Настройка политики безопасности Fedora Linux(Lima VM)

Настройка политики безопасности в Fedora Linux будет состоять из 3 этапов настройки и конфигурации:

1. Ограничение прав суперпользователя через sudo;
2. Изоляция приложений с помощью systemd;
3. Включение и настройка аудита событий безопасности (auditd).

Эти меры снижают риски, связанные с компрометацией учётных записей, утечкой данных через уязвимые приложения и отсутствием прозрачности в действиях пользователей.

Тестируемая система Fedora Linux(Lima VM):

1. Контроль привелегий: настройка sudo с принципом минимальных прав

По умолчанию в Fedora (и многих дистрибутивах) пользователь, добавленный в группу wheel, получает полный доступ к sudo, включая возможность запуска любых команд от root. Это нарушает принцип минимальных привилегий и увеличивает риск случайного или злонамеренного повреждения системы.

Отключим полный доступ для группы wheel. Открываем окно конфига sudoers командой sudo visudo.

```
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL
```

Рисунок 1 – sudoers.tmp

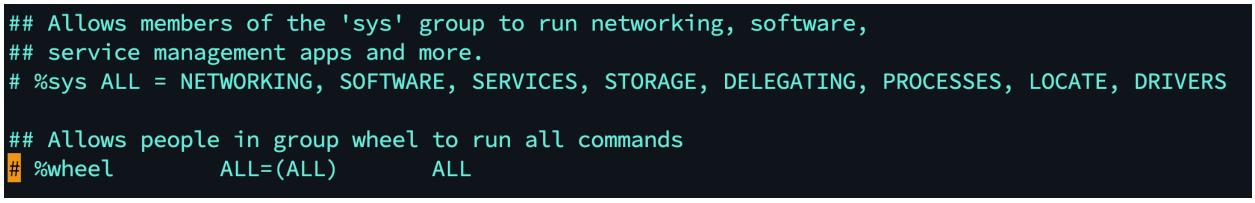

Рисунок 2 – Задокументируем строку %wheel ALL=(ALL) ALL

Рисунок 3 – Создадим отдельного пользователя для сервисов(user)

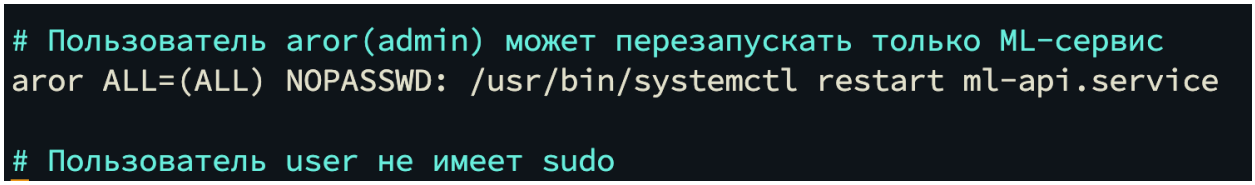

Рисунок 4 –Разрешим только конкретные действия пользователю user, которого раннее добавили

Итог: Даже при компрометации учётной записи aror злоумышленник не сможет, например, изменить /etc/passwd или установить rootkit.

Пользовательские приложения (например, Flask-API для классификации изображений) по умолчанию запускаются с полным доступом к файловой системе, сети и процессам. Уязвимость в таком приложении может привести к полной компрометации системы.

2. Изоляция приложений: использование systemd для ограничения ресурсов и доступа

Используем встроенные механизмы изоляции в systemd для запуска сервиса в «песочнице». Пример: безопасный systemd-юнит для ML-API

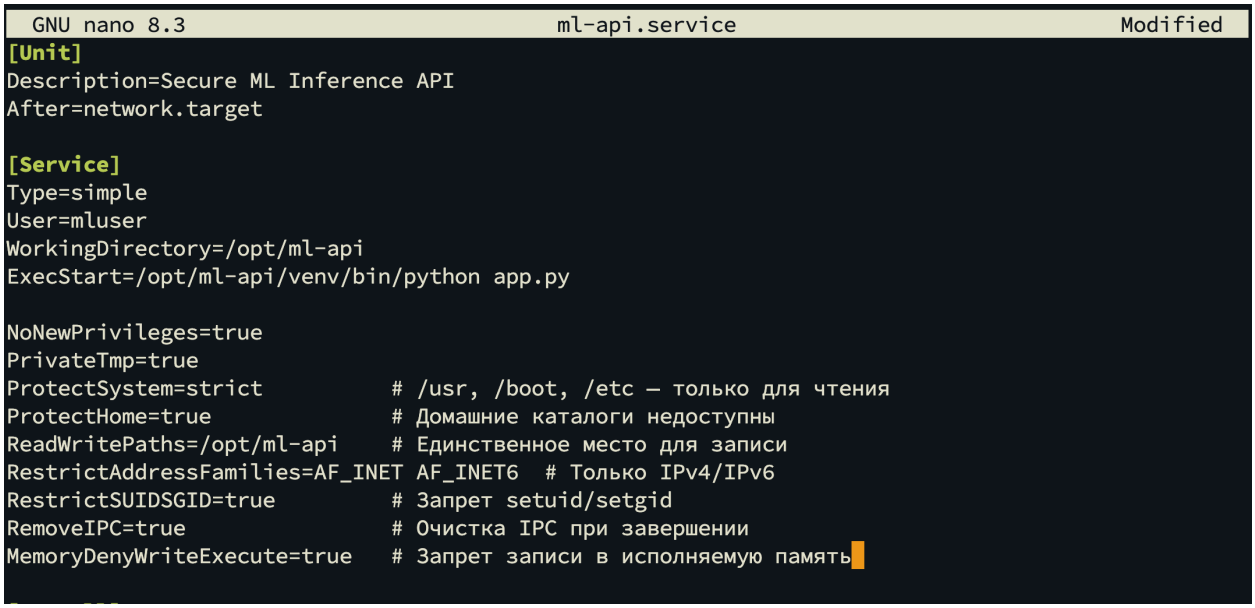

Рисунок 5 – Создадим файл /etc/systemd/system/ml-api.service

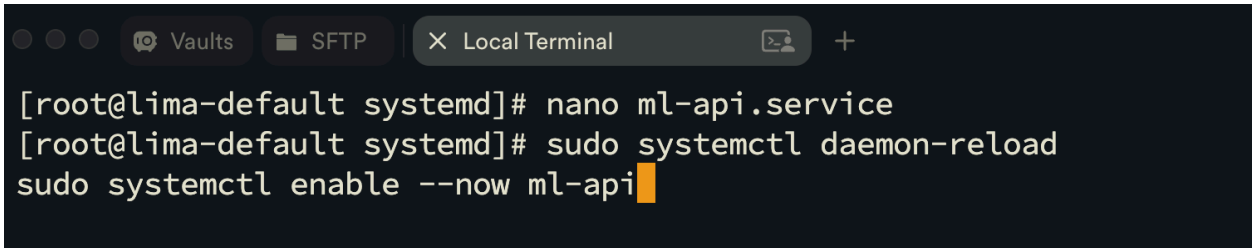

Рисунок 6 – Применим изменения.

Теперь даже если в app.py есть RCE-уязвимость, атакующий не сможет:

- Читать /etc/shadow,

- Записывать файлы вне /opt/ml-api,

- Запускать бинарники с повышенными привилегиями.

3. Аудит безопасности: включение и настройка auditd

При отсутствии журналирования критических событий (попытки входа, изменение привилегий, доступ к чувствительным файлам) невозможно выявить вторжение или проанализировать инцидент.

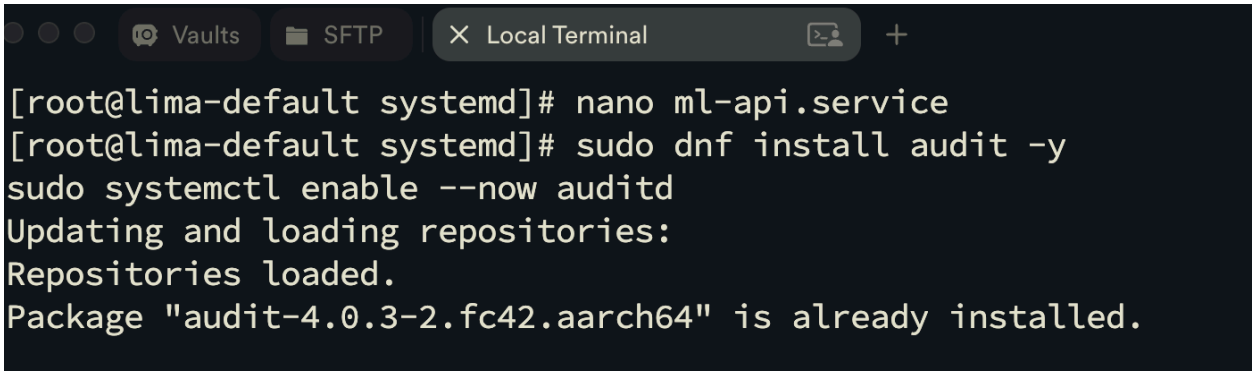

Рисунок 7 – Установим и настроим подсистему аудита ядра Linux – auditd.

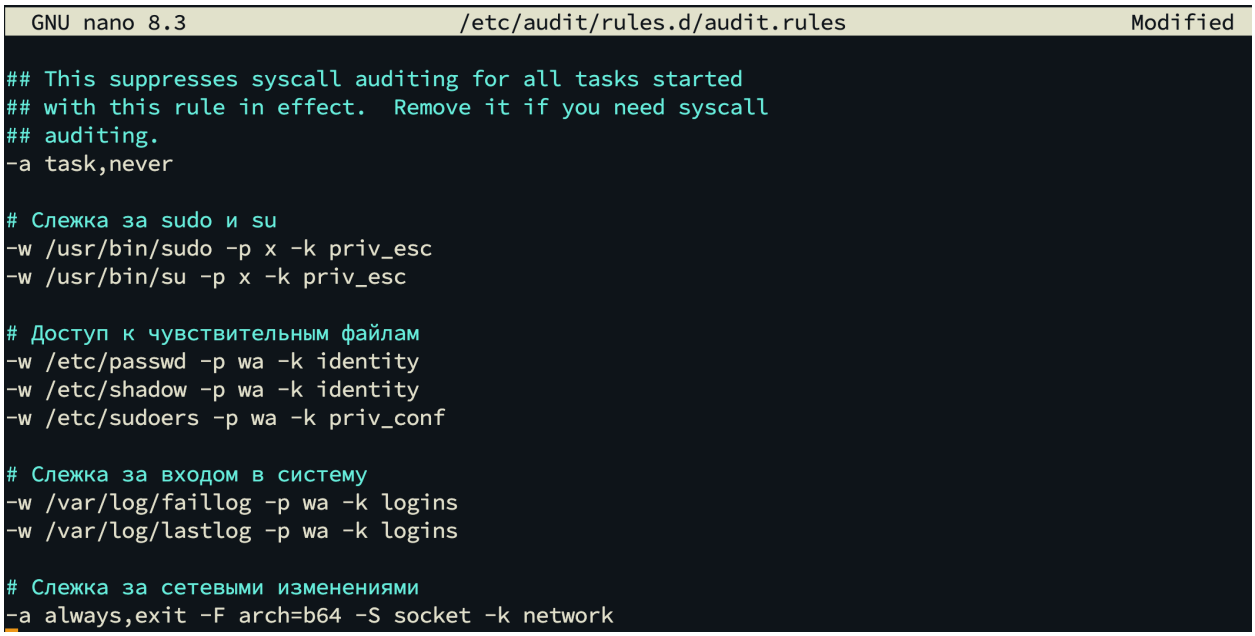

Рисунок 7 – Отредактируем /etc/audit/rules.d/audit.rules:

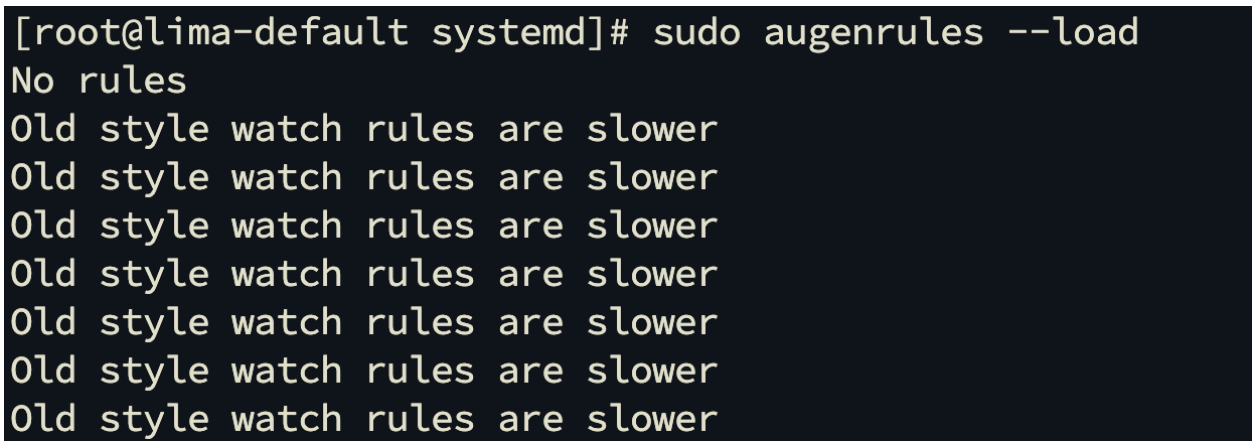

Рисунок 8 – Применим правила отслеживания.

```
[root@lima-default systemd]# sudo aureport -l --success --summary | tail -10

Success Login Summary Report
============================
total  auid
============================
9  501
```

Рисунок 9 – Просмотрим последние события заданными нашими правилами.

Результат: Все критические действия логируются, что позволяет:

- Обнаружить подбор паролей,

- Отследить несанкционированное изменение конфигурации,

- Провести пост-инцидентный анализ.

Итог:

Мера	Эффект
Ограниченный `sudo`	Снижает риски от компрометации учётной записи
Изоляция через `systemd`	Ограничивает ущерб от уязвимостей в приложениях
Аудит через `auditd`	Обеспечивает прозрачность и возможность расследования

Эти практики соответствуют рекомендациям NIST, CIS Linux Benchmarks и активно используются в production-средах.