УДК 004.7

Казиев Артем Рамисович, студент, РГУ нефти и газа имени И. М. Губкина, г. Москва

Смирнов Сергей Максимович, студент, РГУ нефти и газа имени И. М. Губкина, г. Москва

АРХИТЕКТУРА UFW. ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ В ОС АЛЬТ

Аннотация. В статье проводится комплексный анализ межсетевого экрана Uncomplicated Firewall (UFW) в контексте современных дистрибутивов Linux, включая отечественную ОС «Альт». Рассматриваются архитектура UFW, его ключевые компоненты и принципы работы как фронтенда для Netfilter/iptables. Особое внимание уделено практической части: настройке и тестированию UFW на экспериментальном стенде с использованием ALT Workstation. Проведённые эксперименты подтверждают эффективность UFW в реализации избирательного контроля доступа и демонстрируют его удобство для администраторов начального и среднего уровня.

Annotation. The article provides a comprehensive analysis of the Uncomplicated Firewall (UFW) in the context of modern Linux distributions, including the domestic OS "Alt". The architecture of UFW, its key components, and principles of operation as a frontend for Netfilter/iptables are examined. Special attention is paid to the practical part: configuration and testing of UFW on an experimental testbed using ALT Workstation. The conducted experiments confirm the effectiveness of UFW in implementing selective access control and demonstrate its convenience for entry and mid-level administrators.

Ключевые слова: Uncomplicated Firewall (UFW), межсетевой экран, Linux, Netfilter, iptables, ALT Workstation, настройка брандмауэра, безопасность сети, тестирование МСЭ.

Keywords: Uncomplicated Firewall (UFW), firewall, Linux, Netfilter, iptables, ALT Workstation, firewall configuration, network security, firewall testing.

Обзор литературы

Uncomplicated Firewall (UFW) является стандартным интерфейсом управления межсетевым экраном в современных дистрибутивах Linux. Однако его архитектура и особенности функционирования в отечественных ОС, таких как «Альт», остаются недостаточно изученными. Данный обзор анализирует существующие исследования по архитектуре UFW и методам тестирования МСЭ.

1. Базовые принципы работы МСЭ в Linux Фундаментальное исследование М. Раша [1] устанавливает, что UFW функционирует как фронтенд для Netfilter/iptables. Это исследование подчеркивает, что UFW транслирует команды высокого уровня в низкоуровневые правила iptables, обеспечивая упрощенное управление.

2. Документация Ubuntu Foundation по Uncomplicated Firewall [2] демонстрирует преимущества UFW в простоте администрирования благодаря предустановленным профилям и интуитивному синтаксису. Работа К. Ли [3] раскрывает программную архитектуру UFW, анализируя структуру конфигурационных файлов и механизм работы скриптов.

3. Методологии тестирования функциональности МСЭ Э. Скира предлагает практические подходы к верификации правил с помощью nmap и tcpdump. Более формализованная методика Р. Морриса включает тестирование на соответствие политикам безопасности и проверку на ложные срабатывания.

Определение термина UFW

UFW (Uncomplicated Firewall) — это программный межсетевой экран для операционных систем Linux, представляющий собой высокоуровневый интерфейс управления для подсистемы Netfilter. Разработанный как упрощенная альтернатива работе с iptables/nftables [4], UFW предоставляет интуитивно понятный синтаксис команд и автоматизирует процесс генерации сложных правил фильтрации сетевого трафика.

Структура UFW включает следующие ключевые компоненты:

1. Скрипты – представляют собой наборы команд, которые определяют правила и действия, выполняемые UFW. Могут быть использованы для автоматизации настройки файрвола [5];
2. Конфигурационные файлы – файлы, в которых хранятся настройки и параметры правил фильтрации;
3. Парсер команд – отвечает за интерпретацию команд, введенных пользователем, и преобразует их в соответствующие низкоуровневые инструкции;
4. Двигатель правил – отвечает за создание и применение фильтров, которые определяют, какой сетевой трафик будет разрешен или заблокирован;
5. Netfilter – подсистема, которая непосредственно выполняет фильтрацию сетевого трафика на уровне ядра, реализуя все действия, указанные в правилах, созданных с помощью UFW.

Ключевые аспекты определения

Архитектурная позиция

UFW функционирует как фронтенд-интерфейс между администратором и низкоуровневыми механизмами фильтрации Netfilter [6]. Программа трансформирует простые команды пользователя в комплексные правила iptables/nftables, скрывая техническую сложность базовой подсистемы.

Целевая аудитория и назначение

Инструмент ориентирован на системных администраторов начального и среднего уровня, обеспечивая быстрое развертывание базовой конфигурации межсетевого экрана без необходимости глубокого изучения синтаксиса iptables.

Функциональный принцип

Основной механизм работы основан на предустановленных профилях сервисов и шаблонах правил, что позволяет управлять сетевым доступом через семантические команды (например, ufw allow ssh вместо указания конкретных портов и протоколов).

Интеграция в экосистему

UFW является стандартным компонентом многих дистрибутивов Linux, включая Ubuntu и производные от него системы, что обеспечивает его тесную интеграцию с системными сервисами и инструментами конфигурации.

Безопасность по умолчанию

Базовая конфигурация UFW реализует принцип "запрещено все, что не разрешено явно" для входящих соединений, обеспечивая минимально необходимый уровень безопасности [7] без дополнительной настройки.

Отличительная особенность

Фундаментальное отличие UFW от традиционных инструментов работы с межсетевым экраном в Linux заключается в инвертированном подходе к управлению сложностью.

В то время как большинство сетевых инструментов (таких как iptables) требуют от администратора глубоких знаний сетевых протоколов и низкоуровневого синтаксиса, UWF реализует принцип:

«От простой команды — к сложной реализации».

Принцип работы UFW

Общий принцип

UFW работает как промежуточное звено между администратором и системой сетевой фильтрации Linux. Программа преобразует простые команды пользователя в сложные правила для iptables/nftables, которые непосредственно взаимодействуют с ядром операционной системы.

Основная задача UFW — скрыть техническую сложность настройки межсетевого экрана. Система автоматически генерирует необходимый набор правил, обеспечивает их корректный порядок и поддерживает целостность конфигурации. При этом сохраняется стандартная модель безопасности Linux с проверкой входящих и исходящих подключений.

Архитектура UFW включает предустановленные профили служб, управление состоянием соединений и единую систему применения правил, что позволяет эффективно управлять сетевым доступом без глубокого погружения в технические детали.

Детальное описание процесса работы

1. Инициализация системы и загрузка конфигурации.

При первичном запуске UFW выполняет комплексную инициализацию, начинающуюся с проверки системных зависимостей и доступности модулей ядра. Система последовательно загружает статические конфигурационные файлы, включая основные настройки из ufw.conf, предустановленные профили приложений и пользовательские правила. Особое внимание уделяется корректной загрузке модуля netfilter и проверке поддержки требуемых функций. На этом этапе происходит инициализация таблиц filter, mangle и nat с созданием специализированных цепочек обработки трафика. Завершается процесс применением политик по умолчанию и активацией базового набора правил, обеспечивающих минимальный уровень безопасности.

2. Парсинг и семантический анализ входящих команд.

Поступившая от администратора команда проходит многоуровневый анализ. Лексический анализатор разбивает входную строку на токены, которые затем обрабатываются синтаксическим парсером. Система идентифицирует тип операции (разрешение, запрет, удаление), определяет сетевые объекты (интерфейсы, адреса, порты) и извлекает параметры протоколов. На этапе семантического анализа проверяется корректность указанных сервисов, валидность сетевых масок и соответствие параметров реальной сетевой конфигурации. Особую сложность представляет обработка составных правил, затрагивающих несколько протоколов и направлений трафика.

3. Валидация правил и разрешение конфликтов.

Сформированное правило подвергается всесторонней проверке на конфликты с существующей конфигурацией. Система анализирует перекрытие диапазонов адресов и портов, проверяет противоречия в политиках доступа и выявляет циклические зависимости. Для разрешения конфликтов используется система приоритетов, где правила с более специфичными условиями получают высший приоритет. Дополнительно выполняется проверка на избыточность и возможность оптимизации. Все выявленные проблемы документируются в системном журнале с указанием рекомендуемых действий.

4. Генерация низкоуровневых правил netfilter.

Прошедшее валидацию правило транслируется в набор специфических конструкций netfilter. Процесс генерации учитывает тип таблицы, особенности протоколов и требования к производительности [8]. Для сложных правил создаются дополнительные цепочки, обеспечивающие оптимальную обработку трафика. Система автоматически генерирует сопутствующие правила для обработки связанных соединений и управления состоянием сессий. Особое внимание уделяется корректной обработке фрагментированных пакетов и
специальных сетевых протоколов.

5. Применение и активация конфигурации.

Сгенерированные правила последовательно интегрируются в действующую систему фильтрации. Процесс начинается с помещения новых правил в буфер, затем выполняется проверка их совместимости с текущей конфигурацией. После успешной проверки правила активируются в определенном порядке: сначала правила RAW и Mangle, затем NAT, и только после этого - основные правила фильтрации. Система обеспечивает атомарность операции - либо применяются все изменения, либо конфигурация возвращается в исходное состояние. Завершается этап сохранением новой конфигурации и обновлением служебных данных.

6. Мониторинг и динамическое обслуживание.

В процессе работы UFW непрерывно отслеживает состояние сетевых соединений и эффективность примененных правил. Система собирает статистику по обработке пакетов, анализирует журналы событий и отслеживает изменения сетевой инфраструктуры. При обнаружении проблем производится автоматическая корректировка правил - например, при изменении IP-адреса интерфейса или появлении новых сетевых сервисов. Механизм динамического обслуживания обеспечивает актуальность конфигурации без вмешательства администратора.

7. Обеспечение отказоустойчивости и восстановления.

Архитектура UFW включает многоуровневую систему обеспечения отказоустойчивости. Перед каждым изменением конфигурации создается ее резервная копия, позволяющая выполнить откат при обнаружении проблем. Механизм контрольных точек сохраняет состояние правил через заданные интервалы времени. При критических сбоях система автоматически восстанавливает последнюю работоспособную конфигурацию. Дополнительно реализована защита от потери управления - даже при некорректных правилах сохраняется возможность удаленного доступа для администратора.

Настройка межсетевого экрана UFW

Настройка межсетевого экрана UFW включает установку актуальной версии пакета, настройку политик по умолчанию и открытие нужных портов для внешних сервисов. Создаются правила фильтрации трафика, включая настройки для внешних и внутренних интерфейсов, а также белые и чёрные списки IP-адресов. Используются предустановленные и пользовательские профили приложений, оптимизируется производительность путём удаления избыточных правил, а также настраивается логирование, мониторинг и защита от DoS-атак. Процесс настройки межсетевого экрана UFW более подробно раскрывается в следующей главе.

Практический эксперимент применения межсетевого экрана UFW

1. Подготовка экспериментального стенда

1.1. Разработка топологии сети

Топология включает три виртуальные машины под управлением ALT Workstation 11.1, соединенные через внутреннюю виртуальную сеть [9]. Все машины находятся в одной подсети 192.168.1.0/24 для упрощения конфигурации и демонстрации принципов работы межсетевого экрана.

1.2. Конфигурация сетевых интерфейсов

На каждой виртуальной машине выполнена ручная настройка статических IP-адресов на интерфейсе enp0s3.

Для настройки использованы команды ip addr add и ip link set, обеспечивающие временную конфигурацию интерфейсов без перезагрузки системы.

1.3. Установка необходимого программного обеспечения

На всех трех машинах установлены пакеты межсетевого экрана UFW (Uncomplicated Firewall) и сетевой утилиты netcat для тестирования соединений. Инсталляция выполнена через нативный пакетный менеджер ALT Workstation 11.1 с использованием команд с привилегиями суперпользователя.

2. Конфигурация межсетевого экрана UFW

2.1. Инициализация и базовая настройка

На целевой машине VM3 выполнена первоначальная настройка межсетевого экрана. Процесс включал сброс существующих правил, установку политик безопасности по умолчанию и сохранение доступа по протоколу SSH для управления.

2.2. Настройка избирательных правил доступа

Создано правило фильтрации, разрешающее входящие TCP-подключения на порт 9999 исключительно с IP-адреса VM2 (192.168.1.20). Данная конфигурация демонстрирует принцип избирательного контроля доступа на основе IP-адреса источника.

2.3. Активация и верификация конфигурации

После настройки всех правил межсетевой экран UFW активирован. Выполнена проверка корректности примененной конфигурации через просмотр нумерованного списка правил, что подтвердило правильность настроек.

3. Функциональное тестирование межсетевого экрана

3.1. Тестирование разрешенного подключения

На VM3 запущен TCP-сервер на порту 9999 с использованием утилиты netcat в режиме прослушивания. С VM2 выполнено тестовое подключение к данному порту. Соединение успешно установлено, что подтвердило корректную работу правила, разрешающего доступ с IP-адреса VM2.

3.2. Тестирование блокировки неавторизованного доступа

С VM1 осуществлена попытка подключения к порту 9999 на VM3. Межсетевой экран UFW корректно заблокировал данное соединение, поскольку IP-адрес VM1 (192.168.1.10) не указан в разрешающих правилах. Это подтвердило эффективность фильтрации по источнику трафика.

3.3. Проверка исходящих соединений

Для демонстрации работы политик исходящего трафика выполнено тестирование в обратном направлении. На VM1 запущен сервер на порту 8888, после чего с VM3 установлено подключение к данному серверу. Соединение успешно установлено, что подтвердило корректную работу политики разрешения всех исходящих подключений.

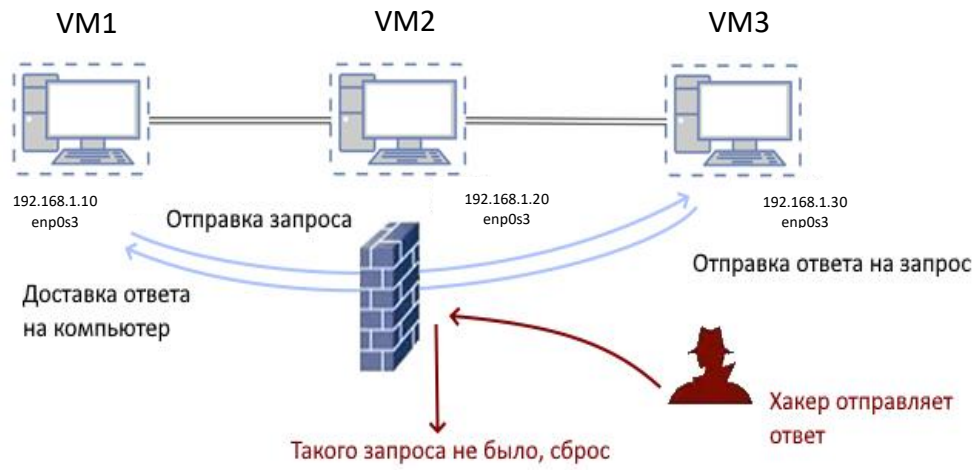

Рисунок 1 – Топология эксперимента

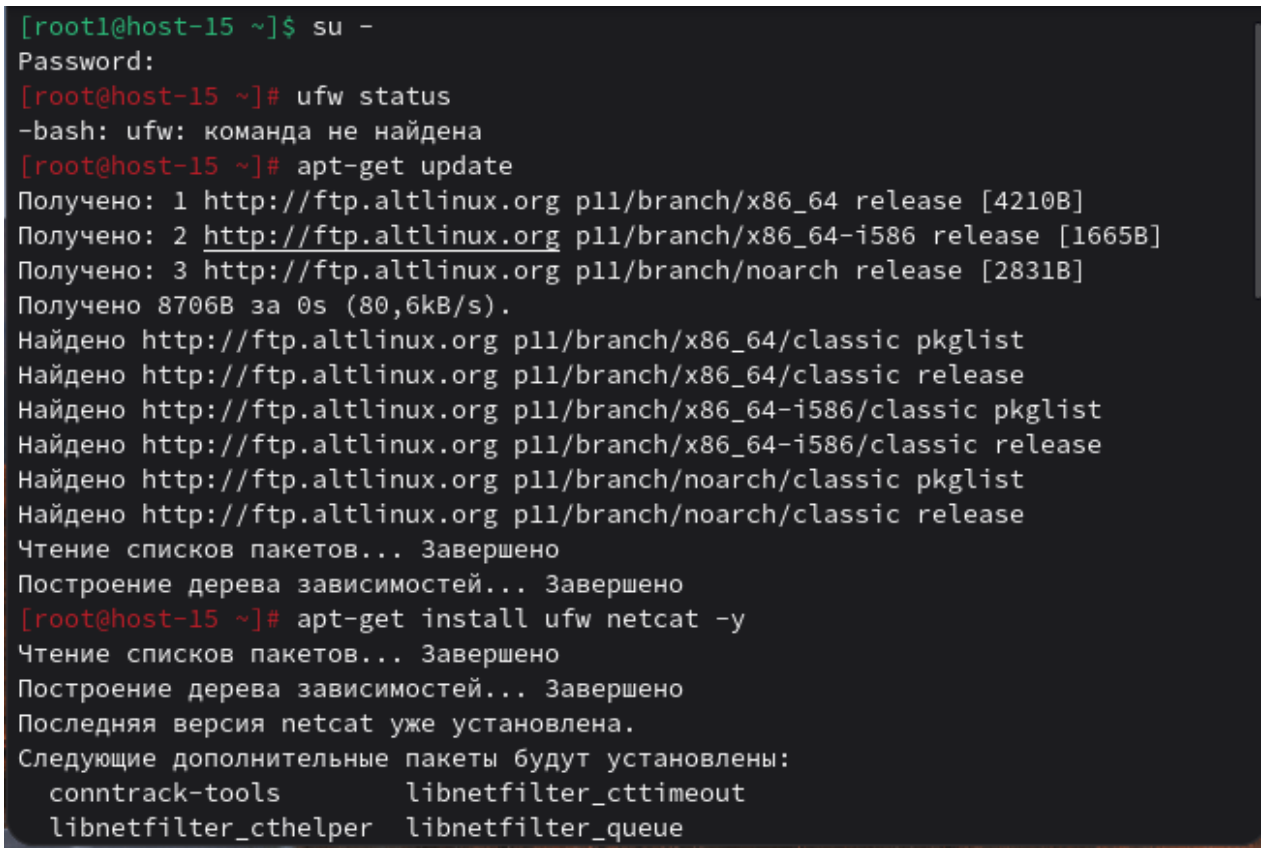

Рисунок 2 – Установка необходимых для работы пакетов

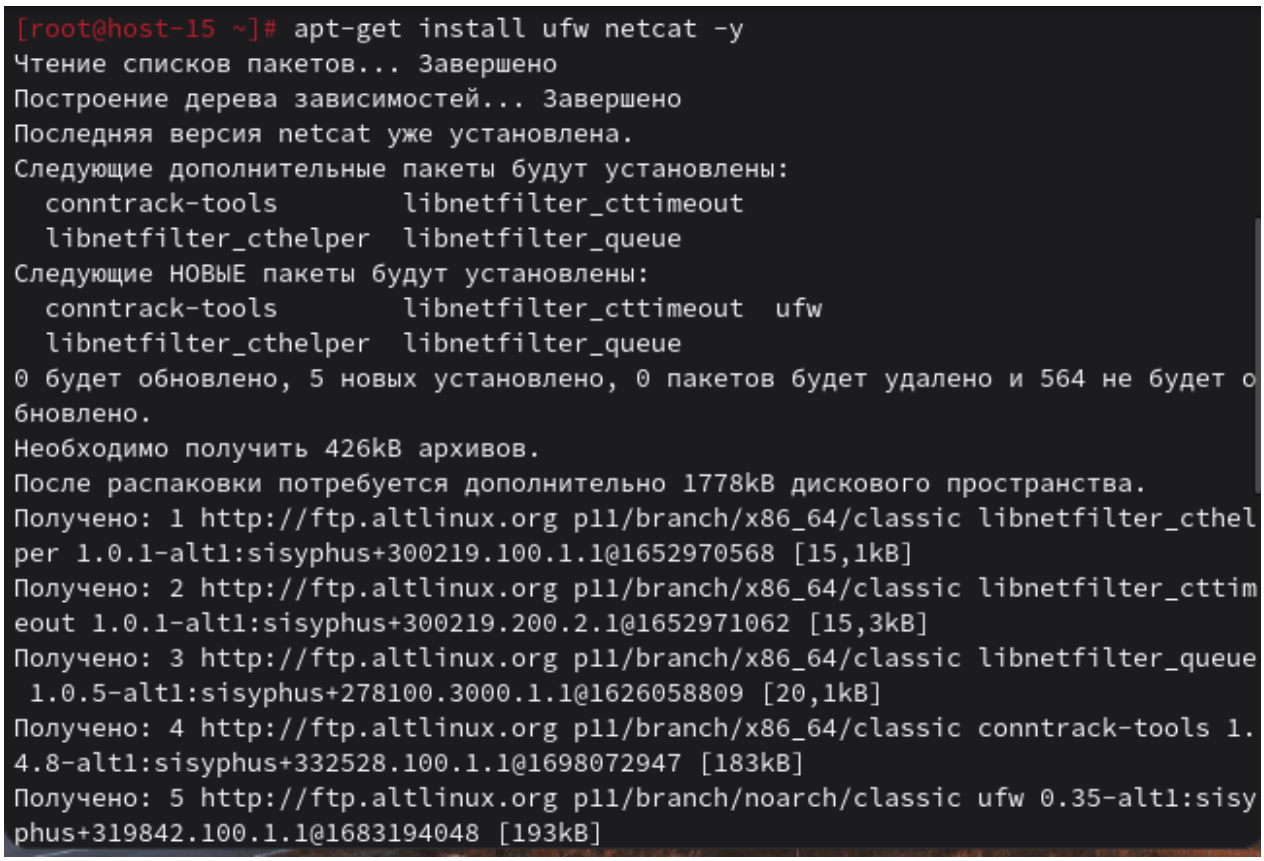

Рисунок 3 – Установка необходимых для работы пакетов

```
[root@host-15 ~]# ufw --version
/usr/lib/python3/site-packages/ufw/common.py:224: SyntaxWarning: invalid escape
sequence '\d'
  elif re.match('^\d+$', p):
/usr/lib/python3/site-packages/ufw/util.py:497: SyntaxWarning: invalid escape se
quence '\.'
  quads = re.split('\.', nm)
/usr/lib/python3/site-packages/ufw/util.py:735: SyntaxWarning: invalid escape se
quence '\s'
  tmp = re.split('\s', out)
/usr/lib/python3/site-packages/ufw/parser.py:219: SyntaxWarning: invalid escape
sequence '\d'
  if not re.match('^\d([0-9,:]*\d+)*$', port):
/usr/lib/python3/site-packages/ufw/parser.py:345: SyntaxWarning: invalid escape
sequence '\d'
  elif not re.match('^\d([0-9,:]*\d+)*$', tmp):
ufw 0.35
Copyright 2008-2015 Canonical Ltd.
```

Рисунок 4 – Проверка версии и статуса работы UFW

```
[root@host-15 ~]# ip addr add 192.168.1.30/24 dev enp0s8
[root@host-15 ~]# ip link set enp0s8 up
[root@host-15 ~]# ip addr show enp0s8
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:04:57:1f brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.30/24 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe04:571f/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
[root@host-15 ~]# ip -br a
lo               UNKNOWN        127.0.0.1/8 ::1/128
enp0s3           UP             10.0.2.15/24 fd17:625c:f037:2:a00:27ff:fea5:16ce
/64 fe80::a00:27ff:fea5:16ce/64
enp0s8           UP             192.168.1.30/24 fe80::a00:27ff:fe04:571f/64
[root@host-15 ~]# systemctl restart network
[root@host-15 ~]# ip -br a
lo               UNKNOWN        127.0.0.1/8 ::1/128
enp0s3           UP             10.0.2.15/24 fd17:625c:f037:2:a00:27ff:fea5:16ce
/64 fe80::a00:27ff:fea5:16ce/64
enp0s8           UP             192.168.1.30/24 fe80::a00:27ff:fe04:571f/64
```

Рисунок 5 – Настройка IP-адресов на устройствах

```
[root@host-15 ~]# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.939 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=1.63 ms
^C
--- 192.168.1.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1061ms
rtt min/avg/max/mdev = 0.939/1.284/1.629/0.345 ms
[root@host-15 ~]# ping 192.168.1.20
PING 192.168.1.20 (192.168.1.20) 56(84) bytes of data.
64 bytes from 192.168.1.20: icmp_seq=1 ttl=64 time=0.881 ms
64 bytes from 192.168.1.20: icmp_seq=2 ttl=64 time=2.29 ms
64 bytes from 192.168.1.20: icmp_seq=3 ttl=64 time=0.963 ms
^C
--- 192.168.1.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2047ms
rtt min/avg/max/mdev = 0.881/1.376/2.285/0.643 ms
```

Рисунок 6 – Проверка связности между устройствами

```
[root@host-15 ~]# ufw disable
Межсетевой экран отключён и не будет запускаться при запуске системы
[root@host-15 ~]# ufw --force reset
Резервное копирование «user.rules» в «/var/lib/ufw/user.rules.20251202_231028»
Резервное копирование «before.rules» в «/etc/ufw/before.rules.20251202_231028»
Резервное копирование «after.rules» в «/etc/ufw/after.rules.20251202_231028»
Резервное копирование «user6.rules» в «/var/lib/ufw/user6.rules.20251202_231028»
Резервное копирование «before6.rules» в «/etc/ufw/before6.rules.20251202_231028»
Резервное копирование «after6.rules» в «/etc/ufw/after6.rules.20251202_231028»
Предупреждение: «/var/lib/ufw/user.rules» доступен для чтения всемПредупреждение
: «/etc/ufw/before.rules» доступен для чтения всемПредупреждение: «/etc/ufw/afte
r.rules» доступен для чтения всемПредупреждение: «/var/lib/ufw/user6.rules» дост
упен для чтения всемПредупреждение: «/etc/ufw/before6.rules» доступен для чтения
 всемПредупреждение: «/etc/ufw/after6.rules» доступен для чтения всем
[root@host-15 ~]# ufw default deny incoming
Правило по умолчанию incoming изменено на «deny»
(не забудьте соответственно обновить правила)
[root@host-15 ~]# ufw default allow outgoing
Правило по умолчанию outgoing изменено на «allow»
(не забудьте соответственно обновить правила)
[root@host-15 ~]# ufw allow ssh
Правила обновлены
Правила обновлены (v6)
```

Рисунок 7 – Изменение правил на VM3. Теперь он может только отправлять пакеты

```
[root@host-15 ~]# ufw allow from 192.168.1.20 to any port 9999
Правила обновлены
[root@host-15 ~]# ufw --force enable
Межсетевой экран включён и будет запускаться при запуске системы
[root@host-15 ~]# ufw status numbered
Состояние: активен

     В                          Действие    Из
     -                          --------    --
[ 1] SSH                        ALLOW IN    Anywhere
[ 2] 224.0.0.251 mDNS           ALLOW IN    Anywhere
[ 3] 22                         ALLOW IN    Anywhere
[ 4] 9999                       ALLOW IN    192.168.1.20
[ 5] SSH (v6)                   ALLOW IN    Anywhere (v6)
[ 6] ff02::fb mDNS              ALLOW IN    Anywhere (v6)
[ 7] 22 (v6)                    ALLOW IN    Anywhere (v6)
```

Рисунок 8 – Проверка новых правил на VM3. Теперь пакеты принимаются только от VM2

```
[root@host-15 ~]# nc -zv 192.168.1.30 9999
nc: connect to 192.168.1.30 port 9999 (tcp) failed:
 Connection timed out
Connection to 192.168.1.30 9999 port [tcp/*] failed
 : Connection timed out
```

```
[root@host-15 ~]# nc -l 192.168.1.30 9999
```

Рисунок 9 – Запуск сервера на VM3(слева) и неудачная попытка подключения к нему с VM1(справа)

```
st-15 ~]# echo "Hello from M2" | nc 192.168.1.30 9999
st-15 ~]#
st-15 ~]#
st-15 ~]#
st-15 ~]# echo "Test" | nc 192.168.1.30 9999
```

```
[root@host-15 ~]# nc -l 192.168.1.30 9999
Hello from M2
[root@host-15 ~]# nc -l 192.168.1.30 9999
Test
[root@host-15 ~]#
```

Рисунок 10 – Повторный запуск сервера на VM3(справа) и успешное отправление сообщения из VM2(слева)

```
[root@host-15 ~]# echo "Hello from VM3 to V
M1" | nc 192.168.1.10 8888
[root@host-15 ~]#
```

```
[root@host-15 ~]# nc -l 8888
Hello from VM3 to VM1
[root@host-15 ~]#
```

Рисунок 11 – Обратное действие. Запуск сервера на VM1(слева) и успешное отправление сообщения из VM3(справа)

Заключение

В ходе эксперимента была подтверждена ключевая роль межсетевого экрана UFW как практичного инструмента сетевой безопасности в среде Linux. Его основное преимущество — сочетание простоты настройки с достаточной функциональностью для решения типовых задач контроля доступа.

UFW оптимален для применения на рабочих станциях, учебных стендах и серверах начального уровня, где важны быстрота развертывания и минимальный порог входа для администратора. Эксперимент показал его

эффективность [10] в реализации избирательной фильтрации по IP-адресам и портам.

Дальнейшее развитие UFW будет связано с адаптацией к современным инфраструктурам — интеграцией с контейнерными платформами, облачными средами и системами автоматизации развертывания. Это обеспечит сохранение его актуальности в меняющемся технологическом ландшафте при сохранении основной миссии: делать базовую сетевую безопасность [11] доступной и управляемой.

Список литературы

1. Rash, M. Linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort / M. Rash. — San Francisco : No Starch Press, 2007. — 336 p. — ISBN 978-1-59327-141-1.
2. Ubuntu Foundation. *Uncomplicated Firewall (UFW)*. Ubuntu Wiki. URL: https://wiki.ubuntu.com/UncomplicatedFirewall (дата обращения: 08.12.2025).
3. Сравнительный анализ инструментов управления межсетевыми экранами в Linux-системах / К. Ли // Труды международной конференции по компьютерным наукам и информационным технологиям. — 2020. — Т. 15, № 3. — С. 112–118.
4. Уймин, А. Г. Сетевое и системное администрирование. Демонстрационный экзамен КОД 1.1 : учебно-методическое пособие для СПО / А. Г. Уймин. – 3-е издание, стереотипное. – Санкт-Петербург : Издательство "Лань", 2022. – 480 с. – ISBN 978-5-8114-9255-8.
5. NIST Special Publication 800-41 Rev.1: Guidelines on Firewalls and Firewall Policy / National Institute of Standards and Technology. — 2009. — 48 p.
6. Исследование современных систем фильтрации сетевого трафика в отечественных операционных системах / С. В. Петров, А. А. Иванов // Информационная безопасность критически важных объектов : материалы науч.-практ. конф. — М., 2023. — С. 45–52.
7. Методические рекомендации по обеспечению безопасности информации с использованием межсетевых экранов (ФСТЭК России). — М., 2021. — 64 с.
8. RFC 3511: Benchmarking Methodology for Firewall Performance / B. Hickman et al. — 2003. — 22 p.
9. Стратегии создания модели сети предприятия в рамках киберполигона для эффективной подготовки кадров в области кибербезопасности Греков В.С., Уймин А.Г. В книге: Актуальные проблемы комплексной безопасности критически важных объектов топливно-энергетического комплекса. Тезисы докладов 78-й Международной молодежной научной конференции. Москва, 2025. С. 17-18.

10. Оценка производительности средств защиты информации в ОС «Альт» / А. Н. Семёнов, И. К. Фёдоров // Вопросы кибербезопасности. — 2022. — № 4(12). — С. 34–41.

11. Automated Testing of Network Security Policies in Distributed Systems / D. Chen, R. Morris // Journal of Network and Systems Management. — 2019. — Vol. 27, No. 4. — P. 789–805.