Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

CrowdSec на сервере клиента: коллективная защита вместо fail2ban

Как мы внедряем Security Engine 1.7.8, AppSec и remediation-компоненты на серверах до 50 мест

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

У клиента до 50 мест периметр — это один-два сервера: RDS с 1С, почта, сайт, VPN-шлюз. fail2ban видит только свой лог и считает попытки с одного IP: распределённый перебор по 1-2 попытки с адреса и обход несуществующих URL он не ловит. Мы переносим детект на CrowdSec: агент читает логи, LAPI хранит решения, bouncer блокирует, CAPI даёт репутацию до первого стука.

Почему это важно бизнесу

- Подбор пароля к RDP терминального сервера 1С — это остановка бухгалтерии и склада на рабочий день, а не «инцидент ИБ»
- Сканеры дают тысячи 404 в час: CPU и диск уходят на мусорный трафик, а живые пользователи получают тормоза в 1С
- Утечка базы 1С или CRM — это 152-ФЗ, штрафы и разбор с контрагентами; восстановление доверия дороже любого железа
- Один скомпрометированный сервер в рассылке = домен в чёрных списках, письма клиентам не доходят неделями
- Ручной разбор логов админом — 3-5 часов в месяц на сервер; автоматика возвращает это время на профильные задачи

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

1.7.8

версия Security Engine, которую ставим как базовую: 1.7.8 — стабильная, 1.8.0 доступна только…

docs.crowdsec.net, Getting Started

8080 / 6060

listen_uri LAPI и prometheus.listen_addr: наружу не публикуем, только loopback и доступ через…

docs.crowdsec.net, Network Management

7422

listen_addr AppSec-компонента (WAF): nginx или HAProxy шлют туда запрос на проверку

docs.crowdsec.net, AppSec General Setup

4h

базовый ban в profiles.yaml; рецидивистам срок растим через duration_expr

docs.crowdsec.net, Profiles Format

131072

дефолт ipset_size в cs-firewall-bouncer 0.0.36 (поднят с 65536 в 0.0.28) — не занижаем, блокли…

docs.crowdsec.net, Firewall Bouncer

14 дней

держим сценарии в simulation.yaml до включения реальных банов на узле клиента

наш стандарт внедрения

РЕАЛИЗАЦИЯ 1 ИЗ 3

Базовая обвязка Linux-узла: агент, LAPI и firewall-bouncer

Что настраиваем

Debian 12 / Ubuntu 24.04: VPS с сайтом, почтовый релей или шлюз OpenVPN — один узел, где агент и LAPI живут вместе

Как мы это делаем

1. Подключаем репозиторий и ставим пакет crowdsec: deb/rpm сам зовёт `cscli setup unattended`; проверяем результат через `cscli setup detect` и файлы в /etc/crowdsec/a…
2. Проверяем разбор: `cscli collections list`, `cscli parsers list`, `cscli metrics` — по секции acquisition видно, читаются ли реально nginx, sshd, postfix
3. Ставим crowdsec-firewall-bouncer-nftables (или -iptables); в /etc/crowdsec/bouncers/ — api_url http://127.0.0.1:8080, api_key из `cscli bouncers add`, update_freque…
4. В db_config включаем use_wal: true для sqlite (по умолчанию выключен) и правим flush.max_items и flush.max_age под узел — дефолт 5000 алертов и 7d
5. Первые 14 дней держим боевые сценарии в /etc/crowdsec/simulation.yaml — в `cscli decisions list` решения идут пометкой (simul)ban

РЕЗУЛЬТАТ

Сервер закрыт на уровне nftables-сета: решения LAPI применяет bouncer, а не человек по утрам. Сканеры и переборщики отсекаются до попадания в nginx и sshd, лог перестаёт быть мусорным, а `cscli metrics` даёт понятную картину — что читается, что срабатывает, сколько банов держится.

КЛЮЧЕВОЙ НЮАНС

Смотрим в раздел Network Management: listen_uri 8080 и prometheus 6060 не должны торчать наружу. И помним про iptables_chains — для контейнеров нужна цепочка DOCKER-USER, иначе трафик идёт мимо INPUT.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Windows: терминальный сервер 1С и защита от RDP-перебора

Что настраиваем

Windows Server 2019/2022 с ролью RDS и 1С: агент из MSI плюс Windows Firewall RC, события берём из Event Log

Как мы это делаем

1. Ставим MSI со страницы релизов: бинарники в C:\Program Files\CrowdSec, конфиг, логи, данные и хаб — в C:\ProgramData\CrowdSec\config|log|data|hub
2. В acquis.yaml описываем source: wineventlog с event_channel Security и event_ids 4625; коллекция crowdsecurity/windows ловит RDP/SMB brute-force
3. Есть MSSQL — `cscli collections install crowdsecurity/mssql` и канал Application, event_ids 18456, event_level information (отказ аутентификации SQL Server)
4. Ставим cs-windows-firewall-bouncer (MSI или Chocolatey, нужен .NET-рантайм): он сам регистрируется, ключ при необходимости — `cscli bouncers add`
5. Логи самого файрвола разбирает crowdsecurity/windows-firewall; публикацию IIS закрываем crowdsecurity/iis с use_time_machine: true на C:\inetpub\logs\LogFiles**.l...

РЕЗУЛЬТАТ

RDP-перебор гасится на файрволе хоста, а не политикой блокировки учёток: пользователи 1С не выбиваются из-за чужих попыток входа. SQL-логины и IIS-запросы попадают в тот же поток решений, а картина по всем узлам видна в одной консоли.

КЛЮЧЕВОЙ НЮАНС

На Windows нет автонастройки acquisition, как на Linux, — acquis.yaml пишем руками. IIS пишет лог блоками раз в минуту: без use_time_machine получаем ложные всплески на границе минуты.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Allowlist, профили решений и WAF: тонкая настройка под клиента

Что настраиваем

LAPI-узел клиента: собственные подсети, офисные адреса, мониторинг, плюс AppSec перед nginx или HAProxy

Как мы это делаем

1. Первым делом `cscli allowlists create itfresh -d 'офис, мониторинг, VPN'` и `cscli allowlists add` с нашими адресами; для временного доступа — флаг -e 7d
2. В /etc/crowdsec/profiles.yaml оставляем ban 4h, а рецидив растим через duration_expr: Sprintf('%dh', (GetDecisionsCount(Alert.GetValue())+1)*4)
3. Туда же вешаем notifications: плагин в /etc/crowdsec/notifications/ с group_wait и group_threshold, чтобы Telegram не залило потоком алертов
4. Для веба поднимаем AppSec: в acquis.d — source: appsec, listen_addr 127.0.0.1:7422, appsec_configs: crowdsecurity/appsec-default, коллекции appsec-virtual-patching…
5. CRS включаем сначала out-of-band (crowdsecurity/appsec-crs), правки — плагином в /var/lib/crowdsec/data/crs-plugins/<имя>/*-before.conf, затем crowdsecurity/appsec-…

РЕЗУЛЬТАТ

Себя и клиента мы не банили ни разу: allowlist работает на уровне LAPI и вырезает адреса до вставки в базу — и локальные алерты, и записи из скачанных блоклистов. WAF закрывает свежие CVE виртуальным патчем, пока обновление CMS или портала ещё согласуется с подрядчиком заказчика.

КЛЮЧЕВОЙ НЮАНС

Allowlist понимает только IP и диапазоны. Исключение по URL или user-agent — это parser whitelist или AppSec-правило. Для bouncer'ов allowlist работает сразу, для AppSec подхватывается раз в минуту.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **LAPI и метрики торчат в интернет**
  listen_uri 8080 и prometheus 6060 по умолчанию слушают loopback, но их вешают на 0.0.0.0 «для удобства». Оставляем loopback, доступ — только через VP...

- ✕ **Правка hub-сценария вместо копии**
  Отредактированный сценарий становится tainted: `cscli hub upgrade` его не перезаписывает, сигналы не уходят в CAPI, а IP по этому сценарию не приходя...

- ✕ **Bouncer не поставили вообще**
  Агент детектит, решения в LAPI есть, а блокировать некому. Проверяем `cscli bouncers list` — там должен быть свежий Last API pull, а api_key прописан...

- ✕ **ipset переполняется на блоклистах**
  Дефолт ipset_size — 131072 (до 0.0.28 было 65536). При подписке на несколько блоклистов и заниженном размере сета часть адресов молча не доезжает до...

- ✕ **Docker обходит правила firewall-bouncer**
  Docker публикует порты своими цепочками: без DOCKER-USER в iptables_chains трафик к контейнеру идёт мимо INPUT и бан не срабатывает.

- ✕ **Боевые баны с первого дня**
  Без прогона через simulation.yaml ловим ложные баны на бэкапах, мониторинге и интеграциях 1С. Две недели смотрим (simul)ban, потом включаем реальные.

- ✕ **Ретеншн базы решений по умолчанию**
  Дефолт flush — 5000 алертов и 7d, use_wal у sqlite выключен. На нагруженном узле это даёт распухший crowdsec.db и тормоза LAPI: лимиты и WAL задаём я...

- ✕ **Обновления агента откладывают**
  В 1.7.8 добавлены лимит размера тела запроса в AppSec и лимит распаковки тела в LAPI. Держим пакет и хаб свежими: `cscli hub update && cscli hub upgr...

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Агент и firewall-bouncer на каждом сервере с внешним IP, LAPI на loopback
- Коллекции под реальные сервисы: sshd, nginx, postfix, iis — сверить cscli metrics
- Allowlist с офисными и нашими адресами заведён до включения банов
- Прогон 14 дней через simulation.yaml, затем перевод в боевой режим

НОРМАЛЬНО

- Подключение к CAPI и `cscli console enroll`: единое окно по всем узлам клиента
- profiles.yaml с duration_expr: рецидивистам срок бана растёт кратно
- Уведомления в Telegram с group_wait и group_threshold, без потока алертов
- db_config: use_wal true, flush.max_items и max_age под объём узла

ХОРОШО

- AppSec на 127.0.0.1:7422 перед nginx/HAProxy: virtual patching по свежим CVE
- CRS сначала appsec-crs (out-of-band), кастомизация плагином, потом appsec-crs-inband
- Централизованный LAPI на площадке, агенты и bouncers по TLS-сертификатам (mTLS)
- prometheus.listen_addr и /metrics на 6060 в Zabbix плюс месячный отчёт по банам

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Стоит Security Engine 1.7.8 и заведён регламент `cscli hub update && cscli hub upgrade`?
- [ ] listen_uri 8080 и prometheus 6060 закрыты снаружи, доступ к LAPI только с доверенных адресов?
- [ ] На каждом сервере есть живой bouncer и `cscli bouncers list` показывает свежий Last API pull?
- [ ] Заведён allowlist с офисными, административными и мониторинговыми адресами?
- [ ] Все боевые сценарии прошли simulation и в `cscli decisions list` больше нет (simul)ban?
- [ ] Сроки банов в profiles.yaml заданы осознанно, а не оставлены дефолтом 4 часа?
- [ ] Кастомные сценарии вынесены в отдельные файлы, а `cscli hub list` не показывает tainted?
- [ ] Для контейнеров в iptables_chains добавлена цепочка DOCKER-USER?
- [ ] Настроен плагин уведомлений в /etc/crowdsec/notifications/ и подключён в profiles.yaml?
- [ ] Для веб-приложений включён AppSec и есть план перевода CRS в appsec-crs-inband?

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).

- Аудит логов и периметра сервера: что реально пишется и что не покрыто ни одним парсером
- Развёртывание Security Engine, bouncers и allowlist на Linux и Windows Server под ключ
- Свои сценарии под 1С, RDS, почту и сайт клиента с обязательным прогоном в simulation
- AppSec перед nginx, HAProxy или IIS: virtual patching и настройка CRS без ложных срабатываний
- Абонентское сопровождение: обновления, разбор ложных банов, отчёт по атакам раз в месяц

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 Configuration: Network Management и CrowdSec Configuration (db_config, flu… (docs.crowdsec.net — v1.7)

02 Local API: Profiles Format — duration, duration_expr, notifications (docs.crowdsec.net — v1.7)

03 Local API: Allowlists — cscli allowlists create/add, флаг -e (docs.crowdsec.net — v1.7)

04 Remediation Components: Firewall Bouncer — mode, ipset_size, iptables_chai… (docs.crowdsec.net — v0.0.36)

05 Getting Started on Windows: wineventlog, IIS use_time_machine, MSSQL, wind… (docs.crowdsec.net — v1.7)

06 AppSec Component: General Setup и CRS Installation (appsec-crs, appsec-crs… (docs.crowdsec.net — v1.7)

07 Log Processor: Simulation и FAQ по tainted-сценариям и Community Blocklist (docs.crowdsec.net — v1.7)

08 Наш шаблон внедрения: acquis.d, simulation 14 дней, чек-лист сдачи (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.