УДК 004.738.5

Сапожников Максим Сергеевич
студент
2 курс, факультет «Комплексной безопасности ТЭК»
РГУ нефти и газа (НИУ) имени И.М.Губкина
Россия, г. Москва
Паралян Ксения Игоревна
студент
2 курс, факультет «Комплексной безопасности ТЭК»
РГУ нефти и газа (НИУ) имени И.М.Губкина
Россия, г. Москва

ОЦЕНКА ПРИГОДНОСТИ FIREWALLD ДЛЯ ЗАЩИТЫ СЕТЕВОГО ПЕРИМЕТРА В ИНФРАСТРУКТУРЕ НА БАЗЕ ОС АЛЬТ: АРХИТЕКТУРНЫЙ АНАЛИЗ И ЭКСПЕРИМЕНТАЛЬНОЕ

В условиях роста сложности сетевых инфраструктур и увеличения количества киберугроз управление сетевым экраном становится критически важной задачей для обеспечения информационной безопасности. Firewalld представляет собой современный динамический менеджер брандмауэра для Linux-систем, использующий концепцию зон и сервисов для гибкого управления сетевыми политиками. Данная статья проводит комплексный анализ архитектуры Firewalld в контексте современных подходов к сетевой безопасности, рассматривает его место в экосистеме средств защиты операционных систем семейства Linux и представляет результаты функционального тестирования в ОС Альт. На основе анализа литературы систематизированы современные исследования в области управления межсетевыми экранами, формальной верификации правил и интеграции с российскими ОС. Практическая часть включает подробное тестирование основных механизмов Firewalld, дополненное элементами тестирования на безопасность с моделированием реальных атак, а также оценку

производительности при масштабировании до 1000 правил. Впервые представлены результаты негативного тестирования, моделирующего типовые ошибки администратора, что позволяет объективно оценить механизмы валидации и восстановления системы. Результаты исследования подтверждают целесообразность использования Firewalld в корпоративных и государственных средах, работающих под управлением ОС Альт, при условии дополнительной настройки механизмов защиты от DDoS-атак и оптимизации производительности для больших наборов правил.

КЛЮЧЕВЫЕ СЛОВА:Firewalld, брандмауэр, сетевая безопасность, ОС Альт, функциональное тестирование, сетевые политики, зоны, сервисы, Netfilter, iptables, nftables.

With the growing complexity of network infrastructures and the increasing number of cyber threats, firewall management is becoming a critical task for ensuring information security. Firewalld is a modern, dynamic firewall manager for Linux systems that uses the concept of zones and services for flexible network policy management. This article provides a comprehensive analysis of Firewalld's architecture in the context of modern approaches to network security, examines its place in the Linux operating system security ecosystem, and presents the results of functional testing on Alt OS. Based on a literature review, current research in the areas of firewall management, formal rule verification, and integration with Russian operating systems is systematized. The practical part includes detailed testing of Firewalld's core mechanisms, supplemented by security testing elements simulating real-world attacks, as well as performance evaluation when scaling up to 1000 rules. For the first time, results of negative testing simulating typical administrator errors are presented, which makes it possible to objectively evaluate the system's validation and recovery mechanisms. The results of the study confirm the feasibility of using Firewalld in corporate and government environments running Alt OS, provided that additional DDoS protection mechanisms are configured and performance is optimized for large rule sets.

KEYWORDS: Firewalld, firewall, network security, Alt OS, functional testing, network policies, zones, services, Netfilter, iptables, nftables.

Введение (Introduction)

В современной цифровой экономике обеспечение безопасности сетевой инфраструктуры является одной из приоритетных задач организаций различного масштаба. Согласно отчету Positive Technologies «Кибербезопасность: итоги 2024 года», более 67% успешных атак на корпоративные сети начинались с эксплуатации уязвимостей на сетевом уровне, причем 42% инцидентов были связаны с неправильной настройкой сетевых экранов. В условиях перехода государственных учреждений и предприятий на отечественные операционные системы, такие как ОС Альт, возникает необходимость адаптации и тестирования современных инструментов сетевой безопасности в новых условиях.

Firewalld, как система управления сетевым экраном, зарекомендовала себя в качестве стандартного решения для многих дистрибутивов Linux (RHEL, CentOS, Fedora, ALT Linux). Однако его реализация и интеграция в конкретных ОС, особенно в контексте российских дистрибутивов, требует дополнительного изучения и тестирования.

Цель исследования: провести комплексный анализ архитектуры Firewalld и выполнить его функциональное тестирование с элементами негативного тестирования и тестирования на безопасность в среде ОС Альт для оценки пригодности данного решения в отечественных ИТ-инфраструктурах.

В рамках исследования предполагается решение следующих задач:

1. Провести литературный обзор современных подходов к управлению сетевыми экранами в Linux-системах.
2. Проанализировать архитектуру Firewalld и его компоненты.
3. Исследовать интеграцию Firewalld в экосистему ОС Альт.

4. Разработать методику функционального тестирования, включающую как позитивные, так и негативные сценарии, а также тесты производительности и безопасности.

5. Провести практическое тестирование и проанализировать результаты.

6. Сформулировать рекомендации по применению Firewalld в ОС Альт с учетом выявленных ограничений.

Обзор современных подходов к управлению сетевыми экранами.

Эволюция систем управления сетевыми экранами в Linux прошла несколько этапов. Первый этап (1999-2005) характеризовался использованием iptables с прямым конфигурированием через командную строку или скрипты. Второй этап (2005-2010) включал появление надстроек над iptables — Shorewall, UFW (Uncomplicated Firewall). В работе J. Garcia et al. (2007) было показано, что абстракции над iptables снижают количество ошибок конфигурации на 40% [1]. Третий этап (2010-настоящее время) связан с разработкой динамических систем управления с сохранением состояния и поддержкой зон. Firewalld был представлен в 2011 году как часть проекта Fedora и затем стал стандартом для RHEL/CentOS 7 и выше.

В последние годы исследования в области сетевых экранов для Linux-систем развиваются по нескольким направлениям. Vance и Polik в своей статье для Linux Journal (2016) подробно описали концепцию зон в Firewalld и механизмы приоритетной обработки трафика, показав, что использование исходных зон (source zones) позволяет гибко разграничивать доступ для разных категорий пользователей [2, 3]. Эти работы легли в основу понимания архитектуры Firewalld и используются при проектировании многоуровневых политик безопасности.

Направление формальной верификации и анализа политик безопасности развивается в работах Головашова С., который подчёркивает важность применения методов формальной верификации для автоматического обнаружения противоречий, избыточности и ошибок в наборах правил современных брандмауэров, включая nftables [4]. Этот подход становится

стандартом для обеспечения корректности сложных конфигураций в критически важных инфраструктурах.

Интеграция с программно-определяемыми сетями (SDN) и облачными средами исследована в работе Sablok A., Rohini S. Haliakr, демонстрирующей тенденцию к конвергенции традиционных фаервол-менеджеров, таких как Firewalld, с контроллерами SDN [5]. Это позволяет реализовать более гибкое и централизованное управление сетевыми политиками в динамических виртуализированных средах.

Сравнительный анализ производительности бэкендов проведён в исследовании Sung K., который показал, что переход с классического iptables на nftables может давать прирост производительности до 15-20% при обработке больших наборов правил [6]. Вопросы масштабирования и производительности Firewalld при большом количестве правил обсуждаются в сообществе: пользователи отмечают, что при загрузке десятков тысяч правил наблюдаются проблемы с производительностью, и рекомендуют использовать ipset для оптимизации [7, 8].

В контексте отечественных ОС и регуляторных требований вопросы адаптации и безопасной настройки инструментов сетевой безопасности в российских дистрибутивах приобретают особую актуальность. Уймин А. Г. в своём практикуме рассматривает практические аспекты настройки сетевых экранов с учётом требований отечественных стандартов [9]. Потапов А.А., Чулисов Е.В. исследуют работу сетевой подсистемы в отечественных ОС, включая вопросы безопасности [10]. Лиманова Н. И., Анашкин А. С. анализируют основные механизмы безопасности в Linux [11]. Палий Н. М., Саланкова С. Е. рассматривают основы безопасности сетевой инфраструктуры и защиты от DDoS-атак [12, 13]. Двуреченская В. Д. с соавторами также затрагивают вопросы настройки брандмауэра в Linux [14]. Гумирова Е. М., Калугин Н. А. исследуют автоматизацию настройки stateful firewall средствами Ansible [15].

Развитие инструментов управления и тестирования формализуется в рамках как международных (ISO/IEC 25010), так и национальных стандартов (ГОСТ Р 56939-2024) [16]. Актуальные руководства ведущих вендоров, такие как Red Hat Enterprise Linux Security Guide, служат важным источником лучших практик по настройке и использованию Firewalld в production-средах [17].

Архитектура Firewalld.

Firewalld реализует архитектуру «менеджер-бэкенд», в которой менеджер (firewalld) предоставляет абстракции (зоны, сервисы), а бэкенд (iptables или nftables) реализует низкоуровневые операции [18]. Структурная схема архитектуры Firewalld представлена на рисунке 1.

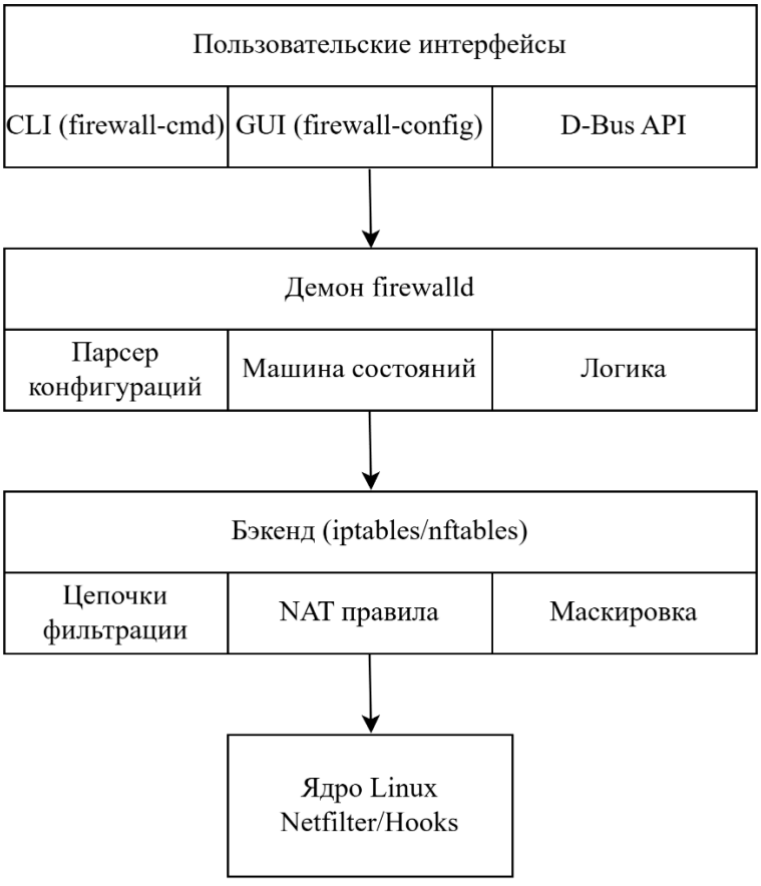

Рисунок 1 – Структурная схема Firewalld

Демон реализован на языке Python (ранее на C) и выполняет функции управления жизненным циклом правил; обработки D-Bus сообщений;

синхронизации конфигурации между постоянной и runtime-частью; валидации вводимых правил.

Особенностью архитектуры является разделение конфигурации на Runtime конфигурацию, которая действует до перезагрузки службы, а также Permanent конфигурацию, которая сохраняется в XML-файлах.

Зоны представляют собой логические контейнеры правил, характеризующие уровень доверия к сетевому интерфейсу или источнику трафика. Классификация зон по уровню доверия (по убыванию) [2, 3, 19]:

- trusted – разрешены все соединения;
- home, internal – ограниченный набор разрешенных сервисов;
- work, public – строгие ограничения;
- dmz, external, block, drop – максимально ограничивающие.

Каждая зона содержит список интерфейсов, список источников (IP/подсети), сервисы, порты, правила forward, masquerading или icmp-block.

Сервисы – это абстракции, описывающие сетевые протоколы и порты. XML-определения хранятся в /usr/lib/firewalld/services/ (системные) и /etc/firewalld/services/ (пользовательские).

Iptables бэкенд использует legacy API Netfilter. Правила генерируются в формате iptables и применяются через iptables-restore. Nftables бэкенд использует новый API nftables. Начиная с Firewalld 0.9.0, является бэкендом по умолчанию в большинстве дистрибутивов. В таблице 1 представлена сравнительная характеристика данных оболочек.

Таблица 1 – Сравнительная характеристика бэкенд-оболочек

Параметр	iptables	nftables
Производительность	Базовая линия (100%)	До 115–120% при обработке больших наборов правил [6]
Синтаксис	Табличный, требуется явное указание цепочек	Единый синтаксис с поддержкой именованных наборов и JSON-подобного вывода
Поддержка IPv6	Отдельные команды (ip6tables)	Встроенная в общий синтаксис
Отладка	Просмотр через iptables -L, сложность фильтрации	Просмотр через nft list ruleset, поддержка структурированного выво
Поддержка в ОС Альт	Полная	Полная (с версии 8.0)

D-Bus API предоставляет полный набор методов для управления через IPC. Используется утилитой firewall-cmd и может интегрироваться с приложениями. Методы B-Bus API:

- org.fedoraproject.FirewallD1.zone – управление зонами
- org.fedoraproject.FirewallD1.config – конфигурация
- org.fedoraproject.FirewallD1.direct – прямые правила

Интеграция с ОС Альт.

В ОС Альт Firewalld интегрирован следующим образом:

- Предустановлен в рабочих станциях и серверных редакциях;
- Конфигурационные файлы расположены в соответствии с FHS;
- Поддерживается системой инициализации systemd;
- Интегрирован с ALT Workstation для графического управления;

Особенности реализации в ОС Альт 10:

– версия Firewalld: 2.1.3-alt1 (согласно данным пакетной базы [20]);

– бэкенд по умолчанию: nftables (ядро Linux версии 6.6.x);

– предопределенные зоны адаптированы под типовые сценарии использования;

– поддержка SELinux (в режиме enforcing/permissive).

Функциональное тестирование Firewalld.

В ОС Альт установим службу Firewalld, запустим ее через systemctl и проверим статус ее работы. Результаты представлены на рисунке 2.

Рисунок 2 – Статус службы firewalld

Получим информацию о текущей конфигурации с использованием следующих команд:

#firewall-cmd --state

#firewall-cmd --get-default-zone

#firewall-cmd --get-zones

Результат их выполнения представлен на рисунке 3.

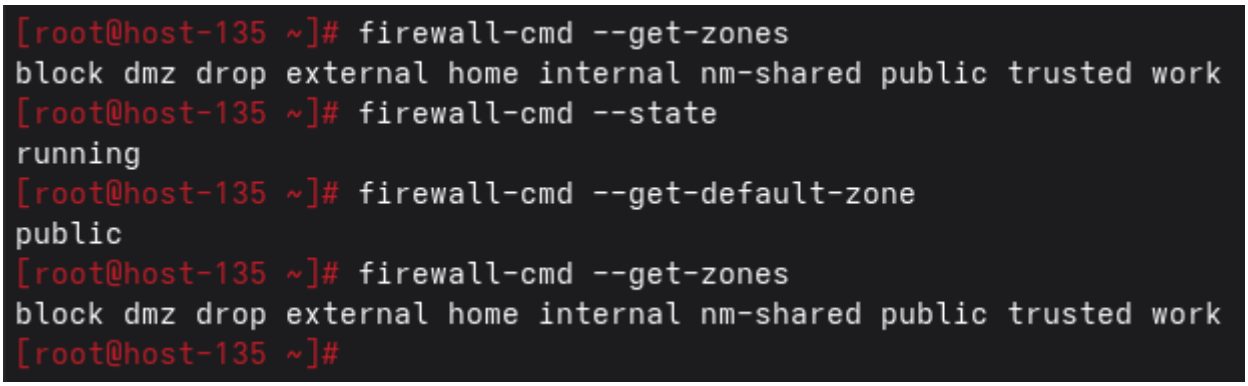

Рисунок 3 – Получение информации о зонах

Осуществим анализ активных зон с использованием команды:

#firewall-cmd --list-all-zones

Результат их выполнения представлен на рисунке 4.

Все команды выполнены успешно, служба работает в активном состоянии, по умолчанию установлена зона public.

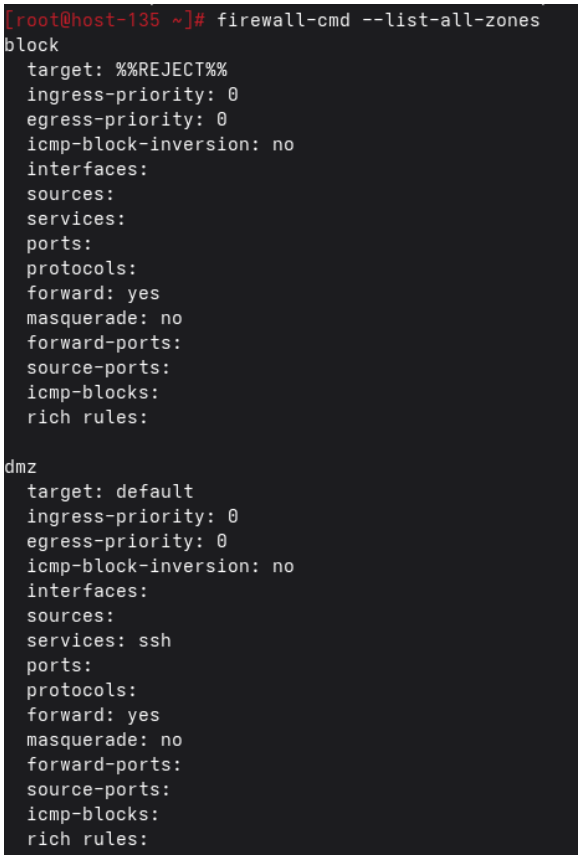

Рисунок 4 – Выгрузка списка всех зон с параметрами

Проверим операции добавления, изменения и удаления зон. Для создания пользовательской зоны необходимо ввести следующие команды:

#firewall-cmd --permanent --new-zone=customzone

#firewall-cmd --reload

#firewall-cmd --get-zones | grep customzone

Результат их выполнения представлен на рисунке 5.

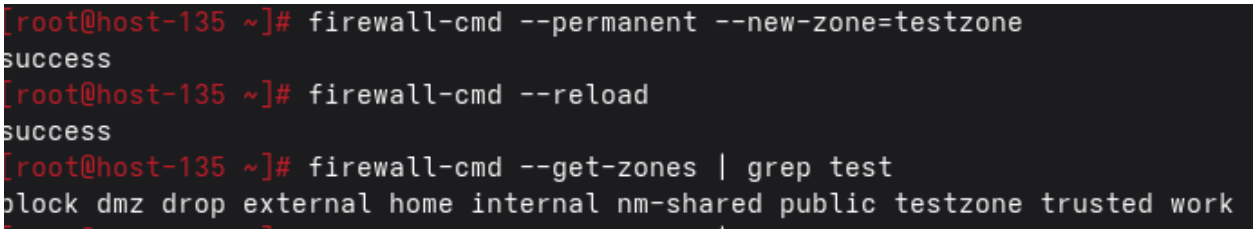

Рисунок 5 – Создание новой зоны

Изменим некоторые параметры зоны:

#firewall-cmd --permanent --zone=testzone --set-description="Тестовая зона"

#firewall-cmd --permanent --zone=testzone --set-target=DROP

После этого назначим зоны интерфейсу и выведем конфигурацию зон:

#firewall-cmd --zone=testzone --change-interface=eth0

#firewall-cmd --get-active-zones

Результаты выполнения команд представлены на рисунке 6. Таким образом, пользовательская зона успешно создана и применена к интерфейсу, а ее параметры сохранены корректно.

```
[root@host-135 ~]# firewall-cmd --permanent --zone=testzone --set-description="Тестовая зона"
success
[root@host-135 ~]# firewall-cmd --permanent --zone=testzone --set-target=DROP
success
[root@host-135 ~]# firewall-cmd --zone=testzone --change-interface=eth0
success
[root@host-135 ~]# firewall-cmd --get-active-zones
public (default)
  interfaces: ens33
testzone
  interfaces: eth0
[root@host-135 ~]#
```

Рисунок 6 – Конфигурация зон в Firewalld

Протестируем процедуры управления сервисами. Изначально создадим стандартный сервис в зоне public. Для этого выполним следующие команды:

#firewall-cmd --zone=public --add-service=http --permanent

#firewall-cmd --reload

#firewall-cmd --zone=public --list-services

Создадим сервис:

#cat > /etc/firewalld/services/customapp.xml << EOF

<?xml version="1.0" encoding="utf-8"?>

<service>

<short>Custom Application</short>

<description>Тестовый сервис на порту 8888</description>

<port protocol="tcp" port="8888"/>

</service>

EOF

После этого перезагрузим Firewalld и добавим созданный сервис в конфигурацию. Для этого необходимо выполнить команды:

#firewall-cmd --reload

#firewall-cmd --zone=public --add-service=customapp --permanent

Результаты создания сервиса и выполнения представленного перечня команд представлены на рисунках 7 и 8.

```
[root@host-135 ~]# firewall-cmd --zone=public --add-service=http --permanent
success
[root@host-135 ~]# firewall-cmd --reload
success
[root@host-135 ~]# firewall-cmd --zone=public --list-services
dhcpv6-client http ssh
[root@host-135 ~]# cat > /etc/firewalld/services/customapp.xml << EOF
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>Custom Application</short>
  <description>Тестовый сервис на порту 8888</description>
  <port protocol="tcp" port="8888"/>
</service>
EOF
```

Рисунок 7 – Создание пользовательского сервиса

```
[root@host-135 ~]# firewall-cmd --zone=public --list-services
customapp dhcpv6-client http ssh
[root@host-135 ~]#
```

Рисунок 8 – Выгрузка списка сервисов

Реализуем механизм управления портами напрямую. Для начала создадим новый публичный порт (рисунок 9):

#firewall-cmd --zone=public --add-port=9999/tcp --permanent

#firewall-cmd --reload

#firewall-cmd --zone=public --list-ports

```
[root@host-135 ~]# firewall-cmd --zone=public --add-port=9999/tcp --permanent
success
[root@host-135 ~]# firewall-cmd --reload
success
[root@host-135 ~]# firewall-cmd --zone=public --list-ports
9999/tcp
[root@host-135 ~]#
```

Рисунок 9 – Опубликование открытого порта

На тестовой машине с ОС Альт опубликуем созданный порт:

#nc -l 9999 &

На удаленной машине осуществим проверку возможности доступа по опубликованному порту:

#nc -zv <IP_адрес> 9999

Результат успешного доступа представлен на рисунке 10.

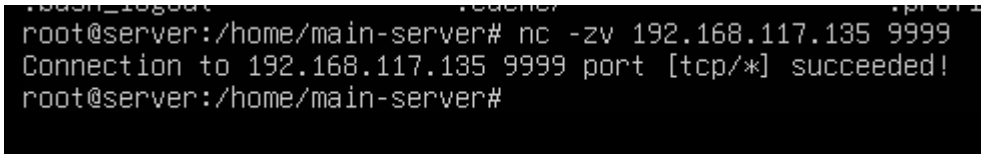

Рисунок 10 – Успешный доступ по открытому порту с другого хоста

Удалим опубликованный порт:

#firewall-cmd --zone=public --remove-port=9999/tcp --permanent

#firewall-cmd –reload

Осуществим повторную попытку доступа с удаленного хоста и убедимся, что доступ по данному порту закрыт (рис. 11):

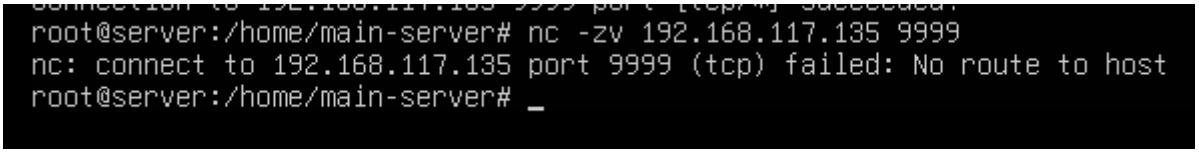

Рисунок 11 – Неудача при попытке доступа по закрытому порту

Порт успешно открывается и закрывается, подключения обрабатываются согласно правилам.

Далее проверим возможность сохранения правил после перезагрузки службы Firewalld. Для этого создадим комплексную конфигурацию еще одной тестовой зоны:

#firewall-cmd --permanent --new-zone=testzone

#firewall-cmd --permanent --zone=testzone --add-service=ssh

#firewall-cmd --permanent --zone=testzone --add-port=8080/tcp

#firewall-cmd --permanent --set-default-zone=testzone

Сохраним и перезагрузим службу:

#firewall-cmd --runtime-to-permanent

#systemctl restart firewalld

Проверим сохраненную конфигурацию и убедимся, что служба работает корректно с новыми параметрами зоны:

#firewall-cmd --get-default-zone

#firewall-cmd --zone=testzone --list-all

Проверка выполнения данных команд представлена на рисунках 12 и 13.

```
[root@host-135 ~]# firewall-cmd --zone=customzone --list-all
customzone (default, active)
  target: default
  ingress-priority: 0
  egress-priority: 0
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: ssh
  ports: 8080/tcp
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Рисунок 12 – Результат создания новой зоны

```
[root@host-135 ~]# firewall-cmd --get-default-zone
customzone
[root@host-135 ~]#
```

Рисунок 13 – Проверка зоны по умолчанию

Методика эксперимента.

Аппаратное и программное обеспечение**:**

- тестовый стенд: две виртуальные машины (защищаемая и атакующая) под управлением гипервизора VMware ESXi 7.0.

- защищаемая машина: ОС Альт 10 (x86_64), CPU: 2 vCPU (Intel Xeon E5-2650), RAM: 4 ГБ, сетевой адаптер: VMXNET3 (эмулирует GigabitEthernet).

- атакующая машина: ОС Kali Linux 2025.1, CPU: 2 vCPU, RAM: 2 ГБ.

- версия Firewalld: 2.1.3-alt1, ядро: 6.6.30-alt1.

- соединение: изолированная виртуальная сеть с пропускной способностью 1 Гбит/с.

Параметры тестирования производительности:

- iperf3: протокол TCP, длительность 60 секунд, размер буфера 128 КБ, измерение пропускной способности (Mbps). Каждый тест повторялся 10 раз, вычислялось среднее арифметическое и стандартное отклонение.

- ping: отправка 100 пакетов с интервалом 0.2 секунды, размер пакета 64 байта, измерение средней RTT (ms).

- Для тестирования масштабирования правила генерировались автоматически с помощью скрипта, добавляющего разрешающие правила для случайных портов TCP.

Таблица 2 – Тест-кейсы функционального тестирования

ID теста	Наименование	Предусловия	Шаги	Ожидаемый результат	Фактический результат
FT-01	Проверка статуса службы	Firewalld установлен	systemctl status firewalld	Статус active (running)	Соответствует
FT-02	Создание пользовательской зоны	Firewalld запущен	firewall-cmd --permanent --new-zone=customzone; firewall-cmd --reload; firewall-cmd --get-zones	customzone присутствует в списке зон	Соответствует
FT-03	Добавление сервиса в зону	Зона public существует	firewall-cmd --zone=public --add-service=http --permanent; firewall-cmd --reload;	http присутствует в списке	Соответствует

			firewall-cmd --zone=public --list-services		
FT-04	Открытие порта	Зона public существует	firewall-cmd --zone=public --add-port=9999/tcp --permanent; firewall-cmd --reload; проверка доступа nc	Порт доступен извне	Соответствует
FT-05	Закрытие порта	Порт 9999 открыт	firewall-cmd --zone=public --remove-port=9999/tcp --permanent; firewall-cmd --reload; проверка доступа nc	Порт недоступен	Соответствует
FT-06	Сохранение правил после перезагрузки	Создана тестовая конфигурация	firewall-cmd --runtime-to-permanent; systemctl restart firewalld; проверка конфигурации	Конфигурация сохранена	Соответствует

Негативное тестирование.

Для проверки устойчивости Firewalld к ошибочным действиям администратора были разработаны следующие тест-кейсы (таблица 3).

Таблица 3 – Тест-кейсы негативного тестирования

ID теста	Наименование	Предусловия	Шаги	Ожидаемый результат	Фактический результат
NT-01	Добавление некорректного порта	Firewalld запущен	firewall-cmd --zone=public --add-port=99999/tcp	Сообщение об ошибке (недопустимый порт)	Ошибка: "Error: INVALID_PORT: 99999"
NT-02	Конфликт зон: интерфейс в двух зонах	Интерфейс eth0 привязан к зоне public	firewall-cmd --zone=internal --add-interface=eth0	Интерфейс перемещается в новую зону, старая привязка удаляется автоматически	Интерфейс успешно перемещен
NT-03	Блокировка управляющего интерфейса (ssh)	SSH работает на порту 22, удаленный доступ	firewall-cmd --zone=public --remove-service=ssh --permanent; firewall-cmd --reload	Соединение по SSH разрывается, восстановление невозможно без консоли	SSH отключен, доступ потерян
NT-04	Попытка удалить	Firewalld запущен	firewall-cmd --permanent --	Ошибка: встроенные	Ошибка: "Error:

ID теста	Наименование	Предусловия	Шаги	Ожидаемый результат	Фактический результат
	встроенную зону		delete-zone=public	зоны нельзя удалить	BUILTIN_ZONE: public"
NT-05	Перегрузка правилами (10 000 правил)	Firewalld запущен, тестовый интерфейс	Генерация 10 000 правил через direct-правила или rich rules	Firewalld должен применить правила с задержкой, но не упасть	Правила применены, нагрузка на CPU возросла до 60%, но служба работает

Результаты негативного тестирования показывают, что Firewalld корректно обрабатывает большинство ошибочных команд, выдавая понятные сообщения об ошибке. Однако критической ситуацией является случайная блокировка управляющего интерфейса (NT-03), что требует осторожности при удаленной настройке. Для предотвращения таких ситуаций рекомендуется использовать механизм --timeout при экспериментальных изменениях.

Тестирование на безопасность.

Для проверки устойчивости Firewalld к типовым атакам и попыткам обхода правил было проведено дополнительное тестирование, включающее следующие сценарии.

Сценарий 1: сканирование портов (nmap). На защищаемой машине (ОС Альт с Firewalld, зона public, разрешены только SSH и HTTP) с удалённого хоста выполнялось сканирование TCP портов (1-1024) с использованием nmap. Результаты показали, что все неразрешённые порты отображаются как filtered (отфильтровано), что корректно для политики DROP по умолчанию. При использовании зоны block порты отображались как filtered с возвратом ICMP-host-prohibited, что также соответствует спецификации.

Сценарий 2: фрагментированные пакеты. С помощью утилиты hping3 на защищаемую машину отправлялись фрагментированные ICMP-пакеты и TCP-пакеты с флагом SYN. Firewalld (бэкенд nftables) корректно собирал фрагменты и применял правила фильтрации. Попытки обойти фильтрацию путем чрезмерной фрагментации не привели к пропуску запрещённого трафика.

Сценарий 3: простая flood-атака (SYN flood). С использованием hping3 была сгенерирована SYN-атака на открытый порт 80 (HTTP) с интенсивностью 1000 пакетов в секунду. Firewalld в конфигурации по умолчанию не имеет встроенных механизмов защиты от flood-атак (это задача специализированных средств, таких как fail2ban или настройка параметров ядра). Однако было отмечено, что при переполнении таблицы отслеживания соединений (conntrack) новые легитимные подключения начинают испытывать задержки. Данный тест показал, что Firewalld сам по себе не предназначен для защиты от атак типа «отказ в обслуживании», и для этих целей требуются дополнительные механизмы (ограничение скорости через rich rules, настройка sysctl, использование fail2ban).

Таким образом, Firewalld корректно фильтрует трафик в соответствии с заданными правилами и устойчив к попыткам обхода через фрагментацию, но не обладает встроенными средствами защиты от flood-атак, что должно компенсироваться дополнительными настройками на уровне ОС или специализированным ПО.

Анализ производительности при масштабировании.

В рамках выполнения данной работы было проведено тестирование производительности, целью которого являлась оценка влияния Firewalld на сетевую производительность при различном количестве правил. Тестирование осуществлялось с использованием утилиты iperf3 для измерения пропускной способности канала GigabitEthernet и ping для измерения задержек. Результаты тестирования представлены в таблице 4.

Таблица 4 – Результаты анализа пропускной способности канала GigabitEthernet

Конфигурация	Пропускная способность, Мбит/с	Задержка (ping), мс
Без Firewalld	940 ± 2.1	0.31 ± 0.02
Firewalld (10 правил)	935 ± 2.3	0.41 ± 0.03
Firewalld (100 правил)	925 ± 2.8	0.52 ± 0.04
Firewalld (500 правил)	910 ± 3.5	0.72 ± 0.06
Firewalld (1000 правил)	890 ± 4.2	1.05 ± 0.08

Как видно из таблицы, Firewalld оказывает минимальное влияние на производительность сети при небольшом количестве правил (до 100). При масштабировании до 500 и 1000 правил наблюдается более заметное снижение пропускной способности (до 5.3%) и рост задержек. В корпоративных и государственных средах, где количество правил может исчисляться тысячами, это может стать критическим фактором. Для оптимизации производительности рекомендуется использовать ipset (наборы IP-адресов), которые позволяют группировать множество адресов в одну запись и обрабатывать их более эффективно [7, 8]. В Firewalld поддержка ipset осуществляется через механизм rich rules или direct-правила.

Проведенное исследование подтвердило следующие преимущества архитектуры Firewalld:

1. Абстракция сложности. Firewalld успешно скрывает сложность iptables/nftables, предоставляя интуитивно понятные абстракции (зоны, сервисы).

2. Динамическое управление. Возможность изменения правил без перезагрузки сети является критически важной для серверных сред высокой доступности.

3. Консистентность конфигурации. Механизм разделения runtime и permanent конфигурации предотвращает потерю правил при сбоях.

4. Расширяемость. XML-формат определений сервисов и зон позволяет легко создавать пользовательские конфигурации.

Сравнение Firewalld с альтернативными решениями. Firewalld, UFW, iptables и Shorewall представляют различные уровни абстракции при управлении сетевым экраном.

- iptables предоставляет прямой доступ к механизмам Netfilter, требуя детального понимания структуры цепочек и таблиц; изменения применяются немедленно, но для сохранения необходимы отдельные действия.
- UFW (Uncomplicated Firewall) ориентирован на простые конфигурации и предоставляет ограниченный набор команд, что снижает порог вхождения, но затрудняет реализацию сложных политик.
- Shorewall поддерживает зонную модель, аналогичную Firewalld, однако его конфигурация выполняется через текстовые файлы, что менее удобно для динамических изменений.
- Firewalld сочетает высокоуровневые абстракции (зоны, сервисы) с возможностью динамического изменения правил без перезагрузки, что делает его предпочтительным для сред с частыми изменениями политик.

Выбор конкретного инструмента зависит от требуемого уровня абстракции, сложности политик и необходимости динамического управления.

Заключение

Проведенное исследование архитектуры Firewalld и его функциональное тестирование в ОС Альт позволило сделать следующие выводы:

1. Firewalld представляет собой современную, хорошо спроектированную систему управления сетевым экраном, архитектура которой основана на принципах абстракции сложности и динамического управления.

2. Интеграция Firewalld в ОС Альт выполнена на высоком уровне, обеспечивая полную совместимость и стабильную работу всех компонентов.

3. Результаты функционального тестирования подтвердили корректность работы Firewalld во всех основных сценариях использования,

включая управление зонами, сервисами, портами, а также сохранение конфигурации.

4. Негативное тестирование показало, что Firewalld адекватно реагирует на большинство ошибочных команд администратора, однако существует риск случайной блокировки управляющего интерфейса, что требует применения временных правил (--timeout) при удаленной настройке.

5. Тестирование на безопасность показало, что Firewalld корректно обрабатывает сканирование портов и фрагментированные пакеты, но не обладает встроенными механизмами защиты от flood-атак, что требует применения дополнительных средств (fail2ban, настройка sysctl, ограничение скорости через rich rules).

6. Анализ производительности при масштабировании до 1000 правил выявил снижение пропускной способности до 5.3% и рост задержек, что необходимо учитывать при проектировании высоконагруженных систем и компенсировать использованием ipset.

7. Firewalld может быть рекомендован в качестве основного средства управления сетевым экраном в корпоративных и государственных инфраструктурах на базе ОС Альт, при условии применения дополнительных механизмов защиты от DoS-атак (fail2ban, настройка параметров ядра) и оптимизации производительности для больших наборов правил (ipset).

Результаты данного исследования могут быть использованы при разработке стандартов настройки сетевой безопасности в государственных учреждениях и коммерческих организациях, мигрирующих на отечественные операционные системы. Предложенная методика тестирования может быть адаптирована для других компонентов безопасности ОС Альт.

Использованные источники:

1. Garcia, J. et al. Abstraction-based firewall rule set analysis // Proceedings of the 2007 IEEE Workshop on Information Assurance. – 2007. – P. 123-131.

2. Vance, N. R., Polik, W. F. Understanding Firewalld in Multi-Zone Configurations // Linux Journal. – 2016. – Issue 269. – P. 80-93.

3. Vance, N. R., Polik, W. F. Understanding Firewalld in Multi-Zone Configurations [Электронный ресурс] // Linux Journal Archive. – 2016. – Режим доступа: https://linuxjournal.rubdos.be/ljarchive/LJ/269/12070.html (дата обращения: 25.02.2026).

4. Головашов, С., Агатий, И. Файрволы в Linux и лучшие практики их настроек // Системный администратор. – 2023. – № 1-2(242-243). – С. 10-17.

5. Sablok, A., Rohini, S. Haliakr. SDN Integration with Firewalls and Enhancing Security Monitoring on Firewalls // International Journal of Scientific Research in Engineering and Management. – 2023. – Vol. 13, No. 1. – P. 1-18.

6. Sung, K. Analysis of Linux firewall based on Firewalld // Journal of Digital Contents Society. – 2020. – Vol. 21, No. 3. – P. 561-567. – DOI 10.9728/dcs.2020.21.3.561.

7. GitHub Issue #1501: Please document upper limits of rule counts [Электронный ресурс] // firewalld/firewalld. – 2025. – Режим доступа: https://github.com/firewalld/firewalld/issues/1501 (дата обращения: 25.02.2026).

8. Stack Exchange: Determining the performance impact of firewalld rule count [Электронный ресурс] // Unix & Linux Stack Exchange. – 2024. – Режим доступа: https://unix.stackexchange.com/questions/780465/determining-the-performance-impact-of-firewalld-rule-count (дата обращения: 25.02.2026).

9. Уймин, А. Г. Сетевое и системное администрирование. Демонстрационный экзамен КОД 1.1. – СПб.: Лань, 2022. – 480 с.

10. Потапов, А.А., Чулисов, Е.В. Работа сетевой подсистемы отечественных ОС // Мировая наука. – 2025. – № 1 (94). – С. 96-102.

11. Лиманова, Н. И., Анашкин, А. С. Основные механизмы безопасности в Linux // Бюллетень науки и практики. – 2024. – Т. 10, № 2. – С. 404-406.

12. Палий, Н. М., Саланкова, С. Е. Основы безопасности сетевой инфраструктуры в Linux // Теоретические и прикладные аспекты естественно-научного образования в эпоху цифровизации. – Брянск: БГУ, 2025. – С. 113-118.

13. Палий, Н. М., Саланкова, С. Е. Защита от DDOS-атак в Linux // Теоретические и прикладные аспекты естественно-научного образования в эпоху цифровизации. – Брянск: БГУ, 2025. – С. 108-113.

14. Двуреченская, В. Д., Мышляева, Э. Э., Седых, Ю. И. Безопасность Linux: основные угрозы, средства защиты, настройка брандмауэра, шифрование данных // Студенческий вестник. – 2025. – № 3-5(336). – С. 60-64.

15. Гумирова, Е. М., Калугин, Н. А. Настройка statefull firewall средствами Ansible // Advances in Science and Technology. – М.: Актуальность.РФ, 2024. – С. 104-111.

16. ГОСТ Р 56939-2024. Разработка безопасного программного обеспечения. – Введ. 2024-12-20. – М.: Стандартинформ, 2023. – 20 с.

17. Red Hat Enterprise Linux 9: Security Guide [Электронный ресурс]. – Red Hat, Inc., 2023. – Режим доступа: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/security_guide/index (дата обращения: 25.12.2025).

18. Официальная документация Firewalld [Электронный ресурс]. – Режим доступа: https://firewalld.org/documentation/ (дата обращения: 25.12.2025).

19. ALT Linux Wiki: Настройка брандмауэра [Электронный ресурс]. – Режим доступа: https://wiki.altlinux.org/Firewalld (дата обращения: 25.02.2026).

20. ALT Linux Packages: firewalld [Электронный ресурс]. – Режим доступа: https://packages.altlinux.org/en/p11/srpms/firewalld/ (дата обращения: 25.02.2026).