

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 привилегий: 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

```
## 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 2 – Задокументируем строку %wheel ALL=(ALL) ALL

```
[root@lima-default /]# sudo useradd -r -s /usr/sbin/nologin user
```

Рисунок 3 – Создадим отдельного пользователя для сервисов(user)

```
# Пользователь aror(admin) может перезапускать только ML-сервис
aror ALL=(ALL) NOPASSWD: /usr/bin/systemctl restart ml-api.service

# Пользователь user не имеет sudo
```

Рисунок 4 –Разрешим только конкретные действия пользователю user, которого ранее 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

```
GNU nano 8.3                               ml-api.service                                Modified
[Unit]
Description=Secure ML Inference API
After=network.target

[Service]
Type=simple
User=mluser
WorkingDirectory=/opt/ml-api
ExecStart=/opt/ml-api/venv/bin/python app.py

NoNewPrivileges=true
PrivateTmp=true
ProtectSystem=strict          # /usr, /boot, /etc – только для чтения
ProtectHome=true              # Домашние каталоги недоступны
ReadWritePaths=/opt/ml-api    # Единственное место для записи
RestrictAddressFamilies=AF_INET AF_INET6 # Только IPv4/IPv6
RestrictsUIDSGID=true         # Запрет setuid/setgid
RemoveIPC=true                # Очистка IPC при завершении
MemoryDenyWriteExecute=true   # Запрет записи в исполняемую память
```

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

The screenshot shows a terminal window with the following content:

```
[root@lima-default systemd]# nano ml-api.service
[root@lima-default systemd]# sudo systemctl daemon-reload
sudo systemctl enable --now ml-api
```

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.

```
[root@lima-default systemd]# nano ml-api.service
[root@lima-default systemd]# sudo dnf install audit -y
sudo systemctl enable --now auditd
Updating and loading repositories:
Repositories loaded.
Package "audit-4.0.3-2.fc42.aarch64" is already installed.
```

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

```
GNU nano 8.3                               /etc/audit/rules.d/audit.rules                               Modified

## This suppresses syscall auditing for all tasks started
## with this rule in effect. Remove it if you need syscall
## auditing.
-a task,never

# Слежка за sudo и su
-w /usr/bin/sudo -p x -k priv_esc
-w /usr/bin/su -p x -k priv_esc

# Доступ к чувствительным файлам
-w /etc/passwd -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/sudoers -p wa -k priv_conf

# Слежка за входом в систему
-w /var/log/faillog -p wa -k logins
-w /var/log/lastlog -p wa -k logins

# Слежка за сетевыми изменениями
-a always,exit -F arch=b64 -S socket -k network
```

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

```
[root@lima-default systemd]# sudo augenrules --load
No rules
Old style watch rules are slower
```

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.