Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Linux Audit System: auditd как чёрный ящик сервера

Правила rules.d, лимиты auditd.conf, ausearch/aureport и вынос событий с хоста по TCP/TLS

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

На Linux-серверах клиентов (1С-публикации, PostgreSQL, файловые шары, веб) логи приложений не отвечают на вопрос «кто и когда открыл /etc/shadow, поменял sudoers или удалил данные». Без журнала уровня ядра инцидент разбирается по догадкам, а при проверке по 152-ФЗ нечего предъявить. Мы ставим auditd с правилами по ключам и выносим события с хоста.

Почему это важно бизнесу

- Разбор «кто удалил данные» без auditd занимает дни и заканчивается ничем; с ключами правил — 20 минут по ausearch -k.
- 152-ФЗ и ISO 27001 требуют регистрацию доступа к ПДн; локальный лог, который может стереть root, проверку не проходит.
- Скомпрометированный деплой-аккаунт или бывший админ в 3 ночи виден по auid, даже если работал через sudo от root.
- Переполненный диск переводит auditd в SUSPEND, и аудит молча останавливается — пороги и алерты закладываем заранее.

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

3.1.2 / 4.2

3.0.9 Debian 12, 3.1.2 Ubuntu 24.04, 4.0.3 RHEL 10, upstream 4.2.2; конфиги пишем под 3.0+

ChangeLog audit-userspace

-b 8192

backlog_limit вместо дефолтных 64: при всплеске событий ядро теряет записи (lost в auditctl -s)

10-base-config.rules

60 000

--backlog_wait_time в 10-base-config (jiffies, ≈60 с при HZ=1000); -f 1 = printk, не panic

по докам auditctl(8)

100 МБ x10

max_log_file / num_logs вместо 8x5 по умолчанию; для ПДн-контуров 200x30 (6 ГБ); ROTATE

наш стандарт

75 / 50 МБ

space_left → SYSLOG (+email), admin_space_left → SUSPEND; с audit 3.0 порог можно задать в %

по докам auditd.conf(5)

auid≥1000

-F auid>=1000 -F auid!=unset в syscall-правилах: пишем действия людей, а не cron и демонов

30-stig.rules

РЕАЛИЗАЦИЯ 1 ИЗ 3

Базовый профиль auditd на каждом Linux-сервере

Что настраиваем

Ubuntu 22.04/24.04, Debian 12, RHEL/Rocky 9–10: все серверы клиента с ПДн, 1С-публикациями и базами данных

Как мы это делаем

1. apt install auditd audispd-plugins / dnf install audit; systemctl enable --now auditd; сверка auditctl -s: enabled=1, backlog_limit, lost=0
2. auditd.conf: log_format=ENRICHED, name_format=hostname, max_log_file=100, num_logs=10, max_log_file_action=ROTATE, flush=INCREMENTAL_ASYNC, freq=50
3. rules.d по номерам: 10-base-config (-D, -b 8192, --backlog_wait_time 60000, -f 1) → 11-loginuid (--loginuid-immutable) → 30-itfresh-security → 99-finalize
4. Watch-правила: /etc/passwd, shadow, group, gshadow -p wa -k identity; /etc/sudoers и sudoers.d -k actions; sshd_config, pam.d, cron — свои ключи
5. augenrules --load, auditctl -l для сверки; smoke-тест перед сдачей: ausearch -k identity --start today -i после тестовой правки /etc/group

РЕЗУЛЬТАТ

На каждом сервере единый набор ключей (identity, actions, perm_mod, delete, privileged, module-load), одинаковый на Debian и RHEL: инженер находит событие одной командой по ключу, а не разбирает сырой audit.log.

КЛЮЧЕВОЙ НЮАНС

Правила через auditctl живут до перезагрузки; постоянные — только rules.d + augenrules. -e 2 включаем последним файлом и после недели обкатки: снять его без ребута нельзя.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Syscall-правила: root-команды, привилегии, удаления, модули

Что настраиваем

Серверы с данными: PostgreSQL под 1С, Samba-шары, веб-корни, каталоги бэкапов

Как мы это делаем

1. execve -F auid>=1000 -F auid!=unset -C euid!=uid (или -F euid=0) -k privileged по arch=b64 и b32: каждая команда через sudo/su с исходным auid
2. unlink,unlinkat,rename,renameat -k delete; chmod/fchmod/fchmodat, chown-семейство -k perm_mod; open,creat,truncate -F exit=-EACCES и -EPERM -k access
3. -w /var/lib/postgresql -p wa -k pg_data (без r!), -w /etc/postgresql -p wa -k pg_config, -w /backup -p rwa -k backup_access; mount, umount2 -k mount_ops
4. 43-module-load.rules: init_module, finit_module, delete_module; 42-injection.rules: ptrace с a0=0x4/0x5/0x6 -k code/data/register-injection
5. 31-privileged.rules по эталону: find /{,usr/}{bin,sbin} -perm -04000 + filecap → -F path=… -F perm=x -F auid>=1000 -k privileged; 21-no32bit при запрете 32-бит

РЕЗУЛЬТАТ

«Кто удалил файл», «кто запускал psql от деплой-аккаунта», «кто грузил модуль ядра» закрываются за минуты: ausearch -k delete -i, ausearch -ua <user> --start yesterday, aureport -x --summary — с реальным auid, а не root.

КЛЮЧЕВОЙ НЮАНС

Порядок полей важен: arch указываем до -S и до -F perm, иначе auditctl не подберёт таблицу syscall. -p r на активных каталогах даёт тысячи событий в секунду и забивает backlog — чтение пишем точечно.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Вынос событий с хоста: audisp-remote, syslog, SIEM

Что настраиваем

Парк от 5 Linux-хостов: центральный auditd-коллектор или rsyslog → Graylog/ELK, статус в Zabbix

Как мы это делаем

1. Клиент: plugins.d/au-remote.conf active=yes; audisp-remote.conf: remote_server, port=60, transport=tcp в VPN или krb5; tls с PSK — только с audit 4.2
2. Очередь: mode=forward, queue_file=/var/spool/audit/remote.log, queue_depth=10240, network_failure_action=syslog (дефолт stop глушит плагин), heartbeat_timeout=60
3. Коллектор: auditd.conf tcp_listen_port=60, tcp_max_per_addr=2, tcp_client_max_idle=300, use_libwrap=yes + allow-лист источников в hosts.allow
4. Параллельно plugins.d/syslog.conf: args=LOG_INFO LOG_LOCAL6 → rsyslog → Graylog; на клиенте name_format=hostname, чтобы события не обезличились
5. Алерты: триггеры на ключи actions/identity/module-load, на «auditd suspended» и lost>0 из auditctl -s в Zabbix; ежедневно aureport --failed --start yesterday

РЕЗУЛЬТАТ

Локальный audit.log перестаёт быть единственной копией: даже если атакующий получил root и вычистил /var/log/audit, события уже на коллекторе; расследование ведётся из одного места по всем хостам.

КЛЮЧЕВОЙ НЮАНС

С audit 3.0 плагины живут в /etc/audit/plugins.d, а не в /etc/audisp — старые инструкции ломают установку. Слово interpret в args syslog-плагина не добавляем: наивный парсер обманывается именами файлов.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Правила только через auditctl**
  Живут до ребута. Всё кладём в /etc/audit/rules.d/*.rules и грузим augenrules --load; auditctl -l используем только для сверки.
- ✕ **Дефолтный backlog 64**
  Всплеск событий → lost>0 в auditctl -s и дыра в журнале. Ставим -b 8192 и --backlog_wait_time 60000 первыми строками 10-base-config.rules.
- ✕ **-p r на живых каталогах**
  Чтение /var/lib/postgresql или веб-корня даёт тысячи записей в секунду, диск и CPU уходят. Чтение включаем точечно под конкретный файл.
- ✕ **Нет фильтра по auid**
  Без -F auid>=1000 -F auid!=unset журнал забит cron и демонами. Действия людей отделяем по auid — он переживает sudo и su.
- ✕ **SUSPEND без алерта**
  admin_space_left_action=SUSPEND тихо останавливает аудит. Алерт на SYSLOG-порог space_left и статус auditd держим в Zabbix.
- ✕ **-e 2 до обкатки**
  Immutable-режим требует ребута для любой правки. Включаем в 99-finalize только после недели наблюдения за объёмом и ложными событиями.
- ✕ **Плагины в /etc/audisp**
  С audit 3.0 конфиги в /etc/audit/plugins.d и /etc/audit/audisp-remote.conf, audispd слит с auditd. По старым путям плагин просто не стартует.
- ✕ **Только b64-правила**
  32-битный execve через arch=b32 обходит журнал. Дублируем правила по b32 либо ставим 21-no32bit.rules, если 32-бит на хосте запрещён.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- auditd включён, rules.d с watch на passwd/shadow/sudoers/sshd_config
- max_log_file=100, num_logs=10, ROTATE; space_left_action=SYSLOG
- -b 8192 и -f 1 в 10-base-config.rules; auditctl -s без lost
- Еженедельный aureport --summary инженеру на почту

НОРМАЛЬНО

- Syscall-правила по 30-stig: identity, perm_mod, delete, access, privileged, modules
- auid-фильтры, --loginuid-immutable, log_format=ENRICHED, name_format=hostname
- Ежедневный ausearch --checkpoint по ключам actions/identity → Telegram
- Zabbix: статус службы, lost в auditctl -s, заполнение /var/log/audit

ХОРОШО

- audisp-remote на коллектор (tcp в VPN/krb5; tls с 4.2), mode=forward, очередь на диск
- syslog-плагин → Graylog/ELK, корреляция по auid и ключам между хостами
- -e 2 в 99-finalize после обкатки; 31-privileged.rules перегенерируем после обновлений
- Регламент разбора инцидента: ключ → auid → сессия → команды, норматив 20 минут

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Правила лежат в /etc/audit/rules.d и переживают перезагрузку (auditctl -l после ребута совпадает)?
- [ ] auditctl -s показывает enabled=1, backlog_limit не меньше 8192 и lost=0?
- [ ] Есть watch на /etc/passwd, /etc/shadow, /etc/sudoers(.d), sshd_config, pam.d, cron с отдельными ключами?
- [ ] execve через sudo/su пишется с реальным auid пользователя, а не только как root?
- [ ] Удаление файлов в каталогах с данными (unlink/rename) попадает в журнал с ключом delete?
- [ ] Известно, что произойдёт при заполнении /var/log/audit (SUSPEND), и кто узнает об этом первым?
- [ ] Копия событий уходит с хоста (audisp-remote или syslog) и недоступна для удаления локальному root?
- [ ] Кто получает ежедневный aureport --failed и алерты по ключам actions/identity?
- [ ] Загрузка модулей ядра и ptrace-инъекции фиксируются (43-module-load, 42-injection)?
- [ ] Проведён учебный разбор: «кто менял sudoers вчера» найдено за 5 минут по ausearch -k actions?

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 серверов и типовой профиль auditd под Debian/Ubuntu/RHEL: rules.d, auditd.conf, ключи под ваши данные
- Правила под конкретные активы: PostgreSQL и 1С-публикации, веб-корни, бэкапы, SSH-ключи, cron — без лишнего шума
- Централизация: коллектор audisp-remote, rsyslog → Graylog/ELK, дашборды и алерты в Zabbix и Telegram
- Регламент и обучение: разбор инцидента по ausearch/aureport за 20 минут, отчёты для 152-ФЗ и ISO 27001
- Сопровождение: контроль lost/backlog, ротация, перегенерация privileged-правил после обновлений пакетов

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): max_log_file, space_left %, tcp_listen_port, transport (github.com/linux-audit — 4.2.2)

02 auditctl(8): -b, --backlog_wait_time, -f, -e, -F auid/perm/exit, -C, arch (github.com/linux-audit — 4.2.2)

03 audisp-remote.conf(5), audisp-syslog(8): mode, queue_depth, tls_psk (github.com/linux-audit — 4.2.2)

04 audit-userspace/rules: 10-base-config, 30-stig, 31-privileged, 42–44, 99 (github.com/linux-audit — 4.2.x)

05 ausearch(8) / aureport(8): --checkpoint, --start, -ua, --failed, --summary (github.com/linux-audit — 4.2.2)

06 RHEL 10 Risk reduction and recovery operations → Auditing the system (docs.redhat.com — RHEL 10)

07 Пакеты auditd: Ubuntu 24.04 (1:3.1.2) и Debian 12 (1:3.0.9) (packages.ubuntu.com — 2024)

08 Шаблон ITfresh: 30-itfresh-security.rules + Zabbix-триггеры auditd (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.