

СКР ГОСТ Р 57580.1 — руководство по установке

Версия 0.7.1 · 26.09.2026

Содержание

Установка СКР и подготовка хостов

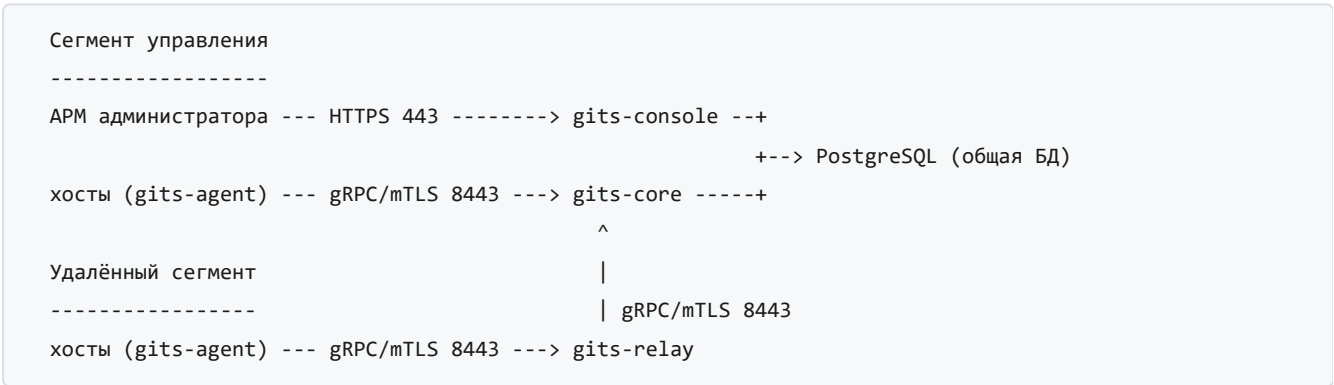
1. Требования
2. Установка сервера
3. Ретранслятор gits-relay
4. Установка агента
5. Подготовка Linux-хоста
6. Подготовка Windows-хоста
7. Диагностика

Установка СКР и подготовка хостов

Руководство администратора по развёртыванию системы контроля соответствия ГОСТ Р 57580.1 (СКР): сервер (ядро и консоль), ретрансляторы, агенты и подготовка контролируемых хостов.

№	Документ	Содержание
1	Требования	ОС, ресурсы, сетевые порты, время, учётные записи
2	Сервер	PostgreSQL, ядро и консоль, УЦ и ключи, первый администратор, резервное копирование
3	Ретранслятор	Сегментный ретранслятор <code>gits-relay</code>
4	Агент	Способы установки агента, проверка, обновление, удаление
5	Подготовка Linux-хоста	<code>auditd</code> , <code>journald</code> , Astra Linux (PARSEC), время, соответствие проверкам
6	Подготовка Windows-хоста	Политика аудита, журналы, время, GPO, WinRM
7	Диагностика	Типовые ошибки и их устранение

Состав системы



- **gits-core** — ядро: шлюз агентов и ретрансляторов, конвейер событий, правила, инциденты, фоновые задания. Хранит УЦ и ключи подписи, применяет миграции БД.
- **gits-console** — REST API и веб-консоль. Секретов ядра не хранит, работает только с БД.
- **gits-server run** — ядро и консоль в одном процессе. Подходит для небольших установок до нескольких сотен хостов.
- **gits-relay** — ретранслятор для изолированных сегментов. Буферизует события и кэширует пакеты обновлений.
- **gits-agent** — агент на контролируемом хосте. Выполняет проверки, собирает инвентарь и события, исполняет подписанные команды.
- **gitsctl** — административная утилита: УЦ, ключи, первый пользователь, токены, публикация пакетов агента.

Порядок работ

1. Проверить требования и согласовать правила межсетевого экрана (01).
2. Установить PostgreSQL и сервер, создать УЦ, ключи и первого администратора (02).
3. Опубликовать пакеты агента: `gitsctl publish-agent` или консоль, раздел «Обновления агентов».
4. Создать информационные системы (ИС) и задать адреса подключения агентов.

5. При необходимости установить ретрансляторы в удалённых сегментах (03).
6. Подготовить хосты: политика аудита, журналы, время (05, 06).
7. Установить агентов (04).
8. Убедиться, что хосты на связи, проверки выполняются, а события поступают (04 §8).

Состав дистрибутива

Файлы релиза vx.Y.Z (Gitea: «Релизы», а также реестры пакетов generic, debian и rpm):

Файл	Назначение
gits-server_<ver>_linux_amd64	ядро и консоль в одном процессе
gits-core_<ver>_linux_amd64, gits-console_<ver>_linux_amd64	ядро и консоль отдельно
gitsctl_<ver>_linux_amd64	административная утилита
gits-relay_<ver>-1_amd64.deb, gits-relay_<ver>-1.x86_64.rpm	ретранслятор (также arm64)
gits-agent_<ver>-1_amd64.deb, gits-agent_<ver>-1.x86_64.rpm	агент Linux (также arm64)
gits-agent_<ver>_linux_amd64	агент Linux: исполняемый файл для gits-agent install
gits-agent_<ver>_windows_amd64.exe	агент Windows 10 / Server 2016 и новее
gits-agent-legacy_<ver>_windows_amd64.exe	агент Windows Server 2008 R2 – 2012 R2
gits-deploy_<ver>.zip	примеры конфигурации, юниты systemd, скрипты подготовки хостов, роль Ansible, перечень мер, это руководство (Markdown и PDF)
SHA256SUMS	контрольные суммы

Сборки для Windows (*_windows_amd64.exe) есть и у серверных компонентов. Установка сервера на Windows описана кратко в 02 §13.

Перед установкой сверьте контрольные суммы:

```
sha256sum -c SHA256SUMS --ignore-missing
```

1. Требования

1.1 Сервер

Параметр	Требование
ОС	Linux x86-64 с systemd. Проверено на Astra Linux SE 1.7.6. Исполняемые файлы статические и не зависят от системных библиотек
СУБД	PostgreSQL 16 (эталонная версия, на ней идут тесты CI). Установка проверена и на PostgreSQL 11 из состава Astra Linux 1.7. Расширения не нужны
ЦП, ОЗУ, диск	Для пилота на 50–100 хостов: 4 vCPU, 8 ГБ ОЗУ, 100 ГБ под БД; сервер и СУБД на одном узле. Ориентир по design/02 §6 для 1 000 хостов: 15–20 млн событий в сутки, БД 200–500 ГБ. Объем зависит от настройки аудита и уточняется по итогам пилота
Резервные копии	отдельный том или хранилище (§ 02 §11)
Размещение	сегмент управления, доступ из пользовательских сетей закрыт (design/02 §4)

Сервер можно запустить и на Windows Server: сборки *_windows_amd64.exe есть. Режимы службы Windows у сервера нет, эталонная установка — Linux (02 §13).

1.2 Контролируемые хосты

ОС	Агент	Примечание
Windows 10 / Server 2016 и новее (x64)	gits-agent_<ver>_windows_amd64.exe	служба gits-agent, LocalSystem; проверено на Windows 10 LTSC 2021
Windows Server 2008 R2 – 2012 R2	gits-agent-legacy_<ver>_windows_amd64.exe	сборка на Go 1.20
Astra Linux SE 1.7+	deb или исполняемый файл	проверено на 1.7.6; дополнительный источник событий — журнал PARSEC
Другие Linux с glibc и systemd	deb, rpm или исполняемый файл	amd64 и arm64

Агент работает от **root** (Linux) или **LocalSystem** (Windows). Без этих прав проверки secedit и auditpol, а также чтение журнала Security завершаются ошибкой.

Целевое потребление ресурсов (design/05 §7): в среднем не больше 1 % ЦП, не больше 150 МБ ОЗУ на APM и 300 МБ на серверах. Агент работает с пониженным приоритетом процесса и ввода-вывода. Дискový буфер событий занимает до 512 МБ: /var/lib/gits-agent/buffer или %ProgramData%\GITS\agent\buffer.

Каталоги агента нужно добавить в исключения антивируса: исполняемый файл и каталог состояния с буфером (пути — в 04 §1). Агент не делает этого сам.

Требования к настройке ОС хоста (аудит, журналы, время) — в 05 и 06.

1.3 Сетевые взаимодействия

Все соединения агента **исходящие**: входящих портов на хостах агент не открывает.

Откуда	Куда	Порт	Протокол	Назначение
Хосты (агенты)	ядро или ретранслятор	TCP 8443	gRPC, TLS 1.2+, mTLS	регистрация, конфигурация, результаты, события, команды
Ретранслятор	ядро	TCP 8443	gRPC, mTLS	трафик агентов сегмента
АРМ администраторов	консоль	TCP 443	HTTPS	веб-консоль и REST API
Ядро, консоль	PostgreSQL	TCP 5432	PostgreSQL (TLS по возможности)	общая БД
Сетевое оборудование, СКУД и др.	ядро	UDP 5514	syslog	необязательно: приём внешних событий
Ядро	SIEM	TCP/UDP 514	CEF / JSON	необязательно: пересылка событий
Ядро	каталог (AD, ALD Pro)	TCP 636	LDAPS	необязательно: сверка учётных записей
Ядро	SMTP	TCP 25/587	SMTP	необязательно: уведомления
Ядро или ретранслятор	хосты	TCP 22 / 5985 / 5986	SSH / WinRM	только на время push-установки агента

Порт шлюза задаётся параметром `gateway.listen`. По умолчанию используется 8443. Можно назначить 443, если политика сети разрешает только его. Адрес, по которому агенты обращаются к ядру или ретранслятору, должен присутствовать в сертификате шлюза как имя DNS или IP-адрес (02 §5).

1.4 Время

Часы сервера, ретрансляторов и хостов должны синхронизироваться с доверенным источником: в домене — с контроллером домена, вне домена — с внутренними серверами NTP.

- Сертификаты выдаются с началом действия «сейчас – 5 минут». Хост, часы которого отстают больше чем на 5 минут, не примет сертификат («certificate not yet valid»).
- Команды с сервера отклоняются, если время выдачи опережает часы агента больше чем на 5 минут («command issued in the future»).
- Проверка CHK-MAS-009a (Windows) требует, чтобы расхождение часов агента и сервера было не больше 5 секунд.

1.5 Учётные записи

Учётная запись	Где	Назначение
<code>gits</code> (системная, без входа)	сервер	служба ядра и консоли, владелец <code>/var/lib/gits</code>
роль <code>gits</code> в PostgreSQL	СУБД	владелец схемы СКР
<code>gits-relay</code> (системная)	ретранслятор	создаётся пакетом
администратор консоли (<code>system_admin</code>)	СКР	первый пользователь, создаётся <code>gitsctl create-user</code>
второй администратор	СКР	утверждение раскатки обновлений и опасных действий («две руки»)
сервисная учётная запись LDAP (только чтение)	каталог	сверка учётных записей, если используется

Учётная запись	Где	Назначение
учётная запись для push-установки	хосты	SSH с sudo или локальный администратор Windows; вводится на одно задание и затем стирается

Пароли пользователей консоли должны быть не короче 16 символов.

2. Установка сервера

Эталонная установка — Linux с systemd, ядро и консоль в одном процессе (`gits-server run`). Раздельная установка ядра и консоли описана в §9. Все команды выполняются от `root`, если не указано иное.

Раскладка файлов:

Путь	Владелец, права	Содержимое
<code>/usr/local/bin/gits-server</code> , <code>gits-console</code> , <code>gitsctl</code>	<code>root</code> , 0755	исполняемые файлы
<code>/etc/gits/server.yaml</code>	<code>root:gits</code> , 0640	конфигурация
<code>/etc/gits/gits.env</code>	<code>root</code> , 0600	необязательно: <code>GITS_DB_DSN</code> с паролем БД
<code>/etc/gits/console.crt</code> , <code>console.key</code>	<code>root:gits</code> , 0644 / 0640	сертификат HTTPS консоли
<code>/var/lib/gits/ca/</code>	<code>gits</code> , 0700	УЦ СКР: <code>ca.crt</code> , <code>ca.key</code> (0600)
<code>/var/lib/gits/keys/</code>	<code>gits</code> , 0700	ключи подписи: <code>commands.key</code> , <code>config.key</code> , <code>update.key</code> , <code>deploy.key</code> (0600)
<code>/var/lib/gits/server/</code>	<code>gits</code> , 0700	сертификат и ключ шлюза агентов

Ядро отказывается запускаться, если файлы ключей доступны группе или остальным пользователям.

1. Подготовка узла

```
# Системная учётная запись и каталоги
useradd --system --home-dir /var/lib/gits --shell /usr/sbin/nologin gits
install -d -m 0750 -o root -g gits /etc/gits
install -d -m 0700 -o gits -g gits /var/lib/gits

# Исполняемые файлы из релиза
install -m 0755 gits-server_<ver>_linux_amd64 /usr/local/bin/gits-server
install -m 0755 gitsctl_<ver>_linux_amd64 /usr/local/bin/gitsctl
gits-server version
```

Синхронизация времени обязательна (01 §1.4). Проверьте её командой `timedatectl` (`System clock synchronized: yes`).

2. PostgreSQL

Нужны база данных и роль-владелец. Расширения не требуются, схему создают миграции СКР.

```
# Debian/Astra: apt install postgresql; RHEL: dnf install postgresql-server && postgresql-setup --initdb
sudo -u postgres psql -v ON_ERROR_STOP=1 -c "CREATE ROLE gits LOGIN PASSWORD 'длинный-случайный-пароль'"
sudo -u postgres createdb -O gits gits
```

Astra Linux SE с мандатным контролем доступа. PostgreSQL из состава Astra отклоняет подключение пользователя без мандатной метки (`error obtaining MAC configuration for user "gits"`). Роли БД `gits`

должна соответствовать учётная запись ОС с меткой:

```
pdpl-user -l 0:0 gits
```

PostgreSQL на отдельном узле. Разрешите подключение в `pg_hba.conf` только с адресов узлов СКР (`hostssl gits gits 10.0.0.10/32 scram-sha-256`) и укажите в DSN `sslmode=require` или `verify-full`.

3. Конфигурация

```
install -m 0640 -o root -g gits server.yaml.example /etc/gits/server.yaml
```

Пример с комментариями ко всем параметрам — `deploy/server.yaml.example` в `gits-deploy_<ver>.zip`. Обязательно задайте:

Параметр	Значение
<code>db_dsn</code>	<code>postgres://gits:<пароль>@127.0.0.1:5432/gits?sslmode=disable</code> (или <code>GITS_DB_DSN</code> в <code>/etc/gits/gits.env</code>)
<code>data_dir</code>	<code>/var/lib/gits</code>
<code>gateway.listen</code>	<code>0.0.0.0:8443</code> . По умолчанию <code>127.0.0.1:8443</code> , и тогда удалённые агенты не подключатся
<code>gateway.agent_endpoints</code>	адреса, по которым агенты подключаются к ядру или ретрансляторам, например <code>gits.bank.local:8443</code>
<code>api.listen</code> , <code>api.tls_cert</code> , <code>api.tls_key</code>	<code>0.0.0.0:443</code> и сертификат консоли (§6)

Пароль БД можно не хранить в `server.yaml`:

```
install -m 0600 /dev/null /etc/gits/gits.env
echo 'GITS_DB_DSN=postgres://gits:<пароль>@127.0.0.1:5432/gits?sslmode=disable' > /etc/gits/gits.env
```

Утилита `gitsctl` работает с теми же файлами и должна запускаться **от имени gits**, иначе ключи получат чужого владельца. Ниже используется сокращение:

```
gctl() { sudo -u gits env GITS_CONFIG=/etc/gits/server.yaml $(cat /etc/gits/gits.env 2>/dev/null) gitsctl "$@"; }
```

4. Схема БД

```
sudo -u gits env GITS_CONFIG=/etc/gits/server.yaml gits-server migrate
# migrations applied
```

Ядро (`gits-server run` / `gits-core`) также применяет миграции при каждом запуске. Консоль (`gits-console`) миграции не применяет и ждёт появления схемы до 2 минут.

5. УЦ, сертификат шлюза и ключи

```
gctl init-ca -name "GITS Root CA <организация>"
gctl issue-server-cert -dns gits.bank.local,gits -ip 10.0.0.10
gctl gen-signing-keys
```

- `init-ca` создаёт корневой сертификат СКР на 10 лет (`/var/lib/gits/ca/ca.crt`) и его ключ. Этим УЦ подписываются сертификаты шлюза (2 года), агентов и ретрансляторов (1 год). Системное хранилище доверенных сертификатов в этой цепочке не используется: агенты доверяют только корню СКР.
- `issue-server-cert` — сертификат шлюза агентов. В `-dns` и `-ip` перечислите **все** имена и адреса, по которым агенты обращаются к ядру, включая адрес из `gateway.agent_endpoints`. По умолчанию подставляются `localhost` и `127.0.0.1`, и удалённые агенты такой сертификат не примут. Если набор адресов изменился, выпустите сертификат заново и перезапустите сервер.
- `gen-signing-keys` создаёт ключи Ed25519 для подписи команд, конфигураций и пакетов обновлений. Ключ `deploy.key` для push-установки ядро создаст при первом запуске.

Проверка:

```
ls -l /var/lib/gits/ca /var/lib/gits/keys /var/lib/gits/server # ключи — 0600, владелец gits
openssl x509 -in /var/lib/gits/server/server.crt -noout -ext subjectAltName
```

6. Сертификат консоли

Консоль открывается в браузере, поэтому её сертификат выпускает **корпоративный УЦ**, которому доверяют APM администраторов, а не УЦ СКР. Имя в сертификате — адрес консоли (`gits.bank.local`).

```
install -m 0644 -o root -g gits gits-console.crt /etc/gits/console.crt # сертификат + промежуточные
install -m 0640 -o root -g gits gits-console.key /etc/gits/console.key
```

Когда задан `api.tls_cert`, cookie сессии получает флаг `Secure`. Обратный прокси перед консолью возможен (`api.listen: 127.0.0.1:8080`, `secure_cookie: true`), но тогда в журнале действий администраторов вместо адреса клиента будет записан адрес прокси: заголовок `X-Forwarded-For` не используется.

7. Первый администратор и перечень мер

```
GITS_PASSWORD='не короче 16 символов' gctl create-user -login admin -roles system_admin
# user created: admin (...) roles=[system_admin]

install -m 0644 tech_measures_level1.csv /tmp/measures.csv # файл должен быть доступен пользователю git
s
gctl load-measures -csv /tmp/measures.csv
# 276 measures loaded
```

Роли: `system_admin`, `checks_admin`, `analyst`, `grizi_lead`, `grizi_dispatcher`, `grizi_secretary`, `auditor`.
Администратор системы не видит событий и свидетельств (разделение полномочий), для этого нужны

отдельные пользователи с ролями `auditor` или `analyst`. Раскатку обновлений агентов утверждает второй пользователь с ролью `system_admin`.

8. Запуск службы

```
install -m 0644 deploy/systemd/gits-server.service /etc/systemd/system/  
systemctl daemon-reload  
systemctl enable --now gits-server  
journalctl -u gits-server -n 20 -o cat
```

В журнале должны появиться строки:

```
level=INFO msg="api listening" addr=0.0.0.0:443 tls=true  
level=INFO msg="gateway listening" component=gateway addr=[::]:8443
```

Юнит запускает сервер от `gits` с `ProtectSystem=strict`. Запись разрешена только в `/var/lib/gits`, а привязка к порту 443 без `root` выполняется через `CAP_NET_BIND_SERVICE`.

Проверка консоли: откройте `https://gits.bank.local/` и войдите как `admin`. Проверка из командной строки:

```
curl -s -o /dev/null -w '%{http_code}\n' https://gits.bank.local/ # 200
```

Межсетевой экран узла: откройте входящие TCP 443 (из сегмента администраторов) и TCP 8443 (от хостов и ретрансляторов); при `syslog.enabled` — UDP 5514.

9. Раздельная установка ядра и консоли

Используется, когда консоль вынесена на отдельный узел или ядер несколько.

Узел	Файлы	Юнит
ядро	<code>gits-core</code> , <code>gitsctl</code> , <code>/etc/gits/server.yaml</code> , весь <code>/var/lib/gits</code>	<code>deploy/systemd/gits-core.service</code>
консоль	<code>gits-console</code> , <code>/etc/gits/server.yaml</code> (секции <code>db_dsn</code> и <code>api</code>), сертификат консоли	<code>deploy/systemd/gits-console.service</code>

- Ядро и консоль связаны только через БД: задания передаются через таблицу `core_jobs`, сигналы — через `LISTEN/NOTIFY PostgreSQL`.
- Ключей и УЦ на узле консоли нет. Команды `gitsctl`, которым нужны ключи (`init-ca`, `issue-server-cert`, `gen-signing-keys`, `publish-agent`), выполняются на узле ядра.
- Несколько ядер: у всех одинаковые `/var/lib/gits` (УЦ и ключи) и одна БД. Фоновые задания выполняет ядро-лидер, выбранное через `advisory lock`. Агенты получают адреса всех ядер в `gateway.agent_endpoints`. Пороговые правила считаются отдельно в каждом ядре.
- Не включайте `gits-server.service` вместе с `gits-core` / `gits-console` на одном узле.

10. Первичная настройка после запуска

1. **Пакеты агента.** Установочные комплекты консоли и обновления собираются только из подписанных пакетов репозитория СКР. Опубликуйте нужные платформы на узле ядра:

```
bash gctl publish-agent -file gits-agent_<ver>_windows_amd64.exe -version <ver> -os windows -arch amd64 -format exe gctl publish-agent -file gits-agent_<ver>_linux_amd64 -version <ver> -os linux -arch amd64 -format bin gctl publish-agent -file gits-agent_<ver>-1_amd64.deb -version <ver> -os linux -arch amd64 -format deb gctl publish-agent -file gits-agent-<ver>-1.x86_64.rpm -version <ver> -os linux -arch amd64 -format rpm
```

Форматы `exe` и `bin` нужны для установочных комплектов и обновления агентов, установленных командой `gits-agent install`. Форматы `deb` и `rpm` нужны для обновления агентов, установленных из пакетов.

Загрузить пакет можно и в консоли, раздел «Обновления агентов». 2. **Информационные системы.** В разделе «Информационные системы» создайте ИС и укажите для каждой «Адреса подключения агентов» (ядро или ретранслятор сегмента). Выдайте пользователям доступ к ИС. То же через CLI: `gctl create-is -name ...`, `gctl grant-is -login ... -is ...`. 3. **Уведомления, LDAP, syslog, SIEM** — секции `notify`, `acc`, `syslog`, `siem` в `server.yaml` и перезапуск службы.

11. Резервное копирование и восстановление

Что	Как	Частота
БД	<code>pg_dump -Fc gits > gits-\$(date +%F).dump</code> или непрерывная архивация WAL (RPO 15 мин, design/02 §7)	ежедневно / непрерывно
УЦ и ключи	<code>tar -C /var/lib/gits -czf gits-keys.tgz ca key s server</code>	после <code>init-ca</code> , <code>gen-signing-keys</code> , <code>issue -server-cert</code> и появления <code>deploy.key</code>
Конфигурация	<code>/etc/gits</code>	после изменений

Архив с УЦ и ключами храните отдельно от копий БД, зашифрованным и с ограниченным доступом. Без него восстановленная из копии система не сможет продлевать сертификаты агентов и подписывать им команды, конфигурацию и обновления. Агенты доверяют ключам, которые получили при регистрации.

Восстановление: развернуть узел по §1–3, восстановить `/etc/gits` и `/var/lib/gits` с сохранением владельца `gits` и прав 0600 на ключи, создать пустую БД (§2), затем `pg_restore -d gits gits-<дата>.dump` и запустить службу.

12. Обновление сервера

1. Прочитайте раздел версии в `CHANGELOG.md`: там перечислены миграции и особые шаги.
2. Сделайте резервную копию БД и `/var/lib/gits` (§11).
3. Замените исполняемые файлы и перезапустите службу: `systemctl restart gits-server`. Миграции применяются при старте. При отдельной установке сначала обновите и запустите ядро, затем консоль.
4. Опубликуйте пакеты агента новой версии (§10) и запустите раскатку в разделе «Обновления агентов».

13. Сервер на Windows

Сборки `gits-server_<ver>_windows_amd64.exe` и `gitsctl_<ver>_windows_amd64.exe` работают с теми же `server.yaml` и командами. Режим службы Windows у сервера нет: запуск при загрузке настраивается планировщиком заданий или внешней обёрткой службы, от имени выделенной учётной записи. Проверка прав на файлы ключей на Windows не выполняется, поэтому ограничьте доступ к `data_dir` списками доступа (SYSTEM, Администраторы, учётная запись службы). Эталонной и проверенной остаётся установка на Linux.

3. Ретранслятор gits-relay

Ретранслятор ставится в удалённом или изолированном сегменте. Агенты сегмента подключаются к нему, а он к ядру. На межсетевом экране достаточно одного правила на сегмент: «ретранслятор → ядро:8443».

Что делает ретранслятор:

- проверяет сертификаты агентов (mTLS) и передаёт ядру их идентичность;
- буферизует события на диске, пока ядро недоступно (`spool_max_mb` , по умолчанию 2 ГБ);
- кэширует пакеты обновлений агентов;
- выполняет push-установку агентов на хосты своего сегмента.

Ретранслятору не нужны права root. Секретов ядра он не хранит: подписи команд и конфигураций проверяют сами агенты.

1. Установка пакета

```
# Debian / Astra
apt install ./gits-relay_<ver>-1_amd64.deb
# RHEL и совместимые
dnf install ./gits-relay-<ver>-1.x86_64.rpm
```

Пакет создаёт системного пользователя `gits-relay` и раскладывает файлы:

Путь	Содержимое
<code>/usr/bin/gits-relay</code>	исполняемый файл
<code>/etc/gits-relay/relay.yaml.example</code>	пример конфигурации
<code>/var/lib/gits-relay</code> (0700)	сертификат, ключ, буфер событий, кэш пакетов
<code>/lib/systemd/system/gits-relay.service</code>	юнит: <code>User=gits-relay</code> , <code>ProtectSystem=strict</code> , <code>CAP_NET_BIND_SERVICE</code>

Пока ретранслятор не зарегистрирован, служба не включается.

2. Конфигурация

```
cp /etc/gits-relay/relay.yaml.example /etc/gits-relay/relay.yaml
cp ca.crt /etc/gits-relay/ca.pem # /var/lib/gits/ca/ca.crt с узла ядра
chown root:gits-relay /etc/gits-relay/relay.yaml /etc/gits-relay/ca.pem
chmod 0640 /etc/gits-relay/relay.yaml /etc/gits-relay/ca.pem
```

Параметр	Значение
<code>name</code>	имя в консоли (по умолчанию — имя хоста)
<code>listen</code>	адрес для агентов, по умолчанию <code>:8443</code>
<code>core</code>	адреса шлюза ядра по порядку предпочтения: <code>["gits.bank.local:8443"]</code>
<code>core_server_name</code>	имя в сертификате ядра, если оно отличается от адреса в <code>core</code>
<code>ca_file</code>	корневой сертификат СКР

Параметр	Значение
<code>state_dir</code>	<code>/var/lib/gits-relay</code>
<code>dns_names</code> , <code>ip_addresses</code>	имена и адреса, по которым к ретранслятору обращаются агенты. Попадают в сертификат ретранслятора; нужно хотя бы одно значение
<code>spool_max_mb</code>	размер буфера событий, МБ

3. Регистрация

1. Выпустите токен вида `relay`: в консоли «Агенты и токены» → «Выдать токен регистрации» (вид «ретранслятор») или на узле ядра:

```
bash gctl create-token -kind relay -name relay-seg1 -uses 1 -ttl 24h
```

2. Зарегистрируйте ретранслятор и запустите службу:

```
bash sudo -u gits-relay gits-relay enroll -config /etc/gits-relay/relay.yaml -token '<токен>'
systemctl enable --now gits-relay journalctl -u gits-relay -n 20 -o cat
```

При регистрации ретранслятор создаёт ключ и получает от ядра сертификат на 1 год с именами из `dns_names` / `ip_addresses`. Сертификат и идентификатор сохраняются в `state_dir`.

3. Ретранслятор появится в консоли, раздел «Ретрансляторы», в состоянии «на связи».

4. Перевод агентов сегмента на ретранслятор

В карточке ИС («Информационные системы») укажите в «Адресах подключения агентов» адрес ретранслятора, например `relay-seg1.bank.local:8443`. Агенты хостов этой ИС получают новый список адресов в подписанной конфигурации и переподключаются сами. Для новых агентов этот адрес попадёт в установочные комплекты.

Межсетевой экран: хосты сегмента → ретранслятор TCP 8443; ретранслятор → ядро TCP 8443.

5. Ограничения

- Сертификат ретранслятора действует 1 год и автоматически **не продлевается**. До истечения срока перерегистрируйте ретранслятор новым токеном: остановите службу, выполните `enroll` (§3) и запустите службу. Если после этого в разделе «Ретрансляторы» осталась прежняя запись без связи, отзовите её. Срок действия сертификата показан там же, в поле «Сертификат до».
- На Windows ретранслятор ставится как служба: `gits-relay.exe service install -config relay.yaml`, затем `service start`.

4. Установка агента

Перед установкой подготовьте хост (05 — Linux, 06 — Windows). Агент заработает и без подготовки, но часть проверок покажет несоответствие, а часть событий не будет собираться.

Способы установки:

Способ	Когда использовать
Установочный комплект из консоли	основной: вручную, через GPO/SCCM, Ansible, ALD Pro
<code>gits-agent install</code> вручную	единичные хосты, свои сценарии развёртывания
Пакет deb/rpm	Linux-хосты, управляемые через репозитории пакетов
Роль Ansible	массовая установка на Linux
Push-установка из консоли	нет своих средств развёртывания; SSH или WinRM до хостов

Все способы идемпотентны: повторный запуск не перерегистрирует уже зарегистрированный агент.

1. Раскладка файлов

	Linux (<code>gits-agent install</code> , комплект, Ansible)	Linux (пакет deb/rpm)	Windows
Исполняемый файл	<code>/usr/local/bin/gits-agent</code>	<code>/usr/bin/gits-agent</code>	<code>%ProgramFiles%\GITS Agent\gits-agent.exe</code>
Конфигурация	<code>/etc/gits-agent/agent.yaml</code> (0600)	<code>/etc/gits-agent/agent.yaml</code>	<code>%ProgramFiles%\GITS Agent\agent.yaml</code>
Состояние, буфер	<code>/var/lib/gits-agent</code> (0700)	<code>/var/lib/gits-agent</code>	<code>%ProgramData%\GITS\agent</code>
Служба	<code>gits-agent.service</code> в <code>/etc/systemd/system</code>	<code>gits-agent.service</code> в <code>/lib/systemd/system</code>	служба <code>gits-agent</code> , LocalSystem, автозапуск

На Windows доступ к каталогам агента есть только у SYSTEM и группы «Администраторы» (наследование отключено). В каталоге состояния хранятся сертификат и ключ агента, кэш подписанной конфигурации, буфер событий и журнал выполненных команд.

2. Перед установкой

1. **Пакет агента опубликован** на сервере (02 §10). Иначе установочный комплект не соберётся.
2. **Адреса подключения** заданы в карточке ИС или в `gateway.agent_endpoints`. Адреса должны быть в сертификате шлюза или ретранслятора.
3. **Токен регистрации.** Токен ограничен по сроку действия (от 1 до 72 часов) и по числу установок. Выдаётся в консоли («Агенты и токены» → «Выдать токен регистрации») или командой `gctl create-token -uses 50 -ttl 72h`. Токен — секрет: с ним любой хост может зарегистрироваться в СКР.
4. **Доступ к серверу:** хост → ядро или ретранслятор по TCP 8443, часы синхронизированы.

3. Установочный комплект из консоли

Консоль: «Агенты и токены» → «Установочный пакет агента». Укажите ОС, архитектуру, ИС (хост сразу попадёт в неё), адреса подключения (если не заданы в ИС), число установок и срок токена. Консоль выпустит токен и соберёт архив:

ОС	Архив	Состав
Linux	<code>gits-agent-<ver>-linux-<arch>.tar.gz</code>	<code>gits-agent</code> , <code>agent.yaml</code> , <code>install.sh</code> , <code>README.txt</code>
Windows	<code>gits-agent-<ver>-windows-<arch>.zip</code>	<code>gits-agent.exe</code> , <code>agent.yaml</code> , <code>install.cmd</code> , <code>README.txt</code>

В `agent.yaml` уже записаны адреса подключения, корневой сертификат СКР и токен. **Храните архив как секрет** и удалите его после истечения срока токена.

Linux:

```
tar xzf gits-agent-<ver>-linux-amd64.tar.gz
sudo ./gits-agent-install/install.sh
```

Windows (от имени администратора):

```
install.cmd
```

GPO. Распакуйте архив в сетевую папку, доступную на чтение учётным записям компьютеров, и назначьте `install.cmd` сценарием автозагрузки компьютера: «Конфигурация компьютера» → «Политики» → «Конфигурация Windows» → «Сценарии» → «Автозагрузка». Число установок в токене должно быть не меньше числа компьютеров, а срок токена — достаточным для перезагрузки всех машин. После регистрации последующие запуски ничего не меняют, а при замене исполняемого файла служба перезапускается.

Результат установки выводится в JSON:

```
{ "enrolled": true, "binary_new": true, "service_new": true, "host_id": "01a0...", "service": "gits-agent.s
ervice" }
```

4. Вручную: gits-agent install

Нужны исполняемый файл агента из релиза, корневой сертификат СКР (`/var/lib/gits/ca/ca.crt` с узла ядра) и токен.

`agent.yaml`:

```
servers:                # ядро и/или ретрансляторы; выбирается случайный, при сбое — следующий
  - gits.bank.local:8443
ca_file: /root/gits-install/ca.pem  # или ca.pem: | ... (PEM прямо в файле)
log_level: info
# server_name: gits.bank.local # если имя в сертификате отличается от адреса
```

```
sudo ./gits-agent install -config agent.yaml -token '<токен>'
```

Команда копирует агента в штатный каталог (`$1`) и пишет итоговый `agent.yaml` с корневым сертификатом внутри. Токен в файл не записывается. Затем команда регистрирует агента, создаёт и запускает службу. На Windows команда выполняется так же, из командной строки администратора: `gits-agent.exe install -config agent.yaml -token <токен>`.

5. Пакет deb/rpm

```
apt install ./gits-agent_<ver>-1_amd64.deb          # или dnf install ./gits-agent-<ver>-1.x86_64.rpm
cp ca.crt /etc/gits-agent/ca.pem
cat > /etc/gits-agent/agent.yaml <<'EOF'
servers: ["gits.bank.local:8443"]
ca_file: /etc/gits-agent/ca.pem
state_dir: /var/lib/gits-agent
log_level: info
EOF
chmod 0600 /etc/gits-agent/agent.yaml
gits-agent enroll -config /etc/gits-agent/agent.yaml -token '<токен>'
systemctl enable --now gits-agent
```

Пакет включает службу автоматически, только когда агент уже зарегистрирован. Иначе после установки он печатает эти шаги. Подключить репозитории пакетов Gitea: README, раздел «CI и релизы».

6. Ansible

Роль `deploy/ansible/roles/gits-agent` (в `gits-deploy_<ver>.zip`) устанавливает агента на Linux-хосты.

```
# из установочного комплекта консоли
ansible-playbook -i inventory -b deploy/ansible/install-agent.yml \
  -e gits_agent_bundle=files/gits-agent-<ver>-linux-amd64.tar.gz \
  -e gits_agent_prepare_host=true
```

Переменная	Назначение
<code>gits_agent_bundle</code>	архив из консоли (вариант 1)
<code>gits_agent_binary</code> , <code>gits_agent_endpoints</code> , <code>gits_agent_ca_pem</code> , <code>gits_agent_enroll_token</code>	отдельные параметры (вариант 2); токен — через <code>ansible-vault</code>
<code>gits_agent_prepare_host</code>	<code>true</code> — перед установкой выполнить <code>prepare-host.sh (05)</code>
<code>gits_agent_prepare_host_args</code>	параметры <code>prepare-host.sh</code> , например <code>--astra-quiet --szi /etc/kes1</code>
<code>gits_agent_prepare_host_strict</code>	<code>true</code> — остановить установку, если после настройки остались несоответствия

Рабочий каталог с токеном удаляется в конце роли.

7. Push-установка из консоли

Консоль: «Агенты и токены» → «Push-установка агента». Укажите ОС, исполнителя (ядро или ретранслятор сегмента), ИС, адреса хостов, учётную запись, пароль и порт. Исполнитель подключается к хостам, копирует установочный комплект и запускает установку. Одновременно обрабатываются 4 хоста. Задание, которое выполняется дольше 30 минут, завершается с ошибкой. Учётные данные шифруются ключом исполнителя и стираются после завершения задания.

Требования к целевым хостам:

	Linux (SSH)	Windows (WinRM)
Порт	TCP 22 от исполнителя	TCP 5985 (HTTP, NTLM с шифрованием сообщений) или 5986 (HTTPS, Basic)
Учётная запись	root или пользователь с sudo на любые команды; вход по паролю или ключу	локальный или доменный администратор
Настройка	для sudo не должен требоваться tty (Defaults requiretty выключен); SFTP не нужен	WinRM включён: prepare-host.ps1 -EnableWinRM или Enable-PSRemoting (06 §6)

Ключ SSH-хоста не проверяется, а только записывается в журнал задания. Сертификат WinRM на порту 5986 тоже не проверяется. Поэтому запускайте push-установку только в доверенном сегменте. Push-установка по WinRM на стенде разработчика вживую не подтверждена: стендовые Windows-хосты отклоняли NTLM. Сначала проверьте её на пилотной группе.

8. Проверка после установки

На хосте:

```
systemctl status gits-agent
journalctl -u gits-agent -n 30 -o cat      # нет ошибок TLS и подключения
```

```
sc query gits-agent
```

В консоли:

1. «Хосты»: хост появился со статусом online, ОС и инвентарь заполнены.
2. Карточка хоста → «Проверки»: результаты появляются в течение минуты. Результат `error` обычно означает, что агенту не хватает прав или на хосте нет нужной утилиты.
3. «События» (пользователь с ролью `auditor` или `analyst`): от хоста приходят события, например вход по SSH или RDP.
4. Проверка CHK-MAS-021 («Исправность конвейера сбора событий») — pass.

9. Обновление

Агенты обновляются централизованно: пакеты публикуются (`gctl publish-agent` или «Обновления агентов»), затем раскатка идёт волнами. Раскатку утверждает второй администратор. Если новая версия не вышла на связь за 2 минуты, агент сам откатывается. Агент, установленный из deb/rpm, обновляется пакетом того же формата, остальные — исполняемым файлом (`bin / exe`).

Повторный запуск установочного комплекта с новым исполняемым файлом тоже обновляет агента: файл заменяется, служба перезапускается.

10. Удаление

Сначала отзовите агента в консоли («Агенты и токены» → «Агенты и сертификаты»), затем удалите его с хоста.

Linux (установка `gits-agent install`):

```
systemctl disable --now gits-agent  
rm -f /etc/systemd/system/gits-agent.service /usr/local/bin/gits-agent && systemctl daemon-reload  
rm -rf /etc/gits-agent /var/lib/gits-agent
```

Linux (пакет): `apt remove gits-agent / dnf remove gits-agent`, затем `rm -rf /var/lib/gits-agent`.

Windows (от имени администратора):

```
"%ProgramFiles%\GITS Agent\gits-agent.exe" service stop  
"%ProgramFiles%\GITS Agent\gits-agent.exe" service uninstall  
rmdir /s /q "%ProgramFiles%\GITS Agent" "%ProgramData%\GITS\agent"
```

11. Windows Server 2008 R2 – 2012 R2

Используйте `gits-agent-legacy_<ver>_windows_amd64.exe`: переименуйте его в `gits-agent.exe` и установите способом из §4 или установочным комплектом (в комплекте замените `gits-agent.exe`). Каталог проверок рассчитан на Windows 10 / Server 2016. Проверки компонентов, которых нет в старых ОС (например, Microsoft Defender), на таких хостах завершатся с результатом `error` или `fail`: оценивайте их вручную.

5. Подготовка Linux-хоста

Агент **ничего не меняет в настройках аудита хоста**. Он читает то, что пишет система, и проверяет настройки. Чтобы события собирались полностью, а проверки отражали реальное состояние, хост готовят заранее: вручную по этому разделу, скриптом `prepare-host.sh` или ролью Ansible с `gits_agent_prepare_host=true`.

1. Что читает агент

Источник	Откуда	Что нужно на хосте
auditd	<code>/var/log/audit/audit.log</code>	auditd включён, <code>log_format = RAW</code> . Записи PAM и shadow-utils (входы, операции с учётными записями) пишутся без дополнительных правил. SYSCALL-записи агент передаёт только с ключами <code>exec</code> , <code>identity</code> , <code>sudoers</code> , <code>sshd</code> , <code>pam</code> , <code>audit_rules</code> , <code>logs</code> , <code>szi</code>
journal d	<code>journalctl -o json -f</code>	фильтр по <code>sshd</code> , <code>sudo</code> , <code>su</code> , <code>login</code> , <code>systemd</code> , ядру. Постоянное хранение журнала, иначе после перезагрузки теряется позиция чтения
astra	<code>/parsec/log/astra/events</code> (только Astra Linux)	syslog-ng с модулем <code>syslog-ng-mod-astra</code> , включённые события PARSEC (<code>astra-admin-events</code>)

2. Быстрая подготовка: prepare-host.sh

Скрипт находится в `deploy/host-prep/linux/` внутри `gits-deploy_<ver>.zip`. Копируйте каталог целиком: рядом со скриптом лежат правила аудита.

```
sudo ./prepare-host.sh --check      # только отчёт; код возврата 1 – есть несоответствия
sudo ./prepare-host.sh              # применить
```

Параметр	Действие
<code>--check</code>	ничего не менять
<code>--no-packages</code>	не устанавливать auditd, если его нет
<code>--no-noise</code>	не ставить правила подавления шума
<code>--astra-quiet</code>	Astra: выключить шумные события PARSEC (запуск процессов, сеть, открытие файлов)
<code>--szi <каталог></code>	контролировать изменения конфигурации СЗИ (ключ <code>szi</code>); можно указать несколько раз

Что делает скрипт (подробности — §3–§7):

- Устанавливает auditd, если его нет.
- В `auditd.conf` задаёт `log_format = RAW`, `max_log_file ≥ 32`, `num_logs ≥ 10`, `max_log_file_action = ROTATE`. Исходный файл сохраняется как `auditd.conf.gits.bak`.
- Кладёт правила `30-gits.rules` и `31-gits-noise.rules` в `/etc/audit/rules.d/` и загружает их (`augenrules --load`).
- Включает и запускает auditd, выставляет права 0750 root на `/var/log/audit`.
- Включает постоянное хранение журнала journald.
- На Astra включает обязательные события PARSEC.

7. Проверяет, что работает служба времени.

Скрипт идемпотентен: повторный запуск ничего не меняет. Отчёт:

```
== Правила аудита (/etc/audit/rules.d)
[ FIX] 30-gits.rules установлен
[ FIX] правила загружены (augenrules --load)
[ OK ] загружено: -w /etc/passwd -p wa
...
== Синхронизация времени
[ FAIL] нет активной службы времени (chrony/ntp/systemd-timesyncd)

Изменений: 9. Несовпадений: 1.
```

Синхронизацию времени скрипт только проверяет: источники времени зависят от сети организации (§7).

3. auditd

Установка: `apt install auditd` (Debian, Astra) или `dnf install audit` (RHEL-совместимые).

`/etc/audit/auditd.conf`, значимые параметры:

```
log_format = RAW                # агент разбирает RAW; ENRICHED не использовать
max_log_file = 32                # МБ; проверка CHK-MAS-013b требует не меньше 8
num_logs = 10                   # CHK-MAS-013b: не меньше 5
max_log_file_action = ROTATE
space_left = 500
space_left_action = SYSLOG
admin_space_left = 100
admin_space_left_action = SUSPEND
disk_full_action = SUSPEND
disk_error_action = SUSPEND
```

auditd запрещает `systemctl restart`. Чтобы применить `auditd.conf`, выполните `service auditd reload`.

Каталог журналов: `chown root /var/log/audit && chmod 0750 /var/log/audit` (CHK-MAS-014b).

4. Правила аудита

`/etc/audit/rules.d/30-gits.rules` — обязательные правила (CHK-MAS-010b):

```

-w /etc/audit/          -p wa -k audit_rules
-w /var/log/audit/      -p wa -k logs
-w /etc/passwd          -p wa -k identity
-w /etc/shadow          -p wa -k identity
-w /etc/group           -p wa -k identity
-w /etc/gshadow         -p wa -k identity
-w /etc/sudoers         -p wa -k sudoers
-w /etc/sudoers.d/      -p wa -k sudoers
-w /etc/ssh/sshd_config -p wa -k sshd
-w /etc/pam.d/          -p wa -k pam
-a always,exit -F arch=b64 -S execve -F euid=0 -F auid!=unset -k exec
-a always,exit -F arch=b32 -S execve -F euid=0 -F auid!=unset -k exec

```

Ключ	Событие СКР
identity	integrity/file_modified — изменение базы учётных записей
sudoers, sshd, pam	policy/security_setting_changed
audit_rules	policy/audit_policy_changed
logs	integrity/log_tampered
exec	process/created — команды, запущенные людьми от root
szl	изменение конфигурации СЗИ: -w /etc/<сзи> -p wa -k szl

Имена ключей менять нельзя: записи с другими ключами агент отбрасывает. Правило `exec` ограничено действиями людей (`auid>=1000`) под root. Без этого ограничения основной поток событий дадут демоны и cron.

`/etc/audit/rules.d/31-gits-noise.rules` — подавление шума. Правила необязательные: на полноту контроля они не влияют, старт и остановку служб агент берёт из journald.

```

-a always,exclude -F msgtype=SERVICE_START
-a always,exclude -F msgtype=SERVICE_STOP
-a always,exclude -F msgtype=1334
#-a always,exclude -F exe=/usr/sbin/cron      # только auditd ≥ 3.0

```

Загрузка и проверка:

```

augenrules --check && augenrules --load
auditctl -l | grep -E 'key=|-k '
auditctl -s | head -3          # enabled 1

```

- **auditd 2.8** (Astra 1.7, Debian 10) не принимает поле `exe` в правилах `exclude`. Строка с `exe` прерывает `augenrules --load`, и следующие за ней правила не загружаются. Поэтому правило для cron закомментировано.
- Не добавляйте `-e 2` (запрет изменения правил до перезагрузки) на Astra: он ломает `astra-admin-events` и проверку CHK-MAS-010с. На других системах с `-e 2` новые правила применяются только после перезагрузки.
- Файлы правил не содержат `-D`, `-b` и `-f`: эти параметры задаёт базовый `audit.rules` дистрибутива.

5. journald

/etc/systemd/journald.conf.d/50-gits.conf:

```
[Journal]
Storage=persistent      # на Astra по умолчанию журнал в памяти
SystemMaxUse=512M
RateLimitIntervalSec=30s
RateLimitBurst=10000    # не терять строки sshd при переборе паролей
```

```
mkdir -p /var/log/journal && systemd-tmpfiles --create --prefix /var/log/journal
systemctl restart systemd-journald && journalctl --flush
```

Проверка CHK-MAS-013b читает Storage из основного /etc/systemd/journald.conf: там не должно быть Storage=volatile или Storage=none.

6. Astra Linux: события PARSEC

Нужны пакеты parsec-base (утилита astra-admin-events), syslog-ng-core и syslog-ng-mod-astra. syslog-ng не отключайте: он пишет журнал /parsec/log/astra/events, который читает агент. Файлы mod-astra*.conf, 10-parsec.rules и astra-syslog.rules принадлежат пакетам, руками их не правьте: состав событий меняется только через astra-admin-events.

Проверка CHK-MAS-010с требует включить 25 событий:

```
astra-admin-events -E succeeded-authorization failed-authorization user-session add-rule remove-rule \
    creating-account deleting-account changing-account-password changing-primary-group changing-uid \
    adding-user-to-group removing-user-from-group creating-group removing-group modifying-group-members \
    audit-log-removed events-log-removed daemon-start daemon-end syslog-ng-stop package-installed \
    new-usb-device usb-mass-storage-detected unsigned-binary-file modifying-file -e
astra-admin-events | grep -E ' 1 '
```

Рекомендуемый для СКР состав шире: всё, что классифицирует агент, плюс значимые события вне его таксономии. prepare-host.sh включает этот состав, а с параметром --astra-quiet выключает шумные события:

```
# Включить
astra-admin-events -G authorization user-accounting user-groups audit security-logs \
    safepolicy general-software system-time system-operation -e
astra-admin-events -E previous-login user-blocked-by-tally modifying-directory package-not-installed \
    package-removing-started usb-disconnect device-mount device-unmount \
    device-mount-attempt-blocked process-ends-abnormally \
    self-diagnostics-critical self-diagnostics-error -e

# Шум (--astra-quiet): запуск процессов, сеть, открытие и удаление файлов
astra-admin-events -G network network-addressing resources access-rights -d
astra-admin-events -E execute-process execute-sudo-process process-ends kernel-module file-opened file-re
moval \
    package-installation-started package-upgrade-started process-out-of-ram \
    self-diagnostics-started self-diagnostics-completed self-diagnostics-warning -d
```

События печати (группа `printing`) оставьте включёнными, если нужен контроль печати (ПУИ.30).

Правила `parsec` на Astra не содержат наблюдения за `/etc/passwd`, `/etc/sudoers` и т. п., поэтому правила `30-gits.rules` на Astra нужны так же, как на других системах. Они дополняют правила `parsec` и не мешают им.

PostgreSQL на Astra пишет мандатный аудит (`USER_AVC`) в `audit.log`. Это даёт большой объём записей, до 40 МБ за 8 часов. Если аудит СУБД ведётся её собственными средствами, раскомментируйте последнее правило в `31-gits-noise.rules`, указав путь к `postgres` своей версии.

7. Время

Нужна служба синхронизации времени с внутренними источниками или с контроллером домена (01 §1.4).

```
# chrony (Debian, RHEL): в /etc/chrony/chrony.conf или /etc/chrony.conf – server ntp1.bank.local iburst
systemctl enable --now chrony          # RHEL: chronyd
# ntp (Astra 1.7):          в /etc/ntp.conf – server ntp1.bank.local iburst
systemctl enable --now ntp
timedatectl                            # System clock synchronized: yes
```

8. Прочее

- **sshd**. Проверки SSH (CHK-RD-030b, 036, 037b) читают эффективную конфигурацию `sshd -T`. На хостах без `openssh-server` они вернут `error`.
- **mokutil**. Проверка Secure Boot (CHK-CZI-025b) использует `mokutil --sb-state`.
- **Антивирус**. Добавьте в исключения `/usr/local/bin/gits-agent` (или `/usr/bin/gits-agent`) и `/var/lib/gits-agent`.

9. Проверки агента и настройки хоста

Проверка	Что должно быть на хосте	Где настроить
CHK-MAS-010b	<code>auditd</code> активен и в автозапуске; наблюдение <code>-p wa</code> за <code>/etc/passwd</code> , <code>/etc/shadow</code> , <code>/etc/group</code> , <code>/etc/sudoers</code> , <code>/var/log/audit</code> , <code>/etc/audit</code> , <code>/etc/ssh/ssh_config</code> , <code>/etc/pam.d</code> ; правило <code>execve</code>	§3, §4
CHK-MAS-010c (Astra)	25 событий PARSEC включены	§6
CHK-MAS-013b	<code>max_log_file ≥ 8</code> , <code>num_logs ≥ 5</code> , <code>max_log_file_action = rotate</code> ; <code>journald</code> не <code>volatile</code>	§3, §5
CHK-MAS-014b	<code>/var/log/audit</code> — владелец <code>root</code> , права 0700, 0740 или 0750	§3
CHK-RD-017b	<code>/etc/shadow</code> , <code>/etc/gshadow</code> — 0000/0600/0640; в <code>/etc/passwd</code> только <code>x</code> ; <code>ENCRYPYPT_METHOD SHA512</code> или <code>YESCRYPT</code> в <code>login.defs</code>	политика ОС
CHK-RD-019	<code>PASS_MAX_DAYS ≤ 365</code> в <code>/etc/login.defs</code>	политика ОС
CHK-RD-021	<code>minlen ≥ 8</code> в <code>pwquality.conf</code> или в параметрах <code>pam_pwquality</code> . На Astra сложность пароля настраивается другим механизмом, и проверка может показать <code>fail</code> при корректной настройке	политика ОС
CHK-RD-023	<code>minclass ≥ 3</code> или <code>ucredit/lcredit/dcredit ≤ -1</code>	политика ОС
CHK-RD-030b, 036, 037b	<code>PermitEmptyPasswords no</code> , <code>PrintLastLog yes</code> , задан <code>Banner</code> , <code>PermitRootLog in no</code> или <code>prohibit-password</code>	<code>sshd_config</code>

Проверка	Что должно быть на хосте	Где настроить
CHK-UZP-010c	нет <code>NOPASSWD: ALL</code> в <code>/etc/sudoers*</code>	<code>sudoers</code>
CHK-RD-033	нет файлов с записью для всех в <code>/etc</code> , <code>/usr</code> , <code>/opt</code>	права файлов
CHK-PUI-018b	<code>install usb-storage /bin/false</code> или <code>blacklist usb-storage</code> в <code>/etc/modprobe.d/</code>	политика ОС
CHK-CZI-025b	Secure Boot включён	UEFI
CHK-ZVK-001L, CHK-ZVK-009	активна и в автозапуске служба антивируса: <code>kes1</code> , <code>drweb-configd</code> или <code>clamav-daemon</code>	AB
CHK-CZI-018, CHK-CZI-024	утверждённый эталон пакетов и целостности конфигурации	консоль, карточка хоста
CHK-MAS-021	буфер агента заполнен меньше чем на 80 %, доставка работает	связь с сервером

10. Самопроверка

```

sudo ./prepare-host.sh --check
systemctl is-active auditd systemd-journald gits-agent
auditctl -l | grep -c -- '-k \key='
# Сгенерировать события и найти их в консоли («События», роль auditor)
sudo id; sudo touch /etc/passwd
ssh nosuchuser@localhost                # неуспешный вход

```

На стенде после подготовки хоста Astra Linux 1.7.6 в консоль пришли события `process/created` (ключ `exec`), `integrity/file_modified` (`identity`), `policy/audit_policy_changed` (`audit_rules`), а также события входа из `journald`, `auditd` и `PARSEC`. Проверка CHK-MAS-010b сменила результат с `fail` на `pass`.

6. Подготовка Windows-хоста

Агент **ничего не меняет в политике аудита хоста**. Он читает журналы и проверяет настройки. По умолчанию Windows не регистрирует большую часть нужных событий (создание процессов, изменение групп, проверку учётных данных), а размер журнала Security — 20 МБ. Такой журнал на сервере перезаписывается за часы.

В домене параметры задаются **групповой политикой** (§2–§5). На отдельных хостах и для проверки можно использовать скрипт `prepare-host.ps1` (§1). Доменная политика имеет приоритет: при следующем `groupdate` она перезапишет локальные значения.

Что читает агент

Канал	Нагрузка	Что нужно
Security	до 2 000 событий/с	расширенная политика аудита (§2), размер журнала (§4)
System	до 500 событий/с	установка служб (7045), изменение режима запуска (7040), остановки, перезагрузки (1074, 6005, 6006, 6008), очистка журнала (104)

Классифицируемые события Security: 4624, 4625, 4634, 4647, 4648, 4672, 4673, 4688, 4697, 4698, 4719, 4720, 4722–4726, 4728, 4732, 4733, 4740, 4756, 4767, 4776, 1100, 1102, 1104, 4946–4948, 6416.

1. Быстрая подготовка: `prepare-host.ps1`

Скрипт находится в `deploy/host-prep/windows/` внутри `gits-deploy_<ver>.zip`. Запуск от имени администратора:

```
powershell -ExecutionPolicy Bypass -File prepare-host.ps1 -Check      # только отчёт; код возврата 1 – нес
оответствия
powershell -ExecutionPolicy Bypass -File prepare-host.ps1           # применить
powershell -ExecutionPolicy Bypass -File prepare-host.ps1 -NtpServer ntp1.bank.local,ntp2.bank.local -Ena
bleWinRM
```

Параметр	Действие
<code>-Check</code>	ничего не менять
<code>-SecurityLogMB</code>	размер журнала Security, МБ (по умолчанию 1024; для серверов рекомендуется 4096)
<code>-OtherLogMB</code>	размер журналов System и Application, МБ (по умолчанию 256)
<code>-NtpServer</code>	источники NTP для хостов вне домена
<code>-EnableWinRM</code>	включить WinRM для push-установки агента (§6)

Скрипт настраивает подкатегории аудита по GUID, поэтому работает при любом языке ОС. Он задаёт параметры реестра из §3, размеры журналов, режим запуска W32Time и проверяет источник времени. Текущие значения подкатегорий читаются так же, как их читает агент: через `auditpol /backup`. Повторный запуск ничего не меняет.

2. Расширенная политика аудита

GPO: «Конфигурация компьютера» → «Политики» → «Конфигурация Windows» → «Параметры безопасности» → «Конфигурация расширенной политики аудита» → «Политики аудита».

Подкатегория	GUID	Требуется	События
Вход в систему (Logon)	{0CCE9215-...}	Успех и сбой	4624, 4625, 4648
Выход из системы (Logoff)	{0CCE9216-...}	Успех	4634, 4647
Блокировка учётной записи (Account Lockout)	{0CCE9217-...}	Сбой	4625 (заблокированная УЗ)
Специальный вход (Special Logon)	{0CCE921B-...}	Успех	4672
Проверка учётных данных (Credential Validation)	{0CCE923F-...}	Успех и сбой	4776
Управление учётными записями пользователей	{0CCE9235-...}	Успех и сбой	4720, 4722–4726, 4740, 4767
Управление группами безопасности	{0CCE9237-...}	Успех и сбой	4728, 4732, 4733, 4756
Создание процесса (Process Creation)	{0CCE922B-...}	Успех	4688
Изменение политики аудита	{0CCE922F-...}	Успех и сбой	4719
Изменение политики проверки подлинности	{0CCE9230-...}	Успех	—
Использование прав, затрагивающее конфиденциальные данные	{0CCE9228-...}	Успех и сбой	4673
Изменение состояния безопасности	{0CCE9210-...}	Успех	—
Расширение системы безопасности	{0CCE9211-...}	Успех (рекоменд.)	4697
Другие события доступа к объектам	{0CCE9227-...}	Успех (рекоменд.)	4698 (задания планировщика)
Изменение политики на уровне правил MPSSVC	{0CCE9232-...}	Успех (рекоменд.)	4946–4948 (правила брандмауэра)
События Plug and Play	{0CCE9248-...}	Успех (рекоменд.)	6416 (внешние устройства)

Суффикс всех GUID: -69AE-11D9-BED3-505054503030. Первые 12 строк проверяет CHK-MAS-010a. Остальные нужны, чтобы соответствующие события вообще появлялись в журнале.

Вручную (имена подкатегорий локализованы, поэтому указывайте GUID):

```
auditpol /set /subcategory:{0CCE922B-69AE-11D9-BED3-505054503030} /success:enable
auditpol /get /category:*
```

3. Параметры реестра аудита

Параметр	GPO	Реестр (DWORD = 1)
Принудительное использование подкатегорий	«Параметры безопасности» → «Локальные политики» → «Параметры безопасности» → «Аудит: принудительно переопределяет параметры категорий политики аудита...»	HKLM\SYSTEM\CurrentControlSet\Control\Lsa\SCENoApplyLegacyAuditPolicy
Командная строка в 4688	«Административные шаблоны» → «Система» → «Аудит создания процессов» → «Включать командную строку в события создания процессов»	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Audit\ProcessCreationIncludeCmdLine_Enabled
Журналирование блоков сценариев PowerShell	«Административные шаблоны» → «Компоненты Windows» → «Windows PowerShell» → «Включить ведение журнала блоков сценариев PowerShell»	HKLM\SOFTWARE\Policies\Microsoft\Windows\PowerShell\ScriptBlockLogging\EnableScriptBlockLogging

Командная строка процессов и блоки сценариев PowerShell могут содержать пароли, переданные в параметрах. Доступ к журналам Security и Microsoft-Windows-PowerShell/Operational ограничьте администраторами. Канал PowerShell/Operational агент сейчас не собирает: CHK-MAS-010a проверяет только сам параметр.

4. Размеры журналов

GPO: «Административные шаблоны» → «Компоненты Windows» → «Служба журнала событий» → Security / System / Application → «Максимальный размер журнала (КБ)».

Журнал	Минимум (CHK-MAS-013a)
Security	1 048 576 КБ (1 ГБ) на APM; для серверов рекомендуется 4 194 304 КБ
System, Application	262 144 КБ (256 МБ)

Без GPO: `wevtutil sl Security /ms:1073741824`. Если размер задан и групповой политикой, и параметром службы, действует значение из политики.

5. Время

Члены домена синхронизируются с контроллером домена (`W32Time\Parameters\Type = NT5DS`). Хосты вне домена синхронизируются с внутренними серверами NTP:

```
w32tm /config /manualpeerlist:"ntp1.bank.local,0x8 ntp2.bank.local,0x8" /syncfromflags:manual /update
w32tm /resync
w32tm /query /source          :: не «Local CMOS Clock» и не «Free-running System Clock»
```

Служба W32Time должна работать и запускаться автоматически. CHK-MAS-009a проверяет тип синхронизации, источник и расхождение с часами сервера (не более 5 секунд).

6. WinRM для push-установки

Нужен только для push-установки агента из консоли (04 §7).

```
Enable-PSRemoting -Force -SkipNetworkProfileCheck # слушатель HTTP 5985, правило брандмауэра
```

- Исполнитель (ядро или ретранслятор) подключается к порту 5985 по NTLM с шифрованием сообщений или к порту 5986 по HTTPS с Basic-аутентификацией. Разрешите ему доступ на межсетевом экране хоста.
- Для входа **локальной** учётной записью администратора по сети нужен `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy = 1`. Скрипт с `-EnableWinRM` задаёт этот параметр. Для доменных администраторов он не требуется.
- После установки агентов WinRM можно отключить, если он не нужен для других задач.

7. Прочее

- **Службы.** Должны работать «Журнал событий Windows» (EventLog) и WMI (Winmgmt). Нужен Windows PowerShell 5.x в `%SystemRoot%\System32\WindowsPowerShell\v1.0`: часть проверок выполняется через него.
- **Антивирус.** Добавьте в исключения `%ProgramFiles%\GITS Agent\` и `%ProgramData%\GITS\agent\`.

8. Проверки агента и настройки хоста

Проверка	Что должно быть	Где настроить
CHK-MAS-010a	подкатегории аудита (§2), параметры реестра (§3)	GPO: расширенная политика аудита
CHK-MAS-013a	размеры журналов (§4)	GPO: служба журнала событий
CHK-MAS-009a	NT5DS или NTP, источник не локальные часы, расхождение не более 5 с	W32Time
CHK-UZP-010a	состав локальной группы «Администраторы» соответствует утверждённому эталону	GPO «Группы с ограниченным доступом», эталон в консоли
CHK-UZP-010b	UAC: <code>EnableLUA=1</code> , <code>ConsentPromptBehaviorAdmin</code> 1 или 2, <code>ConsentPromptBehaviorUser</code> 0 или 1, <code>PromptOnSecureDesktop=1</code>	GPO: «Параметры безопасности» → «Контроль учётных записей»
CHK-RD-011a	блокировка после 1–10 неудачных попыток; длительность блокировки и сброс счётчика — не меньше 30 мин	GPO: «Политика блокировки учётной записи»
CHK-RD-019, 021, 023, 025	срок действия пароля не больше 365 дней и не 0; длина не меньше 8; сложность включена; минимальный срок не меньше 1 дня	GPO: «Политика паролей»
CHK-RD-013a, 014a, 015	блокировка сеанса разрешена; неактивность не больше 900 с; хранитель экрана с паролем	GPO компьютера и пользователя
CHK-RD-018a	<code>LmCompatibilityLevel=5</code> , <code>NoLMHash=1</code> , подпись SMB всегда, <code>LDAPClientIntegrity ≥ 1</code>	GPO: «Параметры безопасности»
CHK-RD-030a	<code>RestrictAnonymous=1</code> , <code>RestrictAnonymousSAM=1</code> , <code>EveryoneIncludesAnonymous=0</code>	GPO: «Сетевой доступ»
CHK-RD-034	заполнены «Отказать в доступе к компьютеру из сети» и «Запретить вход через службы удалённых рабочих столов»	GPO: «Назначение прав пользователя»
CHK-RD-036, 037a	показ сведений о предыдущих входах; не показывать последнего пользователя; Ctrl+Alt+Del не отключён; текст уведомления задан	GPO: «Интерактивный вход»
CHK-RD-031	на общих папках нет прав «Изменение» и «Полный доступ» для Everyone, Users, Authenticated Users	права общих папок

Проверка	Что должно быть	Где настроить
CHK-RD-005a	службы не работают под персональными доменными учётными записями	службы
CHK-CZI-019b	AlwaysInstallElevated=0; SmartScreen включён	GPO
CHK-CZI-017, CHK-CZI-024, CHK-CZI-026b	утверждённые эталоны ПО и целостности; нет Java Runtime без обоснования	консоль, карточка хоста
CHK-ZVK-001b, 008, 009, 010, 006, 012	Microsoft Defender: защита включена, сигнатуры не старше суток, защита в реальном времени, Tamper Protection, проверка съёмных дисков, расписание полной проверки	GPO: Microsoft Defender
CHK-ZUD-010	ноутбуки: BitLocker на системном томе, TPM+PIN	GPO: шифрование диска BitLocker
CHK-MAS-021	буфер агента заполнен меньше чем на 80 %, доставка работает	связь с сервером

9. Самопроверка

```
powershell -ExecutionPolicy Bypass -File prepare-host.ps1 -Check
auditpol /get /category:*
Get-WinEvent -LogName Security -MaxEvents 5 | Format-Table Id, TimeCreated
sc.exe query gits-agent
```

На стенде (Windows 10 LTSC 2021, русская локаль, член домена) скрипт исправил 18 параметров. После этого все требования аудита и журналов выполнены. Осталось одно несоответствие: источник времени Local CMOS Clock, так как хост не синхронизировался с контроллером домена.

7. Диагностика

Журналы компонентов:

Компонент	Журнал
Сервер, ядро, консоль	<code>journalctl -u gits-server (gits-core, gits-console)</code>
Ретранслятор	<code>journalctl -u gits-relay</code>
Агент Linux	<code>journalctl -u gits-agent</code>
Агент Windows	журнал «Приложение», источник <code>gits-agent</code> ; <code>sc query gits-agent</code>

Сервер

Симптом	Причина и решение
<code>run gitsctl init-ca? / run gitsctl gen-signing-keys?</code> при запуске	нет УЦ или ключей в <code>data_dir</code> : выполните 02 §5 или восстановите <code>/var/lib/gits</code> из копии
<code>... is accessible by group/others (mode ...); expected 0600</code>	у файла ключа слишком широкие права: <code>chmod 0600</code> и владелец <code>gits</code> . Обычно так бывает после копирования или запуска <code>gitsctl</code> не от <code>gits</code>
<code>error obtaining MAC configuration for user "gits" (Astra)</code>	нет мандатной метки у пользователя БД: <code>pdpl-user -l 0:0 gits</code>
<code>failed to connect to user=gits database=gits</code>	неверный <code>db_dsn / GITS_DB_DSN</code> , СУБД не запущена или подключение запрещено в <code>pg_hba.conf</code>
консоль не стартует, в журнале ожидание схемы	консоль не применяет миграции: сначала запустите ядро или выполните <code>gits-server migrate</code>
<code>bind: permission denied</code> на порту 443	служба запущена не из штатного юнита (нет <code>CAP_NET_BIND_SERVICE</code>) или порт занят
консоль открывается, но в журнале действий у всех один адрес	консоль работает за обратным прокси (02 §6)
<code>permission denied</code> в <code>gitsctl load-measures</code>	CSV-файл недоступен пользователю <code>gits</code> : скопируйте его в <code>/tmp</code> с правами 0644

Регистрация и подключение агента

Симптом	Причина и решение
<code>x509: certificate is valid for ..., not ...</code>	агент обращается по адресу, которого нет в сертификате шлюза или ретранслятора. Выпустите сертификат с этим адресом (<code>issue-server-cert -dns ... -ip ..., dns_names / ip_addresses</code> ретранслятора) или задайте <code>server_name</code> в <code>agent.yaml</code>
<code>certificate signed by unknown authority</code>	в <code>ca_file / ca_cert</code> не тот корень: возьмите <code>/var/lib/gits/ca/ca.crt</code> с узла ядра
<code>certificate has expired or is not yet valid</code>	часы хоста расходятся с сервером больше чем на 5 минут. Синхронизируйте время (01 §1.4). На виртуальных машинах после паузы или восстановления из снимка часы гостя могут отставать на дни, пока не включена синхронизация с гипервизором

Симптом	Причина и решение
токен отклонён	срок токена истёк или исчерпан лимит установок: выпустите новый. Лимит регистраций с одного адреса — <code>gateway.enroll_per_minute</code>
<code>connection refused</code> / тайм-аут	шлюз слушает <code>127.0.0.1</code> (задайте <code>gateway.listen: 0.0.0.0:8443</code>) или порт закрыт межсетевым экраном
агент долго не переподключается после перезапуска сервера	повтор подключения растягивается до 5 минут, это штатное поведение. Для ускорения перезапустите службу агента
хост зарегистрировался как новый, в консоли предупреждение о клоне	ВМ клонирована вместе с каталогом состояния агента. Решение принимает оператор: карточка хоста → «Экземпляры агента»
пакет <code>deb/rpm</code> установлен, служба не запускается	агент не зарегистрирован: <code>gits-agent enroll ...</code> , затем <code>systemctl enable --now gits-agent</code>

Команды и действия реагирования

Симптом	Причина и решение
<code>command issued in the future</code> / команда просрочена	расхождение часов агента и сервера больше 5 минут
блокировка локальной учётной записи Linux завершается ошибкой записи в <code>/etc/shadow</code>	агент установлен с юнитом старой версии, где был <code>ProtectSystem=full</code> . Выполните <code>gits-agent install</code> новой версии (юнит перезапишется) или удалите строки <code>ProtectSystem/ReadWritePaths</code> из юнита и выполните <code>systemctl daemon-reload && systemctl restart gits-agent</code>

Проверки и события

Симптом	Причина и решение
проверки дают <code>error</code> на Windows	агент запущен не от <code>LocalSystem</code> или не работает WMI/EventLog
CHK-MAS-010a fail при настроенной политике	проверьте <code>auditpol /get /category:*</code> и значение <code>SCENoApplyLegacyAuditPolicy</code> . Доменная GPO могла перезаписать локальные настройки. Старые версии агента на русской Windows неверно разбирали подкатегорию «Использование прав, затрагивающее конфиденциальные данные». Обновите агента
CHK-MAS-009a fail на русской Windows при исправном времени	старые версии каталога читали подпись <code>Source:</code> из <code>w32tm /query /status</code> . Обновите сервер: каталог доставляется агентам автоматически
CHK-MAS-010b fail на Astra	правил <code>parsecs</code> недостаточно: установите <code>30-gits.rules</code> (05 §4)
<code>augenrules --load</code> завершился ошибкой, часть правил не загружена	в правилах есть конструкция, которую не поддерживает <code>auditd</code> , например <code>exclude -F exe=</code> на <code>auditd 2.8</code> . Проверьте построчно: <code>auditctl <строка правила></code>
нет событий <code>auditd</code>	<code>log_format</code> не RAW, либо ключи правил не из списка (05 §1)
после перезагрузки Linux-хоста пропали события	<code>journald</code> хранит журнал в памяти: включите постоянное хранение (05 §5)
нет событий 4688, 4728 и др.	подкатегория аудита не включена (06 §2)
проверка CHK-MAS-021 fail	буфер агента заполняется: нет связи с сервером или мал диск. Проверьте подключение и место в каталоге состояния агента

Push-установка

Симптом	Причина и решение
WinRM: 401	NTLM отклонён: проверьте учётную запись, LocalAccountTokenFilterPolicy для локальных администраторов и политику NTLM домена. Альтернативы — порт 5986 (HTTPS, Basic) или установочный комплект через GPO
SSH: sudo: sorry, you must have a tty	в sudoers включён requiretty: отключите его для учётной записи установки
задание «зависло»	задание, которое выполняется дольше 30 минут, завершается с ошибкой. Проверьте связь исполнителя с хостом