Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Сканирование уязвимостей корпоративной сети на OpenVAS/GVM

Как мы ставим на конвейер еженедельный авторизованный аудит парка серверов на Greenbone

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Инфраструктуру клиента круглосуточно перебирают боты: находят открытый порт с устаревшей версией сервиса и ставят шифровальщик. Ручной nmap раз в квартал видит порты, но не знает про свежие CVE, покрывает лишь пятую часть парка и не даёт трекинга закрытия. Мы ставим на конвейер еженедельное авторизованное сканирование всего парка на OpenVAS/GVM с приоритизацией по CVSS и контролем SLA.

Почему это важно бизнесу

- Шифровальщик через непропатченный сервис = простой на дни и выкуп; закрыть дыру заранее дешевле, чем восстанавливаться из бэкапа
- 683-П ЦБ, PCI DSS и приказы ФСТЭК №17/21 (защита ИСПДн по 152-ФЗ) прямо требуют регулярного анализа уязвимостей — без него не пройти аудит
- Аудиторам и клиентам нужен воспроизводимый PDF-отчёт с динамикой закрытия, а не Excel с тремя колонками
- Без трекинга находки теряются: нашли — записали — забыли, инфраструктура остаётся дырявой месяцами

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

22.4

текущая ветка Greenbone Community Edition; компоненты gvmd 26.x, openvas-scanner 23.x

по докам Greenbone 22.4

70 %

порог QoD: в отчёт идут только находки с достоверностью ≥70%, ниже — вероятный false positive

дефолтный фильтр GVM

100 000+

проверок (VT) в Community Feed: NASL + notus; обновляем ежедневно через greenbone-feed-sync

по докам Greenbone

x3–5

во столько раз больше находит авторизованный скан (SSH/SMB, LSC) против скана снаружи

наш стандарт

24 ч

SLA закрытия для Critical (CVSS 9.0–10.0); High 7.0–8.9 — 7 дней, фиксируем в трекере

наш стандарт SLA

02:00

старт ночного окна сканирования; safe_checks вкл, опасные проверки выкл, чтобы не ронять прод

наш стандарт

РЕАЛИЗАЦИЯ 1 ИЗ 3

Разворачиваем GVM в Docker на выделенном узле

Что настраиваем

Выделенный сервер 4 vCPU / 8 ГБ RAM / 50 ГБ SSD только под сканер, изолирован от прод-нагрузки

Как мы это делаем

1. Поднимаем стек greenbone community-containers через docker compose: gvmd, ospd-openvas + openvas-scanner, notus-scanner с MQTT-брокером, gsad, Redis, PostgreSQL (pg-gvm)
2. Первичная синхронизация фида greenbone-feed-sync тянет 100 000+ VT (NASL+notus) за 40–60 мин, дальше ежедневный rsync-апдейт по cron
3. gsad публикуем только на 127.0.0.1:9392 за реверс-прокси с TLS; GMP-сокет gvmd наружу не выставляем
4. Проверяем связку по OSP: ospd-openvas видит фид, openvas-scanner 23.x отрабатывает тестовый скан

РЕЗУЛЬТАТ

Обновления и бэкап сводятся к пересборке образов и снапшоту тома PostgreSQL; фид всегда свежий, узел изолирован от прод-нагрузки и не тянет соседние сервисы вниз.

КЛЮЧЕВОЙ НЮАНС

GVM не ставим на машину с другими задачами — при активном скане CPU и RAM уходят под ноль; компоненты версионируются раздельно, ориентируемся на ветку Community Edition 22.4.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Настраиваем авторизованный скан (Local Security Checks)

Что настраиваем

На каждом хосте клиента — отдельная read-only учётка SSH/SMB для чтения списка установленных пакетов

Как мы это делаем

1. Заводим Credentials в gvmd (SSH-ключ для Linux, доменная read-only для Windows/SMB) и привязываем к Target
2. Notus-scanner сверяет установленные версии пакетов с уязвимыми — это ловит CVE, которые снаружи по баннеру не видны
3. Скан-конфиг Full and fast с включённым safe_checks: NVT опасных категорий (ACT_DENIAL, ACT_KILL_HOST) на прод-контуре не выполняются
4. Ставим QoD-порог 70%: в отчёт идут только достоверные находки, шум ниже порога отсекаем

РЕЗУЛЬТАТ

Авторизованный скан находит в 3–5 раз больше реальных уязвимостей, чем внешний; отчёт содержит конкретные пакеты и версии для патчинга, а не догадки по баннерам сервисов.

КЛЮЧЕВОЙ НЮАНС

Скан без учётки видит около 10% проблем — только то, что торчит наружу; ключ и пароль учётки храним в секрет-менеджере, права строго read-only.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Ставим на поток отчёты и контроль закрытия по SLA

Что настраиваем

Десятки клиентов — свои группы Target/Schedule: окно, набор IP и получатель отчёта у каждого

Как мы это делаем

1. Расписания (Schedules) в gvmd: 6–7 клиентов в ночь, старт 02:00, чтобы не нагружать рабочую сеть
2. Alerts (метод Email) шлют PDF-отчёт по SMTP техдиректору клиента каждое утро, приоритизация по CVSS v3.1
3. Ранжируем по SLA: Critical 9.0–10.0 → 24 ч, High 7.0–8.9 → 7 дней, Medium 4.0–6.9 → 30 дней
4. Если Critical не закрыт за 24 ч — alert-метод HTTP Get дёргает Telegram Bot API, эскалация руководителю проекта

РЕЗУЛЬТАТ

Клиент получает воспроизводимый PDF для аудиторов и наглядную динамику закрытия; критичные дыры не теряются в почте, а идут по SLA с автоматической эскалацией.

КЛЮЧЕВОЙ НЮАНС

Сканер диагностирует, но не лечит — без команды на устранение и трекинга закрытия отчёты бесполезны; сверяем delta между сканами, а не абсолютные числа находок.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

✕ **Сканер находит, но не лечит**
GVM — инструмент диагностики. Без выделенной команды на патчинг и трекинга закрытия отчёты пылятся в почте, а дыры остаются открытыми.

✕ **Скан без авторизации почти бесполезен**
Без SSH/SMB-учётки сканер видит около 10% проблем — только внешний периметр. Заводим Credentials и Local Security Checks через notus-scanner.

✕ **Устаревший фид пропускает свежие CVE**
Новые уязвимости выходят ежедневно. Если greenbone-feed-sync не гоняется по cron, база VT стареет и свежие классы дыр просто не детектятся.

✕ **Ложноположительные тонут в объёме**
5–8% находок — false positive. Держим QoD-порог 70% и вручную валидируем Critical/High, иначе бизнес тонет в шуме и теряет реальные риски.

✕ **GVM на общей машине роняет прод**
При активном скане процесс забирает CPU и RAM почти полностью. Ставим на выделенный узел 4 vCPU / 8 ГБ, иначе соседние сервисы падают.

✕ **Сканер путают с пентестом**
Сканер ловит известные CVE по базе. Свежую 0-day или логическую дыру найдёт только живой пентест. Для защиты нужны оба, не подменяем одно другим.

✕ **Опасные проверки на боевом контуре**
NVT категорий ACT_DENIAL и ACT_KILL_HOST выполняются при выключенном safe_checks и могут уронить сервис. На проде safe_checks всегда включён, агрессивные прогоны — только на тестовом контуре.

✕ **gsad открыт в интернет**
Веб-интерфейс с полным доступом к сканеру нельзя выставлять наружу. Публикуем на localhost:9392 за реверс-прокси с TLS и доступом по VPN.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Разовый авторизованный скан всего парка на GVM 22.4, отчёт с приоритизацией по CVSS
- Read-only учётки SSH/SMB на ключевых серверах для Local Security Checks
- Ежедневное обновление фида через greenbone-feed-sync по cron

НОРМАЛЬНО

- Еженедельный скан по расписанию, ночное окно, safe_checks, QoD 70%
- PDF-отчёты клиенту по SMTP-alert, группировка Target по клиентам и сегментам
- SLA по CVSS: Critical 24 ч, High 7 дней, трекинг закрытия находок

ХОРОШО

- Выделенный узел в Docker, изоляция от прод, снапшоты тома PostgreSQL
- Эскалация в Telegram при просрочке Critical, delta-отчёт между сканами
- Повторный скан для верификации закрытия и интеграция с патч-менеджментом
- Разделение прод/тест-контуров: прогоны с отключённым safe_checks — только на тесте

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Сканер вынесен на выделенный узел (4 vCPU / 8 ГБ / 50 ГБ SSD) и изолирован от боевых сервисов?
- [ ] Фид Community обновляется ежедневно через greenbone-feed-sync, база VT не старше суток?
- [ ] На хостах заведены read-only учётки SSH/SMB и включены Local Security Checks (авторизованный скан)?
- [ ] Скан идёт по расписанию еженедельно в ночном окне, а не вручную раз в квартал?
- [ ] Задан QoD-порог 70% и включён safe_checks, опасные проверки выключены на проде?
- [ ] Прописан SLA по CVSS (Critical 24 ч, High 7 дней) и есть трекинг закрытия?
- [ ] Настроена автоотправка PDF-отчётов клиенту и эскалация при просрочке Critical?
- [ ] gsad закрыт от интернета (localhost за TLS-прокси/VPN), GMP-сокет наружу не выставлен?
- [ ] Есть отдельный тест-контур для агрессивных прогонов с отключённым safe_checks?
- [ ] Ведётся delta между сканами, а не только абсолютное число находок?

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.

ЧТО МЫ ДЕЛАЕМ

Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Разворачиваем OpenVAS/GVM в Docker на выделенном узле под ключ за ~20 рабочих дней
- Настраиваем авторизованный скан, расписания и группировку Target по клиентам и сегментам
- Ставим конвейер PDF-отчётов, SLA по CVSS и эскалацию просрочек в Telegram
- Разбираем отчёт вместе с вами, отсекаем false positive и приоритизируем закрытие
- Проводим разовый бесплатный аудит сети с письменным приоритизированным отчётом

15+
лет в ИТ-поддержке

50
рабочих мест — наш профиль

МТС
дата-центр, Москва

КОНТАКТЫ

Обсудить вашу задачу

Сайт	**itfresh.ru**
Телефон	**+7 903 729-62-41**
Telegram	**@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.

itfresh.ru

ДОКУМЕНТАЦИЯ И НАШ СТЕК

Техническая база

01 Greenbone Community Documentation — Architecture (greenbone.github.io — 22.4)

02 gvmd Releases — Greenbone Vulnerability Manager Daemon (github.com/greenbone — 26.2.0)

03 openvas-scanner Releases (github.com/greenbone — 23.47)

04 Greenbone Docs — Notus Scanner (Local Security Checks) (greenbone.github.io — 22.4)

05 Greenbone Docs — Feed sync и Quality of Detection (greenbone.github.io — 22.4)

06 Шаблон SLA/QoD и конвейера отчётов ITfresh (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.