графики загрузки процессора и наблюдаемое сетевое поведение маршрутизатора.

Таблица 1 — Сценарии проведения экспериментов

№	Экспериментальный сценарий	Запрещающее\ограничивающее правило nftables	Цель сценария
1	ICMP-трафик	icmp type echo-request drop	Проверка блокировки ICMP-флуда и влияния на легитимные echo-запросы и загрузку CPU
2	SSH	tcp dport 22 drop	Оценка влияния блокировки SSH-доступа на легитимные сессии и нагрузку процессора
3	HTTP	tcp dport 8080 drop	Анализ последствий запрета HTTP-сервиса для легитимных запросов и производительности
4	UDP (iperf3)	udp dport 5201 drop	Исследование влияния фильтров на легитимный UDP-поток и дополнительной нагрузки при его блокировке
5	limit	icmp type echo-request limit rate 8/minute accept	Исследование функционала ограничения частоты обработки ICMP-трафика
6	mangle	ip saddr 192.168.10.10 ip ttl set 64	Исследование функционала mangle по изменению полей IP-заголовка (TTL) легитимных пакетов
7	reject/drop	tcp dport 22 reject with tcp reset tcp dport 22 drop	Исследование функциональных различий действий REJECT и DROP при блокировке SSH-доступа

1. В первом эксперименте оценивается, как ICMP-фильтрация влияет на сохранность легитимного трафика на фоне флуд-атаки. Тест строится по следующей схеме: сначала в течение 30 секунд проходят только редкие ICMP-эхо-запросы от клиента к серверу, что задаёт уровень нормальной работы без заметной нагрузки на процессор маршрутизатора; затем на следующие 30 секунд запускается ping-флуд без правил фильтрации, и всплеск нагрузки до средних значений порядка 16,5% показывает, что атака способна вытеснять любые параллельные легитимные запросы; в заключительные 30 секунд включается правило блокировки ICMP, которое отбрасывает основной поток флуд-пакетов, снижая среднюю загрузку CPU примерно до 5,3%, при этом одиночные ICMP-запросы по-прежнему успешно достигают сервера, что демонстрирует сохранение доступности легитимного трафика при активном фильтре.

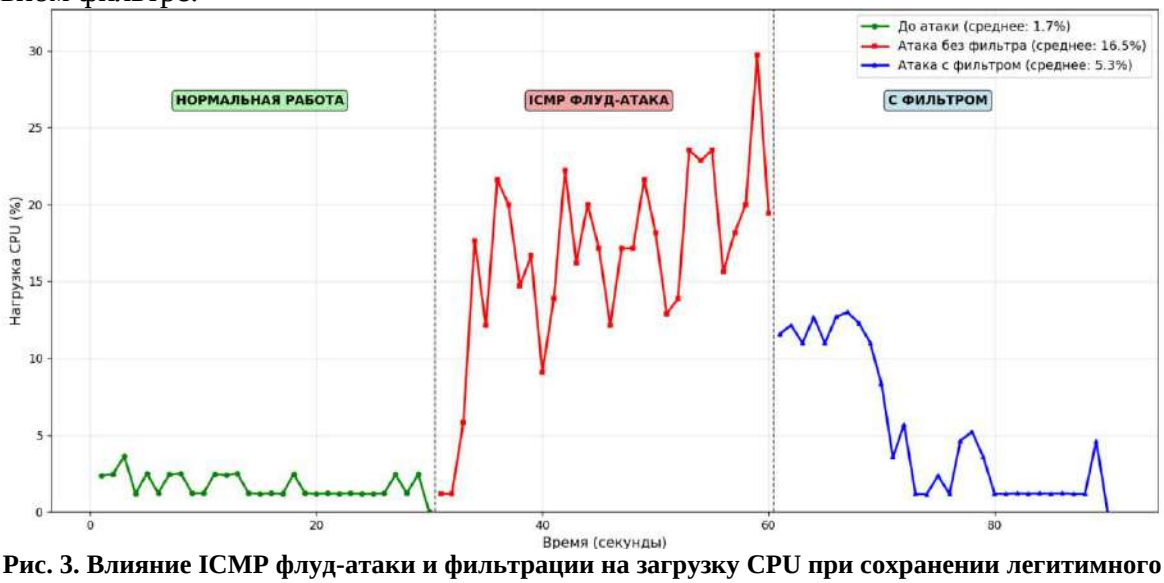


Рис. 3. Влияние ICMP флуд-атаки и фильтрации на загрузку CPU при сохранении легитимного ICMP-трафика

2. Эксперимент с SSH показывает, как фильтрация влияет на легитимные интерактивные сессии при наличии флуд-атаки на тот же сервис. В начале измеряется фоновая нагрузка при одной SSH-сессии между клиентом и сервером: среднее значение около 1,8%sys соответствует нормальной работе легитимного трафика без заметного влияния на процессор. Затем запускается серия частых попыток SSH-подключения на порт 22 при активном запрещающем правиле, и средняя загрузка возрастает до 41,1%sys, при этом новые соединения блокируются, а уже установленная легитимная сессия продолжает работать без разрывов, что демонстрирует способность фильтра изолировать атакующий трафик, не нарушая текущие разрешённые подключения.

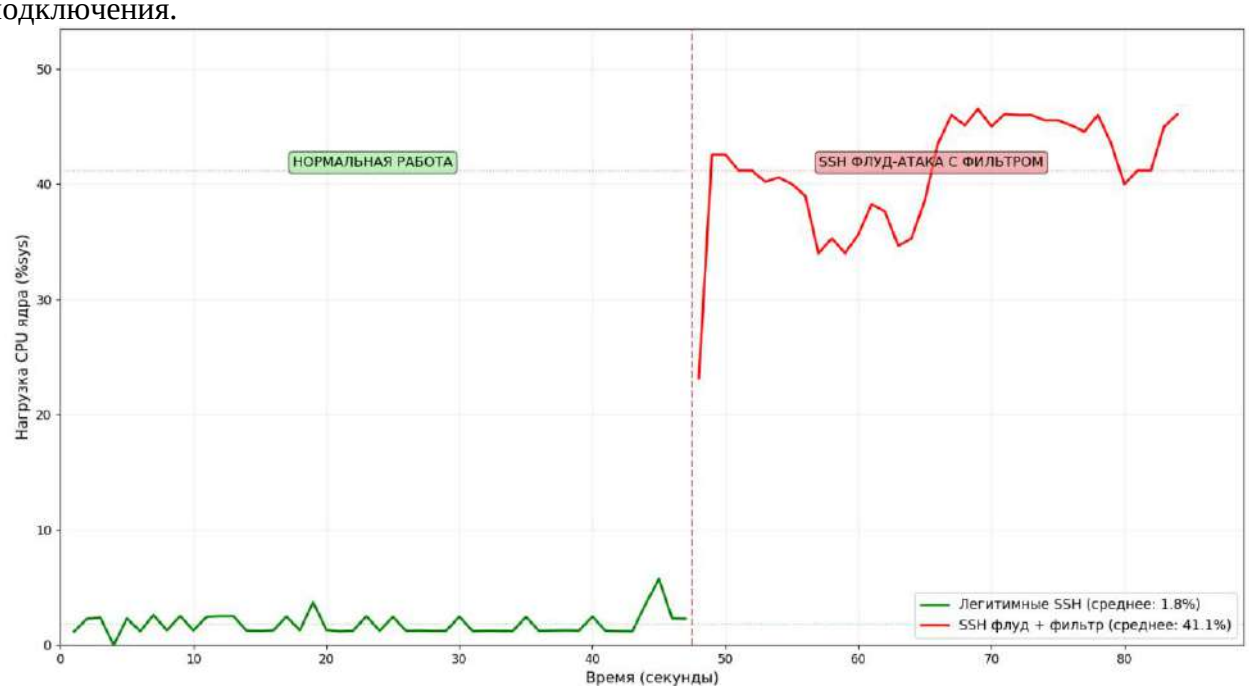


Рис. 4. Влияние SSH флуд-атаки с фильтрацией на загрузку CPU и работу легитимных SSH-сессий

3. В HTTP-эксперименте сравнивается влияние фильтрации на легитимный нагрузочный трафик ApacheBench и на дополнительный «атакующий» поток. В первой половине графика сервер обрабатывает только легитимные HTTP-запросы, при этом загрузка ядра держится на высоком уровне около 78–82%sys, что отражает именно стоимость работы веб-сервера под нагрузкой, а не фильтрации. Во второй половине добавляется блокирующее правило для вредоносного трафика, но профиль нагрузки практически не меняется: среднее значение остаётся в том же диапазоне, а легитимные запросы продолжают успешно обслуживаться, что показывает минимальное влияние фильтра на производительность HTTP-сервиса по сравнению с самой прикладной нагрузкой.

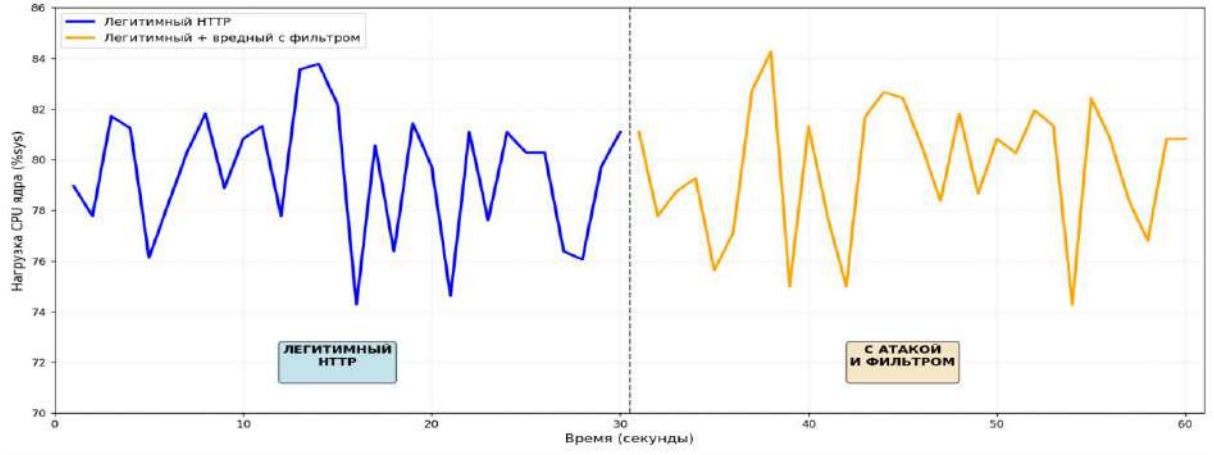


Рис. 5. Нагрузка CPU при легитимном HTTP-трафике и при добавлении атакующего потока с фильтрацией

4. UDP-эксперимент с iperf3 показывает, как фильтрация влияет на легитимный поток и на дополнительный заблокированный трафик. На первом участке графика передаётся только легитимный UDP-поток 10 Мбит/с от клиента к серверу, и нагрузка ядра маршрутизатора стабилизируется на уровне около 2–4%sys, что отражает стоимость нормальной работы при постоянной передаче без атак. После отметки в 30 секунд запускается второй поток iperf3 с роутера с большей скоростью, который отбрасывается правилами nftables, и средняя загрузка возрастает, появляются отдельные пики до 15–16%sys, однако базовый легитимный поток клиента при этом сохраняет целевой битрейт и не деградирует, то есть фильтр добавляет умеренную вычислительную нагрузку, но не вытесняет разрешённый трафик.

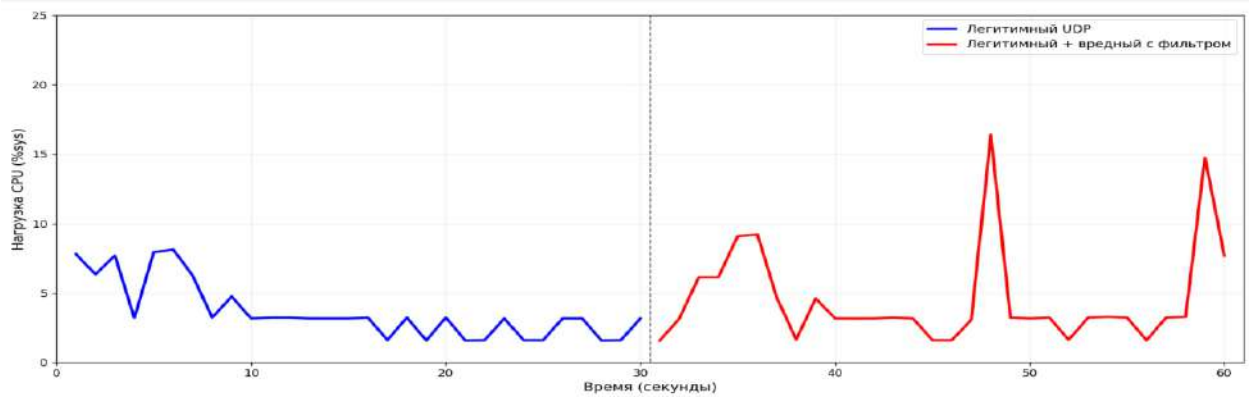


Рис. 6. Нагрузка CPU при легитимном UDP-потоке и при добавлении заблокированного атакующего трафика

5. В эксперименте с limit исследуется, как счётчик ограничивает частоту обработки ICMP-запросов независимо от интенсивности генерации трафика. В цепочке forward задавалось ограничивающее правило, после чего с клиента отправлялась серия из 20 ping-запросов на сервер. Вывод утилиты ping показывает, что при такой конфигурации принимается ровно восемь ответов, оставшиеся пакеты теряются, что подтверждает корректную работу механизма limit и демонстрирует возможность жёстко ограничивать объём легитимного ICMP-трафика без изменения характера отдельных успешно доставленных запросов.

6. Для демонстрации работы mangle использовалось правило, которое изменяет TTL у проходящих ICMP-пакетов без нарушения их доставки. Вывод tcpdump на сервере показывает, что входящие ICMP echo-request приходят уже с TTL=64, хотя изначально клиент отправлял пакеты с TTL=128, при этом все запросы и ответы успешно проходят через маршрутизатор, что подтверждает корректную модификацию заголовка без влияния на легитимный трафик.

7. В эксперименте с REJECT/DROP исследуется, как тип действия влияет на поведение легитимного клиента при попытке доступа к заблокированному SSH-сервису. При использовании правила tcp dport 22 reject with tcp reset клиентская команда time ssh -o ConnectTimeout=5 root@192.168.20.20 exit немедленно получает ответ «Connection refused», как результат, общее время выполнения крайне мало — это означает мгновенное информирование пользователя о том, что доступ запрещён. В случае замены правила на tcp dport 22 drop соединение завершается по тайм-ауту ConnectTimeout, и, как результат, клиент вынужден ждать несколько секунд без какого-либо ответа, хотя уровень сетевой изоляции сервиса при этом остаётся тем же, что подчёркивает функциональное различие между REJECT и DROP именно с точки зрения удобства легитимного пользователя.

Заключение

В ходе исследования была успешно проведена серия функциональных тестов архитектуры netfilter с использованием фреймворка nftables в ОС «Альт». На развернутом стенде экспериментально проверены ключевые механизмы: фильтрация трафика (ICMP, HTTP, SSH), ограничение скорости (limit), манипуляция пакетами (mangle), а также сравнение действий REJECT и DROP. Проведённые эксперименты не выявили особенностей в реализации netfilter, отличных от стандартного ядра Linux.

Результаты подтвердили корректность и предсказуемость работы всех протестированных функций. Netfilter в связке с nftables продемонстрировал высокую эффективность как для реализации базовых политик безопасности, так и для расширенного управления трафиком, что делает его надежной основой для построения защищённой сетевой инфраструктуры на данной платформе.

Библиографический список

1. Netfilter.org project. Netfilter hooks – nftables wiki [Электронный ресурс]. – URL: https://wiki.nftables.org/wiki-nftables/index.php/Netfilter_hooks (дата обращения: 14.12.2025).
2. Документация ALT Linux Team [Электронный ресурс]. – URL: https://docs.altlinux.org/ru-RU/alt-server/11.1/html/alt-server/setup-inet-connection--chapter.html (дата обращения: 06.12.2025).
3. Netfilter.org project [Электронный ресурс]. – URL: https://www.netfilter.org/ (дата обращения: 06.12.2025).
4. Salah, K. Performance Modeling and Analysis of Network Firewalls / K. Salah, K. El-Badawi, A. El-Badawi // International Journal of Network Security & Its Applications. – 2012. – Vol. 4, № 2. – P. 69–82.
5. Kadlecsik, J. Netfilter Performance Testing [Электронный ресурс] / J. Kadlecsik. – Netfilter.org, 2010. – 15 p. – URL: https://people.netfilter.org/kadlec/nftest.pdf (дата обращения: 11.12.2025).
6. Уймин, А. Г. Сетевое и системное администрирование. Демонстрационный экзамен КОД 1.1 : учебно-методическое пособие для СПО / А. Г. Уймин. – 3-е изд. – Санкт-Петербург : Лань, 2022. – 480 с.
7. Benchmarking Methodology for Firewall Performance : RFC 3511 [Электронный ресурс]. – IETF, 2002. – URL: https://www.rfc-editor.org/rfc/rfc3511.html (дата обращения: 11.12.2025).

References

1. Netfilter.org project. (2025). Netfilter hooks – nftables wiki. Retrieved December 14, 2025, from https://wiki.nftables.org/wiki-nftables/index.php/Netfilter_hooks
2. ALT Linux Team Documentation. (2025). Network connection setup (ALT Server 11.1). Retrieved December 6, 2025, from https://docs.altlinux.org/ru-RU/alt-server/11.1/html/alt-server/setup-inet-connection--chapter.html
3. Netfilter.org project. (2025). Official website. Retrieved December 6, 2025, from https://www.netfilter.org/
4. Salah, K., El-Badawi, K., & El-Badawi, A. (2012). Performance modeling and analysis of network firewalls. International Journal of Network Security & Its Applications, 4(2), 69–82.
5. Kadlecsik, J. (2010). Netfilter performance testing. Netfilter.org. Retrieved December 11, 2025, from https://people.netfilter.org/kadlec/nftest.pdf
6. Uymin, A. G. (2022). Network and system administration. Demonstration Exam CODE 1.1: Educational and methodological manual for secondary vocational education (3rd ed.). Lan Publishing House.
7. IETF. (2002). Benchmarking methodology for firewall performance: RFC 3511. Retrieved December 11, 2025, from https://www.rfc-editor.org/rfc/rfc3511.html

Информация об авторах

Тюхтин Кирилл Сергеевич — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: kirill.t0625@gmail.com

NETFILTER ARCHITECTURE. FUNCTIONAL TESTING IN ALT OS

Tyukhtin K. S.[1]

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

Abstract. The article presents a practical study of the netfilter subsystem implementation in the Russian Alt OS. The relevance of the work is driven by the need to verify the correctness and completeness of fundamental network security mechanisms in domestic software. The research addresses the task of comprehensive functional testing of the netfilter architecture through its modern interface -- the nftables framework.

A series of experiments was conducted on an isolated laboratory testbed simulating a corporate network segment. The testing covers key functionalities: packet filtering (ICMP, TCP, UDP), network address translation (NAT), packet header manipulation (mangle), and traffic management using rate limiting (limit) and stateless filtering mechanisms. Particular attention is paid to comparing the behavioral aspects of REJECT and DROP actions when blocking access.

The results experimentally confirm the full correctness and predictability of all tested mechanisms. Load testing allowed for observing the impact of various filtering rules on the router's CPU load. A key practical result is the verification of the reference-standard implementation of netfilter in Alt OS, confirming its suitability for deploying mission-critical network infrastructure. The obtained data and testing methodology can be used for the certification and security audit of domestic Linux distributions.

The study's conclusions indicate that the netfilter/nftables implementation in Alt OS is reference-standard and functionally identical to that in the standard Linux kernel. The presented results demonstrate the platform's readiness to serve as a foundation for building reliable and efficient corporate firewalls.
Keywords: Netfilter, nftables, Alt OS, functional testing, firewall, packet filtering, network security.

Information about the authors

Tyukhtin Kirill Sergeevich – student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: kirill.t0625@gmail.com

ПРОФЕССИОНАЛИТЕТ, 2025, № 4 (008)

***УДК** 004.056.53*

Д. В. Демидова[1], Е. Д. Комнацкая[1]
[1]ФГАОУ ВО «РГУ нефти и газа (НИУ) 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];

– 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 «Магма» (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:

Примечание к конфигурации. В эксперименте использовались пакеты ветки p10 для архитектуры x86_64 на обеих виртуальных машинах; иные ветки и i586 к воспроизводимой конфигурации не относятся.

– 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».

```
# 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
```

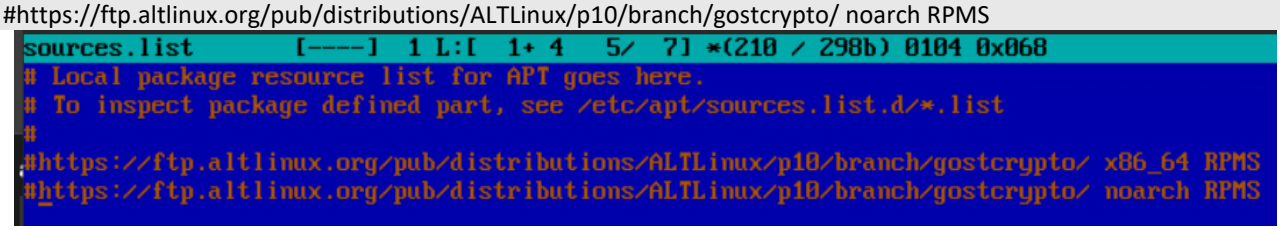


Рис. 1. Добавление репозитория gostcrypto

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/p10/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.

```
#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/%u
#Match Group !secured,*
#       AuthorizedKeysFile /etc/openssh/authorized_keys/%u .ssh/authorized_keys

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

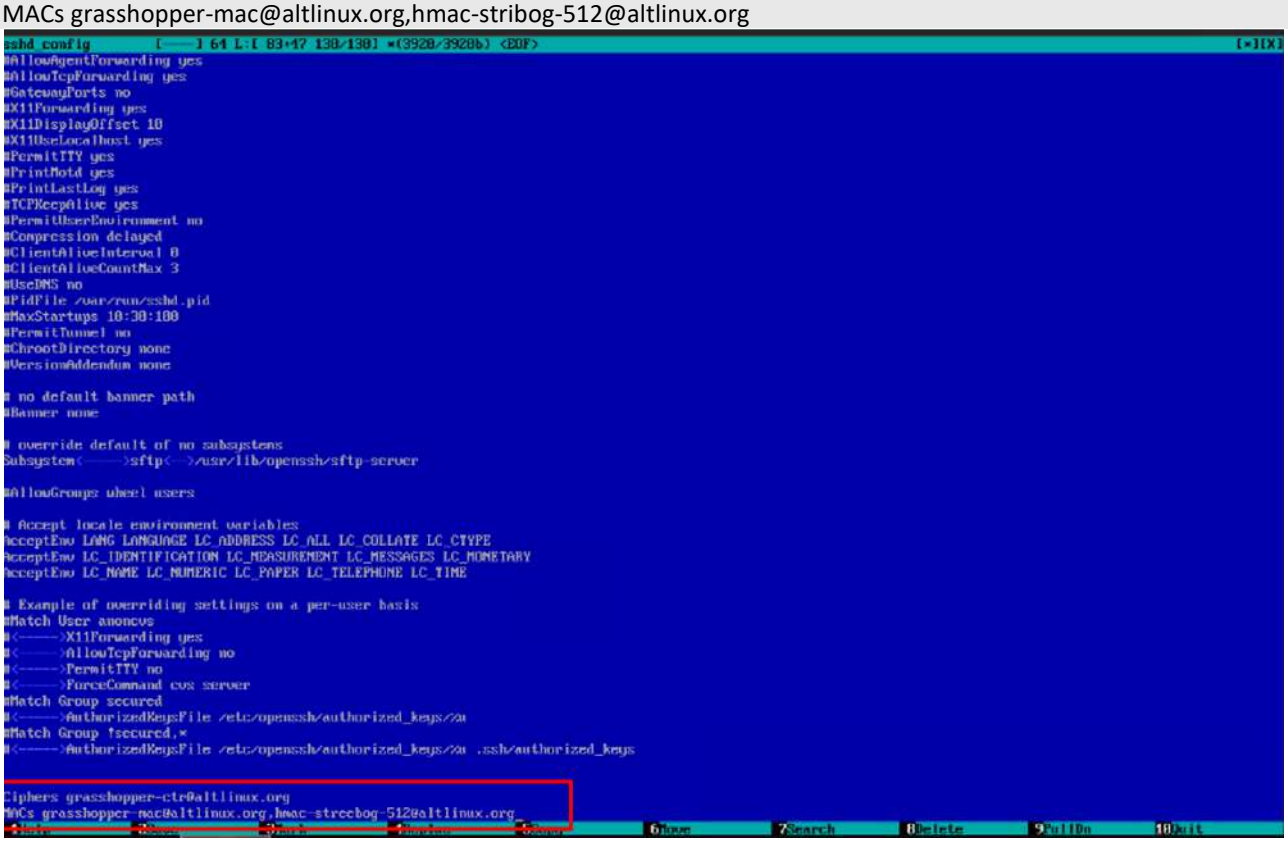


Рис. 2. Настройка sshd_config

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.

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: http://rsync.altlinux.ru/pub/distributions/ALTLinux/p10/branch/x86_64/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).

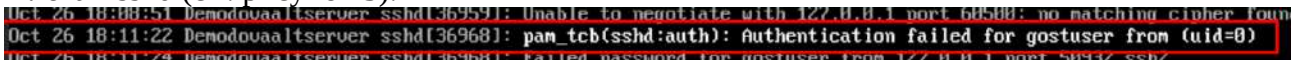


Рис. 3. Сообщение об ошибке несогласования алгоритмов при попытке подключения без ГОСТ-шифров.

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].

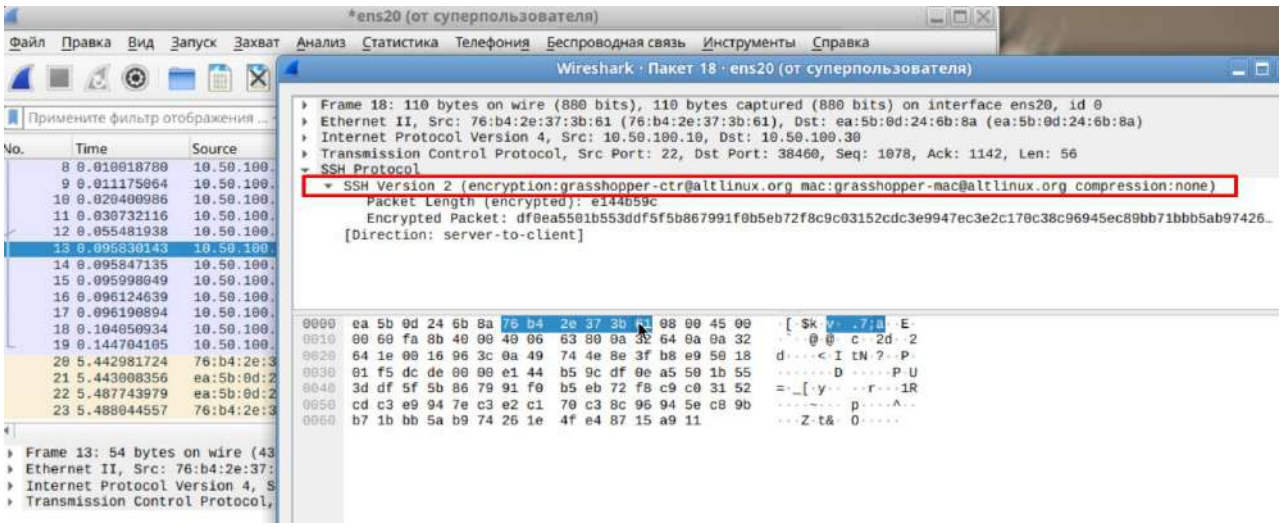


Рис. 4. Анализ SSH-трафика при успешном подключении с ГОСТ-алгоритмами.

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).

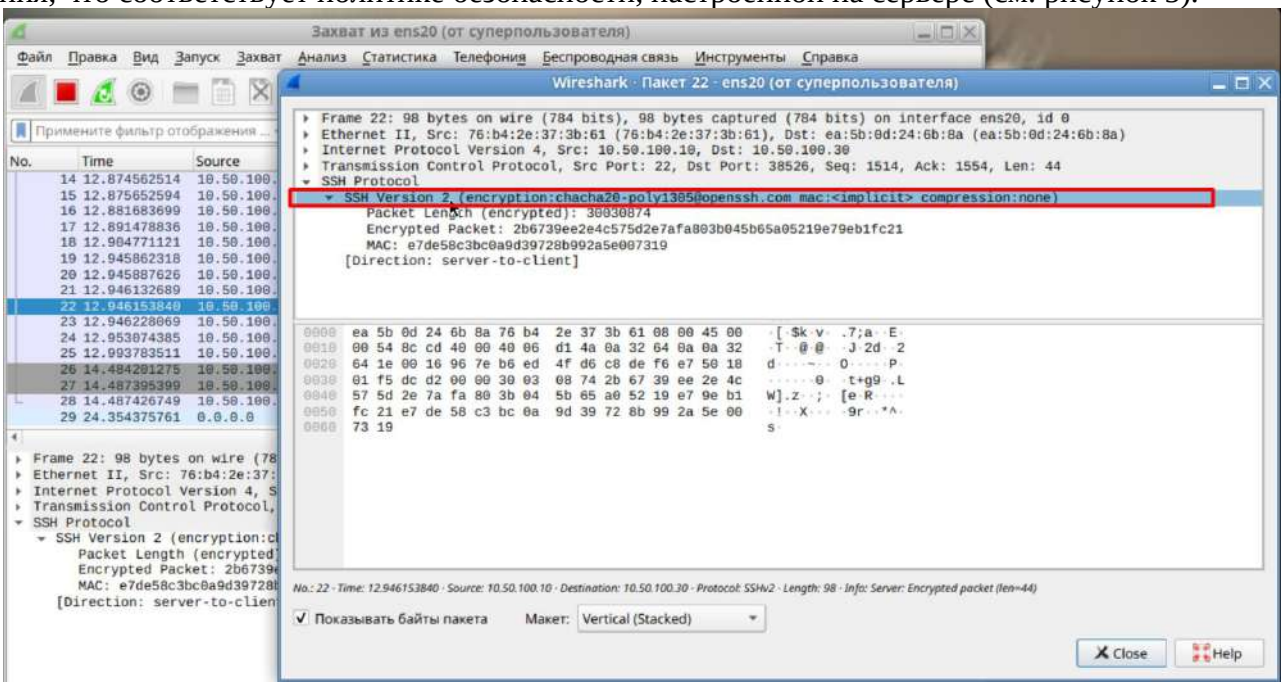


Рис. 5. Анализ SSH-трафика при неудачной попытке подключения с несовместимыми алгоритмами.

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).

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

Тип шифрования	Алгоритм шифрования	Пропускная способность	Средняя пропускная способность	Отклонение	Переданный объем 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		
	chacha20-poly1305@openssh.com	31.9 Гбит/с	31.7 Гбит/с		36.9 ГБ	60-62%	0.2%
		30.6 Гбит/с			35.7 ГБ		

Стан-дарт-ный алгоритм		32.6 Гбит/с		±0.6 Гбит/с (1.8%)	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%).

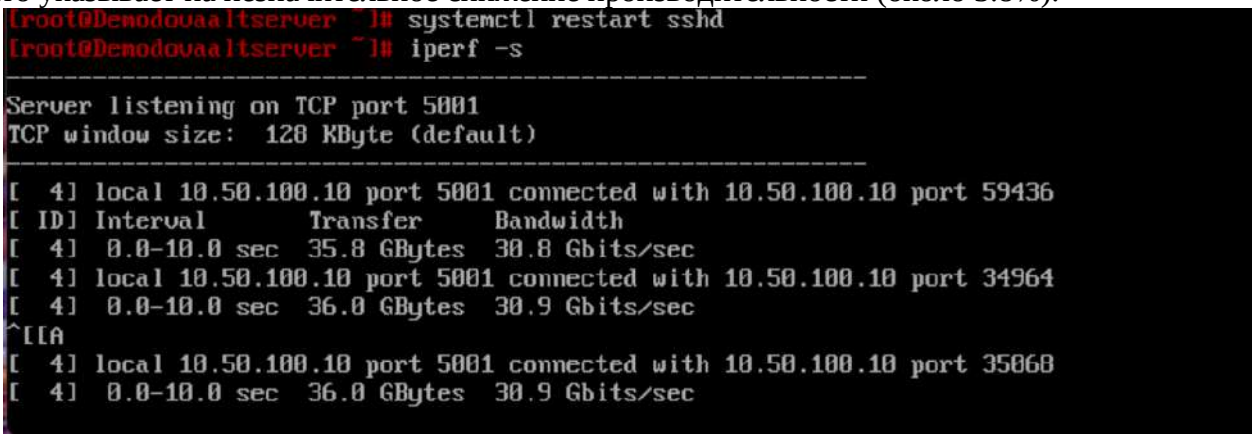


Рис. 6. Измерение пропускной способности SSH-сессии с ГОСТ-шифрованием.

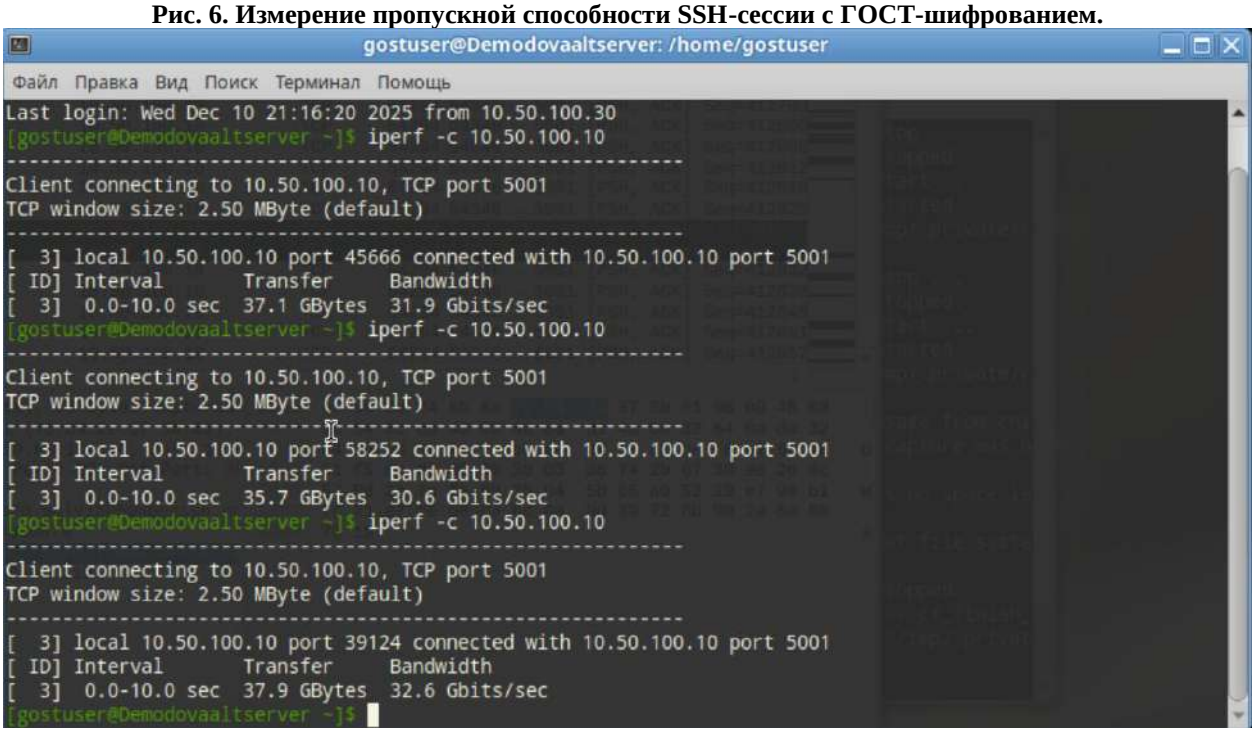


Рис. 7. Измерение пропускной способности SSH-сессии со стандартным алгоритмом.

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.

Вывод

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. Алицар, А. Wireshark для всех. Лайфхаки на каждый день [Электронный ресурс] / А. Алицар // 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. – Москва : Стандартинформ, 2012.
3. ГОСТ Р 34.12–2015. Информационная технология. Криптографическая защита информации. Блочные шифры. – Москва : Стандартинформ, 2015.
4. Официальный репозиторий пакетов ALT Linux [Электронный ресурс] // ALT Linux Packages. – 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» / А. Г. Уймин // 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. (2021, June 14). Wireshark for everyone: Life hacks for every day. Habr. https://habr.com/ru/companies/vdsina/articles/562110/
2. GOST R 34.10–2012. (2012). Information technology. Cryptographic protection of information. Processes of digital signature generation and verification. Standartinform.
3. GOST R 34.12–2015. (2015). Information technology. Cryptographic protection of information. Block ciphers. Standartinform.

4. ALT Linux Packages. (n.d.). Official repository of ALT Linux packages. https://packages.alt-linux.org/
5. ALT Linux Packages. (n.d.). OpenSSH GOSTCrypto. https://packages.alt-linux.org/ru/p10/srpms/openssh-gostcrypto/
6. ALT Linux Packages. (n.d.). OpenSSL GOST Engine. https://packages.alt-linux.org/ru/p10/srpms/openssl-gost-engine/
7. ALT Linux Documentation. (n.d.). Creating SSH tunnels using IP packet header integrity control in accordance with GOST R 34.12-2015. https://docs.altlinux.org/ru-RU/alt-workstation/10.2/html/alt-workstation/ssh_t.html
8. Uimin, A. G. (2024). Definition of a domestic technological platform within the framework of creating a cyber polygon: Network and system administration. In Current challenges in personnel training: Training of specialists in modern areas of information technology, cybersecurity, and ICT electronics relevant to the data economy (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

D. V. Demidova[1], E. D. Komnatskaya[1]

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

Abstract. The paper examines ways to improve the security of remote administration by using Russian cryptographic algorithms when establishing SSH connections in the ALT Linux operating system. The relevance of the study is determined by the need to comply with national information security requirements and to reduce dependence on foreign cryptographic solutions. The research problem consists in assessing the stability of SSH connections based on GOST algorithms and in analyzing how the selected cryptographic policy affects unauthorized access attempts and connection performance. The object of the study is the process of establishing and maintaining an SSH connection, while the subject is the server behavior when the cryptographic algorithm sets of the client and the server do not match. To solve the problem, an experimental SSH server and client configuration was performed in ALT Linux using gostcrypto packages that implement the Grasshopper and Magma algorithms and Stribog-based integrity control functions. The methodology included modeling legitimate and illegitimate connection attempts, analyzing network traffic with Wireshark, and measuring bandwidth and resource load. The experiments showed that a server configured to use only GOST algorithms effectively blocks connections with incompatible cryptographic parameters. When the client is configured correctly, the session is established stably and follows the specified security policy. The performance measurements demonstrate that GOST encryption provides throughput comparable to standard OpenSSH algorithms with a similar computational load. The results confirm the practical applicability of GOST-based SSH encryption 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.

УДК *004.72*

И. А. Чаков [1], М. С. Евтюхов [1], В. В. Павловский [1]

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

ИССЛЕДОВАНИЕ ФУНКЦИОНАЛА И ВРЕМЕНИ ВОССТАНОВЛЕНИЯ VRRP НА БАЗЕ РЕШЕНИЙ ECOROUTER

Аннотация. Актуальность обусловлена расширением применения отечественной программно-аппаратной платформы EcoRouter в условиях импортозамещения сетевой инфраструктуры при недостаточной изученности её реализации протокола VRRP (Virtual Router Redundancy Protocol) с точки зрения производительности и стабильности в реальных сценариях эксплуатации. Цель исследования — оценить функциональные возможности и влияние на производительность сети протокола VRRP при его реализации в платформе EcoRouter, а также выработать практические рекомендации по настройке параметров отказоустойчивости.

Исследование выполнено в виртуализированной лабораторной среде на базе GNS3 и VirtualBox с использованием маршрутизаторов EcoRouterOS. Топология включала два маршрутизатора, два коммутатора и четыре клиентских узла, объединённых в два сегмента сети с самостоятельными VRRP-группами. Методами iperf3 и Wireshark оценивались время переключения ролей Master/Backup, кратковременное прерывание передачи и интервал отправки служебных анонсов при варьировании параметров advertisement-interval, preempt-mode и switch-back-delay.

Примечание к воспроизводимости. Параметры хост-системы и виртуальных машин, полные команды iperf3 и число повторов в исходных материалах не зафиксированы; поэтому количественные результаты рассматриваются как предварительные единичные лабораторные наблюдения для EcoRouterOS 3.2.6.2.

Установлено, что при значении advertisement-interval по умолчанию (50) время переключения составляет около 1,5 с, тогда как увеличение параметра до 3000 повышает это время до 60 с; чрезмерно малые значения (5) приводят к нестабильному «флаппингу» ролей. Показано, что параметр switch-back-delay влияет на время возврата роли Master, но не изменяет интервал отправки анонсов, а отключение preempt-mode блокирует автоматическое восстановление приоритетной роли.

Результаты подтверждают, что реализация VRRP в EcoRouter обеспечивает заявленную отказоустойчивость при корректном выборе конфигурационных параметров, и позволяют сформулировать практические рекомендации для сетевых инженеров по балансу между скоростью реакции на отказ и стабильностью сети. Полученные данные могут использоваться при проектировании отечественных отказоустойчивых сетевых инфраструктур и в учебном процессе по сетевым дисциплинам.

Ключевые слова: протокол VRRP, отказоустойчивость сети, EcoRouter, резервирование маршрутизаторов, время переключения, сетевая производительность, импортозамещение.

Введение

В условиях стремительного роста требований к отказоустойчивости и непрерывности сетевых сервисов вопросы обеспечения высокой доступности шлюзов и маршрутизаторов приобретают особую значимость. Современные корпоративные и провайдерские сети чаще сталкиваются с необходимостью минимизировать время простоя при отказах оборудования в критически важных сценариях — таких как доставка мультимедийного контента, облачные сервисы, системы видеонаблюдения и распределённые приложения реального времени.

Актуальность обусловлена тем, что на практике эффективность VRRP сильно зависит от его реализации в конкретной платформе. Особенно это актуально для отечественных решений, таких как EcoRouter, которые всё шире внедряются в рамках политики импортозамещения, но зачастую недостаточно изучены с точки зрения производительности и стабильности в реальных сценариях эксплуатации. Возникает необходимость оценить, насколько реализация VRRP в отечественных решениях действительно обеспечивает заявленную надёжность, или же в какой степени она сопряжена со значительными накладными расходами, такими как снижение пропускной способности, увеличение задержек и нестабильность работы маршрутизатора.

Объектом данного исследования являются механизмы обеспечения отказоустойчивости сетевой инфраструктуры на основе протокола VRRP (Virtual Router Redundancy Protocol) [1].

Предметом исследования выступает реализация протокола VRRP в программно-аппаратной платформе EcoRouter, включая её функциональные возможности, особенности

конфигурации, а также влияние на производительность сети — такие параметры, как: пропускная способность, задержка передачи данных, потери пакетов и время переключения при отказе основного маршрутизатора.

Цель исследования — оценить функциональные возможности VRRP в EcoRouter, проверить механизм отказоустойчивости и измерить время переключения при отказе. Пропускная способность, задержка и нагрузка CPU в сводном виде не измерялись, поэтому выводы ограничены функциональностью и временем восстановления на лабораторном стенде.

Результаты данного исследования представляют практическую ценность для сетевых инженеров и системных администраторов при проектировании отказоустойчивых сетей на отечественном оборудовании. Кроме того, материалы работы могут быть использованы в учебном процессе по курсам компьютерных сетей и сетевой инженерии для иллюстрации реального поведения протокола, особенностей его настройки и компромиссов между скоростью восстановления и стабильностью сети.

Литературный обзор

1. Анализ основных исследований

Вопрос применения протокола VRRP в сетях поднимался в множестве исследований, но, как выяснилось, в большинстве случаев упор был направлен на изучение отказоустойчивости, безопасности и самой настройке протокола.

Например, Иван Тихомиров в своей статье «Принцип работы протокола VRRP» [2] рассматривает, как очевидно, сам принцип работы протокола.

Симбирцев Е.А. и Галяутдинов Д.А. в своей статье «Настройка и тестирование протокола VRRP. Вопросы безопасности» [3] рассматривают пошаговую конфигурацию протокола VRRP, тестируют механизм переключения виртуального IP-адреса при отказе основного маршрутизатора и приводят анализ уязвимостей протокола.

2. Основные гипотезы исследования

Использование протокола VRRP (Virtual Router Redundancy Protocol) в решениях EcoRouter повышает отказоустойчивость сетевой инфраструктуры без существенного снижения производительности, при условии корректной настройки параметров приоритета, таймеров и распределения нагрузки между основным и резервным маршрутизаторами.

Материалы и методы

Данное исследование носит экспериментальный характер и включает следующие методы:

На первом этапе выполнен теоретический анализ протокола VRRP (RFC 5798) и его реализации в платформе EcoRouter. Проведён обзор существующих решений по обеспечению отказоустойчивости, были изучены литературные и нормативные источники, официальная документация EcoRouter [4], а также выполнен сравнительный анализ применения VRRP в реальных сетях.

Далее была выполнена экспериментальная настройка виртуальной сети на платформе EcoRouter с использованием VRRP для имитации отказов маршрутизаторов. В ходе конфигурации варьировались ключевые параметры протокола – время ожидания, таймеры анонсов и условия переключения на резервный маршрутизатор.

Для оценки влияния VRRP на производительность сети проводились замеры пропускной способности, задержек, потерь пакетов и времени переключения при отказе основного маршрутизатора. В качестве инструментов использовались iperf для генерации трафика и Wireshark для мониторинга и анализа сетевого взаимодействия в реальном времени. Эксперименты включали тестирование различных комбинаций таймеров и приоритетов с целью выявления оптимальных настроек, минимизирующих негативное влияние на производительность.

На заключительном этапе полученные результаты сопоставлялись с теоретическими ожиданиями. Проведена оценка стабильности и эффективности работы системы при отказе основного маршрутизатора, а также рассмотрены возможности оптимизации конфигурации

VRRP для повышения отказоустойчивости без значительного ухудшения пропускной способности и времени отклика.

1.1 Описание исследуемой системы и программно-аппаратной среды

Наше исследование выполнено в виртуализированной лабораторной среде созданной на основе VirtualBox версии 7.2.0 и GNS3 [5] версии 2.2.52, в качестве маршрутизаторов используется EcoRouterOS версии 3.2.6.2.21765, в качестве коммутаторов – дистрибутив Альт Сервер версии 10.4, а в качестве компьютеров – дистрибутив Альт Рабочая Станция версии 10.4. Исследование направлено на оценку влияния протокола VRRP на **производительность сети**: стабильность передачи данных, задержку, потерю пакетов и время переключения при отказе активного маршрутизатора.

Тестовая топология состоит из двух маршрутизаторов EcoRouter (EcoRouter1 и EcoRouter2), двух коммутаторов (SW1 и SW2) и четырёх клиентских ПК (PC1–PC4). Все устройства соединены через гигабитные Ethernet-интерфейсы. Сама топология представлена на рисунке 1.

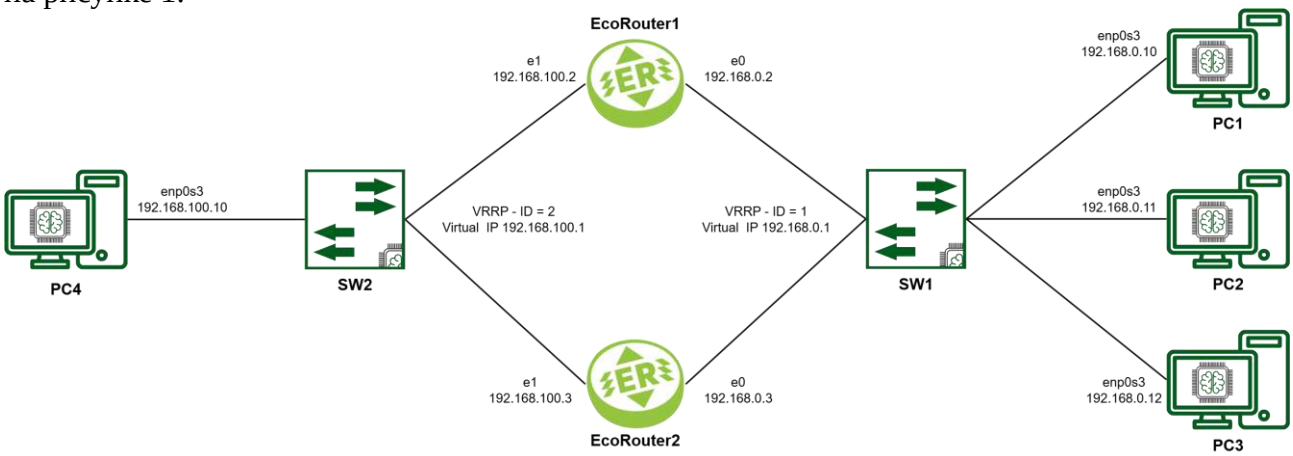


Рисунок 1 — Топология эксперимента

Сеть разбита на два сегмента:

- **LAN-сегмент 192.168.0.0/24**, обслуживаемый через интерфейсы e0 маршрутизаторов;
- **LAN-сегмент 192.168.100.0/24**, обслуживаемый через интерфейсы e1.

На каждом сегменте реализован свой экземпляр VRRP:

- **VRRP ID = 1** на интерфейсах e0 (сегмент 192.168.0.0/24):
  - Виртуальный IP-адрес: 192.168.0.1;
  - Master — EcoRouter1 (192.168.0.2);
  - Backup — EcoRouter2 (192.168.0.3).
- **VRRP ID = 2** на интерфейсах e1 (сегмент 192.168.100.0/24):
  - Виртуальный IP-адрес: 192.168.100.1;
  - Master — EcoRouter1 (192.168.100.2);
  - Backup — EcoRouter2 (192.168.100.3).

Клиентские ПК подключены к коммутаторам:

- PC1, PC2, PC3 — к SW1 (в сегменте 192.168.0.0/24);
- PC4 — к SW2 (в сегменте 192.168.100.0/24).

В качестве источника и приёмника трафика использовались ПК PC2 и PC4, на которых запускалось ПО iperf3 для генерации TCP-потоков [6] между сегментами. Для анализа сетевых характеристик применялся Wireshark — для захвата пакетов на интерфейсах маршрутизаторов и коммутаторов, что позволило фиксировать:

- время переключения VRRP (от момента отказа Master до активации Backup);
- наличие потерь пакетов при переключении;
- изменение задержки отправки пакетов;
- интервал отправки VRRP-анонсов.

Сеть была полностью изолирована от внешнего трафика. Все устройства работали в стабильных условиях, синхронизированных по времени через NTP. Такая конфигурация позволяет имитировать реальные сценарии использования VRRP в корпоративных сетях с несколькими сегментами и оценивать его влияние на **сетевую производительность** в условиях динамического изменения состояния шлюзов.

1.2 Тестирование функциональности и влияния на производительность протокола VRRP

Для подтверждения корректности реализации протокола VRRP на базе решений EcoRouter была разработана и применена серия контролируемых тестовых сценариев, охватывающих как штатные, так и аварийные режимы работы. Целью функционального тестирования являлось подтверждение способности VRRP обеспечивать прозрачное и своевременное переключение шлюза при отказе активного маршрутизатора, а также корректное восстановление после возврата Master-маршрутизатора в строй.

Экспериментальная проверка состояла из нескольких этапов. На первом этапе была проведена и**нициализация и верификация VRRP-сессий.** После настройки пары маршрутизаторов EcoRouter1 и EcoRouter2 на интерфейсах e0 (сегмент 192.168.0.0/24) и e1 (сегмент 192.168.100.0/24) проводилась проверка:

- Назначения ролей Master/Backup в соответствии с приоритетами (EcoRouter1 — приоритет 110, EcoRouter2 — 100);
- Доступности PC4 с клиентских узлов PC1–PC3 посредством ICMP-запросов и ARP-резолва;
- Корректности VRRP-адвертайзментов, захваченных в Wireshark: проверялись тип пакета (Advertisement) и адрес источника.

Помимо этого, на данном этапе была протестирована функция *advertisement-interval*, которая изменяет интервал отправки VRRP-сообщений. Было установлено, что по умолчанию VRRP-сообщения отправляются с интервалом 0.5 сек., что эквивалентно аргументу ~50. Захват VRRP-сообщений в Wireshark представлен на рисунке 2.

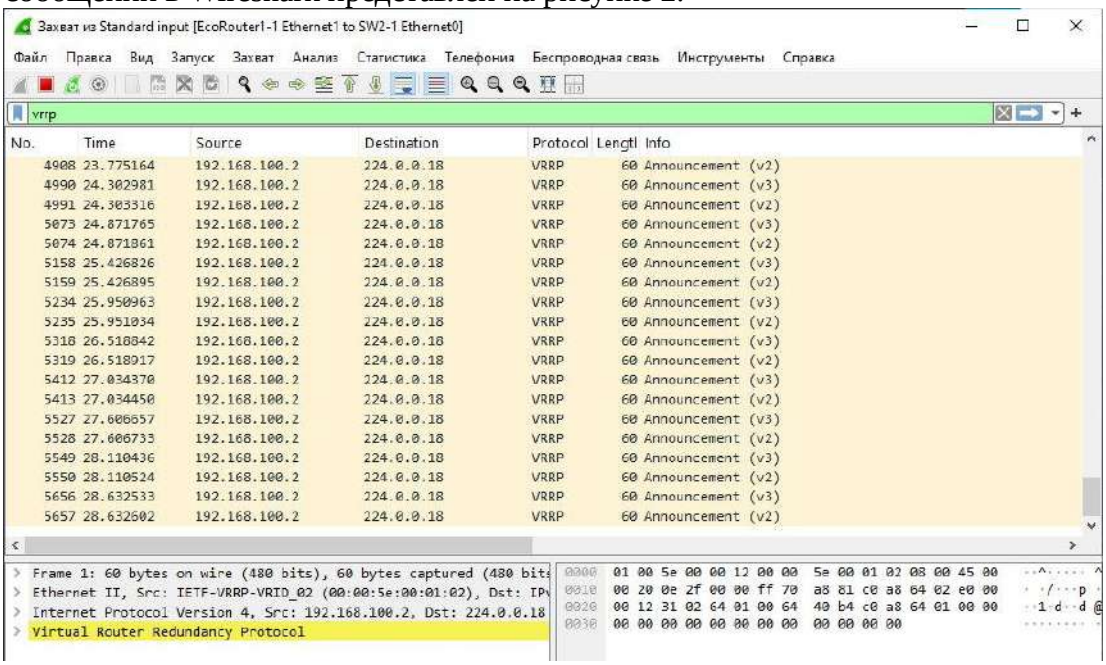


Рисунок 2 – Захват VRRP-сообщений в Wireshark

При значении advertisement-interval 500 ожидаемый интервал составляет около 5 секунд, если параметр задан в сотых долях секунды. В захвате одновременно наблюдались VRRPv2 и VRRPv3; для интерпретации интервалы следует считать отдельно по источнику, VRID и версии, предполагая режим совместимости [1]. Пример выставления таймера и захват пакетов из Wireshark представлены на рисунках 3, 4.

```
router vrrp 1 e0
disable
```

```
advertisement-interval 500
enable
```

```
ecorouter(config)#router vrrp 1 e0
ecorouter(config-router)#disable
ecorouter(config-router)#advertisement-interval 500
ecorouter(config-router)#enable
```

Рисунок 3 – Пример выставления таймера анонсов

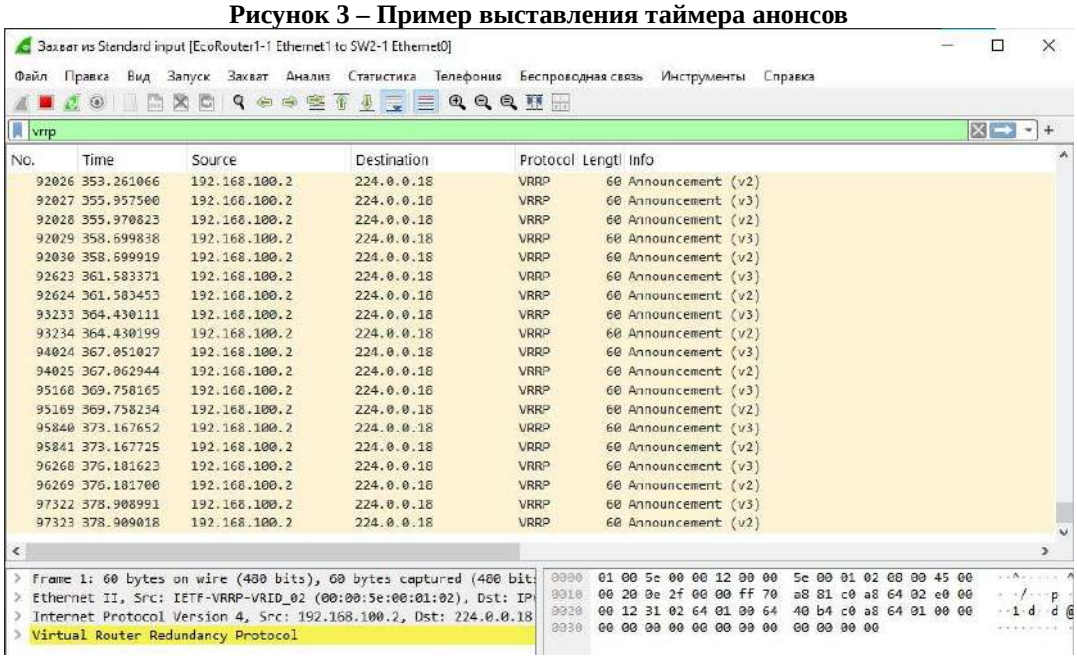


Рисунок 4 – Захват пакетов в Wireshark

Также были протестированы сильно малое и большое значения параметра – итогом стало то, что при параметре интервала, равном 5, второй маршрутизатор ушел в «петлю» и постоянно менял свою роль. Результат представлен на рисунке 5.

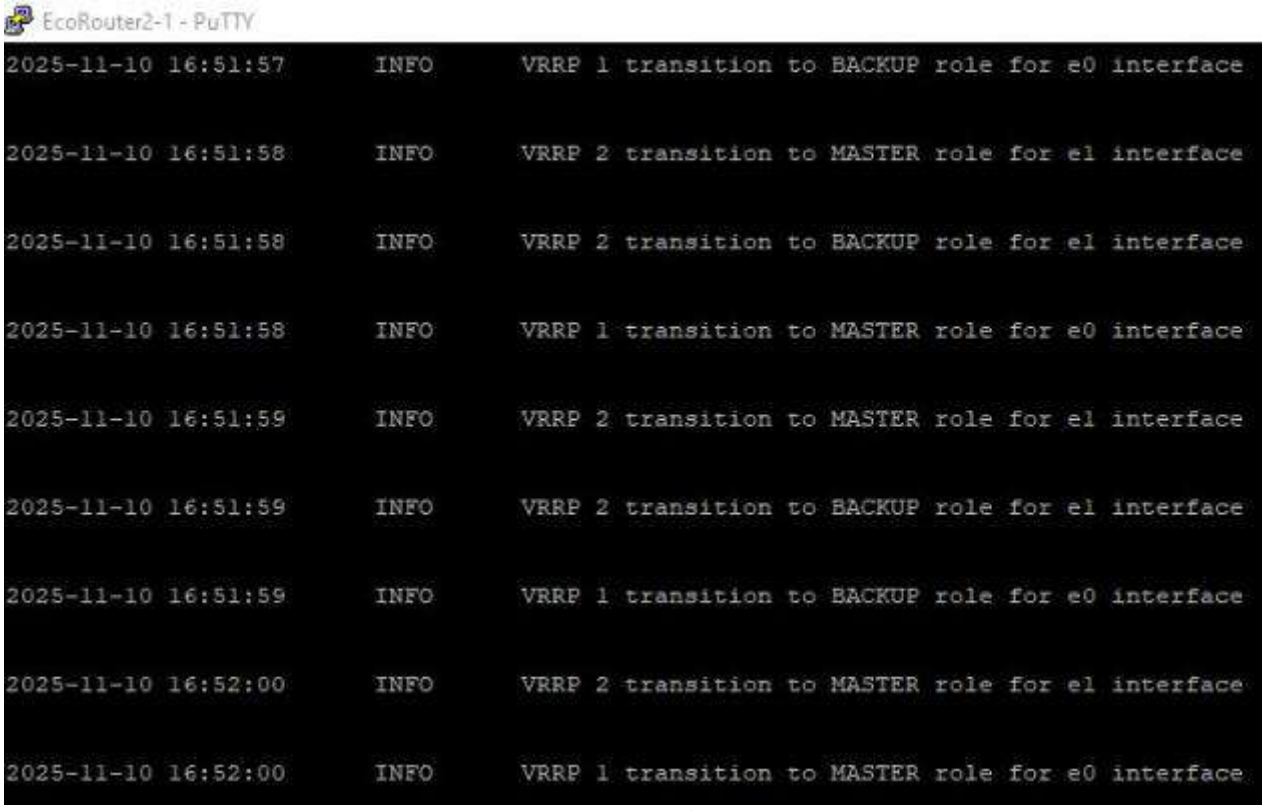


Рисунок 5 – Некорректная работа маршрутизатора

При параметре, равном 3000, интервал отправки VRRP-сообщений увеличился до ~18 секунд, а время перехода к резервному маршрутизатору возросло до одной минуты.

На втором этапе была проведена проверка на отказоусточивость протокола. Была сымитированна ситуация выхода из строя Master-маршрутизатора. Нами был отключен интерфейсы, расположенные в Master-маршрутизатор. При отключении фиксировалось:

- время, прошедшее от момента последнего полученного VRRP-адвертайзмента до появления нового Master;
- момент, когда клиенты впервые получали успешный ответ от нового виртуального шлюза;

При тестировании протокола на отказоустойчивость с параметром *advertisement-interval* равным 250 наблюдаем потерю пакетов при переходе ролей. Результат зафиксирован на рисунке 6.

```
-----------------------------------------------------------
Server listening on 5201 (test #8)
-----------------------------------------------------------
Accepted connection from 192.168.0.11, port 43536
[  5] local 192.168.100.10 port 5201 connected to 192.168.0.11 port 43540
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec   384 KBytes  3.13 Mbits/sec
[  5]   1.00-2.03   sec   384 KBytes  3.07 Mbits/sec
[  5]   2.03-3.47   sec  1.25 MBytes  7.27 Mbits/sec
[  5]   3.47-4.00   sec   384 KBytes  5.95 Mbits/sec
[  5]   4.00-5.62   sec   768 KBytes  3.89 Mbits/sec
[  5]   5.62-6.50   sec   256 KBytes  2.36 Mbits/sec
[  5]   6.50-7.00   sec   256 KBytes  4.23 Mbits/sec
[  5]   7.00-8.13   sec   512 KBytes  3.71 Mbits/sec
[  5]   8.13-9.02   sec   384 KBytes  3.56 Mbits/sec
[  5]   9.02-10.08  sec   896 KBytes  6.88 Mbits/sec
[  5]  10.08-11.33  sec  1.00 MBytes  6.72 Mbits/sec
[  5]  11.33-12.01  sec   896 KBytes  10.8 Mbits/sec
[  5]  12.01-13.01  sec   128 KBytes  1.05 Mbits/sec
[  5]  13.01-14.37  sec  0.00 Bytes  0.00 bits/sec
[  5]  14.37-15.03  sec  0.00 Bytes  0.00 bits/sec
[  5]  15.03-16.16  sec  0.00 Bytes  0.00 bits/sec
[  5]  16.16-17.13  sec  0.00 Bytes  0.00 bits/sec
[  5]  17.13-18.01  sec  0.00 Bytes  0.00 bits/sec
[  5]  18.01-19.03  sec  0.00 Bytes  0.00 bits/sec
[  5]  19.03-20.05  sec  0.00 Bytes  0.00 bits/sec
[  5]  20.05-21.10  sec  0.00 Bytes  0.00 bits/sec
[  5]  21.10-22.23  sec  0.00 Bytes  0.00 bits/sec
[  5]  22.23-23.08  sec  1.12 MBytes  11.1 Mbits/sec
[  5]  23.08-24.02  sec  2.25 MBytes  20.1 Mbits/sec
[  5]  24.02-25.05  sec  2.62 MBytes  21.4 Mbits/sec
[  5]  25.05-26.00  sec  2.75 MBytes  24.3 Mbits/sec
[  5]  26.00-27.00  sec  2.12 MBytes  17.8 Mbits/sec
[  5]  27.00-28.01  sec  3.00 MBytes  25.0 Mbits/sec
```

Рисунок 6 – Кратковременное прерывание передачи при переходе ролей по данным TCP-теста iperf3

Третьим этапом, после стабилизации работы Backup-роутера (EcoRouter2) мы восстанавливали работу EcoRouter1. Поскольку в конфигурации EcoRouter по умолчанию включён механизм **preempt-mode**, ожидается автоматическое возвращение роли Master. Проверялось:

- отсутствие флаппинга (частых переключений ролей за короткий промежуток времени);
- сохранение маршрутизации без петель;
- стабильность соединений с клиентами после возврата Master.

Были рассмотрены следующие функции: *preempt-mode* (вышедший из строя Master-маршрутизатор по возвращению в работу игнорирует тот факт, что назначенное ему значение приоритета выше, чем у Backup-маршрутизатора), *switch-back-delay* (задает время ожидания, в течение которого вернувшийся к работе маршрутизатор с более высоким приоритетом не будет анонсировать служебные сообщения).

Как видно из рисунка 7, для изменения параметра функции preempt-mode мы уже сымитировали отказ маршрутизатора – отключили VRRP процесс, в результате EcoRouter2 стал Master. После включения VRRP-процесса, EcoRouter1, несмотря на более высокий приоритет, так и остался Backup.

```
disable
preempt-mode false
enable
```

```
ecorouter(config-router)#disable
ecorouter(config-router)#preempt-mode false
ecorouter(config-router)#enable
ecorouter(config-router)#
2025-11-10 17:01:50      INFO      VRRP 1 transition to BACKUP role for e0 interface


ecorouter(config-router)#
```

Рисунок 7 – Демонстрация отключения функции preempt-mode

При включении данной функции - режим вытеснения работает. Это отображено на рисунке 8.

```
disable
preempt-mode true
enable
exit
```

```
ecorouter(config-router)#disable
ecorouter(config-router)#preempt-mode true
ecorouter(config-router)#enable
ecorouter(config-router)#
2025-11-10 17:04:25      INFO      VRRP 1 transition to BACKUP role for e0 interface


2025-11-10 17:04:26      INFO      VRRP 1 transition to MASTER role for e0 interface
```

Рисунок 8 – Включение режима вытеснения

Теперь более подробно рассмотрим функцию *switch-back-delay*. При стандартном значении вернувшийся к работе Master почти моментально начал отправлять VRRP-сообщения. Смена ролей маршрутизаторов представлена на рисунке 9.

2423…	2967.184732	192.168.100.3	224.0.0.18	VRRP	60 Announcement (v2)
2423…	2967.692405	192.168.100.3	224.0.0.18	VRRP	60 Announcement (v3)
2423…	2967.692493	192.168.100.3	224.0.0.18	VRRP	60 Announcement (v2)
2423…	2968.196400	192.168.100.3	224.0.0.18	VRRP	60 Announcement (v3)
2423…	2968.196486	192.168.100.3	224.0.0.18	VRRP	60 Announcement (v2)
2423…	2968.570614	192.168.100.2	224.0.0.18	VRRP	60 Announcement (v3)
2423…	2968.570699	192.168.100.2	224.0.0.18	VRRP	60 Announcement (v2)
2423…	2968.858169	192.168.100.2	224.0.0.18	VRRP	60 Announcement (v3)
2423…	2968.858250	192.168.100.2	224.0.0.18	VRRP	60 Announcement (v2)
2423…	2969.182022	192.168.100.2	224.0.0.18	VRRP	60 Announcement (v3)
2423…	2969.182095	192.168.100.2	224.0.0.18	VRRP	60 Announcement (v2)

Рисунок 9 – Смена ролей маршрутизаторов

В качестве эксперимента, в VRRP-id 2 был установлен параметр 5000, а на VRRP-id 1 был установлен параметр 50000. Сымитировав отказ на интерфейсах обоих маршрутизаторов, наблюдаем следующее: в VRRP ID 2 (параметр 5000) роль Master была получена примерно через 4–5 секунд, а в VRRP ID 1 (параметр 50000) — примерно через 30 секунд по временным меткам журнала. Вывод: параметр *switch-back-delay* влияет на время смены ролей, но на интервал отправки VRRP-сообщений не влияет. Началом отсчёта считалось событие подъёма интерфейса, окончанием — запись о получении роли Master; расхождение с настроенным таймером требует повторных прогонов и учитывается как ограничение точности журналирования.

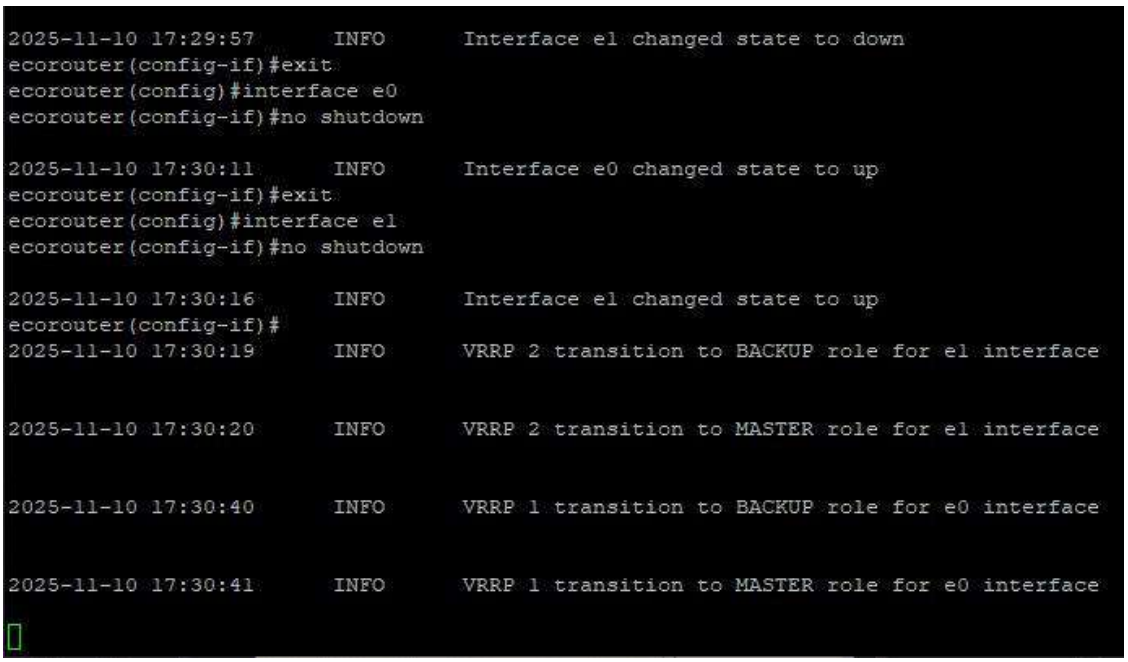


Рисунок 10 – Демонстрация работы функции *switch-back-delay*

1.3 Анализ функционала и влияния на производительность

Анализ экспериментальных данных показал, что корректная настройка ключевых параметров протокола VRRP — таких как приоритеты маршрутизаторов, интервал отправки анонсов (*advertisement-interval*), а также режимы *preempt-mode* и *switch-back-delay* — играет определяющую роль в обеспечении стабильности и эффективности работы VRRP-группы. В частности, чрезмерно короткий интервал анонсов приводит к резкому росту частоты служебных сообщений между устройствами, что значительно увеличивает объем служебного трафика в локальном сегменте. Рост частоты служебных сообщений потенциально может повлиять на пропускную способность и нагрузку CPU, однако эти показатели в работе отдельно не измерялись; на стенде наблюдался риск нестабильного переключения ролей — нестабильные, частые переключения ролей между Master и Backup, особенно в условиях высокой загрузки канала или сетевых задержек [7]. Напротив, завышенные значения таймеров, хотя и уменьшают служебный трафик, замедляют обнаружение отказа активного шлюза и тем самым увеличивают время восстановления сетевого соединения после сбоя. Зависимость времени переключения виртуального IP-адреса от значения параметра отражена на таблице 1.

***Таблица 1.* Влияние параметров VRRP на время переключения**

Параметр	Значения параметра	Время переключения (сек)
advertisement-interval	50 (по умолчанию)	1.5
advertisement-interval	250	7-8
advertisement-interval	500	15-18
advertisement-interval	3000	60
advertisement-interval	5	нестабильно (описано в пункте 1.2)
switch-back-delay	5000	4–5
switch-back-delay	50000	≈30

Таким образом, применённые методы позволили оценить функциональную корректность VRRP и время восстановления на контролируемом виртуализированном лабораторном стенде; влияние на производительность сети требует отдельной серии измерений.

Обсуждение

Полученные результаты согласуются с выводами Симбирцева Е. А. и Галяутдинова Д. А. [3], отмечавших чувствительность протокола VRRP к параметрам таймеров, и дополняют их данными о количественном влиянии advertisement-interval, preempt-mode и switch-back-delay на время восстановления сети применительно к отечественной платформе EcoRouter. В отличие от общего описания принципа работы протокола, приведённого Тихомировым И. [2], настоящее исследование показывает, что выбор параметров носит компромиссный характер: снижение advertisement-interval сокращает время реакции на отказ, но увеличивает служебный трафик и риск флаппинга, тогда как его увеличение снижает нагрузку на сеть ценой роста времени простоя.

Практическая значимость результатов заключается в использовании зависимостей таблицы 1 как предварительных лабораторных ориентиров для EcoRouterOS 3.2.6.2; перед проектированием конкретной сети необходимы повторные испытания под её нагрузкой и SLA. Ограничением исследования является использование изолированного лабораторного стенда с ограниченным числом узлов и отсутствие тестирования при одновременном отказе нескольких маршрутизаторов или при высокой фоновой нагрузке на канал; дальнейшая работа предполагает проверку полученных зависимостей в условиях, приближенных к промышленной эксплуатации, а также исследование поведения VRRP при интеграции с механизмами балансировки нагрузки.

Заключение

В результате выполненного исследования подтверждена работоспособность протокола VRRP в реализации платформы EcoRouter и установлено его существенное влияние на устойчивость сетевой инфраструктуры к отказам маршрутизаторов. Показано, что время переключения ролей Master/Backup зависит от значения параметра advertisement-interval: при значении по умолчанию (50) оно составляет около 1,5 с, при увеличении до 3000 — до 60 с, а при чрезмерном занижении параметра (5) протокол переходит в нестабильный режим флаппинга. Установлено, что параметр switch-back-delay управляет временем возврата роли Master после восстановления основного маршрутизатора, не влияя при этом на интервал отправки VRRP-анонсов, а отключение режима preempt-mode позволяет сохранить роль Master за резервным маршрутизатором даже при более высоком приоритете основного.

На основании полученных данных сформулированы предварительные лабораторные ориентиры для данного стенда и версии EcoRouterOS: для сценариев, критичных к времени простоя (видеонаблюдение, обработка транзакций в реальном времени), целесообразно использовать значения advertisement-interval в диапазоне 50–250 в сочетании с включённым preempt-mode; для сетей с ограниченной пропускной способностью канала управления допустимо увеличение интервала анонсов при условии, что возросшее время восстановления укладывается в требования соглашения об уровне обслуживания (SLA). Таким образом, реализация VRRP в EcoRouter обеспечивает заявленную отказоустойчивость при корректном выборе конфигурационных параметров, что подтверждает выдвинутую гипотезу исследования и может быть использовано сетевыми инженерами при проектировании отказоустойчивых сетей на отечественном оборудовании.

Библиографический список

1. Eckert, T. Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6 [Электронный ресурс] : RFC 9568 / T. Eckert, E. Zhang, A. Lindem. – IETF, 2024. – URL: https://www.rfc-editor.org/rfc/rfc9568.html (дата обращения: 31.12.2025).
2. Тихомиров И. Принцип работы протокола VRRP // Хабр. — 2019. — 19 мая. — URL: https://habr.com/ru/articles/452490 (дата обращения: 07.11.2025).
3. Симбирцев Е. А., Галяутдинов Д. А. Настройка и тестирование протокола VRRP. Вопросы безопасности // Вестник науки. — 2025. — № 6 (87). — С. 1573–1590. — URL: https://cyberleninka.ru/article/n/nastroyka-i-testirovanie-protokola-vrrp-voprosy-bezopasnosti (дата обращения: 07.11.2025).
4. Руководство пользователя EcoRouter: VRRP [Электронный ресурс] // Официальная документация EcoRouter. – 2023. – 12 октября. – URL:

https://docs.ecorouter.ru/%D0%A0%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE/12-VRRP (дата обращения: 07.11.2025).
5. Уймин, А. Г. Обзор систем моделирования: анализ эффективности на примере чемпионата AtomSkills-2023 / А. Г. Уймин, В. С. Греков // Автоматизация и информатизация ТЭК. – 2023. – № 11(604). – С. 25-34. – DOI 10.33285/2782-604X-2023-11(604)-25-34. – EDN QYQRCO.
6. Таненбаум, Эндрю С. Компьютерные сети / Э. Таненбаум,Д. Уэзеролл ; пер. с англ. под ред. В. Н. Четверикова. — 6-е изд. — Санкт-Петербург : Питер, 2023. — 992 с. : ил. — ISBN 978-5-4461-2092-3.
7. Надаш С. Протокол резервирования виртуального маршрутизатора (VRRP) версии 3 для IPv4 и IPv6 [Электронный ресурс]: RFC 5798. — IETF, 2010. — URL: https://www.rfc-editor.org/rfc/rfc5798.html (дата обращения: 31.12.2025).

References

1. *Eckert, T., Zhang, E., & Lindem, A. (2024). Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6 (RFC 9568). IETF. https://www.rfc-editor.org/rfc/rfc9568.html*
2. Tikhomirov, I. (2019). *Principles of VRRP protocol operation*. Habr. Retrieved November 7, 2025, from https://habr.com/ru/articles/452490
3. Simbirtsev, E. A., & Galyautdinov, D. A. (2025). *Configuration and testing of the VRRP protocol. Security issues*. Bulletin of Science, 6(87), 1573–1590. Retrieved November 7, 2025, from https://cyberleninka.ru/article/n/nastroyka-i-testirovanie-protokola-vrrp-voprosy-bezopasnosti
4. EcoRouter User Guide: VRRP. (2023). Official EcoRouter Documentation. Retrieved November 7, 2025, from https://docs.ecorouter.ru/%D0%A0%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE/12-VRRP
5. Uymin, A. G., & Grekov, V. S. (2023). *Review of modeling systems: Efficiency analysis using the AtomSkills-2023 championship as an example*. Automation and Informatization of the Fuel and Energy Complex, 11(604), 25–34. https://doi.org/10.33285/2782-604X-2023-11(604)-25-34
6. Tanenbaum, A. S., & Wetherall, D. J. (2023). *Computer networks* (6th ed.). Piter.
7. Nadas, S. (2010). Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6 (RFC 5798). IETF. https://www.rfc-editor.org/rfc/rfc5798.html

Информация об авторах

Чаков Иван Александрович — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: chakov.vanya@yandex.ru

Евтюхов Михаил Сергеевич — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: eutyuhovmihail@yandex.ru

Павловский Владимир Владимирович — научный руководитель, преподаватель кафедры «Безопасность информационных технологий», ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: pavlovskiy.v@gubkin.ru

INVESTIGATION OF VRRP FUNCTIONALITY AND RECOVERY TIME BASED ON ECOROUTER SOLUTIONS

I. A. Chakov[1], M. S. Evtyukhov[1], V. V. Pavlovskiy[1]

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

Abstract. The study addresses the growing use of the domestic EcoRouter platform amid import substitution of network infrastructure, while its implementation of VRRP (Virtual Router Redundancy Protocol) remains insufficiently studied in terms of performance and stability under controlled virtualized laboratory conditions. The purpose is to evaluate the functional capabilities and VRRP recovery behavior in EcoRouter and to develop practical recommendations for configuring fault-tolerance parameters.

The study was conducted in a virtualized lab environment based on GNS3 and VirtualBox using EcoRouterOS routers. The topology included two routers, two switches, and four client hosts forming two network segments, each with

an independent VRRP group. The iperf3 and Wireshark tools evaluated Master/Backup switchover time, temporary transfer interruption, and advertisement interval while varying the advertisement-interval, preempt-mode, and switch-back-delay parameters.

With the default advertisement-interval value (50), the switchover time is about 1.5 s, while increasing the parameter to 3000 raises it to 60 s; excessively small values (5) cause unstable role “flapping.” The switch-back-delay parameter affects how quickly the Master role returns but not the advertisement interval, while disabling preempt-mode blocks automatic restoration of the priority role.

The results confirm that the VRRP implementation in EcoRouter provides the declared fault tolerance when parameters are configured correctly, enabling practical recommendations for network engineers on balancing failure-response speed against stability. The findings can be applied to designing domestic fault-tolerant network infrastructures and in teaching networking courses.

Keywords: VRRP protocol, network fault tolerance, EcoRouter, router redundancy, failover time, network performance, import substitution.

Information about the authors

Chakov Ivan Aleksandrovich — student, National University of Oil and Gas «Gubkin University», Moscow, Russian Federation, e-mail: chakov.vanya@yandex.ru

Evtyukhov Mikhail Sergeevich — student, National University of Oil and Gas «Gubkin University», Moscow, Russian Federation, e-mail: eutyuhovmihail@yandex.ru

Pavlovskiy Vladimir Vladimirovich — research supervisor, lecturer at the Department of Information Technology Security, National University of Oil and Gas «Gubkin University», Moscow, Russian Federation, e-mail: pavlovskiy.v@gubkin.ru

ПРОФЕССИОНАЛИТЕТ, 2025, № 4 (008)

УДК *004.7:004.056*

И.К. Селиванов [1], Д.А. Леденёв [1]

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

ТЕСТИРОВАНИЕ ШТАТНОЙ ЗАЩИТЫ СЕТЕВЫХ УСТРОЙСТВ L2+ СРЕДСТВАМИ NMAP ПРИ АКТИВНОМ КОНСОЛЬНОМ ДОСТУПЕ

Аннотация. В статье рассматривается вопрос оценки эффективности штатных механизмов защиты сетевых устройств уровня L2+ при проведении сетевого сканирования средствами утилиты nmap. Актуальность работы обусловлена широким использованием управляемых сетевых устройств в корпоративных и учебных сетях, а также необходимостью минимизации поверхности атаки на уровне интерфейсов управления без применения специализированных средств защиты. Особое внимание уделяется анализу management-плоскости устройств в условиях активного консольного доступа администратора.

Целью исследования является практическая оценка сетевой доступности сервисов управления и состояния сетевых портов в начальной конфигурации сетевых устройств различных производителей. В качестве объектов исследования использовались управляемый коммутатор MikroTik CRS326-24G-2S+ и маршрутизаторы Cisco серий 1900 и 800. Экспериментальная методика основана на применении активных методов сетевой разведки с использованием TCP SYN-сканирования, а также сопоставлении результатов сетевого сканирования с фактической конфигурацией устройств, полученной через консольный доступ.

В ходе исследования установлено, что в штатной конфигурации исследуемые сетевые устройства остаются обнаруживаемыми на уровне IP, однако не предоставляют открытых сервисов удалённого управления. Управляющие порты находятся в состояниях filtered или closed, что свидетельствует о функционировании встроенных механизмов фильтрации и ограничения доступа. Показано, что наличие активного консольного доступа не оказывает прямого влияния на результаты сетевого сканирования.

Полученные результаты подтверждают эффективность базовой конфигурации сетевых устройств с точки зрения снижения сетевой поверхности атаки и могут быть использованы при анализе защищённости сетевой инфраструктуры, а также в учебной практике.

Ключевые слова: сетевые устройства L2+, информационная безопасность, management-плоскость, сетевое сканирование, nmap, консольный доступ.

Введение

В современных корпоративных и учебных сетях управляемые сетевые устройства уровня L2+ играют ключевую роль в обеспечении связности, отказоустойчивости и безопасности информационной инфраструктуры. Коммутаторы и маршрутизаторы данного класса обладают развитой системой управления, включающей как консольные, так и сетевые интерфейсы администрирования, что делает их потенциальной целью для атак на уровне management-плоскости. В условиях роста числа инцидентов, связанных с компрометацией сетевого оборудования, вопросы оценки защищённости таких устройств приобретают особую значимость [1, 2].

Практика эксплуатации сетевой инфраструктуры показывает, что уязвимости сетевых устройств зачастую обусловлены не ошибками реализации протоколов, а особенностями конфигурации сервисов управления и механизмов фильтрации доступа. Даже при отсутствии явных уязвимостей некорректные или избыточные настройки могут приводить к расширению сетевой поверхности атаки. В этой связи актуальной задачей является анализ поведения сетевых устройств в их исходной, штатной конфигурации, без применения дополнительных средств защиты и специализированных систем мониторинга.

Одним из наиболее распространённых инструментов первичной сетевой разведки является утилита nmap, широко используемая как в практическом аудите безопасности, так и в учебных лабораторных работах [3, 4]. Результаты сетевого сканирования позволяют выявлять доступные сетевые сервисы, определять состояние портов и косвенно оценивать эффективность встроенных механизмов фильтрации трафика. При этом корректная интерпретация результатов сканирования требует учёта архитектурных особенностей сетевых устройств и способов их администрирования.

Отдельного внимания заслуживает вопрос влияния консольного доступа администратора на результаты сетевого сканирования. Консольное подключение традиционно рассматривается

как изолированный и наиболее защищённый канал управления, однако на практике его наличие нередко вызывает вопросы о возможной взаимосвязи с сетевой доступностью сервисов управления. Анализ данного аспекта позволяет уточнить границы применимости результатов сетевой разведки и избежать ошибочных выводов при оценке защищённости сетевых устройств [5].

Таким образом, рассмотрение штатных механизмов защиты сетевых устройств уровня L2+ в контексте сетевого сканирования и консольного администрирования представляет практический и методический интерес и способствует формированию более корректного подхода к анализу безопасности сетевой инфраструктуры.

Материалы и методы

Настоящее исследование носит прикладной экспериментальный характер и направлено на оценку эффективности штатных механизмов защиты сетевых устройств уровня L2+ в условиях сетевого сканирования. Методологической основой работы является анализ management-плоскости сетевых устройств с использованием активных методов сетевой разведки и последующего сопоставления результатов сканирования с фактической конфигурацией оборудования.

В качестве объектов исследования использовались управляемый коммутатор MikroTik CRS326-24G-2S+ и маршрутизаторы Cisco серий 1900 и 800, относящиеся к классу сетевых устройств уровня L2+ и широко применяемые в корпоративных и учебных сетях. Выбор оборудования обусловлен его типичностью, наличием встроенных механизмов управления и фильтрации доступа, а также возможностью администрирования как через консольный, так и через сетевые интерфейсы.

Для проведения эксперимента применялась утилита nmap, которая является одним из наиболее распространённых инструментов первичной сетевой разведки и используется для выявления доступных сетевых сервисов, определения состояния TCP-портов и анализа особенностей фильтрации сетевого трафика. В рамках работы использовалось TCP SYN-сканирование, позволяющее определить состояния портов open, closed и filtered на основе реакции узла на инициирование соединения. Состояние open интерпретируется как наличие активного сетевого сервиса, closed — как явный отказ в установлении соединения, filtered — как признак функционирования механизмов фильтрации или ограничения доступа [6].

Администрирование исследуемых устройств осуществлялось исключительно через консольный доступ, который не использует сетевой стек TCP/IP и не формирует дополнительных сетевых сервисов. Для обеспечения возможности проведения сетевого сканирования на всех устройствах применялась минимальная начальная сетевая конфигурация: на интерфейсы управления вручную назначались IP-адреса, необходимые для сетевой доступности оборудования. Указанные действия носили исключительно технический характер и не сопровождались включением сервисов удалённого управления, изменением политик фильтрации или настройкой списков контроля доступа. Таким образом, условия эксперимента соответствовали штатной конфигурации устройств за исключением минимально необходимой IP-адресации.

Сетевое сканирование выполнялось с тестовой станции под управлением операционной системы Windows. При проведении эксперимента использовались режимы сканирования, не требующие предварительного ICMP-обнаружения узла, что позволило учитывать возможную фильтрацию эхо-запросов. Полученные результаты фиксировались и сопоставлялись с данными о состоянии интерфейсов и конфигурации устройств, полученными через консольный доступ с использованием стандартных команд операционных систем сетевого оборудования.

Обработка и интерпретация результатов осуществлялись методом качественного анализа. Основное внимание уделялось сопоставлению состояний сетевых портов, выявленных средствами nmap, с фактическим состоянием сервисов управления и механизмов фильтрации доступа. Статистические методы обработки данных в рамках исследования не применялись, поскольку целью работы являлась оценка защитных свойств конкретных конфигураций сетевых устройств, а не получение обобщённых количественных характеристик.

Результаты исследования

Устройство 1. MikroTik CRS326-24G-2S+

На первом этапе эксперимента в качестве исследуемого сетевого устройства использовался управляемый коммутатор MikroTik CRS326-24G-2S+. Перед началом тестирования устройство было сброшено к заводским настройкам, что позволило анализировать его начальную, штатную конфигурацию без дополнительных изменений со стороны администратора.

Подключение к устройству осуществлялось через консольный доступ с использованием клиента PuTTY. Полученная информация о системе свидетельствует, что коммутатор функционирует под управлением операционной системы RouterOS версии 6.48.4 (stable), установленной производителем. Аппаратная платформа основана на архитектуре ARMv7 с одноядерным процессором с тактовой частотой 800 МГц и объёмом оперативной памяти 512 МБ, что соответствует типовым характеристикам устройств данного класса (см. Рисунок 1).

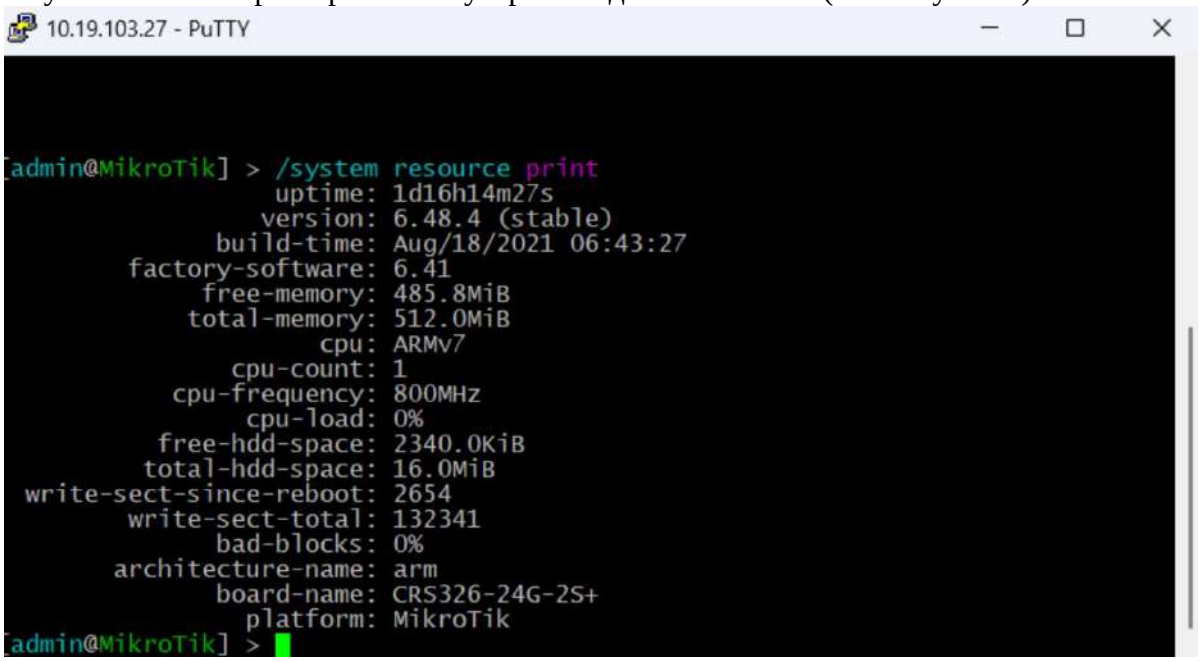


Рис. 1. Коммутатор MikroTik

На втором этапе эксперимента была зафиксирована начальная сетевая конфигурация коммутатора MikroTik после сброса к заводским настройкам и выполнено внешнее сетевое сканирование с использованием утилиты nmap.

С консольной стороны устройства показан вывод команды /interface print, из которого следует, что все физические интерфейсы Ethernet (ether1–ether24), а также два SFP+-интерфейса находятся в отключённом состоянии и объединены в логический bridge с конфигурацией по умолчанию (defconf). Это подтверждает отсутствие пользовательской настройки портов и использование стандартной схемы коммутации.

Вывод команды /ip address print демонстрирует, что на устройстве присутствует IP-адрес 192.168.88.1/24, назначенный интерфейсу bridge в рамках заводской конфигурации RouterOS. Дополнительно отображаются адреса, привязанные к отдельным интерфейсам, что указывает на ранее существовавшие подключения, однако ключевым для эксперимента является именно адрес управления по умолчанию.

С внешней стороны, со стороны тестовой станции под управлением Windows, выполнено сетевое сканирование устройства с помощью nmap. Результаты ping-сканирования подтверждают доступность узла по IP-адресу 192.168.88.1. При выполнении TCP SYN-сканирования (-sS -Pn) было выявлено, что большинство TCP-портов находятся в состоянии фильтрации или закрыты, а стандартные сервисы, такие как SMTP, DNS, MSRPC и Microsoft-DS, не принимают входящие соединения (см. Рисунок 2).

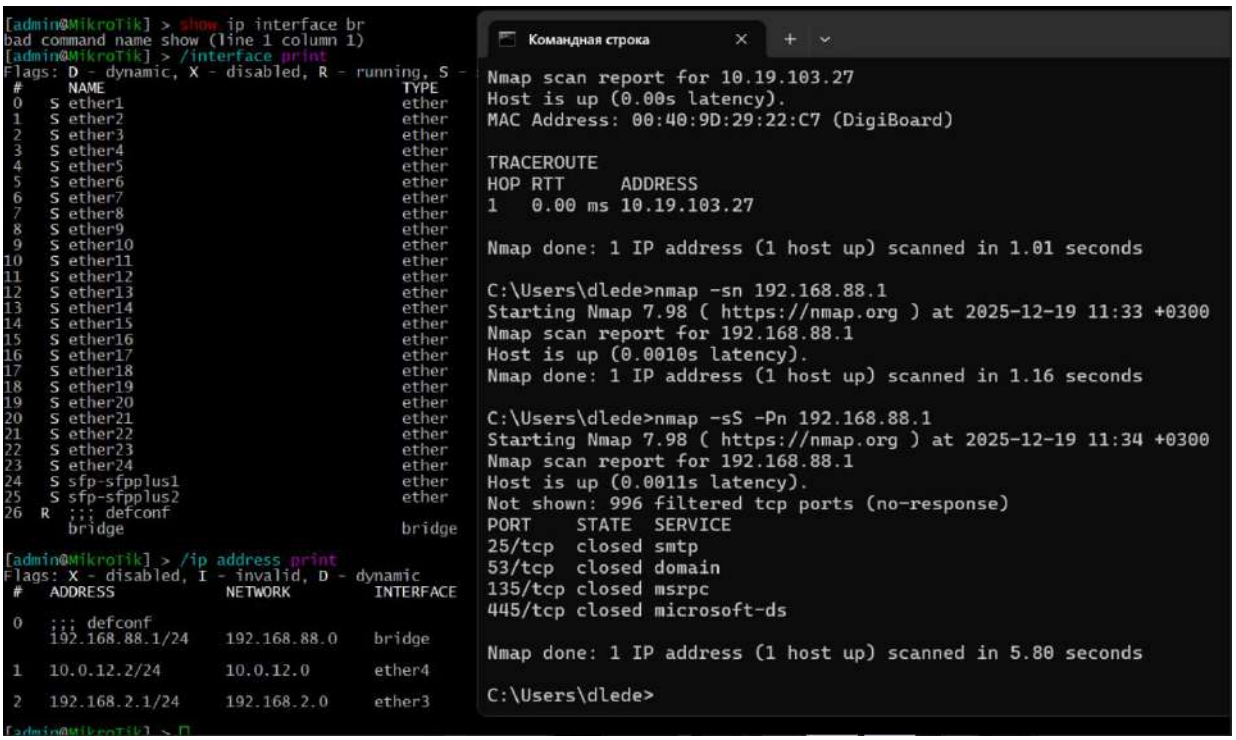


Рис. 2. Сканирование TCP портов

При выборочном сканировании портов управления и прикладных сервисов (22/SSH, 23/Telnet, 80/HTTP, 443/HTTPS, 8291/Winbox) все они определяются как filtered, то есть пакеты блокируются и не получают ответа (см. Рисунок 3).

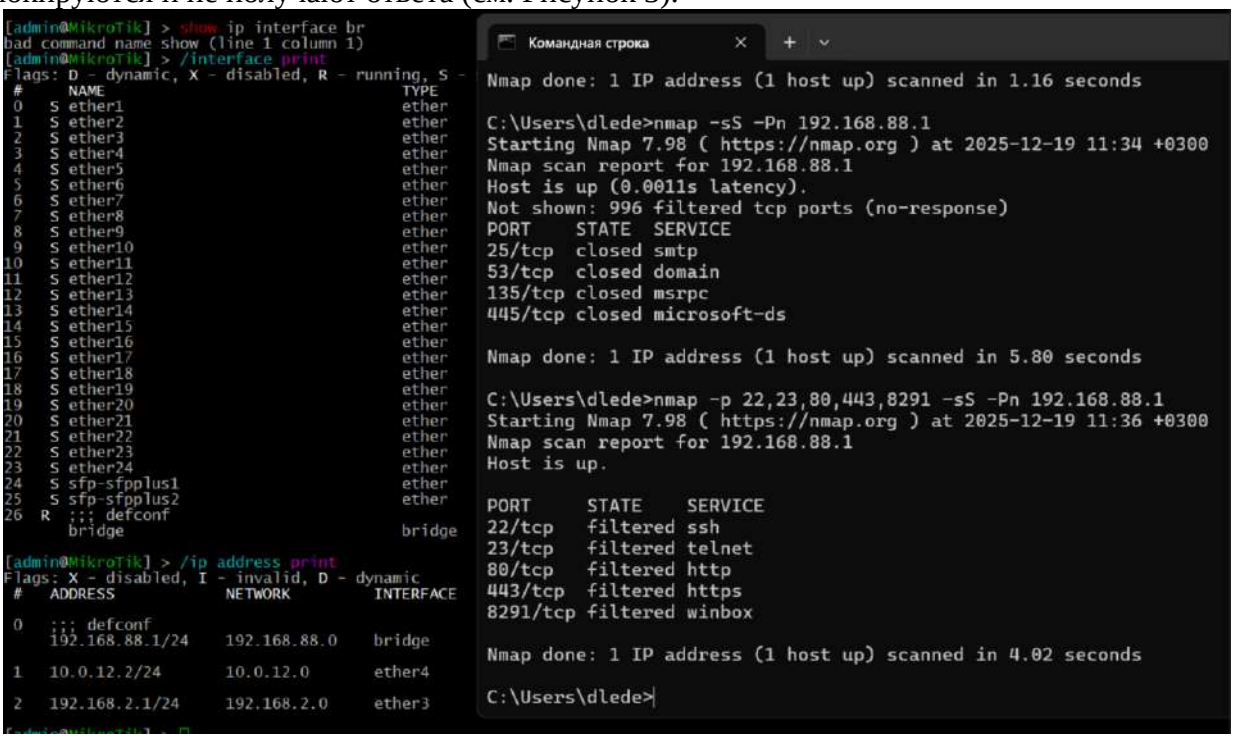


Рис 3. Выборочное сканирование TCP портов

Полученные результаты свидетельствуют о том, что даже при активном сетевом присутствии и назначенном IP-адресе управления MikroTik в начальной конфигурации применяет штатные механизмы фильтрации трафика. Это ограничивает доступ к консольным и веб-сервисам с неавторизованных узлов и существенно снижает поверхность атаки на уровне транспортных протоколов [6].

Устройство 2. Cisco 1941

Устройство функционирует под управлением Cisco IOS Software линейки C1900 (образ C1900-UNIVERSALK9-M), версия операционной системы — 15.7(3)M6, что указывает на

использование универсального образа с поддержкой механизмов маршрутизации и встроенных криптографических функций. Согласно выводу команды show version, маршрутизатор был загружен штатным образом из flash-памяти, без аварийных перезапусков.

Также отображается информация о версии загрузчика (ROM Bootstrap), дате компиляции прошивки и причине последней перезагрузки, которая была выполнена вручную командой перезапуска. Это косвенно подтверждает, что перед началом исследования устройство находилось в контролируемом состоянии и не испытывало сбоев. Наличие стандартного консольного приглашения Router> и переход в привилегированный режим (enable) указывает на активный консольный доступ, используемый для администрирования и мониторинга параметров устройства в рамках эксперимента (см. Рисунок 4).

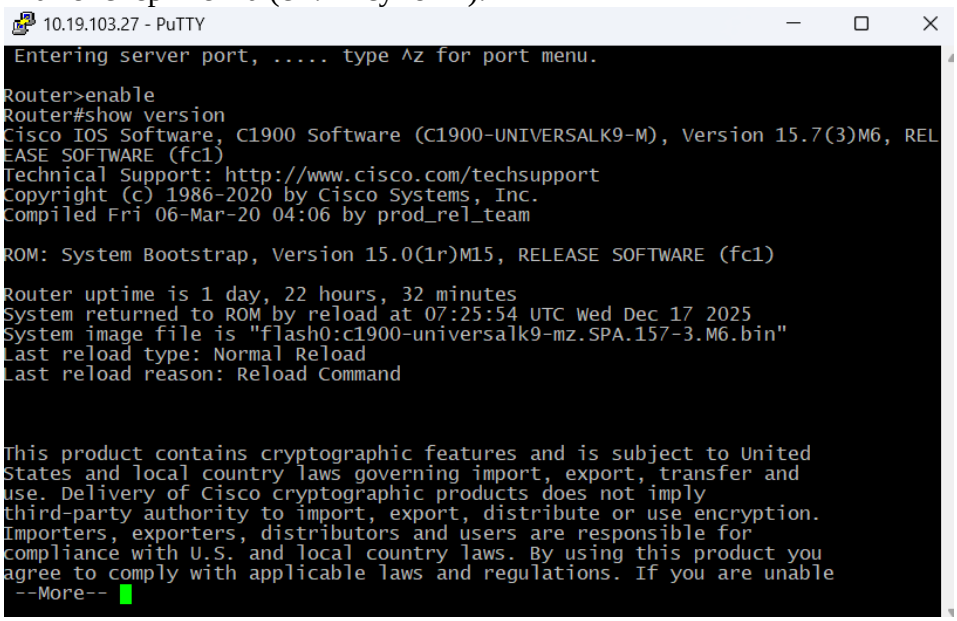


Рис. 4. Маршрутизатор Cisco 1941

Первоначально была выполнена проверка доступности узла (nmap -sn), подтвердившая, что хост находится в сети и отвечает на запросы. Далее осуществлён TCP SYN-скан (nmap -sS -Pn), в ходе которого выявлено, что большинство портов маршрутизатора находятся в состоянии filtered или closed, а ответы от сервисов отсутствуют (см. Рисунок 5). Это свидетельствует о работе встроенных механизмов фильтрации и ограничений доступа, реализованных в штатной конфигурации устройства.

Следует отметить, что результаты сетевого сканирования и вывод команды show ip interface brief, представленные на рисунке 5, были получены в разные моменты времени. Во время проведения сетевого сканирования интерфейс GigabitEthernet0/1 находился во включённом состоянии, что обеспечивало сетевую доступность узла 192.168.1.1. Впоследствии интерфейс был переведён в состояние administratively down, что зафиксировано на приведённом скриншоте. Данное обстоятельство не влияет на интерпретацию результатов эксперимента, однако уточняется для устранения логического противоречия.

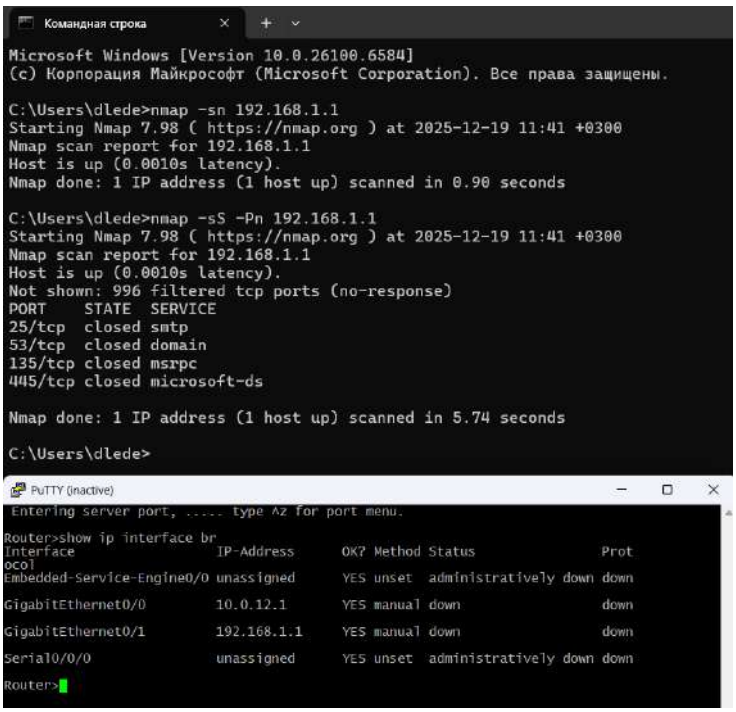


Рис. 5. Сканирование TCP портов Cisco 1941

На следующем этапе эксперимента зафиксирован результат целенаправленного сканирования управляющих портов маршрутизатора Cisco средствами nmap после сброса конфигурации и при наличии только консольного доступа. Сканирование выполнялось по наиболее распространённым портам удалённого управления и администрирования: SSH (22), Telnet (23), HTTP (80), HTTPS (443), а также Winbox (8291). Результаты сканирования показывают, что узел 192.168.1.1 доступен на сетевом уровне (Host is up), однако все проверяемые порты находятся в состоянии filtered (см. Рисунок 6). Это означает, что маршрутизатор не отвечает на SYN-запросы, направленные на данные порты, и не возвращает явных сообщений о закрытии соединения (RST). Такое поведение характерно для работы встроенных механизмов фильтрации трафика, в том числе ACL и базовых политик безопасности Cisco IOS.

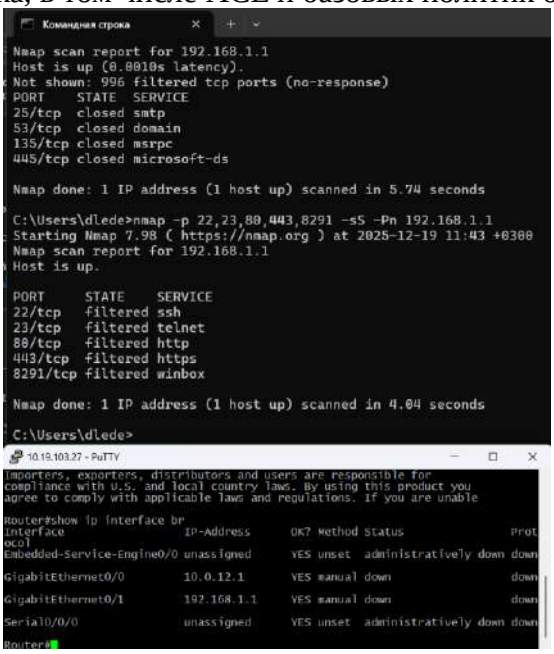


Рис. 6. Выборочное сканирование TCP портов Cisco 1941

Полученные результаты подтверждают, что в начальной (базовой) конфигурации после сброса настроек маршрутизатор Cisco реализует консервативную модель безопасности: сетевое устройство обнаруживается в сети, но все потенциально опасные сервисы управления скрыты или отфильтрованы. Это существенно снижает поверхность атаки и демонстрирует

эффективность штатных средств защиты при отсутствии явного включения веб- и сетевого управления [6].

Устройство 3. Cisco 881

На данном этапе эксперимента представлен вывод информации о втором исследуемом сетевом устройстве — маршрутизаторе Cisco серии 800, полученный по консольному доступу после сброса конфигурации к исходному состоянию. На рисунке 7 представлены сведения о программно-аппаратной платформе устройства. Установлена операционная система Cisco IOS версии 15.8(3)M2, устройство относится к линейке Cisco C800, время непрерывной работы составляет около 15 часов, а последняя перезагрузка была выполнена при включении питания. Тип загрузки указан как штатный (Normal Reload), что также подтверждает отсутствие аварийных состояний (см. Рисунок 7).

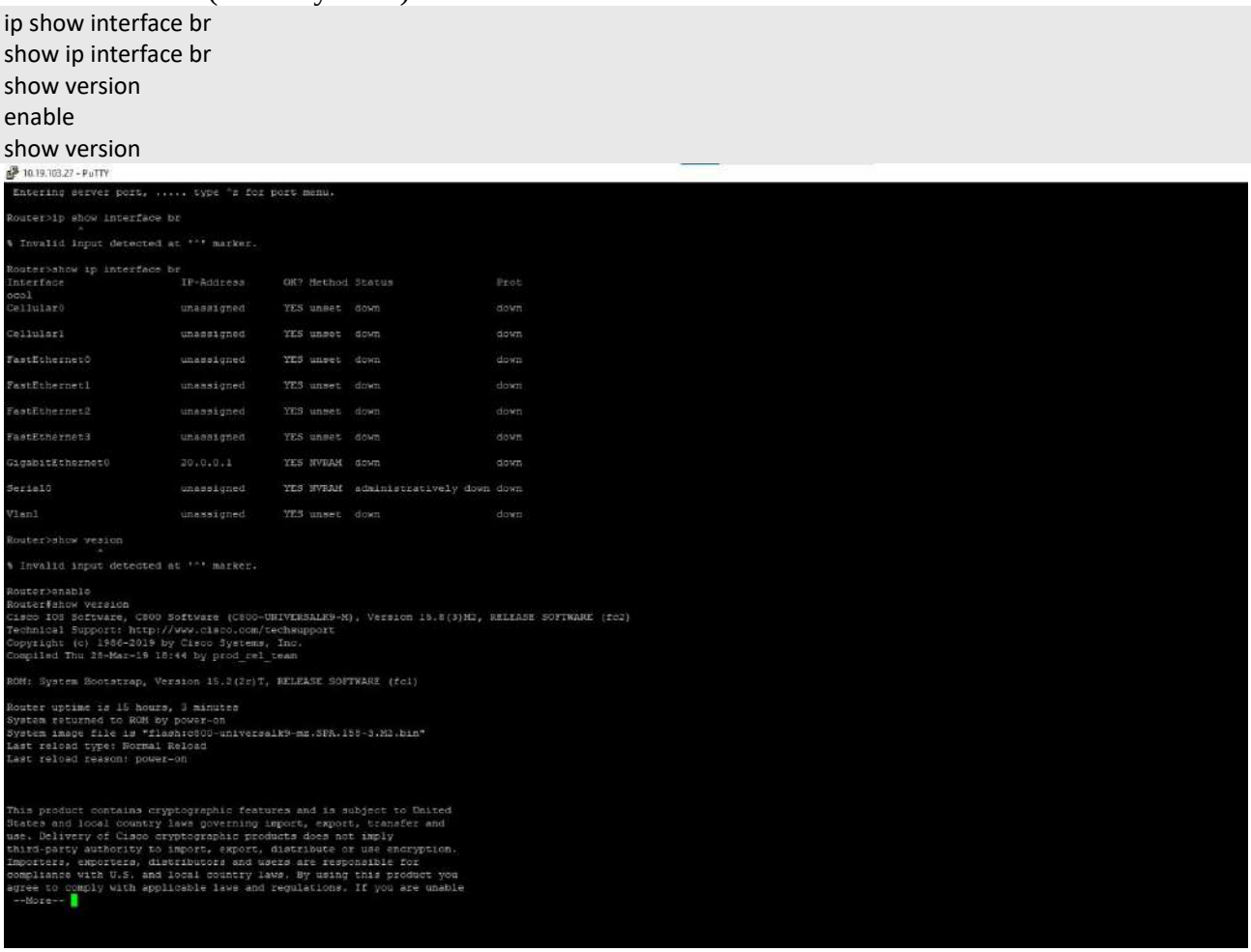


Рис.7. Маршрутизатор Cisco 881

На следующем этапе эксперимента зафиксирован результат сканирования второго исследуемого устройства — маршрутизатора Cisco C800 — с использованием графического интерфейса Zenmap (GUI-оболочка nmap). Сканирование выполнялось по IP-адресу 20.0.0.1, который ранее был назначен интерфейсу GigabitEthernet0 маршрутизатора.

Данный режим предполагает выполнение TCP SYN-сканирования без предварительной проверки доступности хоста ICMP-запросами, что позволяет оценить реакцию устройства на сетевые попытки установления соединений. Результаты сканирования подтверждают, что хост доступен на сетевом уровне (Host is up), при этом задержка отклика минимальна, что указывает на прямую сетевую связность без промежуточных маршрутизаторов [6].

В выводе сканирования указано, что подавляющее большинство TCP-портов (996 из 1000) находятся в состоянии filtered и не возвращают ответа на запросы. Среди отображённых портов зафиксированы порты 25 (SMTP), 53 (DNS), 135 (MSRPC) и 445 (Microsoft-DS), все они находятся в состоянии closed, то есть маршрутизатор явно отклоняет попытки соединения [6,

7]. Открытых портов и активных сетевых сервисов в результате сканирования не выявлено (см. Рисунок 8 и Рисунок 9).

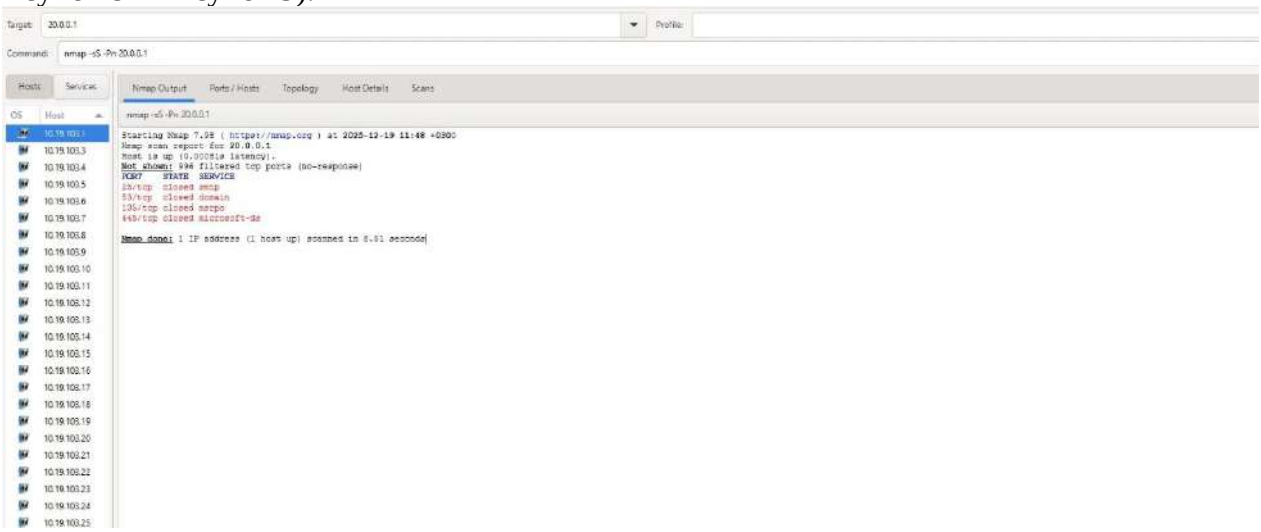


Рис. 8. Сканирование TCP портов Cisco 881

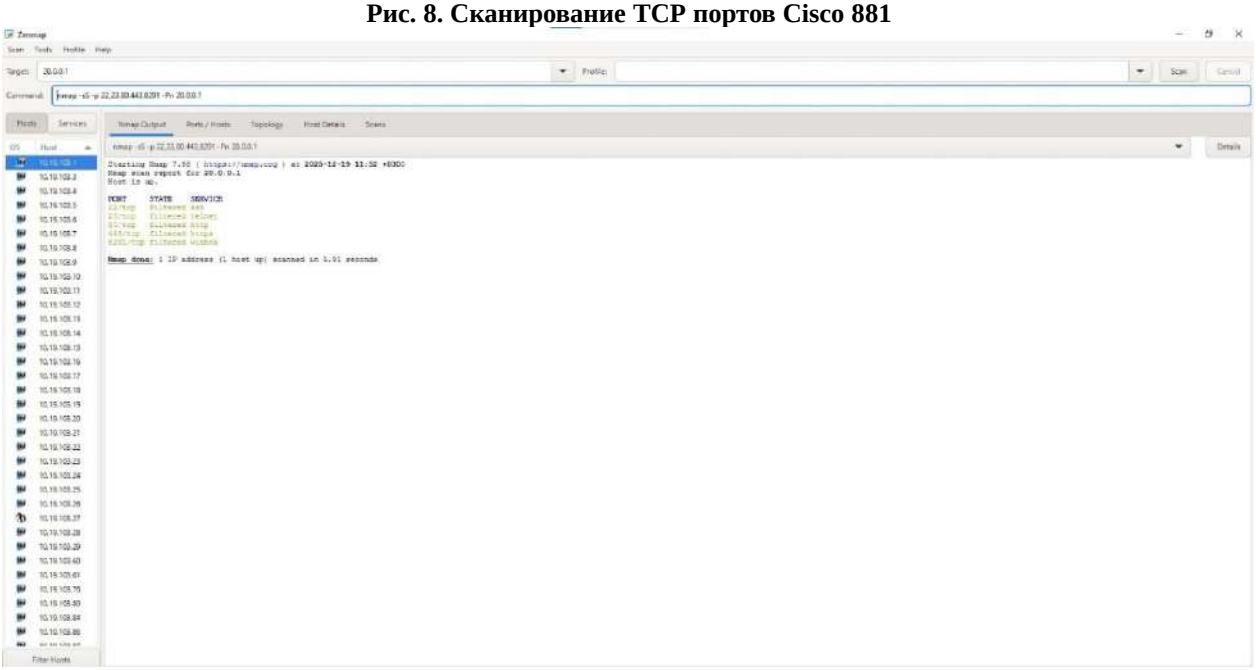


Рис. 9. Выборочное сканирование TCP портов Cisco 881

Аналитический сценарий изменения поверхности атаки при включении сервисов управления

В рамках настоящего исследования экспериментальная часть была ограничена анализом штатной конфигурации сетевых устройств без явного включения сервисов удалённого управления. Вместе с тем, для повышения интерпретационной ценности полученных результатов целесообразно рассмотреть аналитический сценарий изменения поверхности атаки при активации сервисов управления.

При включении на сетевом устройстве сервиса удалённого администрирования, такого как SSH или Web-интерфейс (HTTP/HTTPS), соответствующие TCP-порты переходят из состояния filtered или closed в состояние open, что фиксируется средствами сетевого сканирования nmap. В этом случае утилита nmap получает явный ответ на SYN-запросы, что указывает на наличие активного сервиса и формирует дополнительную поверхность атаки [6].

Данный эффект подробно описан в документации nmap и в работах, посвящённых анализу сетевой безопасности, где подчёркивается, что состояние open является индикатором доступного сетевого сервиса, потенциально подверженного атаке при наличии уязвимостей или слабой аутентификации. Таким образом, включение сервисов управления принципиально изменяет результаты сетевого сканирования и требует применения дополнительных мер защиты,

таких как списки контроля доступа, ограничение источников подключения и использование защищённых протоколов [6, 7].

Рассмотренный аналитический сценарий позволяет наглядно интерпретировать результаты, полученные в рамках эксперимента, и подчёркивает, что эффективность штатной защиты в конфигурации по умолчанию обусловлена именно отсутствием активных сервисов управления, а не влиянием консольного доступа администратора.

Обсуждение

Полученные в ходе исследования результаты подтверждают, что штатная конфигурация сетевых устройств уровня L2+ ориентирована на минимизацию сетевой поверхности атаки за счёт отсутствия активных сервисов удалённого управления. Обнаружение устройств на уровне IP при одновременном отсутствии открытых управляющих портов соответствует консервативной модели безопасности, реализуемой производителями сетевого оборудования.

Важно отметить, что результаты сетевого сканирования, полученные средствами nmap, требуют корректной интерпретации с учётом архитектурных особенностей сетевых устройств и способов их администрирования. В частности, наличие активного консольного доступа не коррелирует с сетевой доступностью сервисов управления и не влияет на состояние TCP-портов, выявляемых в процессе сканирования. Это подчёркивает необходимость сопоставления данных сетевой разведки с фактической конфигурацией устройств, полученной через консольный доступ.

Ограничением проведённого исследования является отсутствие экспериментальной проверки сценариев с включением сервисов удалённого управления. Вместе с тем, аналитический анализ показывает, что активация таких сервисов приводит к принципиальному изменению результатов сетевого сканирования и требует применения дополнительных мер защиты. Указанные сценарии представляют практический интерес и могут быть реализованы в рамках дальнейших исследований.

Заключение

В рамках данной работы проведено экспериментальное исследование штатных механизмов защиты сетевых устройств уровня L2+ с использованием утилиты nmap. Анализ выполнялся на реальном сетевом оборудовании различных производителей при активном консольном доступе и минимальной начальной конфигурации, необходимой для обеспечения сетевой доступности устройств.

В ходе исследования установлено, что в штатной конфигурации сетевые устройства обнаруживаются на уровне IP, однако не предоставляют открытых сервисов удалённого управления. Управляющие порты находятся в состояниях filtered или closed, что свидетельствует о функционировании встроенных механизмов фильтрации и ограничения доступа. Показано, что наличие активного консольного доступа не оказывает прямого влияния на результаты сетевого сканирования.

Полученные результаты подтверждают эффективность базовой конфигурации сетевых устройств с точки зрения снижения сетевой поверхности атаки и подчёркивают значимость корректной интерпретации данных сетевой разведки. Материалы работы могут быть использованы при анализе защищённости сетевой инфраструктуры, а также в образовательной практике при изучении вопросов сетевой безопасности и администрирования.

Библиографический список

1. Компьютерные сети. Принципы, технологии, протоколы : юбилейное издание, доп. и испр. — Санкт-Петербург : Питер, 2024. — 1008 с.
2. Anderson, R. Security Engineering: A Guide to Building Dependable Distributed Systems / R. Anderson. — 3rd ed. — Indianapolis : Wiley, 2020. — 1182 p.
3. Использование Nmap для обнаружения сетевых служб и идентификации сервисов [Электронный ресурс]. — URL: https://labex.io/ru/tutorials/nmap-how-to-use-nmap-for-network-discovery-and-service-identification-in-cybersecurity-415099 (дата обращения: 19.11.2025).

4. Уймин, А. Г. Сетевое и системное администрирование. Демонстрационный экзамен КОД 1.1 : учебно-методическое пособие для СПО / А. Г. Уймин. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 480 с. — ISBN 978-5-8114-9255-8.
5. Denzler, P. An iterative and toolchain-based approach to automate scanning and mapping computer networks / P. Denzler, R. Teixeira // arXiv.org. — 2017. — URL: https://arxiv.org/abs/1710.01026 (дата обращения: 06.12.2025).
6. Lyon, G. F. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning / G. F. Lyon. — Sunnyvale : Insecure.Com LLC, 2009. — 468 p.
7. Полное руководство по сетевому сканированию с использованием Nmap [Электронный ресурс]. — URL: https://codeby.net/threads/kniga-po-nmap-na-russkom-polnoye-rukovodstvo-po-setevomu-skanirovaniyu.67190/ (дата обращения: 21.11.2025).

References

1. Computer networks. Principles, technologies, protocols (2024). Anniversary edition, revised and expanded. Saint Petersburg: Piter.
2. Anderson, R. (2020). Security engineering: A guide to building dependable distributed systems (3rd ed.). Indianapolis, IN: Wiley.
3. LabEx. (2025). How to use Nmap for network discovery and service identification in cybersecurity. Retrieved November 19, 2025, from https://labex.io/ru/tutorials/nmap-how-to-use-nmap-for-network-discovery-and-service-identification-in-cybersecurity-415099
4. Uymin, A. G. (2022). Network and system administration. Demonstration exam CODE 1.1: Study guide for secondary vocational education (3rd ed.). Saint Petersburg: Lan Publishing House.
5. Denzler, P., & Teixeira, R. (2017). An iterative and toolchain-based approach to automate scanning and mapping computer networks. arXiv. Retrieved December 6, 2025, from https://arxiv.org/abs/1710.01026
6. Lyon, G. F. (2009). Nmap network scanning: The official Nmap project guide to network discovery and security scanning. Sunnyvale, CA: Insecure.Com LLC.
7. Codeby.net. (2025). Complete guide to network scanning using Nmap. Retrieved November 21, 2025, from https://codeby.net/threads/kniga-po-nmap-na-russkom-polnoye-rukovodstvo-po-setevomu-skanirovaniyu.67190/

Информация об авторах

Селиванов Иван Константинович — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, Российская Федерация, e-mail: gt0305@mail.ru

Леденёв Дмитрий Алексеевич — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, Российская Федерация, e-mail: dledenev@list.ru

TESTING BUILT-IN SECURITY MECHANISMS OF L2+ NETWORK DEVICES USING NMAP UNDER ACTIVE CONSOLE ACCESS

Selivanov I.K. [1]***, Ledenev D.A.*** [1]

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

Abstract. The article examines the effectiveness of built-in security mechanisms of L2+ network devices under network scanning performed using the nmap utility. The relevance of the study is determined by the widespread use of managed network devices in corporate and educational networks, as well as by the need to minimize the attack surface of management interfaces without deploying additional security tools. Special attention is paid to the analysis of the management plane in the presence of active console access.

The purpose of the study is to assess the network accessibility of management services and the state of network ports in the default configuration of network devices from different vendors. The research objects include a MikroTik CRS326-24G-2S+ managed switch and Cisco routers of the 1900 and 800 series. The experimental methodology is based on active network reconnaissance techniques using TCP SYN scanning and on correlating scanning results with actual device configurations obtained via console access.

The study shows that in the default configuration the examined devices remain detectable at the IP level but do not expose open remote management services. Management ports are identified as filtered or closed, indicating the operation

of built-in traffic filtering and access restriction mechanisms. It is demonstrated that the presence of active console access does not directly affect the results of network scanning.

The obtained results confirm the effectiveness of default security configurations in reducing the network attack surface and may be applied in network security assessments as well as in educational and training environments.

Keywords: L2+ network devices, information security, management plane, network scanning, nmap, console access.

Information about the authors

Selivanov Ivan Konstantinovich — student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, Russian Federation, e-mail: gt0305@mail.ru

Ledenev Dmitriy Alekseevich — student, Gubkin Russian State University of Oil and Gas (National Research University), Moscow, Russian Federation, e-mail: dledenev@list.ru

Рубрика 2. Методы и системы защиты информации, информационная безопасность

УДК 004.056.5

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

ВОПРОСЫ ХРАНЕНИЯ И ЗАЩИТЫ ДОМЕННЫХ УЧЕТНЫХ ЗАПИСЕЙ В ОС СЕМЕЙСТВА WINDOWS. ИНСТРУМЕНТЫ ПОЛУЧЕНИЯ ПАРОЛЕЙ

Аннотация. В статье рассматриваются вопросы хранения и защиты доменных учетных записей в инфраструктуре Active Directory на базе Windows Server 2019. Актуальность исследования обусловлена тем, что доменная учетная запись остается одним из ключевых объектов атак на корпоративную сеть: при компрометации учетных данных злоумышленник получает возможность расширять привилегии, закрепляться в системе и получать доступ к критически важным ресурсам. Цель работы состоит в практической оценке рисков извлечения аутентификационных данных и анализе эффективности базовых защитных мер. В ходе исследования рассмотрены механизмы хранения секретов в LSASS и базе NTDS.dit, а также проведен лабораторный эксперимент с использованием Mimikatz в контролируемой среде. Сопоставлены результаты для слабого и криптостойкого пароля, а также для конфигурации с включенной защитой LSA Protection. Показано, что сложный пароль снижает вероятность быстрого восстановления пароля по хешу, однако не устраняет риск оффлайн-подбора и не предотвращает сам факт утечки хеша. Наиболее значимым результатом является подтверждение необходимости комплексной защиты: включения LSA Protection и Credential Guard, применения группы «Защищенные пользователи», разграничения привилегий и модели многоуровневого администрирования. Полученные выводы могут быть использованы при построении учебных стендов и при разработке организационно-технических мер защиты доменной инфраструктуры Windows. Дополнительно подчёркивается роль профилактических настроек в снижении вероятности повторной компрометации.

Ключевые слова: Active Directory, доменные учетные записи, информационная безопасность, уязвимости, доменная инфраструктура Windows.

Введение

В современных корпоративных сетях технология Active Directory (AD) обеспечивает централизованное управление доменной аутентификацией и доступом, фактически являясь стандартом администрирования в ОС Windows. По оценкам Microsoft, AD используется в более чем 90% предприятий (включая 95% компаний из списка самых крупных компаний США по версии американского журнала Fortune) для управления учётными записями и идентификацией пользователей [1]. Это означает, что безопасность доменных учетных данных напрямую влияет на защиту всей сети организации – компрометация контроллера домена даёт злоумышленнику привилегированный доступ и может привести к краже критически важных данных и атаке на всю инфраструктуру [2][3]. Одним из ключевых инструментов поиска угроз безопасности является утилита Mimikatz[1] — — она позволяет извлекать из памяти процесса LSASS открытые пароли, хэши, PIN-коды и Kerberos-токены [4].

Известно, что учётные данные доменных пользователей хранятся в базе AD (файл NTDS.dit) в виде зашифрованных хешей паролей (NTLM) и соответствующих Kerberos-ключей. При аутентификации и на рабочих станциях эти данные также попадают в память процесса LSASS, откуда их могут считывать компрометированные локальные процессы. Чтобы ограничить эти риски, в Windows Server 2019 и Windows 10/11 были внедрены различные механизмы защиты. Например, Credential Guard, который использует виртуализацию (Virtualization-based Security) для изоляции секретов в специальном защищённом процессе (LSAIso), так что обычный LSASS становится «пустым» для остальной системы [5]. Это защищает хранимые в памяти пароли и билеты Kerberos от прямого считывания. При включении LSA Protection (RunAsPPL) процесс LSASS становится «защищённым» (PPL), и только доверенные подписанные системные компоненты могут взаимодействовать с его памятью. В результате внедрение кода в LSASS и чтение его памяти вредоносными утилитами (в нашем случае Mimikatz) становится значительно сложнее [6]. Группа «Защищенные пользователи»

[1]Созданная французским исследователем Бенжаменом Дельпи (Benjamin Delpy). Изначально проект задумывался как исследовательский — просто чтобы доказать, что пароли в Windows реально извлечь из памяти [4].

— встроенная глобальная группа AD, которая при добавлении в неё аккаунта запрещает кеширование пароля, ограничивает протоколы аутентификации (только AES-Kerberos) и делает аккаунт менее уязвимым к атакам (например, отключает поддержку NTLM для этих пользователей) [7].

Тем не менее сохраняются значительные пробелы в защите доменных аккаунтов. Так, Credential Guard защищает локальные устройства, но не защищает саму базу AD на контроллерах домена и не влияет на протоколы внешней аутентификации (например, RDP Gateway) [5]. Многие организации до сих пор не активировали эти функции из-за совместимости или нехватки ресурсов на обновление ОС/уровня домена. Атаки типа Pass-the-Hash, Overpass-the-Hash, DCSync и особенно подделка Kerberos-билетов (Golden/Silver Ticket) остаются эффективными средствами обхода большинства текущих мер защиты. По данным Verizon, в 88% случаев успешных взломов используются украденные учётные данные, что подчёркивает недостаточную стойкость существующих решений. Кроме того, рост гибридных сред (смешанные On-Premise AD и Azure AD, синхронизация учётных записей и т.д.) создаёт новые векторы атак и усложняет обнаружение нарушений [2]. Таким образом, несмотря на внедрённые средства защиты, остаётся высокий риск компрометации доменных учётных записей, что требует дальнейших исследований уязвимостей и совершенствования архитектуры безопасности.

Объект исследования — информационная безопасность доменных инфраструктур Windows.

Предмет исследования — механизмы хранения, защиты и извлечения доменных учётных данных в среде Windows Server 2019/AD.

Цель исследования — анализ уязвимостей, архитектурных особенностей и существующих защитных мер с целью предотвращения компрометации доменных учётных записей.

Литературный обзор

В научной и практической литературе описаны многочисленные методы атак и защиты инфраструктуры, построенной на технологии AD. Многие специалисты (например, Sean Metcalf из ADSecurity, Red Canary, CIS и др.) публикуют исследования и рекомендации по защите домена. В ряде недавних работ [8][9][10][11] анализируются уязвимости AD и предлагаются алгоритмы обнаружения. Практические руководства и блоги (CIS, SecurityLab, Microsoft) детально освещают механизмы кражи паролей (DCsync, Kerberoasting, Golden Ticket и др.) и способы их предотвращения [3].

Технические материалы и белые книги Microsoft, MITRE и CIS, а также отчёты и блоги специализированных команд (Red Team/Blue Team) подчёркивают необходимость защиты ключевых компонентов AD. В руководствах Microsoft также описан подход многоуровневого управления привилегиями (AD Tiering). Согласно этому подходу, учётные записи доменных администраторов (уровень 0) не должны выполнять вход на сервера нижних уровней (уровни 1 и 2), чтобы ограничить распространение компрометированных учётных данных. Это отражено в рекомендациях по выделению «тысячелеток» безопасности (Tier0–Tier2) и ограничению доступа (например, Tier-0 админы не логинятся на рабочие станции Tier-2) [12]. Параллельно механизм AdminSDHolder служит встроенным средством непрерывного контроля прав на особо защищённых учётных записях; объект AdminSDHolder регулярно применяется (через процесс SDProp) к ACL членов привилегированных групп, обеспечивая консистентность прав [13].

Обзор инструментов атаки и защиты в литературе подчёркивает широту решения задачи. С одной стороны, Mimikatz и аналогичные утилиты позволяют осуществлять различные формы кражи учётных данных (из памяти, из Kerberos-тиков, из реестра и пр.). С другой стороны, помимо упомянутых LSA Protection и Credential Guard описаны защитные технологии по разграничению прав: минимизация числа аккаунтов в группах «Domain Admins», внедрение мультилогиновой модели, регулярная смена пароля. Изучение академических исследований и руководств позволяет сформулировать гипотезы исследования: во-первых, активация средств защиты LSA и Credential Guard резко снижает объем извлекаемых учётных данных (например, минимизируя успешные PtH-атаки); во-вторых, строжайшее соблюдение AD Tiering (запрет

логина Domain Admin на машины низших уровней) и защита AdminSDHolder ограничивают возможности злоумышленника по эскалации привилегий после утечки учётных данных; наконец, сравнение конфигураций «с защитой/без защиты» покажет качественные различия в полученных хешах и доступах, подтверждая эффективность этих мер [14].

Методы исследования

Исследование носило прикладной характер и включало практическую реализацию атаки Mimikatz для демонстрации рисков. В экспериментальной среде была развернута виртуальная лаборатория на базе Windows Server 2019 с контроллером домена Active Directory и двумя тестовыми учётными записями (одна — с правами доменного администратора, другая — локального администратора). Для сбора данных выполнялся захват NTLM-хешей. Такая форма лабораторной работы соответствует учебно-методическому подходу к изучению сетевого и системного администрирования [15].

Авторы подтверждают, что все учётные записи, пароли и хэши относятся к изолированному синтетическому учебному стенду; публикуемые строковые значения замаскированы и не являются действующими реквизитами доступа.

Процедура эксперимента включала следующие шаги: 1. Создание пользователей и настройка среды. Были созданы новые учётные записи пользователей с заданными паролями и распределены по группам (Domain Admins и Administrators). 2. Затем на контроллере домена были запущены команды Mimikatz для извлечения хешей NTLM. Результаты работы Mimikatz (извлечённые учётные данные) сохранились для анализа. 3. С помощью сторонних ресурсов был проведен анализ NTLM-хешей. 4. Использовали две практики защиты инфраструктуры (усложнение пароля и включение LSA Protection), построенной на технологии AD, провели повторную имитацию атаки и сравнили результаты.

Результаты исследования

В ходе практической части работы была успешно реализована имитация атаки на инфраструктуру Active Directory с последующим анализом защитных мер. На этапе развертывания стенда в оснастке «Active Directory — пользователи и компьютеры» была создана учетная запись пользователя с именем Stealer, которая для целей эксперимента была включена в привилегированные группы, включая Administrators и доменную группу Mim. Проверка состава групп подтвердила корректность настройки прав доступа, необходимых для проведения дальнейших манипуляций в системе.

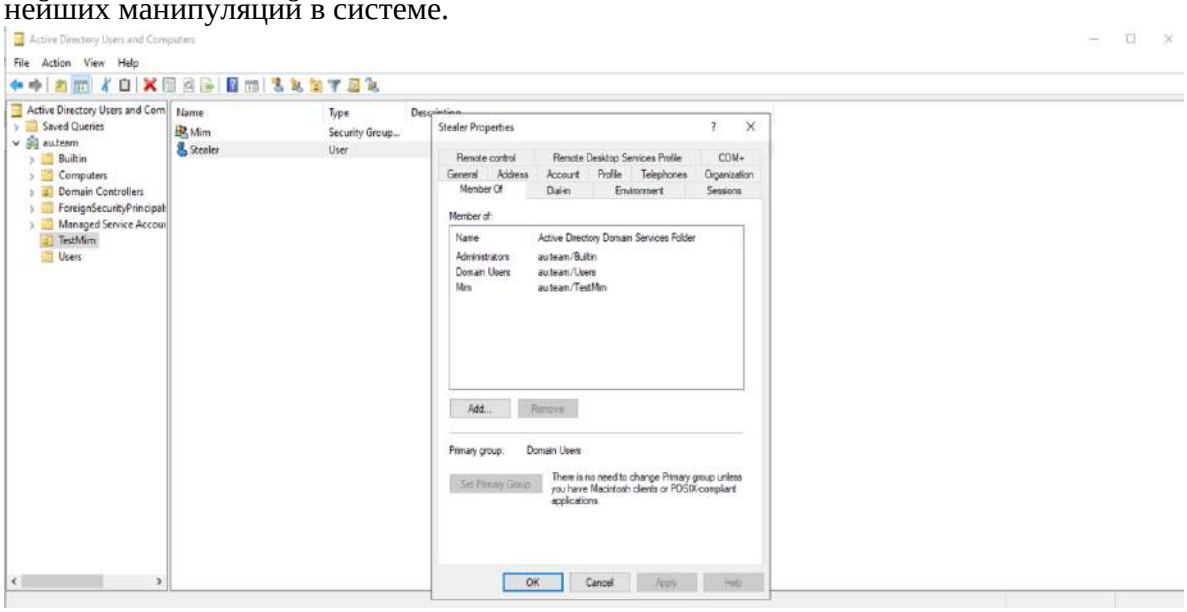


Рис. 1. Настройка прав доступа и членства пользователя Stealer в группах администраторов.

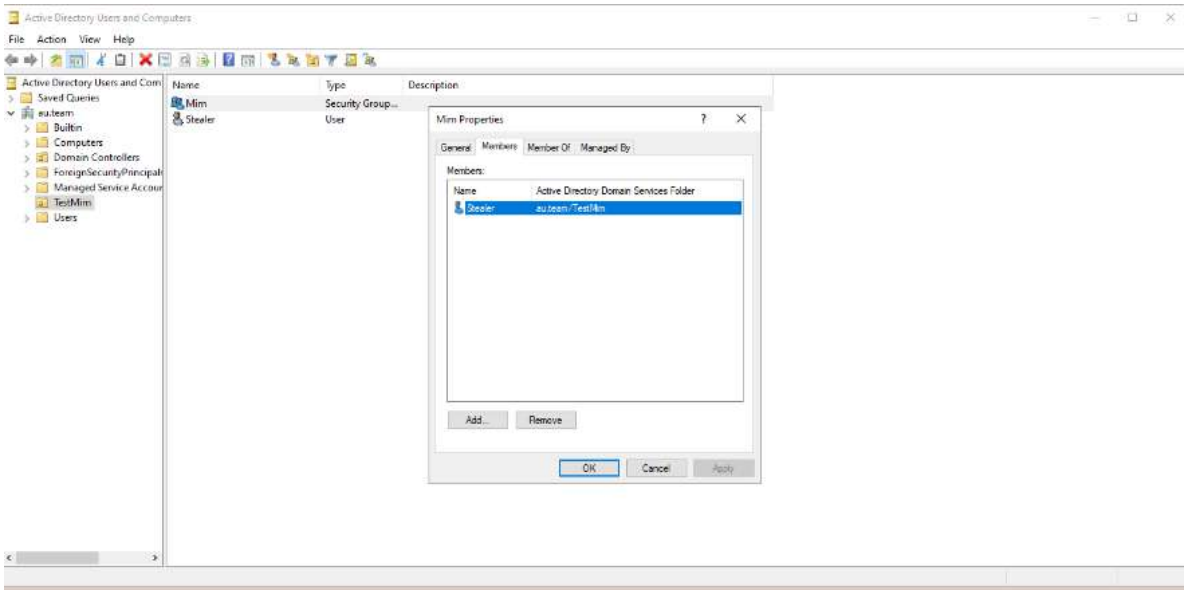


Рис. 2. Список учетных записей группы Mim

После подготовки среды был произведен запуск утилиты Mimikatz с правами локального администратора (из-под учетной записи Stealer). С помощью последовательного ввода команд privilege::debug и sekurlsa::logonpasswords был осуществлен доступ к памяти процесса LSASS [4]. В результате работы утилиты из оперативной памяти были успешно извлечены аутентификационные данные активных сессий. В частности, для учетной записи Administrator был получен NTLM-хеш: [учебное значение скрыто редакцией]. Полученные данные свидетельствуют о возможности полной компрометации учетной записи без знания исходного пароля в открытом виде.

```
cd C:\Users\Stealer\Desktop
cd .\mimikatz_trunk\
cd .\x64\
dir
.\mimikatz.exe
privilege::debug
sekurlsa::logonpasswords
```

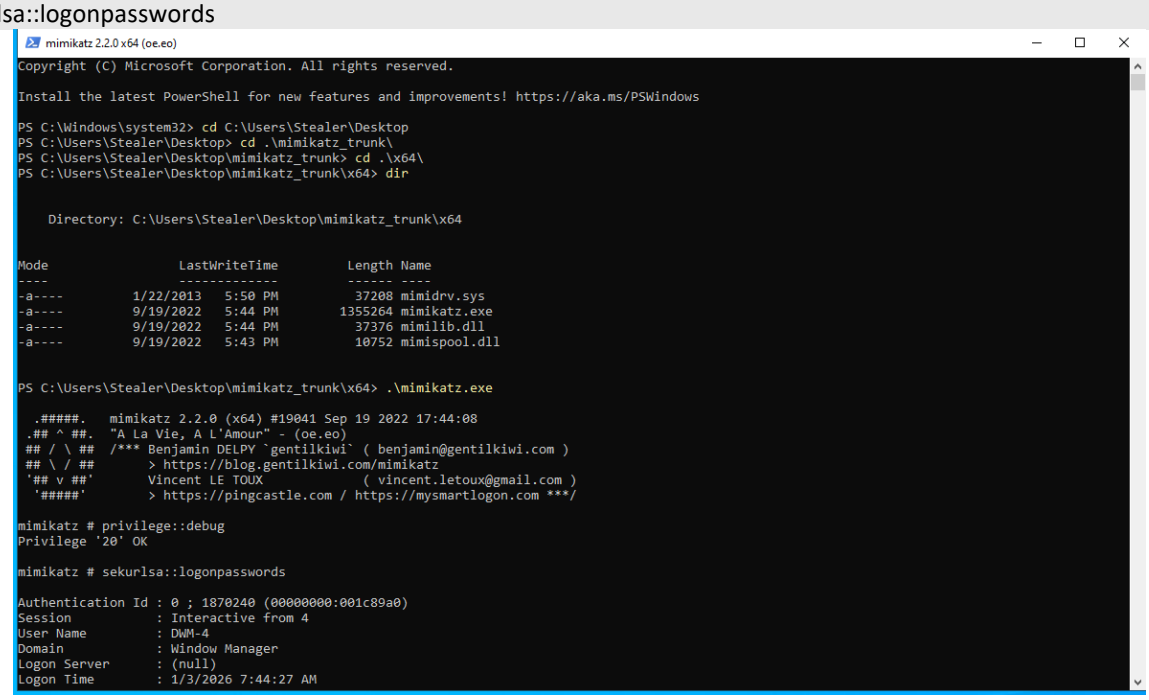


Рис. 3. Ввод команд Mimikatz

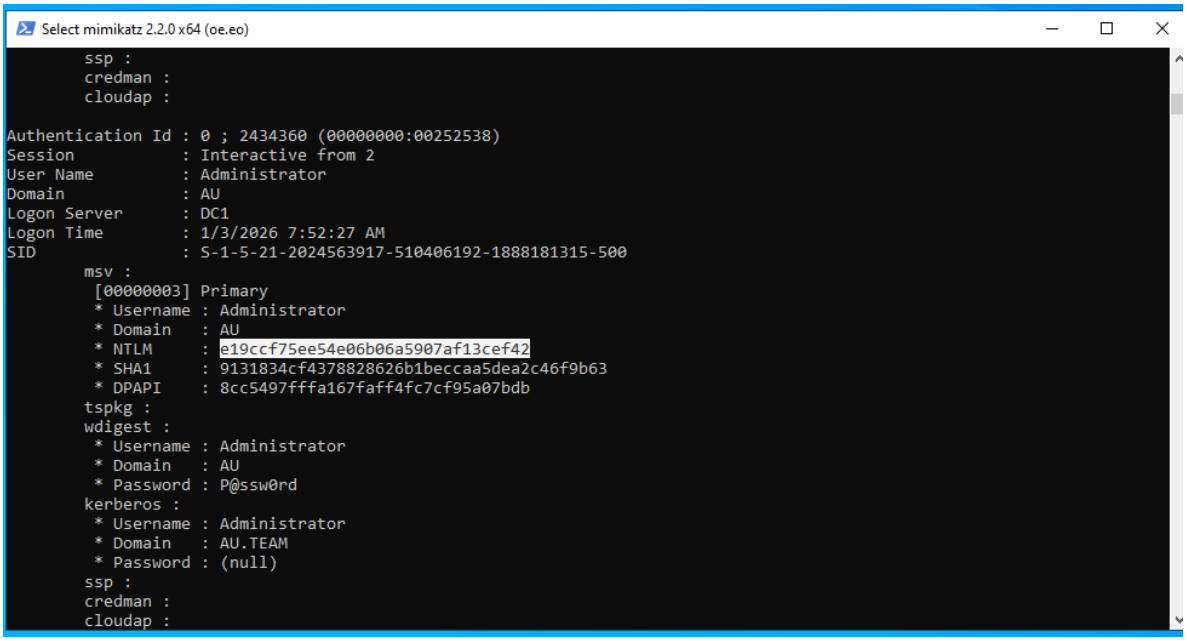


Рис. 4. Извлечение NTLM-хеша учетной записи Administrator с помощью Mimikatz

Для оценки критичности выявленной уязвимости был проведен анализ извлеченного хеша с использованием специализированных онлайн-сервисов Hashes.com и CrackStation. В обоих случаях проверка по базам данных известных паролей заняла менее секунды. Хеш был успешно дешифрован, результатом чего стал пароль «[слабый учебный пароль скрыт редакцией]». Это подтверждает, что использование простых, словарных паролей делает инфраструктуру уязвимой не только к атакам типа Pass-the-Hash, но и к быстрому восстановлению пароля злоумышленником.

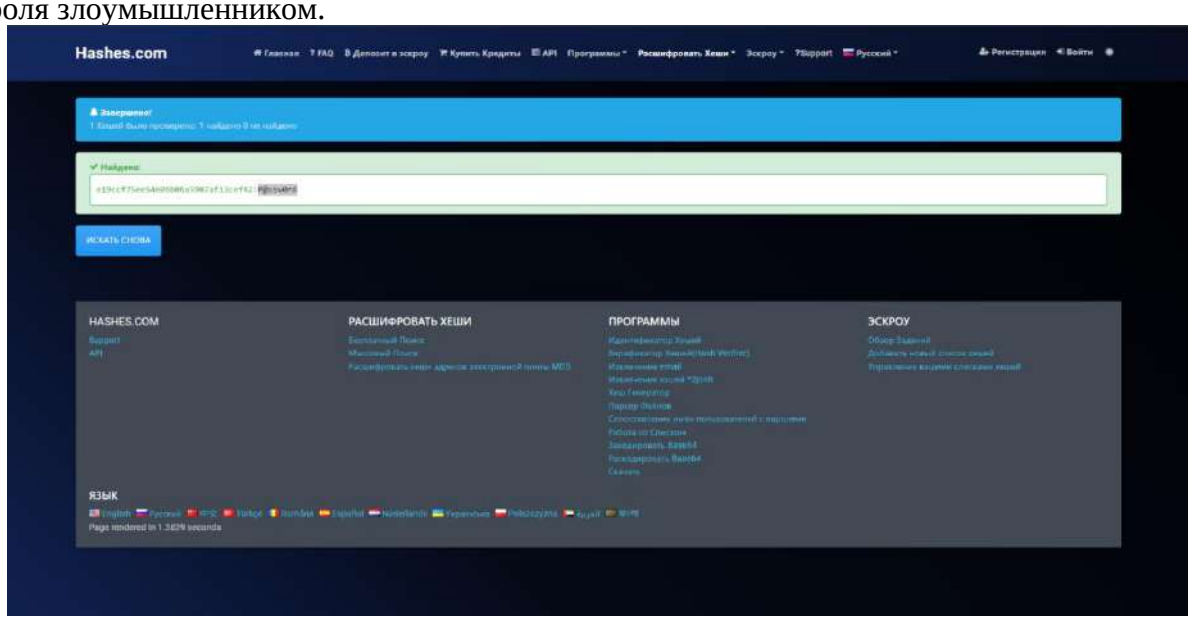


Рис. 5. Успешная дешифровка слабого пароля через онлайн-сервис (Hashes.com)

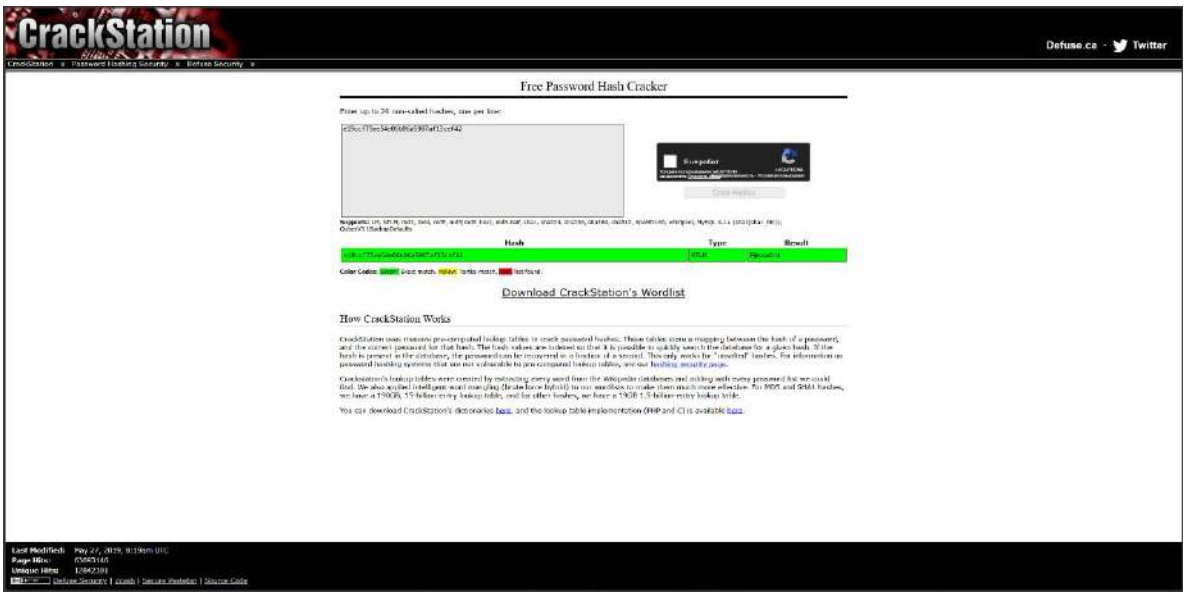


Рис. 6. Успешная дешифровка слабого пароля через онлайн-сервис (CrackStation.net)

Далее была применена практика усиления парольной политики. Для учетной записи Administrator был установлен новый, криптостойкий пароль: [стойкий учебный пароль скрыт редакцией]. После смены пароля была проведена повторная имитация атаки. Несмотря на то, что Mimikatz снова смог извлечь актуальный NTLM-хеш ([учебное значение скрыто редакцией]), попытка его анализа через сервис CrackStation завершилась неудачей — хеш не был найден в базе. Сравнительный анализ результатов показал, что использование сложных паролей и регулярная их ротация значительно повышают трудоемкость атаки, делая невозможным восстановление пароля методами простого перебора по словарям.

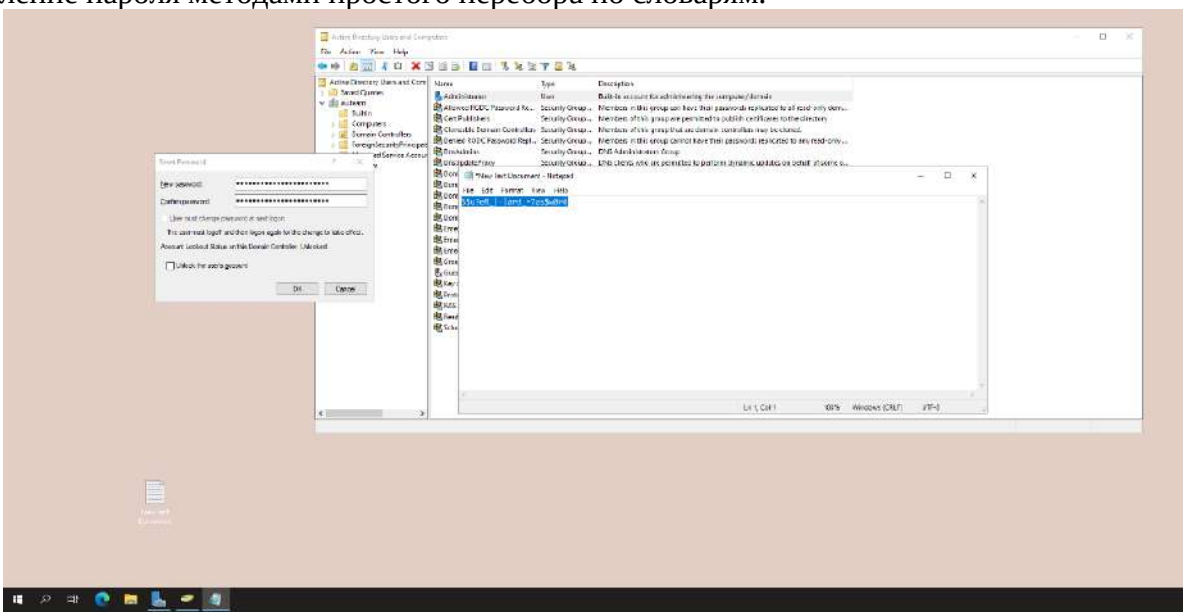


Рис. 7. Установка сложного пароля S$u?eR_|-|ard_=7as$w0r6

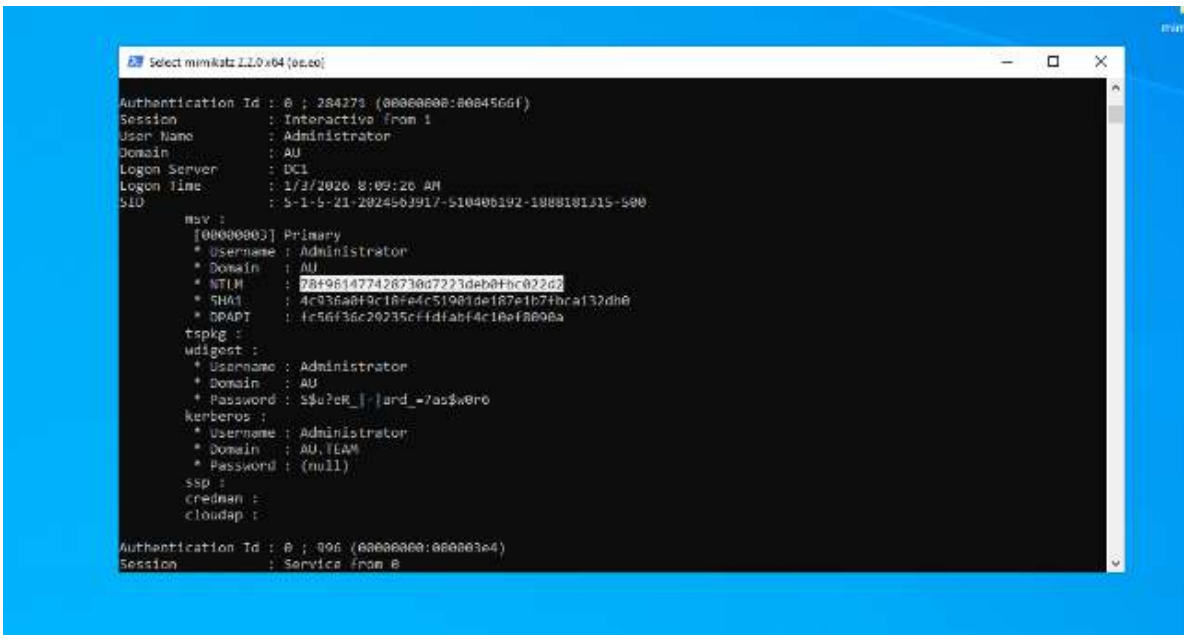


Рис. 8. Извлечение NTLM-хеша учетной записи Administrator с помощью Mimikatz после смены пароля

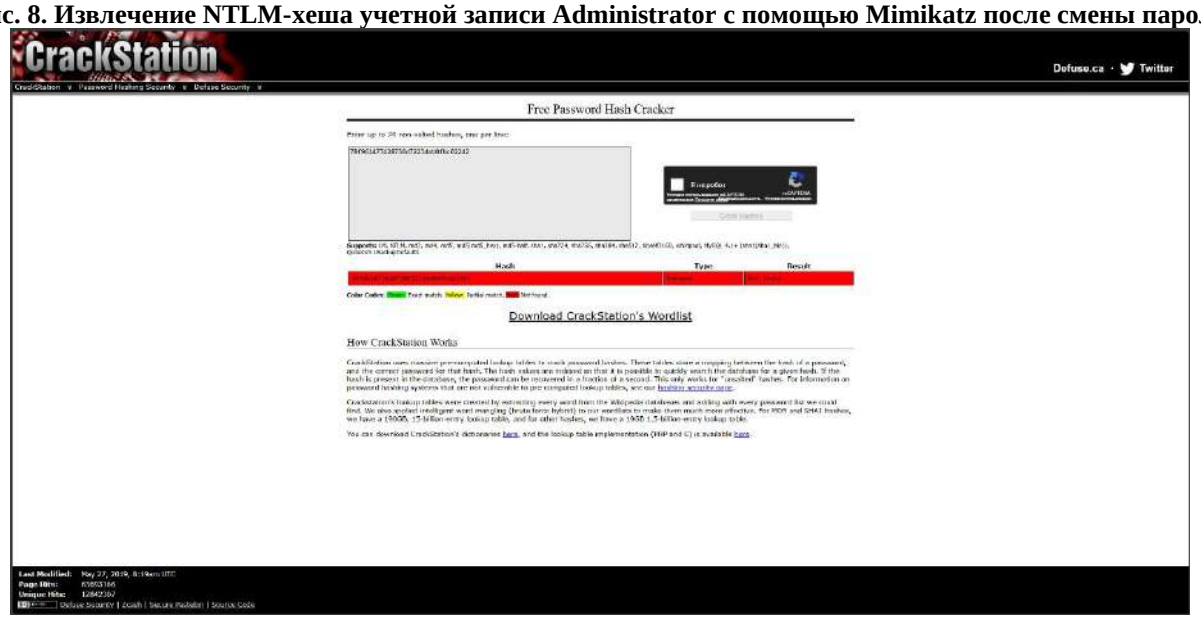


Рис. 9. Неуспешная дешифровка сильного пароля через онлайн-сервис (CrackStation.net)

На заключительном этапе эксперимента включили LSA Protection.Мы вошли в учетную запись Administrator. В редакторе реестра перешли по пути HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa, создали ключ RunAsPPL, изменили его значение данных на 00000001, чтобы настроить функцию защиты LSA с переменной UEFI. Перезапустили контроллер домена, открыли окно просмотра событий (Event Viewer) и в журнале системы нашли событие «WinInit : 12: LSASS.exe был запущен как защищенный процесс с уровнем 4». Это значит, что LSA запускается в защищенном режиме при запуске Windows.

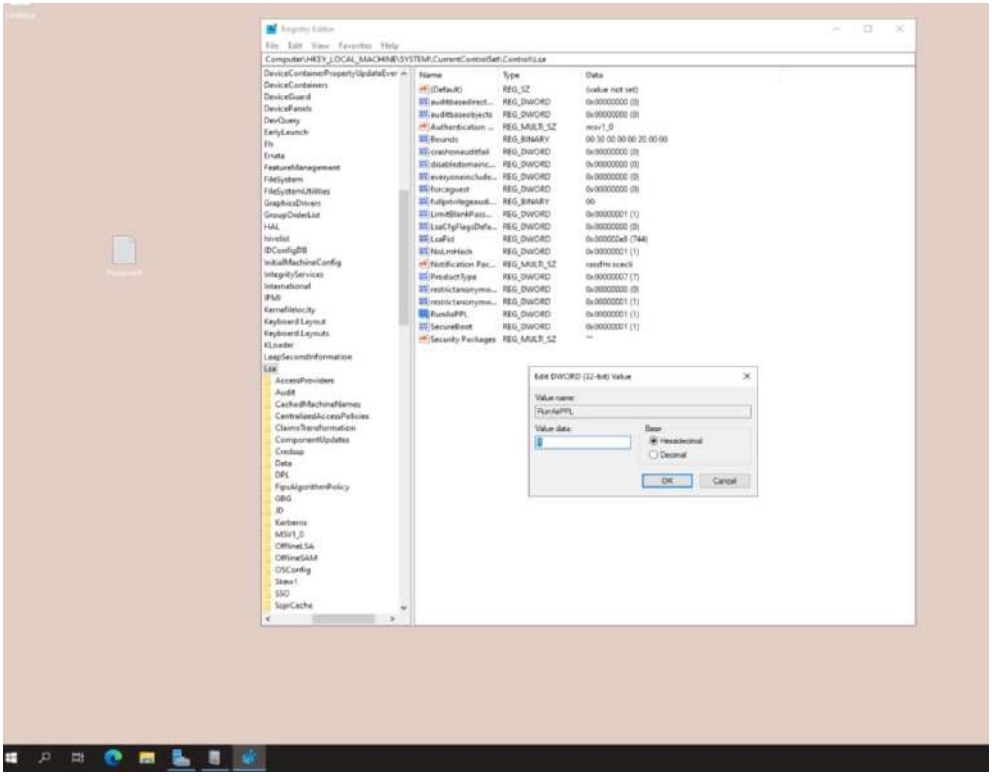


Рис. 10. Изменение значения ключа RunAsPPL в редакторе реестра

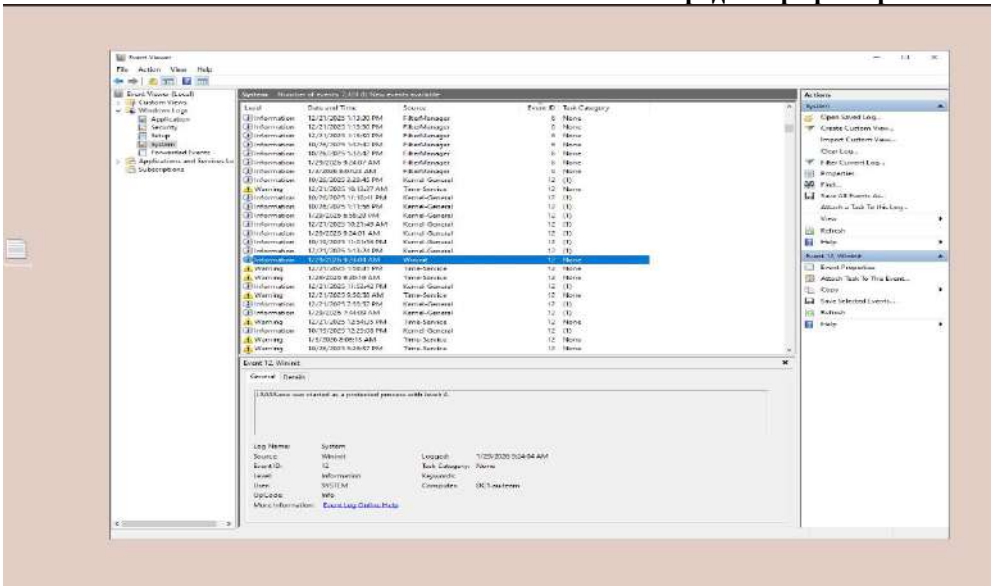


Рис. 11. Просмотр журнала событий системы

Перешли на учетную запись Stealer. Попробовали повторно извлечь NTLM-хеш, но программа выдала ошибку (см. рис. 12).

```
cd C:\Users\Stealer\Desktop
cd .\mimikatz_trunk\
cd .\x64\
.\mimikatz.exe
privilege::debug
sekurlsa::logonpasswords
```

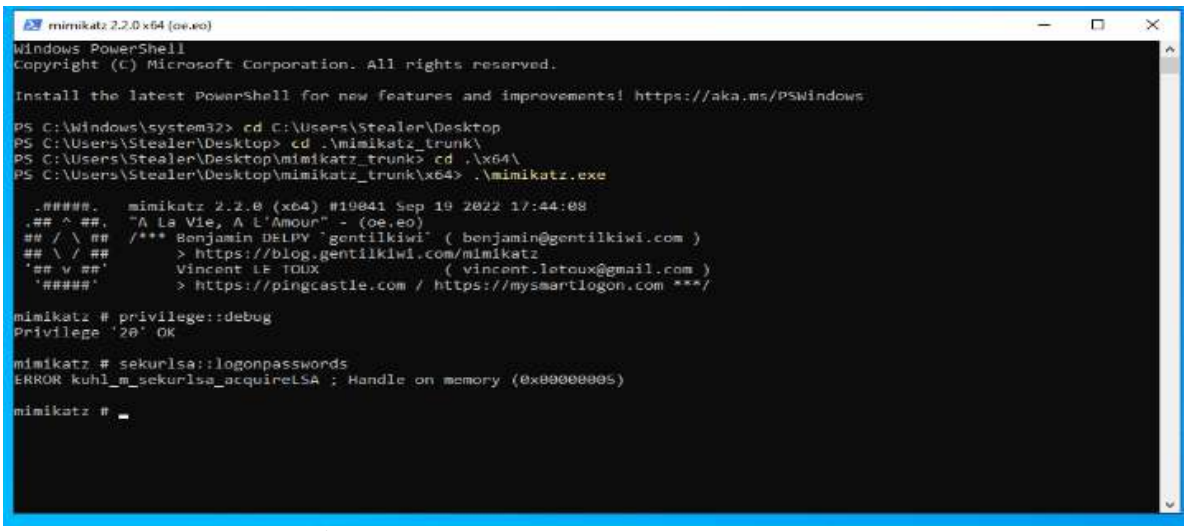


Рис. 12. Ошибка при попытке извлечь NTLM-хеш с помощью Mimikatz

Заключение

Проведенное исследование наглядно продемонстрировало уязвимость стандартных механизмов аутентификации в инфраструктуре Active Directory перед инструментами извлечения учетных данных из оперативной памяти. Экспериментальным путем было подтверждено, что использование утилиты Mimikatz позволяет злоумышленнику, обладающему правами администратора, успешно обходить локальные механизмы защиты и получать доступ к NTLM-хешам активных сессий пользователей.

Анализ результатов первой фазы эксперимента показал, что использование простых словарных паролей фактически сводит защиту к нулю, так как наличие доступных онлайн-баз позволяет восстановить пароль в открытом виде практически мгновенно. Это дает атакующему возможность не только использовать технику Pass-the-Hash, но и полноценно закрепиться в системе под легитимными данными.

Внедрение политики криптостойких паролей во второй части исследования подтвердило свою эффективность как базового эшелона защиты: несмотря на успешное извлечение хеша из памяти, его дешифровка с помощью известных словарей и радужных таблиц стала невозможной. Радужные таблицы представляют собой заранее рассчитанные сопоставления «пароль → хеш», которые позволяют быстро находить пароль по хешу без полного перебора, если пароль входит в предварительно вычисленное множество. Отсутствие записи в подобных таблицах означает только защиту от таких предопределённых словарных атак, но не исключает возможности его подбора методом полного перебора (brute-force) или подбором по маске с использованием современных GPU-акселераторов (например, Hashcat) и других специализированных инструментов. Таким образом, даже достаточно длинный и сложный пароль, оставаясь вне словарей и радужных таблиц, не гарантирует абсолютной стойкости, если злоумышленник обладает доступом к NTLM-хешу и мощным вычислительным ресурсам. Важнейшей мерой защиты является предотвращение самой утечки хеша: если хеш попадёт в руки злоумышленника, он становится объектом оффлайн-подбора вне зависимости от исходной сложности пароля. Соответственно, следует подчеркнуть важность мер защиты, предотвращающих утечки хешей: использование группы «Защищенные пользователи», многоуровневое управление привилегиями (AD Tiering) и внедрение технологий LSA Protection, Credential Guard и подобных, что в совокупности значительно повышает трудоемкость атаки и минимизирует риски компрометации сети.

В ходе эксперимента было доказано, что активация механизма LSA Protection (через параметр реестра RunAsPPL) является критически важным барьером. После настройки и подтверждения запуска процесса LSASS в защищенном режиме утилита Mimikatz не смогла получить доступ к памяти процесса даже с правами администратора. Это подтверждает, что перевод LSASS в статус защищенного процесса (PPL) эффективно блокирует попытки несанкционированного чтения секретов из памяти.

Следует отметить, что сценарий, реализованный в эксперименте, изначально стал возможен из-за фундаментального нарушения принципа разделения привилегий. В нем администратор домена (учетная запись уровня Tier 0) выполнял вход и работал на обычной рабочей станции, отнесенной к более низкому уровню доверия (Tier 1 или 2). Внедрение модели многоуровневого администрирования (Tiering) предотвратило бы саму возможность проведения такой атаки, поскольку учетные данные администраторов уровня Tier 0 (пароли, хеши, Kerberos-тикеты) никогда не кэшировались бы и не обрабатывались в памяти менее защищенных систем нижних уровней. Таким образом, даже если злоумышленник получит полный контроль над рабочей станцией, в ее памяти просто не окажется высокопривилегированных учетных данных, представляющих ценность для атак на критическую инфраструктуру. Внедрение модели Tiering является организационно-технической мерой, дополняющей технические средства вроде LSA Protection и создающей глубокоэшелонированную защиту, где компрометация одного уровня не приводит к катастрофическим последствиям для всей сети.

Библиографический список

1. Success with Enterprise Mobility + Identity [Электронный ресурс] : блог-пост // Microsoft Security Blog. Microsoft Tech Community. — 2016. — URL: https://techcommunity.microsoft.com/blog/microsoft-security-blog/success-with-enterprise-mobility-identity/248613 (дата обращения: 31.12.2025).
2. Active Directory under siege: Why traditional hardening isn’t enough [Электронный ресурс] // The Hacker News. — 2025. — URL: https://thehackernews.com/2025/11/active-directory-under-siege-why.html (дата обращения: 31.12.2025).
3. Хакер, С. Mimikatz — легенда хакерских инструментов: как работает, зачем нужен и почему его боятся все админы [Электронный ресурс] / С. Хакер // SecurityLab. — 2024. — URL: https://www.securitylab.ru/blog/personal/SimlpeHacker/355268.php (дата обращения: 31.12.2025).
4. Fazel, M. A. P. Comprehensive Mimikatz Ebook: A Practical Guide to Post-Exploitation / M. A. P. Fazel, M. Rashidi // SSRN Electronic Journal. — 2025. — DOI: 10.2139/ssrn.5190361.
5. How It Works (Credential Guard) [Электронный ресурс] // Microsoft Learn. — URL: https://learn.microsoft.com/en-us/windows/security/identity-protection/credential-guard/how-it-works (дата обращения: 31.12.2025).
6. Why You Should Enable LSA Protection [Электронный ресурс] // Lepide. — 2023. — URL: https://www.lepide.com/blog/why-you-should-enable-lsa-protection/ (дата обращения: 31.12.2025).
7. Protected Users Security Group [Электронный ресурс] // Microsoft Learn. — URL: https://learn.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/protected-users-security-group (дата обращения: 31.12.2025).
8. Binduf, H. O. Active Directory and Related Aspects of Security / H. O. Binduf, H. Alamoudi, S. Balahmar, H. Alshamrani, H. Al-Omar, N. Nagy // 2018 21st Saudi Computer Society National Computer Conference (NCC). — Riyadh, 2018. — P. 4474–4479. — DOI: 10.1109/NCG.2018.8593188.
9. Ebad, S. A. Lessons learned from offline assessment of security-critical systems: the case of Microsoft’s Active Directory / S. A. Ebad // International Journal of System Assurance Engineering and Management. — 2022. — Vol. 13. — P. 535–545. — DOI: 10.1007/s13198-021-01236-2.
10. Grillenmeier, G. Protecting Active Directory against modern threats / G. Grillenmeier // Network Security. — 2021. — Vol. 2021, iss. 11. — P. 15–17. — DOI: 10.1016/S1353-4858(21)00132-X.
11. Kumar, C. Active Directory and Its Security Testing / C. Kumar, S. S. Debnath, A. Kar, P. Debbarma, S. Piramanayagam // Proceedings of International Conference on Advanced

Communications and Machine Intelligence. MICA 2023. — Singapore : Springer, 2024. — DOI: 10.1007/978-981-97-6222-4_43.
12. Tier model [Электронный ресурс] // Microsoft Learn. — URL: https://learn.microsoft.com/en-us/microsoft-identity-manager/pam/tier-model-for-partitioning-administrative-privileges (дата обращения: 31.12.2025).
13. Reducing the Active Directory attack surface [Электронный ресурс] // Microsoft Learn. — URL: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/reducing-the-active-directory-attack-surface (дата обращения: 31.12.2025).
14. T1558.001 Golden Ticket [Электронный ресурс] // MITRE ATT&CK. — URL: https://attack.mitre.org/techniques/T1558/001/ (дата обращения: 31.12.2025).
15. Уймин, А. Г. Сетевое и системное администрирование. Демонстрационный экзамен КОД 1.1 : учебно-методическое пособие для СПО / А. Г. Уймин. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 480 с. — ISBN 978-5-8114-9255-8.

References

1. Microsoft Security Blog. (2016). Success with Enterprise Mobility + Identity. Microsoft Tech Community. Retrieved December 31, 2025, from https://techcommunity.microsoft.com/blog/microsoft-security-blog/success-with-enterprise-mobility-identity/248613
2. The Hacker News. (2025). Active Directory under siege: Why traditional hardening isn't enough. Retrieved December 31, 2025, from https://thehackernews.com/2025/11/active-directory-under-siege-why.html
3. Khaker, S. (2024). Mimikatz — the legend of hacking tools: How it works, why it is needed, and why all administrators fear it. SecurityLab. Retrieved December 31, 2025, from https://www.securitylab.ru/blog/personal/SimlpeHacker/355268.php
4. Fazel, M. A. P., & Rashidi, M. (2025). Comprehensive Mimikatz Ebook: A practical guide to post-exploitation. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5190361
5. Microsoft Learn. (n.d.). How It Works (Credential Guard). Retrieved December 31, 2025, from https://learn.microsoft.com/en-us/windows/security/identity-protection/credential-guard/how-it-works
6. Lepide. (2023). Why you should enable LSA Protection. Retrieved December 31, 2025, from https://www.lepide.com/blog/why-you-should-enable-lsa-protection/
7. Microsoft Learn. (n.d.). Protected Users Security Group. Retrieved December 31, 2025, from https://learn.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/protected-users-security-group
8. Binduf, H. O., Alamoudi, H., Balahmar, S., Alshamrani, H., Al-Omar, H., & Nagy, N. (2018). Active Directory and related aspects of security. In 2018 21st Saudi Computer Society National Computer Conference (NCC) (pp. 4474–4479). IEEE. https://doi.org/10.1109/NCG.2018.8593188
9. Ebad, S. A. (2022). Lessons learned from offline assessment of security-critical systems: The case of Microsoft's Active Directory. International Journal of System Assurance Engineering and Management, 13, 535–545. https://doi.org/10.1007/s13198-021-01236-2
10. Grillenmeier, G. (2021). Protecting Active Directory against modern threats. Network Security, 2021(11), 15–17. https://doi.org/10.1016/S1353-4858(21)00132-X
11. Kumar, C., Debnath, S. S., Kar, A., Debbarma, P., & Piramanayagam, S. (2024). Active Directory and its security testing. In S. L. Peng et al. (Eds.), Proceedings of International Conference on Advanced Communications and Machine Intelligence. MICA 2023. Springer. https://doi.org/10.1007/978-981-97-6222-4_43
12. Microsoft Learn. (n.d.). Tier model. Retrieved December 31, 2025, from https://learn.microsoft.com/en-us/microsoft-identity-manager/pam/tier-model-for-partitioning-administrative-privileges

13. Microsoft Learn. (n.d.). Reducing the Active Directory attack surface. Retrieved December 31, 2025, from https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/reducing-the-active-directory-attack-surface
14. MITRE ATT&CK. (n.d.). T1558.001 Golden Ticket. Retrieved December 31, 2025, from https://attack.mitre.org/techniques/T1558/001/
15. Uymin, A. G. (2022). Network and system administration. Demonstration exam CODE 1.1: Study guide for secondary vocational education (3rd ed.). Lan Publishing House.

Информация об авторах

Богданов Родион Андреевич — студент ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: bogdanov.r@gubkin.ru.

Гададов Гаджирамазан Джамалутинович — студент ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: gadadov.g@gubkin.ru

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

Bogdanov R. A.[1], Gadadov G. D.[1]

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

Abstract. The article examines the storage and protection of domain accounts in a Windows Server 2019 Active Directory infrastructure. The relevance of the study is determined by the fact that domain credentials remain one of the primary targets in attacks against corporate networks: once such data is compromised, an attacker may escalate privileges, maintain persistence, and gain access to critical resources. The purpose of the work is to assess, in practice, the risks associated with extracting authentication data and to analyze the effectiveness of basic protective measures. The study considers the mechanisms for storing secrets in the LSASS process and in the NTDS.dit database, and includes a controlled laboratory experiment using the Mimikatz utility. The results are compared for a weak password, a cryptographically stronger password, and a configuration with LSA Protection enabled. It is shown that a strong password reduces the probability of rapid password recovery from a hash, but it does not eliminate the possibility of offline cracking and does not prevent the leakage of the hash itself. The most significant result is the confirmation that domain infrastructure protection requires an integrated approach: enabling LSA Protection and Credential Guard, using the Protected Users group, enforcing strict privilege separation, and applying a tiered administration model. The findings may be used in educational laboratories and in the development of organizational and technical measures for protecting Windows domain infrastructures.

Keywords: Active Directory, domain accounts, information security, vulnerabilities, Windows domain infrastructure**.**

Information about the authors

Bogdanov Rodion Andreevich — student of Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: bogdanov.r@gubkin.ru

Gadadov Gadzhiramazan Djamalutinovich — student of Gubkin Russian State University of Oil and Gas (National Research University), Moscow, e-mail: gadadov.g@gubkin.ru

УСЛОВИЯ И ПОРЯДОК НАПРАВЛЕНИЯ МАТЕРИАЛОВ

Редакция научного журнала «ПРОФЕССИОНАЛИТЕТ» принимает к рассмотрению статьи, исправленные версии рукописей, ответы на замечания редакции и рецензентов, а также сопроводительную переписку по конкретной статье. Направление материалов осуществляется в электронной форме в соответствии с внутренним регламентом редакции.

Каждое письмо должно содержать информативную тему и заполненный сопроводительный текст. Переписка ведется в официально-деловом стиле. В теме письма указываются тип обращения, фамилия автора, краткое название статьи и, при необходимости, номер итерации исправления.

В тексте письма обязательно приводятся полное название статьи, сведения обо всех авторах, дисциплина (если применимо), номер итерации исправления и контактные данные ответственного автора. Для исправленных версий дополнительно рекомендуется указывать, на какое письмо редакции или рецензии дается ответ, и кратко обозначать внесенные исправления.

Файл рукописи направляется в формате .docx, на кириллице, по установленной форме наименования. Исправленные версии направляются исключительно ответом на письмо редакции с обязательным указанием номера итерации в теме письма, тексте письма и имени файла.

Материалы, оформленные с нарушением установленных требований, могут быть возвращены без рассмотрения до устранения технических замечаний.

Контактные данные редакционной коллегии

Почтовый адрес для корреспонденции: Москва, Ленинский пр-кт, д. 65/1, отделение почтовой связи № 119296.

До востребования.

Получатель: Павловский Владимир Владимирович.

Тел. +7(950) 632-04-38

e-mail: organizers@au-team.ru

главный редактор – Уймин Антон Григорьевич;

ответственный секретарь - Уймина Ольга Ивановна;

технический редактор - Козлов Глеб Васильевич.

ПРОФЕССИОНАЛИТЕТ, 2025, № 4 (008)

Научное электронное издание.

Сведения о программном обеспечении, использованном для создания электронного издания:
LibreOffice — набор, вёрстка текста, генерация PDF
https://ru.libreoffice.org

Техническая обработка и подготовка материалов выполнены авторами.

Подписано к использованию: 31.12.2025.

Объём издания: 76,6 Мб.

Комплектация издания: pdf.

Запись на физический носитель: Уймин А. Г., тел. +7 (950) 632-04-38.

Издатель — редакция научного журнала «ПРОФЕССИОНАЛИТЕТ».

Место издания: Москва.

Электронная версия подготовлена редакцией журнала для распространения в локальной и сетевой форме.

Носитель электронного издания: URALOLIMP.WEBSITE