ОПЕРАЦИОННАЯ СИСТЕМА
РОСА ХРОМ 12 СЕРВЕР
Руководство по эксплуатации
Версия 1.0

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ

1.1. Аннотация документа

Этот документ содержит описание эксплуатации операционной системы РОСА ХРОМ 12 — сервер (далее – ОС), Данная ОС зарегистрирована в реестре российского ПО от 05.09.2016 №1607.

1.2. Условные обозначения

«Курсивом» в кавычках выделены термины из интерфейса системы.

Жирным шрифтом выделяются команды в консоли.

<Текст в угловых скобках> должен быть заменен на команду или её параметр.

[Квадратными скобками] обозначаются клавиши на клавиатуре.

1.3. Состав ОС

Ключевые особенности ОС Роса Хром 12 сервер:

- Кодовая база GNU Linux, совместимость с LSB
- Собственные репозитории, пакетная база на основе RPM
- Менеджер пакетов DNF/DNFdragora
- Ядра 5.10, 5.15, 6.1 с поддержкой SMP и HugeTLB
- Менеджер загрузки сервисов - systemd 249,
- Инсталлятор на основе anaconda с поддержкой сценариев установки
- Доступны графические оболочки KDE Plasma, GNOME, XFCE, LXQT
- Поддержка сетевого стека TCP/IPv4, TCP/IPv6, 803.2ad, Jumbo Frames
- Совместимость с Kaspersky Endpoint Security
- Совместимость с КриптоПро
- Поддерживаются тексты в кодировках UTF8, UTF16, CP866, WIN1251

2. ОБЩИЕ СВЕДЕНИЯ О РАБОТЕ В ОС

2.1. Установка и настройка

Установку ОС рекомендуется выполнять в соответствии с руководством по установке.

2.2. Интерфейсы операционной системы

По умолчанию ОС работает в текстовом консольном интерфейсе tty, при необходимости можно установить графическую оболочку с помощью менеджера пакетов, например командой **sudo dnf install task-iso-plasma5**

2.3. Справка

В консольном интерфейсе для получения справки о какой-то команде, нужно ввести **man <команда>,** например **man mc**. Для текстовых команд в консоли доступно автодополнение по кнопке [tab], например если ввести **справка** и нажать [tab] то получим все аргументы команды **справка**. Если же после команды **справка** ввести через пробел букву **ф** и нажать **[tab],** то автоматически введется параметр **файл.**

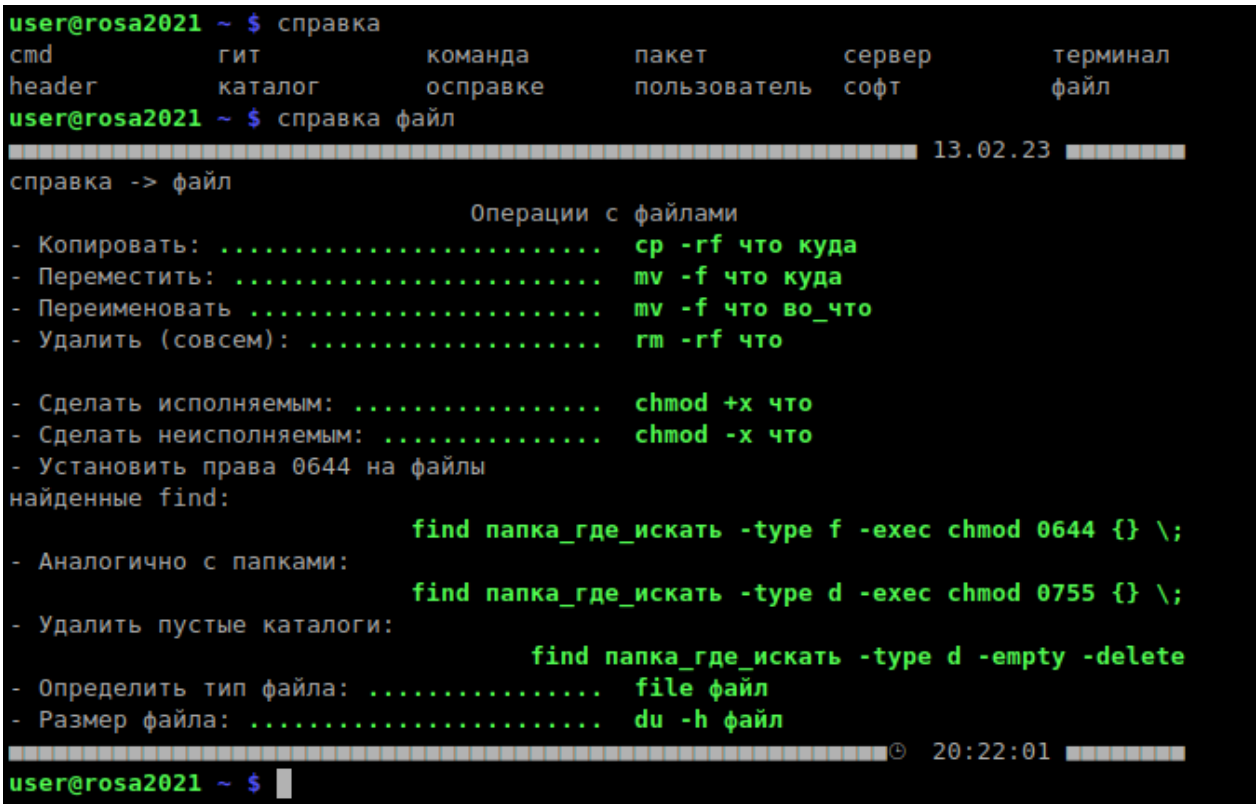


Рис. 1: Получение справки об операциях с файлами через консоль

2.4. Работа в сети

После установки системы, если компьютер подключен с помощью проводного соединения, ОС автоматически подключается к сети TCP/IP (при наличии в ней сервера DHCP) с помощью **NetworkManager**. Если соединение с сетью обеспечивается с помощью беспроводного соединения, его можно легко настроить с помощью псевдографического интерфейса **nmtui**, также доступен консольный интерфейс **nmcli**.

2.5. Администраторы и пользователи

При установке по-умолчанию, первый, созданный в инсталляторе, пользователь является пользователем-администратором с правом повышения своих полномочий до административных командой **sudo** с вводом пароля**.**

Как типичная Linux-система, ОС является многопользовательской. Администратор может управлять учетными записями других пользователей с помощью консольного интерфейса. Наиболее употребимые команды:

sudo useradd -m username (создать пользователя user1 и его профиль в /home)

sudo userdel -r username (удалить пользователя user1 и его профиль в /home со всеми файлами)

sudo usermod -ag wheel username (дать пользователю возможность повышения прав до администратора, включив его в группу wheel)

2.6. Установка и удаление программ

Для установки и удаления программ в ОС используется файловый менеджер **dnf.** Его часто используемые команды:

sudo dnf install <имяпакета> - установка

sudo dnf erase <имяпакета> - удаление

sudo dnf update - обновление системы из подключенных репозиториев

полный список команд **dnf** доступен в интерактивной справочной системе по команде **man dnf**.

3. ЦЕНТРАЛИЗОВАННОЕ УПРАВЛЕНИЕ УЧЕТНЫМИ ЗАПИСЯМИ

3.1. Настройка сервера домена IPA

Для централизованной аутентификации пользователей в локальной сети в ОС применяется технология IPA.

Перед установкой сервера IPA необходимо настроить сеть.

1. **sudo hostnamectl dc1.test.dom** (устанавливаем имя сервера)
2. используя интерфейс **nmtui** задаем для сервера статический IP
3. перезагружаемся, чтоб изменения вступили в силу

После перезагрузки и входа в систему необходимо установить пакет сервера IPA

sudo dnf install ipa-server

и запустить его инсталляцию командой

ipa-server-install

При инсталляции программа задаст ряд вопросов, рекомендуется использовать встроенный DNS а остальные настройки взять по-умолчанию. Установка может занять значительное время, для инсталляции и работы сервера IPA рекомендуется не менее 4 GiB оперативной памяти!

Управлять IPA-сервером можно через web-интерфейс по адресу **dc1.test.dom.**

3.2. Настройка клиента домена IPA

Перед установкой клиента IPA необходимо настроить сеть:

1. **sudo hostnamectl station.test.dom** (устанавливаем имя станции в домене)
2. используя интерфейс **nmtui** задаем для станции шлюз и DNS-сервер как IP IPA-сервера а также указываем домен поиска в test.dom
3. перезагружаемся, чтоб изменения вступили в силу

Если все сделано правильно, команда **ping dc1** должна отрабатывать корректно, показывая соединение с сервером IPA.

После перезагрузки и входа в систему необходимо установить пакет клиента IPA

sudo dnf install ipa-client

далее нужно запустить инсталляцию и вход в домен командой

sudo ipa-client-install --mkhomedir

понадобится ввести пароль администратора домена!

После перезагрузки можно входить доменным пользователем.

4. АУДИТ ПОЛЬЗОВАТЕЛЕЙ В СИСТЕМЕ

4.1. Введение

Аудит (audit) — это механизм аудита безопасности в ядре Linux, который позволяет отслеживать события в системе. Он может использоваться для мониторинга доступа к файлам, сетевых подключений, а также других системных событий.

В пространстве пользователя работает сервис auditd, который получает от ядра ОС информацию о произошедших событиях и обрабатывает ее в соответствии с настройками.

Ведение логов аудита обычно решает 2 задачи:

- протоколирование происходящего для разбора в случае действий злоумышленника;
- протоколирование происходящего для отладки работы программ.

4.2. Управление службой аудита

Служба auditd по-умолчанию включена, однако при необходимости ее можно включить в автозагрузку при старте вручную:

```
sudo systemctl enable auditd
```

Выключить так:

```
sudo systemctl mask auditd
```

Остановить так:

```
sudo systemctl kill auditd
```

4.3. Настройка конфигурации аудита

Конфигурационный файл находится по адресу **`/etc/audit/auditd.conf`**. Его содержимое по умолчанию подходит для работы.

Некоторые из наиболее важных параметров:

- *max_log_file*: максимальный размер журнального файла аудита в мегабайтах.
- *num_logs*: максимальное количество журнальных файлов аудита.
- *log_file*: путь к журнальному файлу аудита.
- *log_group*: группа, которой принадлежит журнальный файл аудита.

- *write_logs*: писать ли (yes — писать, no — не писать) события аудита в файл **`/var/log/audit/audit.log`**. Можно выставить no, тогда события аудита будут попадать в **systemd-journald**, но не будут дублироваться в еще один файл. Утилита **`ausearch`** читает именно этот файл, поэтому ей нельзя будет пользоваться.

  Больше информации можно получить в руководстве **`man auditd.conf`.**

4.4. Настройка правил аудита

Правила аудита — это инструкции, какие события нужно подвергать аудиту, то есть логировать.

Правила аудита задаются с помощью утилиты **`auditctl`**, которая взаимодействует с сервисом **auditd** и передает ему команды.

При запуске **auditd** читаются правила из файлов **`/etc/audit/audit.rules`** и **`/etc/audit/rules.d/*.rules`**. Каждая строка в этих файлах представляет собой аргументы для **`auditctl`**. Например, строка:

-w /etc/passwd -p wa -k passwd-file

эквивалентна запуску команды:

sudo auditctl -w /etc/passwd -p wa -k passwd-file

Запись в файл позволяет загружать правила аудита автоматически при запуске ОС.

4.5. Выборка из логов аудита - ausearch

Утилита **ausearch** — это инструмент для анализа журналов аудита, созданных с помощью системы аудита Linux. Журналы аудита содержат записи о действиях, совершаемых в системе, таких как изменение файлов, запуск процессов или использование сетевых ресурсов. Утилита **ausearch** предназначена для поиска и анализа этих записей.

Она может выполнять поиск записей журнала аудита по различным критериям, таким как время, пользователь, тип события и др.

Для примера, можно использовать следующую команду для поиска всех записей журнала аудита, связанных с изменением файла **/etc/passwd**:

sudo ausearch -f /etc/passwd

Результаты поиска будут выведены на экран. Каждая запись журнала аудита будет содержать информацию о времени события, типе события, пользователе, который совершил действие, и другие подробности.

Одной из наиболее полезных функций утилиты **ausearch** является возможность использования ключей для выполнения расширенного поиска. Например, следующая команда выполняет поиск записей журнала аудита, связанных с запуском процесса с именем "httpd":

sudo ausearch -c httpd

Другим полезным ключом является ключ -i, который позволяет просмотреть подробную информацию о конкретной записи журнала аудита. Например, следующая команда выводит подробную информацию о записи журнала аудита:

sudo ausearch -i 12345

где 12345 — это идентификатор записи журнала аудита.

Утилита **ausearch** также может использоваться для выполнения аудита безопасности. Например, можно настроить систему аудита таким образом, чтобы она записывала информацию о всех неудачных попытках входа в систему. Затем можно использовать утилиту **ausearch** для поиска этих записей журнала аудита и выявления потенциальных угроз безопасности.

Также утилита **ausearch** поддерживает несколько форматов вывода, включая форматы текста и JSON. Это может быть полезно для автоматизации процесса анализа журналов аудита.

4.6. Примеры

4.6.1. Отслеживание изменений в директории или файле

Для отслеживания всех изменений в директории **/var/log** добавьте следующее правило:

sudo auditctl -w /var/log -p w -k var_log_changes

Это правило отслеживает любые операции записи (w) в директорию **/var/log** и связывает их с ключевым словом (tag) var_log_changes.

Проверьте, что правила аудита корректно добавлены:

sudo auditctl -l

Эта команда выводит список текущих правил аудита.

Запишите текст в файл внутри этой директории:

echo text | sudo tee /var/log/testfile

Просмотрите журнальные файлы аудита, используя утилиту ausearch. Чтобы просмотреть все записи аудита, связанные с директорией /var/log, используйте следующую команду:

sudo ausearch -k var_log_changes

Эта команда выводит все записи аудита, связанные с ключевым словом **var_log_changes**, в т.ч. событие записи в файл **`/var/log/testfile`.**

4.6.2. Отслеживание запуска определенного приложения

Правило аудита:

sudo auditctl -a exit,always -F path=/usr/bin/myapp -F perm=x -k myapp_execution

Это правило отслеживает каждый выход (exit) из приложения **/usr/bin/myapp** с правами на выполнение (x) и связывает его с ключевым словом **myapp_execution**.

Поиск событий осуществляется по аналогии:

sudo ausearch -k myapp_execution

4.6.3. Отслеживание системных вызовов

Правило аудита для отслеживания 64-битных системных вызовов execve() от root, т.е. отслеживания запуска программ от root:

sudo auditctl -a exit,always -F arch=b64 -S execve -F uid=0 -k authentication_events

Это правило отслеживает каждый вызов функции execve (используется для запуска исполняемых файлов) с идентификатором пользователя (uid) равным 0 (root) и связывает его с ключевым словом authentication_events. Т.к. 64-битные ОС поддерживают запуск и 32-битных программ, подвергнем аудиту то же самое в 32-битном варианте:

sudo auditctl -a exit,always -F arch=b32 -S execve -F uid=0 -k authentication_events

4.6.4. Отслеживание доступа к конфиденциальным файлам

Правило аудита:

sudo auditctl -a exit,always -F arch=b64 -S open -F dir=/etc/sensitive -F success=0 -k sensitive_files_access

Это правило отслеживает каждый вызов функции open для любого файла в директории **/etc/sensitive**, который заканчивается неудачей (success=0) и связывает его с ключевым словом **sensitive_files_access**.

4.6.5. Отслеживание сетевых подключений

Правило аудита:

sudo auditctl -a exit,always -F arch=b64 -S bind -S connect -F success=0 -k network_events

Это правило отслеживает каждый вызов функций bind() и connect() для любого сетевого подключения, которое заканчивается неудачей (success=0) и связывает его с ключевым словом network_events.

4.6.6. Запись в журнал аудита при подключении устройства USB

Правило аудита:

sudo auditctl -w /dev/bus/usb -p rwxa -k usb

Теперь система будет записывать информацию при подключении устройства USB в журнал аудита.

Чтобы найти связанные с подключением устройства USB записи журнала аудита, выполните следующую команду **ausearch**:

sudo ausearch -f /dev/bus/usb -k usb -i

Эта команда найдет все записи журнала аудита, связанные с ключом usb и событием OPEN, которое относится к подключению устройства USB.

4.7. Интеграция с SIEM

Интеграция auditd со специализированными системами сбора событий информационной безопасности (SIEM) возможна двумя способами, в зависимости от SIEM:

- через плагин audisp (необходимо установить пакет audispd-plugins)
- через отсылку событий в SIEM как сервер syslog (RFC 5424)

5. СЕТЕВОЕ ЭКРАНИРОВАНИЕ (ФАЙРВОЛЛ, БРАНДМАУЭР)

5.1. О NFTables

Стандартным сетевым экраном в ОС принят nftables.

nftables — проект netfilter, целью которого ставится замена существующего набора межсетевых экранов iptables/ip6tables/arptables. Была разработана новая система фильтрации пакетов, добавлена пользовательская утилита ntf, а также создан слой совместимости с iptables/ip6tables. nftables использует набор хуков, систему отслеживания соединений, систему очередей и подсистему логирования netfilter.

Как и iptables, nftables построен на правилах, определяющих действия. Эти правила содержаться в таблицах (table), к которым прикреплены цепочки (chain). Цепочка может содержать набор правил и исполняется в хуках (hook) netfilter. Таблица специфична для каждого из протоколов (arp/ipv4/ipv6). Одно из основных отличий от iptables заключается в том, что больше нет предопределенных таблиц и цепочек.

Таблицы (table):

- table - это контейнер для ваших цепочек (chain).

Семейства таблиц (family):

- ip: используется для цепочек, связанных с IPv4.
- ip6: используется для цепочек, связанных с IPv6.
- arp: используется для цепочек, связанных с ARP.
- bridge: используется для соединения связанных цепей.
- inet: смешанные цепочки ipv4 / ipv6
- netdev: используется для цепочек, которые фильтруют в начале стека

Цепочки (chain):

- chain - используются для группировки правил.
- base chain - имеет зарегистрированный крючок, тип и приоритет.
- other chain - не привязаны к хуку и по умолчанию не видят никакого трафика.

Существует три вида цепочек:

1. filter - стандартный тип. фильтрует пакеты

2. nat - протоколы ip/ip6/inet. хуки prerouting, input, output, postrouting. преобразование NAT
3. route - протоколы ip/ip6. хуки output. маршрутизация

Хук (hook):

hook - это представление того, где происходит обработка, на каком этапе.

- prerouting: это происходит до принятия решения о маршрутизации, все пакеты, входящие в машину, попадают в это правило.
- input: все пакеты для локальной системы попадают в это правило.
- forward: пакеты не для локальной системы, те, которые необходимо пересылать, попадают в это правило.
- output: пакеты, исходящие из локальной системы, попадают в это правило.
- postrouting: эта ловушка возникает после того, как решение о маршрутизации было принято, все пакеты, покидающие машину, попадают в это правило.
- priority- относится к числу, используемому для упорядочивания цепочек или для установки их между некоторыми операциями.
- policy- это утверждение для тех пакетов, которые явно не приняты или отклонены в содержащихся правилах.
- accep: принять пакет.
- drop: отбросить пакет.
- queue: поставить пакет в очередь в пользовательское пространство и остановить оценку набора правил..
- continue: продолжение
- return: возврат из текущей цепочки и переход к следующему правилу последней цепочки. В базовой цепочке это эквивалентно accept.

Правила (rules):

rule - определяют, какое действие необходимо выполнить для каких пакетов. Правила прикреплены к цепочкам . Каждое правило может иметь выражение для сопоставления пакетов и одно или несколько действий, выполняемых при сопоставлении.

statements - представляют собой действие, которое должно быть выполнено при совпадении правила.

- accept : принять пакет и остановить оценку набора правил.
- drop : отбросить пакет и остановить оценку набора правил.
- reject : отклонить пакет с сообщением icmp.

- queue: поставить пакет в очередь в пользовательское пространство и остановить оценку набора правил.
- continue: продолжить
- return : возврат из текущей цепочки и переход к следующему правилу последней цепочки. В базовой цепочке это эквивалентно accept.
- jump : продолжить с первого правила . Он будет продолжен в следующем правиле после выдачи оператора возврата.
- goto : аналогично прыжку, но после новой цепочки оценка продолжится с последней цепочки вместо той, которая содержит оператор goto.
- handle - внутренний номер идентифицирующий определенное правило.
- position - это внутренний номер используемый для вставки правил до пределенного handle.

5.2. Установка и запуск Nftables

В РОСА Хром по умолчанию уже установлен **nftables**.

Если по каким-то причинам его пакет удален, **nftables** можно установить следующей командой:

sudo dnf install nftables

Запуск сервиса nftables:

sudo systemctl start ntfables.service

Чтобы включить его запуск при старте системы:

sudo systemctl enable nftables.service

5.3. Работа с nftables

Рассмотрим базовый пример закрытие/открытие порта.

Для начала, создадим таблицу:

sudo /usr/sbin/nft add table inet my_table

формат: nft add table <family> <name>

- <family> - один из протоколов ip/ip6/inet/arp/bridge/netdev
- <name> - имя нашей новой таблицы

Помотрим, что она появилась у нас:

sudo nft -a list ruleset

Теперь добавим цепочку:

sudo /usr/sbin/nft add chain inet my_table input { type filter hook input priority 0 \; policy accept \; }

формат: nft add chain <family> <name table> <name chain> {hook}

- [family] - один из протоколов ip/ip6/inet/arp/bridge/netdev
- [name table] - имя нашей таблицы
- [name chain] - имя нашей новой цепочки

Помотрим, что она появилась у нас:

sudo /usr/sbin/nft -a list ruleset

Добавим правило, которым ограничим доступ к 80 порту:

sudo /usr/sbin nft add rule inet my_table input tcp dport 80 drop

формат: nft add rule <family> <name table> <protocol> <dport> <port> <event>

- [family] - один из протоколов nftable ip/ip6/inet/arp/bridge/netdev
- [name table] - имя нашей таблицы
- [name chain] - имя нашей новой цепоки
- [protocol] - протокол tcp/udp/icmp и т.д.
- [dport] - входящий порт (destination port}
- [port] - номер входящего порта
- [event] - что делать drop/accept/jump и т.д.

Логично предположить, что для открытия порта нам нужно выполнить:

sudo /usr/sbin nft add rule inet my_table input tcp dport 80 accept

Но получить доступ к 80 порту после этого нам не удастся, и если мы заглянем в список правил, будет видно, что сначала все пакеты на порт 80 блокируются, а потом "пропускаются". Но в нашем случае , пропускать уже нечего, т.к. изначально на этом порту всё блокируется.

Нам нужно удалить правило с drop, для этого мы обращаем внимание на отметку handle 2 в выводе, и с её помощью удаляем запрещающее правило:

sudo /usr/bin/nft delete rule inet my_table input handle 2

проверяем, что у нас получилось:

sudo /usr/sbin/nft -a list ruleset

Теперь доступ к 80 порту у нас есть в полном объеме.

Если мы хотим полной очистки таблицы, то выполняем:

sudo /usr/sbin/nft delete table inet my_table

Если хотим полностью очистить файрвол (все цепочки, таблицы), то выполняем:

sudo /usr/sbin/nft flush ruleset

6. Списки контроля доступа ACL

6.1. Описание технологии

ACL (Access Control List, список контроля доступа) - это механизм в Linux, который позволяет более гранулярно управлять доступом к файлам и папкам, чем это позволяют права доступа по умолчанию. С ACL вы можете установить права доступа для конкретных пользователей и групп пользователей на конкретные файлы и папки.

ACL в Linux обычно состоит из двух типов прав: пользовательские права и групповые права. Пользовательские права позволяют устанавливать права доступа для конкретных пользователей, а групповые права - для групп пользователей.

Преимущество использования ACL заключается в том, что он позволяет установить более точные права доступа, чем это возможно с помощью стандартных прав доступа на основе пользователей и групп. Например, если у вас есть файл, к которому нужен доступ только для нескольких пользователей, вы можете установить ACL для этого файла, чтобы ограничить доступ только для этих пользователей, не давая права доступа всем остальным пользователям на системе.

Команды для работы с ACL в Linux включают **getfacl** и **setfacl**. Команда **getfacl** используется для просмотра текущих прав доступа ACL на файлы и папки, а команда **setfacl** используется для изменения или добавления новых прав доступа ACL на файлы и папки.

Знание ACL может быть полезно для системных администраторов, которые хотят более тонко настраивать доступ к файлам и папкам на своих серверах.

В Linux ACL хранятся в расширенных атрибутах (extended attributes, xattr), которые являются дополнительными метаданными, хранящимися в файловой системе. Большинство файловых систем, в т.ч. ext4, BTRFS, XFS, поддерживают хранение ACL в xattr. Однако, не все файловые системы поддерживают ACL и хранение их в xattr. Например, файловая система FAT32 не поддерживает ACL вообще.

6.2. Установка пакетов

Обычно пакет с утилитами для работы с ACL уже предустановлен, при необходимости установите его командой **sudo dnf install acl**.

6.3. Пример настройки ACL

6.3.1. Постановка задачи

Пусть имеется:

- каталог /media/buhi (на поддержерживающей ACL файловой системе: ext4, btrfs, xfs и др.)
- группа пользователей buhi (бухгалтерия)
- пользователи ivanova, petrova, sidorova, входящие в группу buhi

Нужно сделать так, чтобы все пользователи из группы buhi — ivanova, petrova, sidorova — могли:

- читать файлы в /media/buhi
- создавать новые файлы и каталоги
- изменять существующие файлы

6.3.2. Настройка ACL для существующих файлов

Выставьте ACL на каталог /media/buhi для группы "buhi", чтобы ее пользователи могли записывать и читать файлы в этом каталоге. Для этого выполните команду:

sudo setfacl -R -m g:buhi:rwx /media/buhi

Эта команда добавляет права для группы "**buhi"** на папку **/media/buhi** и все ее подкаталоги и файлы. Опция **-R** означает, что изменения должны применяться рекурсивно для всех подкаталогов и файлов. Опция **-m** означает, что мы меняем существующие права. **g:buhi:rwx** означает, что мы выставляем права для группы **"buhi"** на чтение, запись и выполнение.

Проверьте, что ACL настроены правильно. Для этого выполните команду:

sudo getfacl /media/buhi

В выводе должно быть что-то похожее на:

```
# file: /media/buhi/
# owner: <owner>
# group: buhi
user::rwx
group::r-x
other::r-x
default:user::rwx
```

default:group::r-x

default:group:buhi:rwx

default:other::r-x

Здесь мы видим, что для группы "buhi" выставлены права на чтение, запись и выполнение.

Теперь группа "buhi" имеет права на чтение и запись файлов в папке /media/buhi и все ее подпапки и файлы.

6.3.3. Автоустановка ACL для создаваемых файлов и папок

Ключ **-d** у команды **setfacl** позволяет установить ACL по умолчанию на вновь создаваемые файлы и папки в указанной директории (ранее мы выставили ACL просто на уже существующие файлы и каталоги).

Для того, чтобы установить ACL по умолчанию для группы "buhi" на каталог **/media/buhi**, выполните следующую команду:

sudo setfacl -R -d -m g:buhi:rwx /media/buhi

Здесь ключ **-d** указывает, что мы устанавливаем ACL по умолчанию, а ключ **-m** указывает, что мы изменяем существующие права. Аргумент g:buhi:rwx задает права для группы "buhi" на чтение, запись и выполнение. Теперь при создании новых файлов и каталогов в каталоге **/media/buhi**, они будут автоматически получать ACL с правами, выставленными по умолчанию.

Вы можете проверить, что ACL по умолчанию настроены правильно, выполнив команду:

sudo getfacl /media/buhi

В выводе должно быть что-то похожее на:

file: /media/buhi/

owner: <owner>

group: buhi

user::rwx

group::r-x

other::r-x

default:user::rwx

default:group::r-x

default:group:buhi:rwx

default:other::r-x

Здесь мы видим строку "default:group:buhi:rwx", которая указывает, что группа "buhi" имеет права на чтение, запись и выполнение для всех вновь создаваемых файлов и каталогов в каталоге /media/buhi.

7. ZABBIX — СБОР ОТЧЕТОВ О ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ

7.1. Введение

Система Zabbix предназначена для организации мониторинга инфраструктуры.

Типовая конфигурация сервера с Zabbix:

- сервер Zabbix
- веб-интерфейс через веб-сервер Apache HTTPD
- база данных MySQL (MariaDB)

Также необходимо будет настроить агент Zabbix, собирающий данные.

7.2. Настройка Zabbix

Подробную инструкцию по настройке сервера и агентов Zabbix в ОС РОСА ХРОМ можно прочитать на нашей вики по адресу http://wiki.rosalab.ru/ru/index.php/Zabbix

ПРИЛОЖЕНИЕ 1

ОПИСАНИЕ ФУНКЦИЙ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

1) ПОЛЬЗОВАТЕЛИ И ПАРОЛИ

Создание пользователей

Для создания пользователя выполните:

sudo useradd username

Домашним каталогом пользователя будет /home/username.

Установите пароль новому пользователю:

sudo passwd username

При вводе пароль не отображается (защита обратной связи).

Удаление пользователей

Чтобы удалить пользователя, необходимо выполнить команду userdel:

sudo userdel username

При этом удаление произойдет безопасно: домашняя директория пользователя будет сохранена на диске, а его файлы будут доступны для администратора. Если вы хотите удалить пользователя и его файлы с диска, выполните команду с ключом -r:

sudo userdel -r username

Создание группы пользователей

sudo groupadd groupname

где groupname — имя группы.

Добавления пользователя в группу

sudo usermoad -a -G groupname username

где groupname — имя группы, в которую добавить пользователя username.

Числовые идентификаторы пользователей

Каждый пользователь в Linux имеет уникальный идентификатор (UID) и групповой идентификатор (GID). UID - это числовой идентификатор пользователя, который используется для определения прав доступа к файлам и каталогам. GID - это числовой идентификатор группы, к которой принадлежит пользователь.

Чтобы узнать UID и GID пользователя, выполните команду id:

id username

При создании нового пользователя система автоматически назначает ему UID. Однако, если вы хотите назначить пользователю конкретный UID, вы можете использовать команду useradd с опцией -u. Например:

sudo useradd -u 1001 username

В этом примере мы создаем пользователя username и назначаем ему UID 1001.

Файлы с информацией о пользователях

Информация о локальных пользователях и группах хранится в файлах: /etc/shadow, /etc/passwd и /etc/group. Утилиты useradd, usermod и др. вносят изменения в эти файлы, а NSS-модуль files читает их.

/etc/shadow

Файл /etc/shadow содержит зашифрованные пароли пользователей. Этот файл доступен только для чтения и записи администратору системы (обычному пользователю доступ к этому файлу закрыт). Это позволяет предотвратить несанкционированный доступ к паролям пользователей.

Формат файла /etc/shadow следующий:

username:password:lastchanged:minimum_age:maximum_age:warning_days:inactive_days:expire_date:reserved

где:

username - имя пользователя;

password - зашифрованный пароль пользователя;

lastchanged - дата последнего изменения пароля в формате дней с 1 января 1970 года;

minimum_age - минимальный срок действия пароля (в днях);

maximum_age - максимальный срок действия пароля (в днях);

warning_days - количество дней, за которое пользователь будет предупрежден о необходимости изменения пароля;

inactive_days - количество дней неактивности, после которых учетная запись пользователя будет заблокирована;

expire_date - дата истечения срока действия учетной записи в формате дней с 1 января 1970 года;

reserved - зарезервированный параметр.

/etc/passwd

Файл /etc/passwd содержит основную информацию о пользователях. Каждая строка в файле представляет одного пользователя и имеет следующий формат:

username:password:UID:GID:comment:home_directory:default_shell

где:

username - имя пользователя;

password - зашифрованный пароль пользователя (значение поля x в файле /etc/shadow);

UID - числовой идентификатор пользователя;

GID - числовой идентификатор группы пользователя;

comment - комментарий о пользователе (обычно содержит имя пользователя или другую полезную информацию);

home_directory - домашний каталог пользователя;

default_shell - командная оболочка по умолчанию.

/etc/group

Файл /etc/group содержит информацию о группах пользователей. Каждая строка в файле представляет одну группу и имеет следующий формат:

groupname:password:GID:user_list

где:

groupname - имя группы;

password - зашифрованный пароль группы (не используется в ROSA, имеет значение "x");

GID - числовой идентификатор группы;

user_list - список пользователей, входящих в данную группу (разделенные запятой).

Задание срока действия пароля пользователя

В ROSA Linux можно настроить срок действия пароля пользователя, чтобы обеспечить безопасность системы. Для этого можно использовать команду chage. Например, чтобы задать срок действия пароля пользователя на 90 дней, выполните следующую команду:

sudo chage -M 90 username

При этом пользователь будет вынужден изменить свой пароль при следующем входе в систему.

Блокировка учетной записи

При наличии подозрений на нарушение безопасности системы или если вы хотите заблокировать доступ к учетной записи пользователя по другим причинам, можно выполнить команду блокировки учетной записи пользователя. Например:

sudo usermod -L username

Эта команда заблокирует учетную запись пользователя username и запретит ему доступ к системе. Для разблокировки учетной записи выполните команду usermod с ключом -U:

sudo usermod -U username

Принудительная смена пароля пользователя при следующем входе

Чтобы пользователь сменил свой пароль при следующем входе в систему, выполните команду passwd с опцией -e:

sudo passwd -e username

Пользователю будет предложено сменить свой пароль при следующем входе в систему.

Настройка pwquality

pwquality - это библиотека и инструмент для проверки и настройки параметров безопасности паролей в Linux. С помощью pwquality можно задавать критерии сложности пароля, его время жизни, период блокировки и т.д.

pwquality обычно предустановлена в дистрибутивах ROSA, при необходимости установить можно так:

sudo dnf install pam_pwquality libpwquality-common libpwquality-tools

PAM-модуль pam_pwquality прописан в конфигурационных файлах /etc/pam.d/* из коробки в большинстве дистрибутивов ROSa, но может быть при необходимости добавлен в них вручную.

После установки вы можете отредактировать файл /etc/security/pwquality.conf, чтобы настроить параметры проверки безопасности паролей. Например, чтобы задать минимальную длину пароля в 8 символов, выполните следующие шаги:

Откройте файл /etc/security/pwquality.conf для редактирования:

sudo nano /etc/security/pwquality.conf

Найдите строку, которая начинается с minlen и измените ее значение на 8:

minlen = 8

Сохраните изменения и закройте файл.

Этот и другие параметры описаны в man pwquality.conf.

Также вы можете использовать команду pwpolicy, чтобы настроить параметры безопасности паролей на уровне конкретного пользователя или группы пользователей. Например, чтобы задать максимальное время жизни пароля для пользователя username в 90 дней, выполните следующие шаги:

Выполните команду pwpolicy с опцией -u и указанием имени пользователя:

sudo pwpolicy -u username -setpolicy "maxMinutesUntilChangePassword=129600"

При этом максимальное время жизни пароля для пользователя username будет установлено на 90 дней (129600 минут).

Блокирование консольного сеанса по неактивности

Рассмотрим настройки автоматической блокировки консольного сеанса по неактивности.

Установим необходимые пакеты:

```
sudo dnf install tmux vlock
```

Файл /etc/tmux.conf приведем к следующему виду:

```
set -g lock-command "vlock -c"
set -g lock-after-time 900
bind L lock-session
set -g mouse on
set-option -g history-limit 30000
```

В начало файла /etc/bashrc вставьте следующий код:

```
# $PS1 is set if it is an interactive shell
# $XDG_SESSION_TYPE is set by pam_systemd
if [ -n "$PS1" ] &&
  [ "$XDG_SESSION_TYPE" = tty ] &&
  [ -z "$TMUX" ] &&
  ! [[ "$TERM" =~ screen ]] &&
  ! [[ "$TERM" =~ tmux ]] &&
  command -v tmux >/dev/null 2>&1
then
  exec tmux
```

fi

Теперь при входе в TTY и по SSH консоль будет блокироваться через 900 секунд неактивности, для разблокировки нужно будет ввести пароль, когда консоль заблокирована, ее содержимое не будет видно. При работе внутри графической сессии предполагается работа блокировщика экрана вместо блокировщика консоли.

2) АУДИТ

Аудит (audit) — это механизм аудита безопасности в ядре Linux, который позволяет отслеживать события в системе. Он может использоваться для мониторинга доступа к файлам, сетевых подключений, а также других системных событий.

В пространстве пользователя работает демон auditd (рассматриваем именно его, хотя возможны иные реализации), который получает от ядра ОС информацию о произошедших событиях и обрабатывает ее в соответствии с настройками.

Ведение логов аудита обычно решает 2 задачи:

- протоколирование происходящего для разбора в случае действий злоумышленника;

- протоколирование происходящего для отладки работы программ.

Поскольку получивший доступ к компьютеру (серверу) злоумышленник может получить доступ к удалению логов аудита, рекомендуется в режиме реального времени отправлять их на внешний сервер. Пример такой настройки рассмотрен в статье Сервер логирования Journald.

Установка auditd

Пакет предустановлен в большинстве дистрибутивов ROSA, при необходимости установка производится так:

sudo dnf install audit

Управление службой auditd

Служба auditd включается автоматически, однако при необходимости ее можно включить в автозапуск так:

sudo systemctl enable auditd

Выключить так:

sudo systemctl mask auditd

Остановить так:

sudo systemctl kill auditd

Настройка auditd

Конфигурационный файл находится по адресу /etc/audit/auditd.conf. Его содержимое по умолчанию подходит для работы.

Некоторые из наиболее важных параметров:

max_log_file: максимальный размер журнального файла аудита в мегабайтах.

num_logs: максимальное количество журнальных файлов аудита.

log_file: путь к журнальному файлу аудита.

log_group: группа, которой принадлежит журнальный файл аудита.

write_logs: писать ли (yes — писать, no — не писать) события аудита в файл /var/log/audit/audit.log. Можно выставить no, тогда события аудита будут попадать в systemd-journald, но не будут дублироваться в еще один файл. Утилита ausearch читает именно этот файл, поэтому ей нельзя будет пользоваться.

Больше информации в man auditd.conf.

Настройка правил аудита

Правила аудита — это инструкции, какие события нужно подвергать аудиту, то есть логировать.

Правила аудита задаются с помощью утилиты auditctl, которая по сокету взаимодействует с демоном auditd и передает ему команды.

При запуске auditd читаются правила из файлов /etc/audit/audit.rules и /etc/audit/rules.d/*.rules. Каждая строка в этих файлах представляет собой аргументы для auditctl. Например, строка:

-w /etc/passwd -p wa -k passwd-file

эквивалентна запуску команды:

sudo auditctl -w /etc/passwd -p wa -k passwd-file

Запись в файл позволяет прогружать правила аудита автоматически при запуске ОС.

Выборка из логов аудита с помощью ausearch

Утилита ausearch — это инструмент для анализа журналов аудита, созданных с помощью системы аудита Linux. Журналы аудита содержат

записи о действиях, совершаемых в системе, таких как изменение файлов, запуск процессов или использование сетевых ресурсов. Утилита ausearch предназначена для поиска и анализа этих записей.

Она может выполнять поиск записей журнала аудита по различным критериям, таким как время, пользователь, тип события и др.

Для примера, можно использовать следующую команду для поиска всех записей журнала аудита, связанных с изменением файла /etc/passwd:

sudo ausearch -f /etc/passwd

Результаты поиска будут выведены на экран. Каждая запись журнала аудита будет содержать информацию о времени события, типе события, пользователе, который совершил действие, и другие подробности.

Одной из наиболее полезных функций утилиты ausearch является возможность использования ключей для выполнения расширенного поиска. Например, следующая команда выполняет поиск записей журнала аудита, связанных с запуском процесса с именем "httpd":

sudo ausearch -c httpd

Другим полезным ключом является ключ -i, который позволяет просмотреть подробную информацию о конкретной записи журнала аудита. Например, следующая команда выводит подробную информацию о записи журнала аудита:

sudo ausearch -i 12345

где 12345 — это идентификатор записи журнала аудита.

Утилита ausearch также может использоваться для выполнения аудита безопасности. Например, можно настроить систему аудита таким образом, чтобы она записывала информацию о всех неудачных попытках входа в систему. Затем можно использовать утилиту ausearch для поиска этих записей журнала аудита и выявления потенциальных угроз безопасности.

Также утилита ausearch поддерживает несколько форматов вывода, включая форматы текста и JSON. Это может быть полезно для автоматизации процесса анализа журналов аудита.

Примеры

Отслеживание изменений в директории или файле

Для отслеживания всех изменений в директории /var/log добавьте следующее правило:

sudo auditctl -w /var/log -p w -k var_log_changes

Это правило отслеживает любые операции записи (w) в директорию /var/log и связывает их с ключевым словом (tag) var_log_changes.

Проверьте, что правила аудита корректно добавлены:

sudo auditctl -l

Эта команда выводит список текущих правил аудита.

Запишите текст в файл внутри этой директории:

echo text | sudo tee /var/log/testfile

Просмотрите журнальные файлы аудита, используя утилиту ausearch. Чтобы просмотреть все записи аудита, связанные с директорией /var/log, используйте следующую команду:

sudo ausearch -k var_log_changes

Эта команда выводит все записи аудита, связанные с ключевым словом var_log_changes, в т.ч. событие записи в файл /var/log/testfile.

Отслеживание запуска определенного приложения

Правило аудита:

sudo auditctl -a exit,always -F path=/usr/bin/myapp -F perm=x -k myapp_execution

Это правило отслеживает каждый выход (exit) из приложения /usr/bin/myapp с правами на выполнение (x) и связывает его с ключевым словом myapp_execution.

Поиск событий осуществляется по аналогии:

sudo ausearch -k myapp_execution

Отслеживание системных вызовов

Правило аудита для отслеживания 64-битных системных вызовов execve() от root, т.е. отслеживания запуска программ от root:

sudo auditctl -a exit,always -F arch=b64 -S execve -F uid=0 -k authentication_events

Это правило отслеживает каждый вызов функции execve (используется для запуска исполняемых файлов) с идентификатором пользователя (uid) равным 0 (root) и связывает его с ключевым словом authentication_events. Т.к. 64-битные ОС поддерживают запуск и 32-битных программ, подвергнем аудиту то же самое в 32-битном варианте:

sudo auditctl -a exit,always -F arch=b32 -S execve -F uid=0 -k authentication_events

Отслеживание доступа к конфиденциальным файлам

Правило аудита:

sudo auditctl -a exit,always -F arch=b64 -S open -F dir=/etc/sensitive -F success=0 -k sensitive_files_access

Это правило отслеживает каждый вызов функции open для любого файла в директории /etc/sensitive, который заканчивается неудачей (success=0) и связывает его с ключевым словом sensitive_files_access.

Отслеживание сетевых подключений

Правило аудита:

sudo auditctl -a exit,always -F arch=b64 -S bind -S connect -F success=0 -k network_events

Это правило отслеживает каждый вызов функций bind() и connect() для любого сетевого подключения, которое заканчивается неудачей (success=0) и связывает его с ключевым словом network_events.

Запись в журнал аудита при подключении устройства USB

Правило аудита:

sudo auditctl -w /dev/bus/usb -p rwxa -k usb

Теперь система будет записывать информацию при подключении устройства USB в журнал аудита.

Чтобы найти связанные с подключением устройства USB записи журнала аудита, выполните следующую команду ausearch:

sudo ausearch -f /dev/bus/usb -k usb -i

Эта команда найдет все записи журнала аудита, связанные с ключом usb и событием OPEN, которое относится к подключению устройства USB.

Интеграция с SIEM

Интеграция auditd со специализированными системами сбора событий информационной безопасности (SIEM) возможна двумя способами, в зависимости от SIEM:

через плагин audisp (необходимо установить пакет audispd-plugins)

через отсылку событий в SIEM как сервер syslog (RFC 5424)

через свой собственный агент SIEM-системы

Для отсылки событий в syslog-сервер можно использовать службу rsyslog. Установка пакета производится так:

sudo dnf install rsyslog

В файле /etc/rsyslog.conf необходимо задать адрес сервера:

. @@<IP адрес сервера SIEM>:<порт>

Перезапустить службу rsyslog:

sudo systemctl restart rsyslog

Проверить ее работу:

sudo systemctl status rsyslog

Добавить ее в автозапуск:

sudo systemctl enable rsyslog

3) СПИСКИ КОНТРОЛЯ ДОСТУПА (ACL)

ACL (Access Control List, список контроля доступа) - это механизм в Linux, который позволяет более гранулярно управлять доступом к файлам и папкам, чем это позволяют права доступа по умолчанию. С ACL вы можете установить права доступа для конкретных пользователей и групп пользователей на конкретные файлы и папки.

ACL в Linux обычно состоит из двух типов прав: пользовательские права и групповые права. Пользовательские права позволяют устанавливать права доступа для конкретных пользователей, а групповые права - для групп пользователей.

Преимущество использования ACL заключается в том, что он позволяет установить более точные права доступа, чем это возможно с помощью стандартных прав доступа на основе пользователей и групп. Например, если у вас есть файл, к которому нужен доступ только для нескольких пользователей, вы можете установить ACL для этого файла, чтобы ограничить доступ только для этих пользователей, не давая права доступа всем остальным пользователям на системе.

Команды для работы с ACL в Linux включают getfacl и setfacl. Команда getfacl используется для просмотра текущих прав доступа ACL на файлы и папки, а команда setfacl используется для изменения или добавления новых прав доступа ACL на файлы и папки.

Знание ACL может быть полезно для системных администраторов, которые хотят более тонко настраивать доступ к файлам и папкам на своих серверах.

В Linux ACL хранятся в расширенных атрибутах (extended attributes, xattr), которые являются дополнительными метаданными, хранящимися в файловой системе. Большинство файловых систем, в т.ч. ext4, BTRFS, XFS, поддерживают хранение ACL в xattr. Однако, не все файловые системы поддерживают ACL и хранение их в xattr. Например, файловая система FAT32 не поддерживает ACL вообще.

Установка пакетов

Обычно пакет с утилитами для работы с ACL уже предустановлен, при необходимости установите его:

sudo dnf install acl

Настройка ACL на примере

Постановка задачи

Пусть имеется:

каталог /media/buhi (на поддержерживающей ACL файловой системе: ext4, btrfs, xfs и др.)

группа пользователей buhi (бухгалтерия)

пользователи ivanova, petrova, sidorova, входящие в группу buhi

Нужно сделать так, чтобы все пользователи из группы buhi — ivanova, petrova, sidorova — могли:

читать файлы в /media/buhi

создавать новые файлы и каталоги

изменять существующие файлы

Настройка ACL для существующих файлов

Выставьте ACL на каталог /media/buhi для группы "buhi", чтобы ее пользователи могли записывать и читать файлы в этом каталоге. Для этого выполните команду:

setfacl -R -m g:buhi:rwx /media/buhi

Эта команда добавляет права для группы "buhi" на папку /media/buhi и все ее подкаталоги и файлы. Опция -R означает, что изменения должны применяться рекурсивно для всех подкаталогов и файлов. Опция -m

означает, что мы меняем существующие права. g:buhi:rwx означает, что мы выставляем права для группы "buhi" на чтение, запись и выполнение.

Проверьте, что ACL настроены правильно. Для этого выполните команду:

getfacl /media/buhi

В выводе должно быть что-то похожее на:

file: /media/buhi/

owner: <owner>

group: buhi

user::rwx

group::r-x

other::r-x

default:user::rwx

default:group::r-x

default:group:buhi:rwx

default:other::r-x

Здесь мы видим, что для группы "buhi" выставлены права на чтение, запись и выполнение.

Теперь группа "buhi" имеет права на чтение и запись файлов в папке /media/buhi и все ее подпапки и файлы.

Автоматическая установка ACL и вновь создаваемых файлов и каталогов

Ключ -d у команды setfacl позволяет установить ACL по умолчанию на вновь создаваемые файлы и папки в указанной директории (ранее мы выставили ACL просто на уже существующие файлы и каталоги).

Для того, чтобы установить ACL по умолчанию для группы "buhi" на каталог /media/buhi, выполните следующую команду:

sudo setfacl -R -d -m g:buhi:rwx /media/buhi

Здесь ключ -d указывает, что мы устанавливаем ACL по умолчанию, а ключ -m указывает, что мы изменяем существующие права. Аргумент g:buhi:rwx задает права для группы "buhi" на чтение, запись и выполнение. Теперь при создании новых файлов и каталогов в каталоге /media/buhi, они будут автоматически получать ACL с правами, выставленными по умолчанию.

Вы можете проверить, что ACL по умолчанию настроены правильно, выполнив команду:

getfacl /media/buhi

В выводе должно быть что-то похожее на:

file: /media/buhi/

owner: <owner>

group: buhi

user::rwx

group::r-x

other::r-x

default:user::rwx

default:group::r-x

default:group:buhi:rwx

default:other::r-x

Здесь мы видим строку "default:group:buhi:rwx", которая указывает, что группа "buhi" имеет права на чтение, запись и выполнение для всех вновь создаваемых файлов и каталогов в каталоге /media/buhi.

4) ПАКЕТНЫЙ МЕНЕДЖЕР (DNF)

Приведённые команды запускать от root или через sudo:

$ sudo dnf <...>

DNF и RPM

Управление программными пакетами осуществляется с помощью утилит командной строки rpm и dnf. RPM является "низкоуровневым" пакетным менеджером, производящим установку, удаление и обновление пакетов, DNF "высокоуровневым" пакетным менеджером, в задачи которого входит разрешение зависимостей между пакетами, их скачивание и установка с использованием "низкоуровневого" RPM.

Основные операции с пакетами

Синтаксис консольной утилиты dnf имеет следующий вид:

dnf <опции> <команда> <пакет>

Команда Описание

install Установка пакета

reinstall Переустановка пакета

check-update Проверка наличия обновлений

update Обновление пакета

remove Удаление пакета

list Вывод имен всех доступных и установленных пакетов

search Поиск пакета

info Вывод информации о пакете

groupinstall Установка группы пакетов

groupupdate Обновление группы пакетов

groupremove Удаление группы пакетов

grouplist Вывод информации о группах

repolist Вывод списка включённых репозиториев

repolist all Вывод списка всех подключённых репозиториев

history Дает информацию о выполненных командах, о датах и времени их выполнения, о числе затронутых пакетов, о том, были ли эти транзакции успешными или же были прерваны, и была ли изменена база данных RPM в промежуток между транзакциями.

distro-sync По мере необходимости обновляет, понижает версию или сохраняет выбранные установленные пакеты в соответствии с последней версией, доступной в любом включенном репозитории. Если пакет не указан, учитываются все установленные пакеты.

Пример установки пакета mc:

$ sudo dnf install mc

Подробнее о работе dnf можно узнать во встроенной справке:

$ dnf --help

И в расширенной документации:

$ man dnf

Регулярные выражения

Регулярные выражения (regexp или regex) — это механизм поиска и замены с помощью шаблонов-символов. Все команды DNF предоставляют возможность поиска и фильтрации результата с помощью добавления одного или нескольких шаблонов выражений в качестве аргумента.

Шаблоны выражений содержат один или несколько символов подстановки — символ « * » расширяет поиск до соответствия любому поднабору знаков, а « ? » до соответствия любому одиночному символу. Чтобы команда всегда

отрабатывала корректно, искомое выражение должно быть в прямых одинарных или двойных кавычках.

$ dnf repoquery 'gea*'

Покажет все доступные в репозитории пакеты имена которых начинаются на "gea":

geany-0:1.38-1.x86_64

.....

geany-plugins-0:1.38-4.x86_64

$ dnf repoquery "gea??"

Покажет все пакеты имена которых начинаются на "gea" плюс ещё любых два символа:

geany-0:1.38-1.x86_64

geary-0:40.0-10.x86_64

Найти пакет по имени файла

$ dnf rq --whatprovides /usr/bin/gzip

...

gzip-0:1.12-1.x86_64

$ dnf provides /usr/bin/gzip

...

gzip-1.12-1.x86_64 : The GNU data compression program

Репозиторий : mirror-rosa-x86_64-main

Совпадения с:

Имя файла : /usr/bin/gzip

$ dnf rq --whatprovides libfontconfig.so.1

libfontconfig1-0:2.13.94-3.i686

$ dnf provides libfontconfig.so.1

libfontconfig1-2.13.94-3.i686 : Font configuration and customization library

Репозиторий : @System

Совпадения с:

Предоставьте : libfontconfig.so.1

...

Показать список файлов в пакете

$ dnf rq -l zip

Подключить тестовые репозитории

$ sudo dnf install rosa-repos-testing

Подключить контейнер и обновить пакет из него

$ sudo dnf --repofrompath name,url update pkg

5) НАСТРОЙКА ЗАМКНУТОЙ ПРОГРАММНОЙ СРЕДЫ (IMA)

Установка пакетов

Установите необходимые пакеты:

sudo dnf install ima-evm-utils libressl audit ima-inspect audit

Во многих дистрибутивах ROSA все эти пакеты уже установлены, в таком случае пакетный менеджер dnf сообщит об этом при выполнении приведенной выше команды.

Создание ключей

Создается пара ключей: закрытый и открытый ключи.

Закрытым ключом подписываются файлы, следовательно, к нему не должно быть доступа у тех, кто не должен иметь возможности подписать файлы, и рекомендуется хранить его отдельно, а не на машине, где выполняется запуск подписанных исполняемых файлов.

Открытый ключ должен быть установлен на каждой системе, на которой производится запуск подписанных исполняемых файлов, он используется для проверки валидности подписи. Открытый ключ хранится на диске и загружается в ключницу ядра в initrd.

Ниже описано создание ключевой пары по ГОСТ. Возможно применение RSA.

Откройте консоль root командой:

sudo -i

или:

su -

Создайте каталог «ima»:

mkdir -p ima

Переведите терминал в него:

cd ima

Создайте файл x509.conf с приведенным ниже текстом. Для создания файла откройте консольный текстовый редактор, например, nano, командой:

nano x509.conf

и вставьте текст ниже:

```
[ req ]
distinguished_name = req_distinguished_name
prompt = no
string_mask = utf8only
x509_extensions = myexts

[ req_distinguished_name ]
O = IMA
CN = Executable Signing Key
emailAddress = ivan@petrov.tld

[ myexts ]
basicConstraints=critical,CA:FALSE
keyUsage=digitalSignature
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid
```

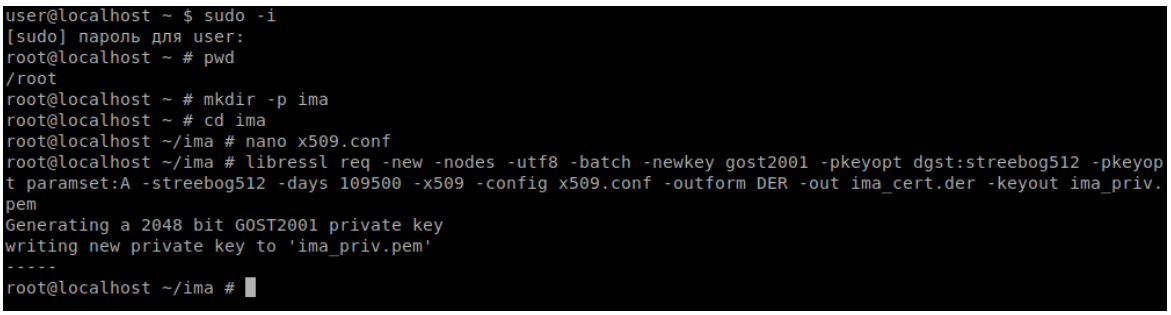
Значения полей O, CN, emailAddress являются произвольными, можно заменить их на свои.

Сохраните файл. Если используется редактор nano, то последовательно нажмите: Ctrl+O, Enter, Ctrl+X.

Теперь создайте пару ключей:

libressl req -new -nodes -utf8 -batch -newkey gost2001 -pkeyopt dgst:streebog512 -pkeyopt paramset:A -streebog512 -days 109500 -x509 -config x509.conf -outform DER -out ima_cert.der -keyout ima_priv.pem

ima_cert.der — открытый ключ, ima_priv.pem — закрытый.



Создайте каталог с открытыми ключами:

mkdir -p /etc/keys/ima

Все ключи из этого каталога будут импортированы в ядро на этапе initrd.

Скопируйте созданный открытый ключ в этот каталог:

cp -v ima_cert.der /etc/keys/ima/ima_cert.der

И дополнительно в другой каталог для работы evmctl ima_verify:

cp -v ima_cert.der /etc/keys/x509_evm.der

Создание политики IMA

Продолжая работать в консоли root, создайте файл /etc/sysconfig/ima-policy

Если используется редактор nano, то для этого выполните:

nano /etc/sysconfig/ima-policy

Скопируйте и вставьте текст ниже, нажмите Ctrl+O, Enter, Ctrl+X.

```
# PROC_SUPER_MAGIC
dont_measure fsmagic=0x9fa0
dont_appraise fsmagic=0x9fa0
# SYSFS_MAGIC
dont_measure fsmagic=0x62656572
dont_appraise fsmagic=0x62656572
# DEBUGFS_MAGIC
dont_measure fsmagic=0x64626720
dont_appraise fsmagic=0x64626720
# TMPFS_MAGIC
dont_measure fsmagic=0x01021994
dont_appraise fsmagic=0x01021994
# RAMFS_MAGIC
dont_appraise fsmagic=0x858458f6
# DEVPTS_SUPER_MAGIC
dont_measure fsmagic=0x1cd1
dont_appraise fsmagic=0x1cd1
# BINFMTFS_MAGIC
dont_measure fsmagic=0x42494e4d
dont_appraise fsmagic=0x42494e4d
```

```
# SECURITYFS_MAGIC
dont_measure fsmagic=0x73636673
dont_appraise fsmagic=0x73636673
# SELINUX_MAGIC
dont_measure fsmagic=0xf97cff8c
dont_appraise fsmagic=0xf97cff8c
# CGROUP_SUPER_MAGIC
dont_measure fsmagic=0x27e0eb
dont_appraise fsmagic=0x27e0eb
# CGROUP2_SUPER_MAGIC
dont_measure fsmagic=0x63677270
dont_appraise fsmagic=0x63677270
# NSFS_MAGIC
dont_measure fsmagic=0x6e736673
dont_appraise fsmagic=0x6e736673
measure func=MMAP_CHECK mask=MAY_EXEC
measure func=BPRM_CHECK mask=MAY_EXEC
appraise func=BPRM_CHECK mask=MAY_EXEC appraise_type=imasig
appraise func=MMAP_CHECK mask=MAY_EXEC appraise_type=imasig
```

Выставьте права на этот файл:

```
chmod 0400 /etc/sysconfig/ima-policy
```

В целом все равно, какие на него права, но в процессе работы системы содержимое политики доступно по адресу /sys/kernel/security/ima/policy с правами 0400, поэтому для единообразия запретим всем, кроме root, читать этот файл. Так нельзя будет узнать применяемую политику IMA без root-прав.

Подписывание файлов в системе

Теперь необходимо подписать все исполняемые файлы, в т.ч. скрипты, иные файлы, которые mmap()-ятся с PROT_EXEC, в т.ч. разделяемые библиотеки *.so*.

Продолжая работать в root-консоли, создайте файл sign.sh

Если используется редактор nano, то выполните команду:

```
nano sign.sh
```

Скопируйте и вставьте в редактор текст ниже, нажмите Ctrl+O, Enter, Ctrl+X.

```
#!/bin/bash
set -xe
FS="$(findmnt --output FSTYPE / | tail -n1)"
echo "$FS"
echo > failed.log
find / -fstype "$FS" -type f -executable | sort -u | while read -r line
do
  if ! evmctl ima_sign --hashalgo streebog512 --key ima_priv.pem "$line" ; then
    echo "$line" >> failed.log
  fi
done

for D in /lib /lib64 /usr/lib /usr/lib64
do
  find "$D" -fstype $FS -\! -executable -type f -name "*.so*" | sort -u | while read
-r line
    do
       if ! evmctl ima_sign --hashalgo streebog512 --key ima_priv.pem "$line";
then
        echo "$line" >> failed.log
      fi
    done
done
```

Сделайте этот скрипт исполняемым:

```
chmod 755 sign.sh
```

И запустите его:

```
./sign.sh
```

Выведите лог ошибок:

```
cat failed.log
```

и убедитесь, что этот файл пуст, а значит в процессе подписывания не возникло ошибок.

Подготовка к файловой системы к запуску со включенной IMA

Если используется файловая система ext4, то необходимо добавить опцию монтирования "iversion". На большинстве типовых установок ROSA это можно сделать следующей командой:

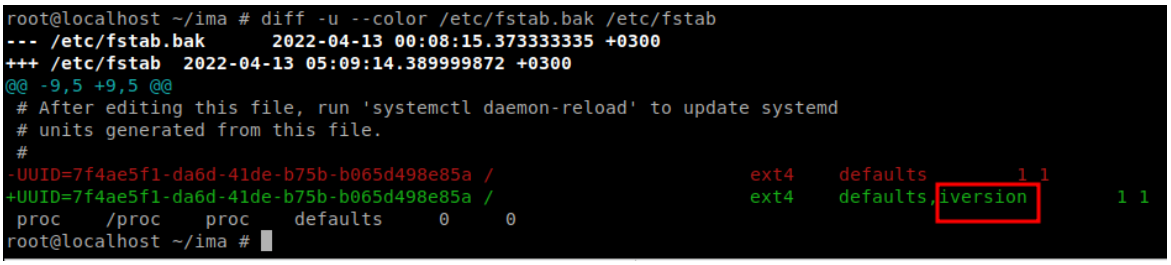
sed -i.bak -e 's!\([[:space:]]/[[:space:]].*ext4[[:space:]].*defaults\)!\1,iversion!' /etc/fstab

Она добавит "iversion" в опции монтирования корня системы в файле /etc/fstab. Резервная копия файла до изменений будет сохранена по адресу /etc/fstab.bak. Выполните:

diff -u --color /etc/fstab.bak /etc/fstab

и удостоверьтесь, что изменения между файлами правильные. Ниже приведен пример правильных изменений.

В случае необходимости отредактируйте файл /etc/fstab самостоятельно вместо применения приведенной выше команды.



Включение auditd

Выполните от root:

systemctl enable auditd

Если команда ничего не выдала, то значит служба auditd уже была включена, никаких дополнительных действий предпринимать не надо.

Сборка initrd с поддержкой IMA

Добавьте модуль integrity в initrd:

echo 'add_dracutmodules+=" integrity "' | tee /etc/dracut.conf.d/10-ima.conf

Пересоберите initrd:

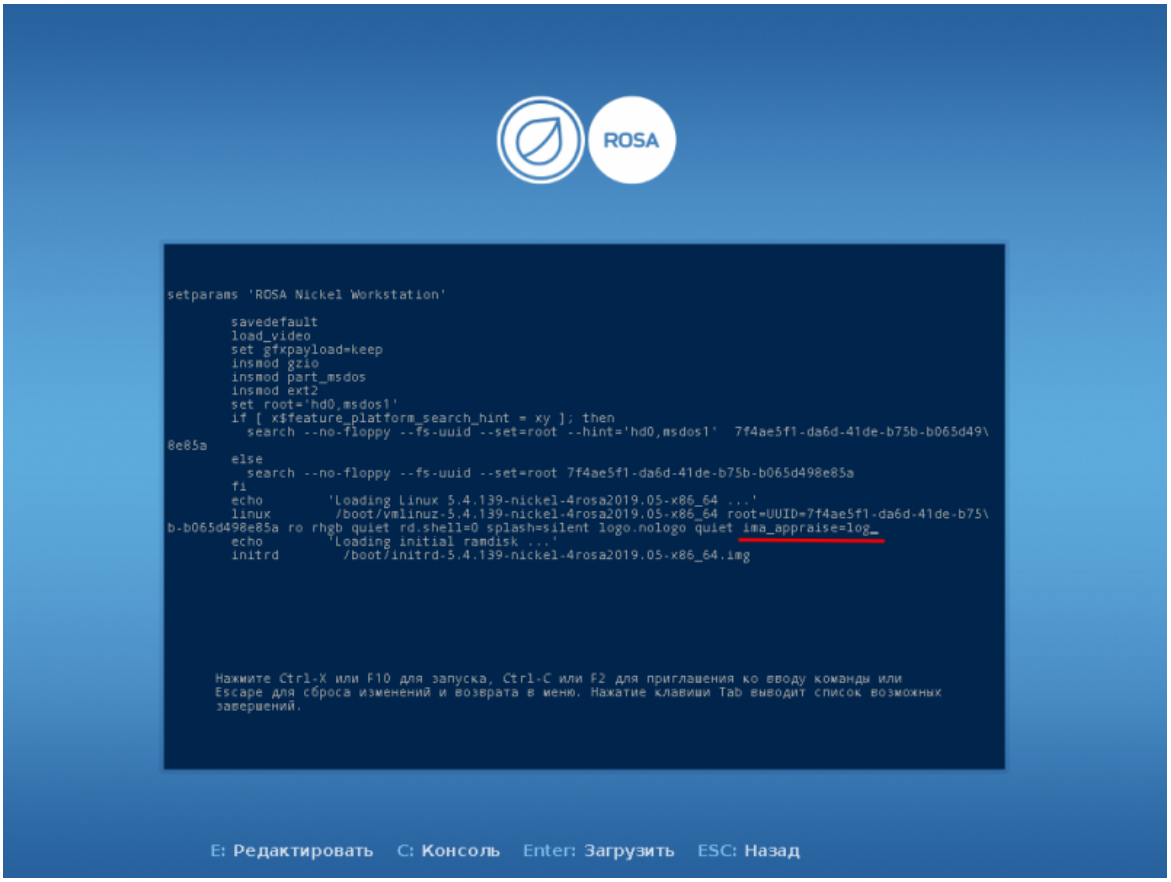
systemd-initramfs-gen

Обратите внимание, что в initrd должны попасть уже подписанные файлы, поэтому данный шаг выполняется после подписывания файлов в системе.

Пробный запуск подписанной системы

Осуществите пробный запуск системы с добавлением "ima_appraise=log" в cmdline ядра, для этого при включении системы в загрузчике Grub выберите

нужный пункт меню и нажмите клавишу с латинской буквой E, при необходимости введите логин и пароль от загрузчика, в открывшемся текстовом редакторе с помощью клавиш со стрелками вверх, вниз, вправо, влево переместите курсор ввода текста в конец строки, начинающейся со слова «linux», и допишите в конец строки: « ima_appraise=log».



Нажмите Ctrl+X или F10 для запуска ОС с измененными параметрами ядра.

В этом режиме система не будет запрещать запуск неподписанных файлов, но будет записывать попытки их запуска.

Откройте консоль root с помощью команды:

sudo -i

или:

su -

Если вы используете ОС с мандатной защитой (Никель), то в этой консоли после входа в root также выполните:

newrole -r secadm_r

Посмотрите, нет ли в логе событий безопасности сообщений от IMA:

strings /var/log/audit/audit.log | grep IMA

Должно быть пусто.

Проверьте наличие подписи у файлов:

[root@rosa2019 ~]# ima_inspect /bin/bash

/bin/bash

security.ima

---

digital signature version 2

digest algorithm: streebog512

key-id v2 (gpg compatible): 5ac982c4

signature length: 1024 bits

signature data:

afbe5f078a9a052c eda2683037c92b40 dab77e6f7fcf2d42 bdce159ef64097ad 9db7b39124fa9f5c ea04c010248e65d4 6ce4d8dcf692d124 62447ccdf53e979c

16ca24090c175f55 81218c5e024a5d6e 57c81d2511bc1311 0b9479aaa04605a5 cbf0b5c1f64aca04 47ab95abd392de8c 08d2683bc1f3c4a4 e6355635c1608671

security.evm

---

no such attribute

[root@rosa2019 ~]# evmctl ima_verify --key /etc/keys/ima/ima_cert.der /bin/bash

key 1: 5ac982c4 /etc/keys/x509_evm.der

/bin/bash: verification is OK

[root@rosa2019 ~]#

Боевой запуск и проверка работы

Запустите без ima_appraise=log, то есть просто перезагрузите систему. Система должна запуститься и работать штатно.

Теперь проверим работу защиты от запуска неподписанных исполняемых файлов.

Войдите в root-консоль с помощью команды:

sudo -i

или:

su -

Создайте копию существующего исполняемого файла:

cp -v /bin/cat cat.copy

Дайте ему права на исполнение:

chmod +x cat.copy

Попробуйте его запустить командой:

./cat.copy

Получите ошибку:

-bash: ./cat.copy: Отказано в доступе

При этом в логе аудита появляется запись о запрете запуска неподписанного файла.

Если вы используете ОС с мандатной защитой (Никель), то в этой консоли выполните:

newrole -r secadm_r

В логе будет сообщение с текстом "cause=IMA-signature-required", чтобы его увидеть, в этой же консоли выполняем команду:

strings /var/log/audit/audit.log | grep IMA

Ее вывод будет похож на следующий:

type=INTEGRITY_DATA msg=audit(1610966070.898:250): pid=7021 uid=0 auid=1000 ses=2 op=appraise_data cause=IMA-signature-required comm="bash" name="/root/cat.copy" dev="sda2" ino=2759743 res=0 errno=0

Это говорит о работоспособности замкнутой программной среды (IMA).

Удалите более ненужный файл:

rm -fv cat.copy