Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Аудит действий root на Linux: auditd + Wazuh под требования ЦБ

Неотключаемый журнал действий администраторов: правила auditd, центральный сбор, корреляция в Wazuh

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Регуляторы (ЦБ, ФСТЭК, PCI DSS, 152-ФЗ) требуют журналировать действия привилегированных пользователей, а типовой auditd «из коробки» пишет только факт логина. Без журнала инцидент нерасследуем, а на предписание регулятора даётся жёсткий срок. Мы строим неотключаемый контур: правила auditd под стандарт, иммутабельная конфигурация, копия логов на внешний коллектор в реальном времени, корреляция в Wazuh.

Почему это важно бизнесу

- Предписание регулятора имеет срок (обычно 60–90 дней); срыв — штрафы и ограничение операций
- Без журнала инцидент нерасследуем: простой растягивается с часов на дни, причину не найти
- Журнал защищает самих админов: легко доказать, что сбой вызвал не «последний вошедший»
- Подрядчики и уволенные: root-доступ без записи действий — слепое доверие и забытые ключи
- Попытка замести следы (чистка логов, стоп auditd) фиксируется раньше, чем завершится

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

4.1.2

актуальная ветка audit-userspace; синтаксис правил совместим с 3.x в текущих дистрибутивах

github.com/linux-audit

-e 2

иммутабельный режим: правки правил только через перезагрузку — аудит не отключить на ходу

man auditctl(8)

8192

backlog_limit вместо kernel default 64 — не теряем события при пиках системных вызовов

наш стандарт / man auditctl(8)

12 мес

ретенция журналов по Req 10.5.1, последние 3 месяца — в горячем доступе для анализа

PCI DSS v4.0.1, Req 10

4.14.6

Wazuh: localfile log_format=audit разбирает журнал, ключи -k матчатся CDB-списком audit-keys

documentation.wazuh.com

200 МБ/сут

объём журнала на сервер при полном наборе правил — база сайзинга хранилища коллектора

наш стандарт

РЕАЛИЗАЦИЯ 1 ИЗ 3

Базовый контур auditd на каждом сервере

Что настраиваем

Все Linux-серверы с критичными данными: web, приложения, БД, batch-обработка

Как мы это делаем

1. auditd.conf: log_format=ENRICHED, flush=incremental_async + freq=50, max_log_file_action=keep_logs
2. Пороги диска: space_left=25% и space_left_action=email, admin_space_left=10% → single, disk_full_action=halt (PCI-профиль)
3. pam_loginuid + audit=1 в cmdline ядра: каждый sudo-сеанс несёт неизменяемый auid реального пользователя
4. Финальные строки набора правил: -f 1 на сбои и -e 2 — конфигурация иммутабельна до перезагрузки

РЕЗУЛЬТАТ

Журнал переживает пики нагрузки и заполнение диска по сценарию регулятора; каждая запись содержит auid реального инициатора, а не обезличенного root — расследование sudo-сеансов занимает минуты.

КЛЮЧЕВОЙ НЮАНС

halt для space_left_action в актуальных версиях deprecated — жёсткую остановку вешаем только на disk_full_action/disk_error_action, предупреждающие пороги обязаны сработать раньше.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Правила: критичные файлы и все команды root

Что настраиваем

Набор правил под PCI DSS Req 10 + профиль ЦБ/ГОСТ Р 57580 на каждом хосте

Как мы это делаем

1. -w /etc/passwd, shadow, sudoers -p wa -k identity; отдельные ключи на sshd_config, PAM, cron — поиск ausearch -k за секунды
2. -a always,exit -F arch=b64 -S execve -F euid=0 -F auid!=unset -k root_cmd — все команды root с живым auid; дублируем для arch=b32
3. Исключения от шума: -a never,exit по exe/uid мониторинговых агентов — ставим первыми, правила отрабатываются по порядку
4. Обкатка на стенде: augenrules --load, неделя замера EPS и объёма через aureport --summary, только потом прод и -e 2

РЕЗУЛЬТАТ

67 правил закрывают PCI DSS Req 10 и профиль ЦБ: identity-файлы, SSH/PAM/cron, все команды root. После тюнинга исключений полезный сигнал не тонет в шуме — объём стабилен около 200 МБ/сутки на сервер.

КЛЮЧЕВОЙ НЮАНС

Список правил фильтруется последовательно: never-исключения размещаем до always-правил, иначе шум агентов мониторинга съедает до 90% объёма журнала и маскирует реальные события.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Централизация и корреляция: audisp-remote + Wazuh

Что настраиваем

Изолированный лог-коллектор + Wazuh manager/agents 4.14

Как мы это делаем

1. Плагин au-remote: remote_server=коллектор, port=60, local_port<1024; на приёмнике tcp_listen_port=60 + tcp_client_ports
2. network_failure_action и очередь плагина: обрыв сети не роняет поток, события копятся локально и досылаются
3. Wazuh-agent: localfile log_format=audit на /var/log/audit/audit.log; наши ключи -k матчатся против CDB audit-keys
4. Алерты на изменение sudoers/PAM/authorized_keys и любые попытки тронуть auditd — в Telegram-канал службы ИБ
5. aureport --login, --failed, -x в cron — ежедневная утренняя сводка руководителю СБ

РЕЗУЛЬТАТ

Локальный журнал можно стереть — центральная копия уже записана; сама попытка тронуть auditd поднимает алерт. Служба безопасности получает готовые события и утреннюю сводку, а не сырые логи.

КЛЮЧЕВОЙ НЮАНС

transport=tcp у audisp-remote — открытый текст: между площадками заворачиваем поток в KRB5 либо изолированный management-VLAN/VPN, коллектор держим вне доменных учёток админов.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Шум мониторинговых агентов**
  Zabbix/osquery генерируют тысячи однотипных syscall-событий и забивают журнал. Ставим -a never,exit по exe/uid агента в начало списка правил.

- ✕ **auid=unset у сервисных сеансов**
  Без pam_loginuid в стеке sshd/login/su auid не проставляется, и фильтр auid!=unset режет события. Проверяем session-стек PAM на всех точках входа.

- ✕ **keep_logs без порогов диска**
  max_log_file_action=keep_logs копит файлы бесконечно. Пороги space_left/admin_space_left с алертами обязаны сработать задолго до disk_full_action.

- ✕ **-e 2 включён до отладки**
  Иммутабельный режим требует перезагрузки для любой правки. Включаем его последним шагом, после недели обкатки правил на стенде и замера объёма.

- ✕ **Жёсткий halt не согласован**
  disk_full_action=halt останавливает сервер — для PCI это норма, для остальных шок. Режим (halt/single/suspend) фиксируем с заказчиком письменно.

- ✕ **Единственная локальная копия**
  Локальная ротация и root-доступ убивают доказательность: журнал можно стереть. audisp-remote шлёт копию на изолированный коллектор в реальном времени.

- ✕ **Пропущенный arch=b32**
  На x86_64 системные вызовы идут в двух ABI: правило только для arch=b64 пропустит 32-битные вызовы. Дублируем каждое syscall-правило для b32 и b64.

- ✕ **Сырой журнал никто не читает**
  Записи audit.log многострочны и нечитаемы глазами. Только ausearch/aureport по ключам -k плюс корреляция в SIEM — иначе журнал мёртвый груз.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- auditd с watch-правилами на /etc/passwd, shadow, sudoers, sshd_config (-p wa)
- Логирование всех execve под euid=0 с ключом root_cmd и живым auid
- Ротация keep_logs, хранение 90 дней, разбор через ausearch по ключам

НОРМАЛЬНО

- Иммутабельная конфигурация -e 2, backlog 8192, пороги диска с алертами
- Центральный сбор audisp-remote на изолированный коллектор (port 60)
- Wazuh: декодер audit + алерты на sudoers, PAM, authorized_keys

ХОРОШО

- Полный профиль PCI DSS Req 10 / ГОСТ Р 57580: 60–70 правил с обкаткой на стенде
- Ретенция 12 мес (3 мес hot), disk_full_action по требованиям регулятора
- Ежедневные aureport-сводки + алерты в мессенджер ИБ с разбором каждого
- Регламент: любая попытка тронуть auditd = инцидент высокого приоритета

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Проставляется ли auid реального пользователя в каждой записи (pam_loginuid на sshd, login, su)?
- [ ] Заблокировано ли отключение аудита на ходу (-e 2 последней строкой набора правил)?
- [ ] Уходит ли копия журнала на внешний коллектор в реальном времени?
- [ ] Покрыты ли оба ABI — arch=b64 и arch=b32 — в каждом syscall-правиле?
- [ ] Согласовано ли с бизнесом поведение при заполнении диска (halt/single/suspend)?
- [ ] Хватает ли хранилища: ~200 МБ/сутки на сервер при ретенции 12 месяцев?
- [ ] Стоят ли never-исключения для мониторинговых агентов раньше always-правил?
- [ ] Алертит ли SIEM на изменения sudoers, PAM, authorized_keys и попытки тронуть auditd?
- [ ] Получает ли служба безопасности ежедневную сводку aureport?

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).

- Оценим разрыв текущего журналирования с требованиями ЦБ, ФСТЭК, PCI DSS, 152-ФЗ
- Внедрим auditd под ключ: правила, иммутабельность, пороги диска, центральный сбор
- Интегрируем с Wazuh: декодеры, CDB-списки, алерты в Telegram службы безопасности
- Сопроводим: разбор алертов, тюнинг правил, ежемесячный отчёт руководству
- Подготовим к проверке регулятора: чек-лист, доказательная база, прогон сценариев

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 auditd.conf(5) — пороги диска, flush, действия при отказах (man7.org — 4.1)

02 auditctl(8) — синтаксис правил, -e 2, backlog, фильтры auid (man7.org — 4.1)

03 audisp-remote.conf(5) — удалённая отправка журнала (github.com/linux-audit — 4.1)

04 Release notes audit-userspace 4.1.x (github.com/linux-audit — 2025)

05 Wazuh: Monitoring system calls, log_format audit (documentation.wazuh.com — 4.14)

06 PCI DSS v4.0.1, Requirement 10 (Log and Monitor) (pcisecuritystandards.org — 2024)

07 Наш шаблон правил под профиль ЦБ / ГОСТ Р 57580 (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.