

Runbook: доступ к secret-файлам, смонтированным в контейнер

Происхождение: OLV-3708 (май 2026) — ключ Firebase лежал 600 root:root, процесс в контейнере под www-data его не читал, push на мобильные не уходил неделю. Закреплено в OLV-3729.

Применяется к любому секрету (service-account JSON, ключи, сертификаты), который кладут на хост и монтируют в контейнер через bind mount (-v /etc/<сервис>/...:/app/...:ro).

1. Кто читает файл

Файл читает **фактический пользователь процесса, который его использует** — не root-пользователь контейнера и не пользователь хоста, выполнившего деплой. uid/gid определяй в контейнере, не предполагай:

```
docker exec <container> ps -eo user,uid,gid,args      # кто реально запущен
docker exec <container> id <user>                  # uid/gid этого
пользователя
```

Для bind mount права проверяются по **числовым** uid/gid: имя пользователя на хосте и в контейнере может не совпадать.

Пример back-auth: push отправляет консюмер шины и FPM-воркеры под www-data (uid 33, gid 33); файлы — /etc/firebase/back-auth/<brand>.json на хосте → /app/config/firebase/ в контейнере.

2. Как выдать доступ

Доступ ограничивается необходимыми субъектами средствами ОС. Чтение «для всех» стандартом не является.

Предпочтительно — группа с gid процесса:

```
chown root:<gid процесса> /etc/<сервис>/<file>      # например root:33
chmod 640 /etc/<сервис>/<file>
chmod 750 /etc/<сервис>                             # каталог: root и та же группа
chown root:<gid процесса> /etc/<сервис>
```

Если группу менять нельзя — ACL на конкретный uid:

```
setfacl -m u:<uid процесса>:r /etc/<сервис>/<file>
setfacl -m u:<uid процесса>:x /etc/<сервис>
```

Не оставлять 600/700 root:root: процесс не под root файл не прочитает, а ошибка часто уходит только в лог.

3. Проверка после деплоя (обязательна)

```
# процесс-потребитель читает — должно быть успешно (rc=0)
docker exec -u <user> <container> test -r <путь в контейнере> && echo OK

# посторонний пользователь хоста не читает — должен быть отказ (rc≠0)
sudo -u nobody test -r <путь на хосте> && echo "ПЛОХО: читается всеми" || echo OK
```

Пример back-auth (test, srv-app):

```
docker exec -u www-data tpark-it.back-auth.latest test -r
/app/config/firebase/tpark-it.json && echo OK
sudo -u nobody test -r /etc/firebase/back-auth/tpark-it.json && echo
"ПЛОХО" || echo OK
```

После смены прав перезапусти процессы сервиса: back-auth кеширует неудачу чтения ключа на время жизни процесса.

Известное состояние: Firebase back-auth

На 01.10.2026 на srv-app (test): каталог /etc/firebase/back-auth — 711 root:root, tpark-it.json — 644 root:root. Сервис ключ читает, но **известное ограничение**: любой локальный пользователь хоста, знающий путь к файлу, может прочитать ключ (проверка «посторонний — отказ» из п. 3 не проходит). Перевод на 640 с группой gid 33 или ACL — отдельное решение владельца; массово менять права на площадках этим runbook не предписывается.

Нечитаемый ключ Firebase в back-auth проявляется ERROR «Firebase credentials file is missing or unreadable»; контейнер и вход при этом продолжают работать.

From:

<https://wiki.olvery.ru/> - Olvery

Permanent link:

<https://wiki.olvery.ru/doku.php?id=runbooks:secret-files-docker-mount>

Last update: 2026/10/01 15:14

