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 НАЙС.ОС

Развертывание, зависимости, верификация, эксплуатация

1. Область применения

Настоящий документ устанавливает порядок установки, первичной инициализации, запуска, проверки работоспособности и базовой эксплуатации стека Greenbone Vulnerability Management (далее — GVM) в НАЙС.ОС.

Управление компонентами выполняется средствами systemd через единый таргет `gvm-stack.target`. Конфигурационные файлы поставляются как RPM-конфиги с защитой от перезаписи обновлениями.

2. Термины и сокращения

- **GVM** — Greenbone Vulnerability Management.
- **GVMD** — Greenbone Vulnerability Manager Daemon (центральный менеджер).
- **GSAD** — Greenbone Security Assistant Daemon (веб-интерфейс управления).
- **OSPD** — Open Scanner Protocol Daemon (демон протокола сканера).
- **Feed** — наборы данных уязвимостей (NVT/SCAP/CERT), используемые

компонентами GVM.

3. Архитектура и принципы управления

3.1. Единая точка управления

Все сервисы стека интегрируются в systemd-таргет `gvm-stack.target` через зависимости и привязку: `WantedBy=gvm-stack.target` и/или `PartOf=gvm-stack.target`. Запуск и останов выполняются командами systemd и распространяются на все компоненты, входящие в таргет.

3.2. Политика запуска

Автоматический запуск сервисов при установке пакетов не выполняется. Активация стека производится администратором. Инициализационные операции выполняются oneshot-юнитами с контролем повторного запуска через маркеры состояния в `/var/lib/gvm/`.

3.3. Политика конфигураций

- Файлы конфигурации в `/etc` поставляются как RPM-конфиги с режимом защиты от перезаписи при обновлениях (`%config(noreplace)`).
- Параметры сервисов задаются через `/etc/sysconfig/*` и/или декларативные файлы (TOML) в зависимости от компонента.
- После изменения конфигурации требуется перезапуск соответствующего сервиса systemd.

4. Состав пакетов и назначение компонентов

Компонент	Назначение и ключевые точки интеграции
`gvm-common`	Создаёт системного пользователя/группу `gvm`; подготавливает каталоги данных и runtime (`/var/lib/gvm`, `/run/gvm`); устанавливает `gvm-stack.target`.

Компонент	Назначение и ключевые точки интеграции
redis-openvas	Изолированный экземпляр Redis, используемый компонентами сканирования. Предпочтительная схема доступа — UNIX-сокет /run/redis-openvas/redis.sock (без TCP-публикации).
openvas-scanner	Движок сканирования уязвимостей. Конфигурация: /etc/openvas/openvas.conf.
python-ospd-openvas	OSPD-демон. Обеспечивает интерфейс сканера через сокет /run/ospd/ospd-openvas.sock.
openvasd	Альтернативный демон (Scanner API/Notus) для конфигураций без OSPD. Параметризация через sysconfig и/или TOML.
gvmd	Центральный компонент управления: задания, пользователи, политики, интеграция с БД, оркестрация сканирования. Требует доступность PostgreSQL и сканерного интерфейса.
gsad	Веб-интерфейс управления. Параметры адреса/порта и TLS задаются через /etc/sysconfig/gsad (при наличии).
python3-greenbone-feed-sync	Инструмент синхронизации фидов (NVT/SCAP/CERT) для первичной и периодической актуализации данных.
postgresql-server	Сервер СУБД PostgreSQL для хранения данных менеджера gvmd.

5. Зависимости и порядок запуска

Запуск выполняется через gvm-stack.target. systemd обеспечивает порядок запуска на основе зависимостей (Requires=, Wants=, After=) и условий выполнения.

Логическая модель зависимостей:

```
network-online.target
└─ redis-openvas.service
└─ (ospd-openvas.service | openvasd.service)
└─ gvmd-db-setup.service       [oneshot; при отсутствии /var/lib/gvm/.db-initialized]
```

```
□□ gvmd-provision.service      [oneshot; при отсутствии /var/lib/gvm/.admin-provisioned]
□□ gvmd.service                [требует PostgreSQL и доступность сканера]
□□ gsad.service                [после gvmd]
```

Примечание
Для взаимоисключающих схем (OSPD и `openvasd`) должны использоваться `Conflicts=` и/или `Condition*` в unit-файлах. PostgreSQL должен быть установлен и доступен на момент запуска `gvmd`.

6. Установка и первый ввод в эксплуатацию

6.1. Установка пакетов

Установка выполняется штатным менеджером пакетов НАЙС.ОС (`tdnf`).

```
sudo tdnf install -y \
  gvm-common redis-openvas openvas-scanner python-ospd-openvas openvasd \
  gvmd gsad python3-greenbone-feed-sync postgresql-server
```

6.2. Первичная синхронизация фидов

Первичная синхронизация выполняется oneshot-сервисом. Контроль выполнения осуществляется через журнал systemd.

```
sudo systemctl start greenbone-initial-sync.service
sudo journalctl -u greenbone-initial-sync.service -f
```

6.3. Запуск стека

```
sudo systemctl enable --now gvm-stack.target
```

6.4. Регулярная синхронизация фидов

При наличии таймера синхронизации требуется включить `greenbone-feed-sync.timer`.

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

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

7. Проверка работоспособности

7.1. Статус таргета и сервисов

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

7.2. Проверка Redis

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

7.3. Проверка GVMD

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

7.4. Проверка GSAD

Локальная проверка доступности сокета/порта выполняется сетевыми утилитами. При использовании TLS проверка выполняется через HTTPS.

```
sudo ss -lntp | grep -E '(:9392)\s'
# При необходимости:
curl -kI https://127.0.0.1:9392/
```

8. Конфигурации

Компонент	Файлы конфигурации (типовой перечень)
`openvas-scanner`	`/etc/openvas/openvas.conf`
`ospd-openvas`	`/etc/sysconfig/ospd-openvas`

Компонент	Файлы конфигурации (типовой перечень)
`openvasd`	`/etc/sysconfig/openvasd`, `/etc/openvasd/openvasd.toml`
`gvmd`	`/etc/sysconfig/gvmd`, `/etc/gvm/*.conf`
`gsad`	`/etc/sysconfig/gsad`
`feed-sync`	`/etc/sysconfig/greenbone-feed-sync`

После изменения конфигурации требуется перезапуск соответствующего сервиса: `sudo systemctl restart <service>`.

9. Требования безопасности и ограничения доступа

9.1. Исполнение от непривилегированного пользователя

Сервисы стека исполняются от системного пользователя `gvm` и не должны требовать `root` для штатных операций. Повышенные привилегии допускаются только в строго определённых местах (capabilities) и должны быть зафиксированы в unit-файлах и политике пакета.

9.2. Каталоги runtime и данных

- `/run/gvm`, `/run/ospd`, `/run/redis-openvas` — каталоги runtime, права доступа ограничены владельцем/группой.
- `/var/lib/gvm`, `/var/lib/openvas`, `/var/lib/notus` — каталоги данных, владельцы и режимы должны соответствовать требованиям сервиса.

9.3. Сетевой доступ к GSAD

Если `gsad` слушает `0.0.0.0:9392`, внешняя доступность должна контролироваться правилами входящего трафика. Открытие порта допускается только адресно (по IP/подсети источника), с обязательным использованием TLS и регламентированным управлением учетными записями.

Требование
Запрещается публиковать интерфейс управления в неконтролируемые сети без TLS и ограничений по источнику. Допускается публикация через reverse proxy с проверкой сертификата и дополнительными ACL.

10. Диагностика

10.1. Общая проверка состояния

```
sudo systemctl status gvm-stack.target
sudo systemctl list-dependencies gvm-stack.target
sudo systemctl --no-pager --failed
```

10.2. Анализ журналов

```
sudo journalctl -b -u redis-openvas -u ospd-openvas -u openvasd -u gvmd -u gsad --no-pager
```

10.3. Проверка PostgreSQL

```
sudo systemctl status postgresql --no-pager
sudo -u postgres psql -c 'SELECT version();'
# Дополнительно (пример проверки готовности):
sudo -u postgres pg_isready
```