Руководство по запуску GVM (Greenbone)

Этот документ описывает, что именно было реализовано в пакетах, как запустить весь стек одной командой через gvm-stack.target, как устроены зависимости, и как проверить работу. Конфигурации спроектированы так, чтобы никогда не перезаписывались при обновлениях, а сервисы стартовали только после перезагрузки или вручную через таргет.

GVM (Greenbone Vulnerability Management) в NiceOS: развертывание, зависимости и верификация

PDF Print

Этот документ является неотъемлемой частью проекта **НАЙС.ОС SOC** и детально описывает интеграцию Greenbone Vulnerability Management (GVM) в операционную систему NiceOS. Основная концепция — предоставление полностью готового к использованию стека Greenbone "из коробки", который можно развернуть за

считанные минуты без необходимости в ручной конфигурации, с встроенной поддержкой надежных обновлений. Мы обеспечиваем самостоятельную поставку актуальных обновлений, круглосуточную техническую поддержку, и на рынке просто **не существует** аналогов по уровню полноты, удобства и надежности.

Архитектура спроектирована таким образом, чтобы системный администратор мог активировать всю инфраструктуру единственным действием — через systemd-таргет `gvm-stack.target`. Конфигурационные файлы защищены от перезаписи при обновлениях с помощью директивы `%config(noreplace)`, а сервисы запускаются исключительно по инициативе администратора или при перезагрузке системы.

Что реализовано

Вся инфраструктура консолидирована под единой точкой управления — systemd-таргетом `gvm-stack.target`. Все сервисы стека интегрированы в него через директивы `WantedBy=gvm-stack.target` и `PartOf=gvm-stack.target`, обеспечивая атомарное управление. Автоматический запуск при установке (`%post`) намеренно отключен, предоставляя администратору полный контроль над моментом активации. Последовательность инициализации сервисов гарантируется зависимостями `Requires=` и `After=` в unit-файлах systemd, минимизируя риски конфликтов.

Политика управления конфигурационными файлами

- **Защита от перезаписи:** Все файлы в директории `/etc` помечены как `%config(noreplace)`, сохраняя пользовательские изменения во время обновлений.
- **Настройка параметров:** Индивидуальные опции задаются через файлы `/etc/sysconfig/*` или TOML-конфигурации для гибкости и читаемости.
- **Одноразовые операции:** Инициализация (создание БД, учетной записи администратора, первичная синхронизация фидов) реализована через *oneshot*-юниты с маркерами в `/var/lib/gvm/`, предотвращая повторные выполнения.

Инициализация активируется строго по требованию, обеспечивая предсказуемость. Например, `gvmd-db-setup.service` создает базу данных и роли PostgreSQL только при отсутствии маркера `/var/lib/gvm/.db-initialized`, а `gvmd-provision.service` provision'ит администратора и владельца фидов, если не установлен маркер `/var/lib/gvm/.admin-provisioned`. Первичная синхронизация фидов выполняется сервисом `greenbone-initial-sync.service`.

Все сервисы исполняются от системного пользователя `gvm` (группа `gvm`) с жесткими ограничениями безопасности systemd, включая изоляцию ресурсов и контроль привилегий. По умолчанию веб-интерфейс GSAD (`gsad.service`) настроен на прослушивание всех интерфейсов (`0.0.0.0:9392`), однако политика iptables в NiceOS по умолчанию применяет DROP ко всем входящим соединениям, кроме SSH (порт 22). Это обеспечивает базовую защиту "из коробки".

Доступ к веб-интерфейсу извне

Для публикации GSAD наружу (рекомендуется только в контролируемой среде с TLS) администратор должен явно открыть порт 9392 в iptables, ограничив доступ по IP-адресу хоста. Чтобы узнать IP-адрес сервера NiceOS, выполните:

```
ip addr show | grep inet
```

Ищите строку с `inet /` для активного интерфейса (например, `eth0`). Для добавления правила (замените `YOUR_IP` на IP клиента или подсеть, например `192.168.1.0/24`):

```
sudo iptables -I INPUT -s YOUR_IP -p tcp --dport 9392 -m conntrack --ctstate NEW -m comment --comment "Allow GSA (gsad) 9392/tcp from YOUR_IP" -j ACCEPT
```

Чтобы сохранить правила persistently, используйте `iptables-save > /etc/systemd/scripts/ip4save` и настройте автозагрузку в systemd. Всегда применяйте TLS (через reverse proxy, например Nginx) для производства.

Состав пакетов

- gvm-common
  Инициализирует системного пользователя и группу `gvm`, создает необходимые runtime-каталоги (`/var/lib/gvm`, `/run/gvm`) и определяет корневой таргет `gvm-stack.target`.
- redis-openvas
  Изолированный экземпляр Redis для сканера, доступный по UNIX-сокету `/run/redis-openvas/redis.sock` для повышения безопасности.
- openvas-scanner
  Ядро vulnerability-сканера с конфигурацией в `/etc/openvas/openvas.conf`;

поддерживает параллельные сканирования.

- python-ospd-openvas
  OSPD-демон (Open Scanner Protocol Daemon), обеспечивающий взаимодействие сканера через сокет `/run/ospd/ospd-openvas.sock`.
- openvasd
  Современный демон на Rust для Notus и Scanner API, оптимизированный для производительности и безопасности.
- gvmd
  Greenbone Vulnerability Manager Daemon — центральный менеджер задач, инициализирующий БД, provision'ящий администратора и оркестрирующий сканирования.
- gsad
  Greenbone Security Assistant Daemon — веб-интерфейс (UI) для управления GVM, по умолчанию на `0.0.0.0:9392`.
- python3-greenbone-feed-sync
  Утилита для автоматизированной синхронизации фидов уязвимостей (первичная и ежедневная).

Зависимости и порядок запуска

Все компоненты интегрированы через `gvm-stack.target`, который при активации инициирует последовательный запуск в соответствии с зависимостями systemd. Это гарантирует, что каждый сервис стартует только после готовности предшественников, минимизируя простои и ошибки.

Упрощённый граф зависимостей:

```
network-online.target
□□ redis-openvas.service (обязательно)
□□ openvasd.service □□ (альтернатива OSPD)
□ □□(сканер API)
□□ ospd-openvas.service □□
□□ gvmd-db-setup.service [oneshot, если нет /var/lib/gvm/.db-initialized]
□□ gvmd-provision.service [oneshot, если нет /var/lib/gvm/.admin-provisioned]
□□ gvmd.service (требует PostgreSQL)
□□ gsad.service (после gvmd)
```

PostgreSQL развертывается отдельным пакетом (рекомендуется `postgresql-server`), а его инициализация интегрируется автоматически в `gvmd-db-setup.service`. Для

альтернативных конфигураций (OSPD vs. openvasd) используйте `Conflicts=` в unit-файлах.

Установка и первый запуск

Для развертывания полного стека на базе OSPD выполните установку пакетов из репозитория NiceOS:

```
sudo dnf install  gvm-common openvasd redis-openvas openvas-scanner python-ospd-openvas gvmd gsad python3-greenbone-feed-sync
```

После установки инициализируйте систему:

```
sudo systemctl start greenbone-initial-sync.service
sudo greenbone-feed-sync
sudo systemctl enable --now gvm-stack.target
sudo systemctl enable --now greenbone-feed-sync.timer
```

Примечание: Пароль администратора генерируется автоматически и сохраняется в `/var/lib/gvm/.admin-password` с правами `0600` (доступен только `gvm`). Для безопасности смените его сразу после первого входа в GSAD.

Проверка работы

Общий статус стека:

```
sudo systemctl status gvm-stack.target
sudo systemctl list-dependencies gvm-stack.target
```

Верификация Redis и сканера:

```
sudo -u gvm redis-cli -s /run/redis-openvas/redis.sock ping
# Ожидаемый вывод: PONG

sudo -u gvm gvmd --get-scanners
```

Верификация менеджера GVMD:

```
sudo -u gvm gvmd --get-users
```

Верификация веб-интерфейса:

Локально на сервере:

```
xdg-open https://127.0.0.1:9392
```

Извне (с другого компьютера в сети): Предварительно откройте порт 9392 по IP сервера (см. раздел "Доступ к веб-интерфейсу извне" выше). Затем в браузере на клиентской машине введите `https://<IP_СЕРВЕРА>:9392` (замените `<IP_СЕРВЕРА>` на реальный IP, полученный командой `ip addr show`). Используйте логин `admin` и пароль из `/var/lib/gvm/.admin-password`. Рекомендуется игнорировать предупреждение о самоподписанном сертификате для тестов; в продакшене настройте валидный TLS.

Синхронизация фидов

Первичная синхронизация фидов (NVT, SCAP, CERT) выполняется однократно для загрузки актуальных данных об уязвимостях:

```
sudo systemctl start greenbone-initial-sync.service
```

Мониторьте прогресс: `sudo journalctl -u greenbone-initial-sync -f`. Процесс может занять до 30 минут в зависимости от сети.

Для автоматизированного ежедневного обновления активируйте таймер:

```
sudo systemctl enable --now greenbone-feed-sync.timer
sudo systemctl list-timers greenbone-feed-sync.timer
```

Совет: Таймер запускается в 02:00 UTC; настройте `/etc/sysconfig/greenbone-feed-sync` для кастомных расписаний.

Конфигурации

- `openvas-scanner`: Основной конфиг `/etc/openvas/openvas.conf` (опции сканирования, логирование).
- `ospd-openvas`: Системные параметры в `/etc/sysconfig/ospd-openvas` (сокеты, таймауты).
- `openvasd`: `/etc/sysconfig/openvasd` и TOML `/etc/openvasd/openvasd.toml` (API, аутентификация).
- `gvmd`: `/etc/sysconfig/gvmd` и `/etc/gvm/*.conf` (БД, пользователи).
- `gsad`: `/etc/sysconfig/gsad` (порт, сертификаты).

Все конфигурации защищены `%config(noreplace)`. После модификаций примените изменения:

```
sudo systemctl restart <service>
```

Лучшая практика: Используйте `diff` для сравнения с дефолтными файлами перед редактированием.

Безопасность

- **Пользователь/группа:** `gvm:gvm` с минимальными привилегиями.
- **Runtime-каталоги:** `/run/gvm`, `/run/ospd`, `/run/redis-openvas` (режим `0750`, владелец `gvm`).
- **Хранилища данных:** `/var/lib/openvas`, `/var/lib/notus` (режим `02775`, SGID для совместного доступа).
- **Redis-сокет:** Права `0660`, группа `redis`; `gvm` включен в группу для изоляции.
- **File capabilities:** `cap_net_bind_service=ep`, `cap_net_raw=ep` на `/usr/sbin/openvas` для избежания root.
- **Сетевой доступ:** GSAD на `0.0.0.0:9392`, но заблокирован iptables (DROP все кроме 22). Открывайте selectively; внедряйте TLS и RBAC.

Предупреждение: Никогда не отключайте iptables глобально в продакшене!

Диагностика

Если сервисы не стартуют

```
sudo systemctl status gvm-stack.target
sudo systemctl list-dependencies gvm-stack.target
sudo journalctl -b -u redis-openvas -u ospd-openvas -u openvasd -u gvmd -u gsad --no-pager
```

Ищите ошибки SELinux или портов; используйте `systemd-analyze verify gvm-stack.target`.

Проблемы с БД

```
sudo journalctl -u gvmd-db-setup -u gvmd-provision --no-pager
sudo -u postgres psql -d gvmd -c 'SELECT version();'
```

Проверьте подключение: `sudo -u gvm pg_isready -d gvmd -h localhost`.

Если фиды не синхронизируются

```
sudo journalctl -u greenbone-initial-sync -u greenbone-feed-sync --no-pager
sudo -u gvm greenbone-feed-sync --help
```

Проверьте прокси в `/etc/sysconfig/greenbone-feed-sync`; протестируйте `curl https://feed.community.greenbone.net`.

Для мониторинга в реальном времени:

```
sudo journalctl -u <service> -f
```

Поддержка и ресурсы

Для углубленной помощи обратитесь в службу поддержки **НАЙС.ОС SOC**.
Дополнительно: официальная документация Greenbone на greenbone.github.io.