Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

AppArmor: изоляция сервисов в Linux без остановки бизнеса

Как мы профилируем веб-стек, демоны и контейнеры на Ubuntu/Debian и держим MAC под контролем

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Взлом одного сервиса (PHP-FPM, самописный демон, контейнер) на сервере клиента без MAC-политики превращается в захват всего хоста: чтение /etc/shadow, подмена SSH-ключей, cron-закладки, доступ к бэкапам и базе 1С. Мы закрываем это AppArmor-профилями: даже при RCE процесс остаётся в перечисленных путях и портах.

Почему это важно бизнесу

- Простой сайта/1С после компрометации — от 4 часов до недели; профиль ограничивает ущерб одним сервисом
- Утечка клиентской базы = штрафы 152-ФЗ и репутация; MAC не даёт www-data дотянуться до дампов и бэкапов
- Реагирование на инцидент дороже внедрения в 3–5 раз: профиль пишется за 2–3 дня, живёт годами
- Требования заказчиков/аудиторов к харденингу серверов закрываются штатным LSM без покупки СЗИ
- Журнал DENIED — ранний сигнал атаки за часы до того, как её заметят по симптомам

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

4.1.8

Стабильная LTS-ветка 4.1 (релиз 08.2026); 5.0.x — development-ветка; Ubuntu 24.04 LTS несёт 4....

Release Notes, wiki gitlab.com/apparmor

abi/4.0

abi <abi/4.0> в каждом профиле — без него парсер берёт default ABI без фич AppArmor 3+: userns...

apparmor.d(5), раздел Feature ABI

14 сут

Минимум complain до enforce: успевают отработать logrotate, недельные cron, месячные отчёты

наш стандарт

-Q -W

parser -Q (skip-kernel-load) проверяет синтаксис в CI без root, -W пишет кэш в /var/cache/appa...

apparmor_parser(8)

5 мин

Окно триггера Zabbix на apparmor="DENIED" в audit.log для enforce-хостов; complain — дайджест

наш стандарт

= 0

Целевое число процессов «unconfined but have a profile defined» в aa-status после деплоя

aa-status, наш стандарт

РЕАЛИЗАЦИЯ 1 ИЗ 3

Профиль PHP-FPM и nginx для сайта на Bitrix/WordPress

Что настраиваем

Ubuntu 24.04 LTS, nginx + php8.3-fpm, отдельный pool на сайт; профиль php-fpm из apparmor-profiles + хэты на pool, свой профиль n...

Как мы это делаем

1. Ставим apparmor-utils, apparmor-profiles, auditd; aa-status: mysqld (usr.sbin.mysqld) уже в enforce, штатного профиля nginx нет — пишем свой (aa-genprof)
2. php-fpm из apparmor-profiles (attach /usr/{bin,sbin}/php-fpm*): в pool apparmor_hat = site1, хэт ^site1 — в /etc/apparmor.d/php-fpm.d/site1; abstractions/php, base
3. Руками ужимаем: запись только в @{DOCROOT}/upload/** и /tmp/**; исполнение — deny @{DOCROOT}/upload/** x, /tmp/** x; сеть — network inet stream + unix
4. aa-complain на 14 суток, ежедневно aa-logprof по /var/log/audit/audit.log; затем aa-enforce и apparmor_parser -r -W
5. audit deny /etc/shadow r, audit deny /root/** rwkx, audit deny @{HOME}/.ssh/** rw — чтобы попытки были видны в журнале, а не тонули

РЕЗУЛЬТАТ

Webshell в upload/ не может ни выполниться, ни прочитать конфиг БД соседнего сайта, ни подложить ключ в authorized_keys; каждая попытка — строка type=AVC apparmor="DENIED" с профилем и путём, по которой мы ловим атаку в первые минуты.

КЛЮЧЕВОЙ НЮАНС

Профиль из apparmor-profiles — conffile, перезапишется при apt upgrade: свои правила кладём в /etc/apparmor.d/php-fpm.d/ и local/php-fpm (include if exists), а не в основной файл; nginx — отдельный usr.sbin.ng...

РЕАЛИЗАЦИЯ 2 ИЗ 3

Связка AppArmor + systemd для самописных демонов и интеграций 1С

Что настраиваем

Python/Go-сервисы обмена (OData-шлюзы, боты, очереди) в /opt/<svc> на Debian 12 и Ubuntu 24.04, запуск через systemd

Как мы это делаем

1. Профиль /etc/apparmor.d/opt.svc.bin.svc: abi <abi/4.0>, profile svc /opt/svc/bin/svc; переменные @{SVC_HOME}, @{SVC_LOG}; бинарь mrix, данные rwk, конфиг r, TLS-кор…
2. В юните: AppArmorProfile=svc без «-» — если профиль не загружен, юнит не стартует; это страховка от случайного apparmor_parser -R
3. Дублируем изоляцию средствами systemd: NoNewPrivileges=true, ProtectSystem=strict, ReadWritePaths=/var/lib/svc /var/log/svc, ProtectHome=true, PrivateTmp=true
4. CapabilityBoundingSet= сводим к CAP_NET_BIND_SERVICE; в профиле — только capability net_bind_service, и network inet stream
5. Отладка запуска: aa-exec -p svc -- /opt/svc/bin/svc --check и journalctl -k -g DENIED

РЕЗУЛЬТАТ

Два независимых слоя: даже если профиль ослабят при правке, ProtectSystem=strict и NoNewPrivileges не дадут писать в /usr и /etc и поднять привилегии; и наоборот. Деплой нового хоста воспроизводим из репозитория без ручных шагов.

КЛЮЧЕВОЙ НЮАНС

apparmor.service (Before=sysinit.target) грузит /etc/apparmor.d раньше сервисов, поэтому AppArmorProfile= работает; профиль, выкаченный Ansible «на горячую», сразу грузим -r, иначе restart юнита упадёт.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Контейнеры Docker, userns-ограничения и мониторинг DENIED

Что настраиваем

Docker CE на Ubuntu 24.04 (kernel.apparmor_restrict_unprivileged_userns=1), Zabbix 7.0, профили в Git/Ansible

Как мы это делаем

1. docker-default остаётся базой; для nginx/redis-контейнеров пишем свой профиль, грузим apparmor_parser -r -W, запускаем --security-opt apparmor=<имя>
2. --privileged и apparmor=unconfined запрещаем политикой; для rootless-инструментов (bwrap, Electron) — профиль flags=(unconfined) с правилом userns, а не сброс sysctl
3. auditd включён: AVC пишется в /var/log/audit/audit.log; ротация в auditd.conf: num_logs = 8, max_log_file_action = ROTATE
4. Zabbix: item log[/var/log/audit/audit.log,"apparmor=\"DENIED\"",,,skip], триггер по count за 5m для enforce-хостов; aa-notify -s 1 в утренний дайджест
5. Все профили — в репозитории роли Ansible; pipeline гоняет apparmor_parser -Q --abort-on-error на каждый MR, деплой — copy + parser -r -W

РЕЗУЛЬТАТ

Побег из контейнера через volume или уязвимость рантайма упирается в профиль хоста; попытки видны в Zabbix в течение 5 минут. Восстановление сервера после сбоя возвращает те же профили, что и в проде, — без «забыли включить».

КЛЮЧЕВОЙ НЮАНС

Профиль Docker защищает контейнеры, а не dockerd: профиль демона в deb-пакете не ставится, его при необходимости берём из contrib/apparmor исходников Engine и ведём отдельно.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **AppArmor «есть», но ядро его не грузит**
  На старых Debian/кастомных ядрах aa-enabled даёт No. Смотрим /sys/module/apparmor/parameters/enabled=Y, при нужде apparmor=1 security=apparmor в GRUB

- ✕ **deny-правила работают и в complain**
  Явный deny применяется даже в режиме обучения (aa-complain(8)) — сервис ломается «внезапно». В complain-фазе deny держим закомментированными, включае…

- ✕ **Обновление пакета затёрло правки**
  Профили из apparmor-profiles идут как conffile. Дополнения только в /etc/apparmor.d/local/<профиль>, который подключён через include if exists.

- ✕ **userns-ограничение ломает софт**
  Ubuntu 24.04: apparmor_restrict_unprivileged_userns=1 бьёт bwrap/Electron/rootless. sysctl не трогаем — точечный профиль flags=(unconfined) с правило…

- ✕ **Неделя complain — мало**
  Месячные cron, ротация логов и разовые операции админки всплывают позже. Наш минимум 14 суток плюс ручной прогон всех джобов из crontab перед enforce.

- ✕ **Логов DENIED «нет»**
  Без auditd сообщения идут в journal/kern.log: aa-notify откатится на kern.log, а item Zabbix смотрит в /var/log/audit/audit.log. auditd ставим на все…

- ✕ **Устаревший кэш профилей**
  После смены ядра политика из /var/cache/apparmor может не совпасть по features. Перегружаем apparmor_parser -r -T -W, в профиле фиксируем abi.

- ✕ **Символическая ссылка вместо бинаря**
  Профиль привязан к пути exec. После usrmerge или переименования (/bin → /usr/bin) он не цепляется — attachment пишем с альтернацией /{usr/,}{s,}bin/<…

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- aa-enabled=Yes, штатные профили mysqld/cupsd в enforce, свой профиль nginx, aa-statu…
- auditd установлен, ротация через auditd.conf (num_logs/ROTATE), DENIED виден в journ…
- Docker только с docker-default, --privileged запрещён политикой
- Профили под контролем версий, деплой через Ansible, а не руками

НОРМАЛЬНО

- Свои профили для nginx, apache/1С-публикаций и самописных демонов; php-fpm — хэты на…
- 14 суток complain + aa-logprof, затем enforce; audit deny на shadow/root/.ssh
- AppArmorProfile= + NoNewPrivileges + ProtectSystem=strict в юнитах своих сервисов
- Zabbix-триггер на apparmor="DENIED" за 5 минут для enforce-хостов

ХОРОШО

- abi <abi/4.0>, дочерние профили (Cx) для воркеров, capability и network по минимуму
- CI: apparmor_parser -Q --abort-on-error на каждый MR, тест aa-exec в staging
- Свои профили контейнеров через --security-opt apparmor=, отдельный профиль dockerd
- Ежеквартальный пересмотр профилей после обновлений приложений и ядра

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] aa-enabled отвечает Yes на всех Linux-серверах, а в aa-status нет unconfined-процессов с профилем?
- [ ] Для nginx/apache и каждого самописного демона есть свой профиль в enforce, а у php-fpm — хэт на каждый pool?
- [ ] Профили прошли не менее 14 суток complain и разбор aa-logprof, включая cron и ротацию логов?
- [ ] Свои правки лежат в /etc/apparmor.d/local/, а не в файлах, которые перезапишет apt?
- [ ] В юнитах systemd прописан AppArmorProfile= без «-» и включён NoNewPrivileges=true?
- [ ] auditd пишет /var/log/audit/audit.log, а Zabbix поднимает триггер по DENIED в течение 5 минут?
- [ ] Контейнеры запускаются с docker-default или своим профилем, --privileged и unconfined запрещены?
- [ ] На Ubuntu 24.04 apparmor_restrict_unprivileged_userns оставлен =1, исключения оформлены профилями?
- [ ] Все профили хранятся в Git/Ansible и проходят apparmor_parser -Q перед выкаткой?
- [ ] После обновления ядра или приложения профили перепроверены и кэш перегружен (-T -W)?

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 состояния LSM на серверах клиента: aa-status, unconfined-процессы, кэш, GRUB, auditd — отчёт с приоритетами
- Написание и обкатка профилей под веб-стек, публикации 1С, самописные интеграции и контейнеры: complain → enforce
- Харденинг юнитов systemd (AppArmorProfile, ProtectSystem, NoNewPrivileges, Capability) в связке с профилями
- Мониторинг DENIED в Zabbix 7.0 с триггерами и утренним дайджестом aa-notify, реагирование по регламенту
- Роль Ansible с профилями в Git, CI-проверка apparmor_parser -Q и воспроизводимый деплой на новые хосты

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

Техническая база

01 apparmor.d(5) — синтаксис профилей, флаги режимов, Feature ABI, правила fi… (manpages.ubuntu.com — 24.04)

02 apparmor_parser(8) — ключи -r/-R/-C/-Q/-T/-W, --abort-on-error, кэш /var/c… (manpages.ubuntu.com — 24.04)

03 AppArmor Release Notes 4.1.8 и 5.0.2 — статус веток: 4.1 LTS, 5.0 developm… (gitlab.com/apparmor (wiki) — 2026)

04 Release Notes 4.1.0 — priority-модификатор, port range в network, unconfin… (gitlab.com/apparmor (wiki) — 2025)

05 aa-complain(8), aa-notify(8), auditd.conf(5) — deny в complain, источник л… (manpages.ubuntu.com — 24.04)

06 systemd.exec(5) — AppArmorProfile=, NoNewPrivileges=, ProtectSystem=, Capa… (freedesktop.org — v255)

07 Docker Engine security — AppArmor, docker-default, --security-opt apparmor… (docs.docker.com — 2026)

08 Ubuntu 24.04 release notes — userns restriction, флаг unconfined; php.net… (ubuntu.com / php.net — 24.04 LTS)

09 Роль Ansible itfresh.apparmor — шаблоны профилей, CI-проверка, Zabbix (наш шаблон — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.