SELinux в НАЙС.ОС

SELinux (Security-Enhanced Linux) — это система мандатного контроля доступа, встроенная в ядро Linux и расширяющая стандартную модель безопасности. Она обеспечивает чёткое разграничение прав на основе политик безопасности, а не только пользовательских прав. В НАЙС.ОС SELinux включён по умолчанию и работает в режиме targeted, ограничивая действия системных служб, пользовательских приложений и контейнеров. В редакциях с повышенными требованиями безопасности доступна и политика MLS (многоуровневая защита). Что получает пользователь: Защиту от эксплойтов, эскалации прав и нарушений изоляции; Контроль доступа даже для процессов с правами root; Гарантию, что один сервис не сможет прочитать данные другого — даже при уязвимости; Логи и аудит событий безопасности через journalctl и auditd. Итог: SELinux делает систему стойкой к взлому на архитектурном уровне. В НАЙС.ОС он работает «из коробки», без дополнительной настройки.

SELinux в НАЙС.ОС

Политика SELINUXTYPE=default, тип TYPE=mcs, модульная сборка.

Документ: эксплуатационная документация (администрирование)
Назначение: включение, проверка, настройка и сопровождение SELinux в НАЙС.ОС

SELinux в НАЙС.ОС: архитектура,

режимы работы и эксплуатация

Документ устанавливает техническое описание реализации SELinux в НАЙС.ОС, включая состав политики, модель MCS, режимы enforcing/permissive, управление метками, диагностику AVC и требования к вводу в эксплуатацию.

1. Область применения

SELinux (Security-Enhanced Linux) в НАЙС.ОС используется как механизм обязательного контроля доступа (MAC, Mandatory Access Control), реализующий разграничение прав на основе меток безопасности (security context) и формализованных правил политики.

Реализация ориентирована на профиль **контейнерного хоста** и минимального серверного окружения, с обеспечением совместимости типовых системных сценариев (загрузка, вход, журналирование, сеть, управление сервисами, контейнерный рантайм).

Важно. В режиме `permissive` SELinux не блокирует операции, но фиксирует нарушения (AVC) для последующего анализа. Для продуктивного применения режима `enforcing` требуется стендовая валидация рабочих нагрузок и регламент обработки AVC.

2. Термины и сокращения

- **MAC** — обязательный контроль доступа на основе политики и меток.
- **TE** (Type Enforcement) — модель “домен тип объекта” как основа правил SELinux.
- **Domain** — домен процесса (тип процесса), например `sshd_t`, `systemd_t`.
- **Type** — тип объекта ФС/IPC, например `etc_t`, `var_log_t`.
- **Context** — SELinux-контекст вида `user:role:type:level`.
- **MCS** — Multi-Category Security (категории), облегчённая модель изоляции (в т.ч. для контейнеров).
- **AVC** — событие отказа/нарушения политики (в permissive — логируется, в enforcing — блокируется).

3. Профиль политики SELinux в НАЙС.ОС

3.1. Тип политики

В НАЙС.ОС используется модульная refpolicy-политика в конфигурации: **TYPE = mcs**, **NAME = default**, **DISTRO = niceos**. Политика предназначена для минимального окружения и контейнерного хоста с MCS-изоляцией.

3.2. Параметры сборки политики

Параметры определяют формат бинарной политики, модель меток и совместимость с ядром:

```
# Политика SELinux в НАЙС.ОС (профиль default)
#OUTPUT_POLICY = 32
TYPE = mcs
NAME = default
DISTRO = niceos

# Совместимость с неизвестными разрешениями (см. раздел 9)
UNK_PERMS = allow

DIRECT_INITRC = n
MONOLITHIC = n
UBAC = n

# Категории MCS
MCS_CATS = 1024

QUIET = n
```

Пояснение. `MONOLITHIC = n` означает модульную структуру политики: функции распределены по модулям (base/contrib/container), что упрощает сопровождение и целевое включение подсистем.

3.3. Режим SELinux по умолчанию

В НАЙС.ОС состояние SELinux управляется через `/etc/selinux/config`. Типовой профиль задаётся как `SELINUXTYPE=default`. Режим по умолчанию — `permissive` (эксплуатационный переход в `enforcing` выполняется после валидации).

```
# /etc/selinux/config
SELINUX=permissive
SELINUXTYPE=default
```

4. Состав политики: базовые подсистемы и модули

Политика НАЙС.ОС построена на refpolicy и разделена на: (1) базовые модули (ядро правил), (2) функциональные модули системных служб, (3) модули контейнерного профиля. Это позволяет обеспечивать предсказуемость минимальной установки и масштабировать покрытие при расширении состава ПО.

4.1. Базовые подсистемы

Базовый слой охватывает доменную модель, файловые типы, взаимодействие с ядром, сетью и терминалами, а также MCS/MLS-структуры уровня.

4.2. Системные сервисы и инфраструктурные модули

В профиле `default` предусмотрены модули для типовых задач: загрузка/инициализация, systemd, udev, ssh, iptables, журналирование, управление пользователями, подсистемы хранения (LVM/FS tools), сетевой стек, политики для “unconfined” режима, а также вспомогательные модули администрирования и аудита.

4.3. Контейнерный профиль

Контейнерный модуль обеспечивает: (1) домены процессов контейнерного рантайма, (2) типы файлов/томов контейнерного хранилища, (3) MCS-изоляцию контейнеров, (4) совместимость контейнерных оркестраторов и компонентов кластерной инфраструктуры при условии корректной разметки меток.

5. Модель безопасности: контексты, домены, роли, уровни

5.1. Формат контекста

SELinux-контекст имеет вид:

```
user:role:type:level
```

Пример (типовой):

```
system_u:system_r:sshd_t:s0
```

5.2. Пользователи SELinux и соответствие Unix-учётным записям

В профиле НАЙС.ОС определены базовые SELinux-идентичности и роли для системных процессов и пользователей. Конфигурация рассчитана на минимальную установку и административные сценарии.

```
# Концептуально (минимальный профиль)
system_u     — системные процессы
user_u       — пользователь без специальной роли
staff_u      — оператор/администратор (staff_r + возможность перехода)
sysadm_u     — администратор (sysadm_r)
root         — расширенный набор ролей (unconfined_r/sysadm_r/staff_r/system_r)
unconfined_u — совместимый профиль “без ограничений” для переходного режима
```

Эксплуатационная рекомендация. Для enforcing-режима следует минимизировать использование `unconfined_r` и переводить административные операции в предсказуемые роли (`staff_r`/`sysadm_r`), фиксируя регламент.

5.3. MCS в НАЙС.ОС

В НАЙС.ОС используется MCS-модель с числом категорий `MCS_CATS=1024`. Уровень задаётся как `s0` с категориями `c0..c1023`. Назначение: изоляция контейнеров и групп объектов без полной MLS-модели.

```
# Примеры MCS-уровней
s0
s0:c0
s0:c3,c10
```

```
s0:c100.c200
```

Для контейнеров применяется принцип: процесс контейнера и его данные получают совпадающие категории MCS. Это предотвращает доступ контейнера к данным другого контейнера при корректной маркировке объектов.

6. Режимы работы SELinux и управление состоянием

6.1. Проверка состояния

```
getenforce
sestatus
```

6.2. Временное переключение enforcing/permissive

```
# Включить enforcement до перезагрузки
setenforce 1

# Отключить enforcement до перезагрузки
setenforce 0
```

6.3. Постоянная настройка

Постоянный режим задаётся в `/etc/selinux/config`. Изменение вступает в силу после перезагрузки.

6.4. Стартовые параметры ядра (диагностика/восстановление)

Для аварийных сценариев допускается временное отключение SELinux на уровне загрузки (например, при некорректной маркировке). Применение следует регламентировать, поскольку отключение SELinux снижает уровень защиты.

```
selinux=0
enforcing=0
```

7. Метки объектов: файловая система, порты, процессы

7.1. Просмотр меток

```
# Файлы и каталоги
ls -Z /etc /var /usr | head

# Процессы
ps -eZ | head
```

7.2. Восстановление контекстов по политике

Базовая операция приведения меток к ожидаемым значениям — `restorecon`. Используется при переносе файлов, ручном редактировании путей, восстановлении после миграций и при подготовке enforcing-режима.

```
# Восстановить контекст для одного объекта
restorecon -v /etc/ssh/sshd_config

# Рекурсивно для каталога
restorecon -Rv /var/lib
```

7.3. Назначение контекстов для нестандартных путей

Для постоянного назначения контекста нестандартным путям применяется `semanage fcontext` с последующим `restorecon`.

```
# Пример: добавить правило контекста для каталога приложения
semanage fcontext -a -t var_lib_t "/opt/myapp(/.*)?"
restorecon -Rv /opt/myapp
```

7.4. Управление портами

SELinux разделяет сетевые сервисы по типам портов. Для нестандартных портов используется `semanage port`.

```
# Просмотр
semanage port -l | head

# Пример: назначить порт для ssh (пример приведён как методика)
semanage port -a -t ssh_port_t -p tcp 2222
```

8. Интеграция с системными службами и контейнерной подсистемой

8.1. systemd и доменные переходы

В НАЙС.ОС запуск сервисов осуществляется через systemd с использованием доменных переходов (domain transition), при которых процесс получает целевой домен на основе типа исполняемого файла и правил политики. Это обеспечивает разделение прав системных компонентов и предсказуемое поведение в enforcing-режиме.

```
# Диагностика домена процесса сервиса
systemctl status sshd
ps -eZ | grep -E 'sshd|systemd' | head
```

8.2. Контейнеры и MCS-изоляция

Контейнерный профиль использует: (1) отдельные домены процессов контейнеров и рантайма, (2) типы данных контейнерного хранилища, (3) MCS-категории для изоляции контейнеров друг от друга.

Для корректной работы требуется соблюдение двух условий: (а) каталоги/тома контейнеров должны иметь согласованные SELinux-метки, (б) процессы контейнера должны стартовать в доменах, предусмотренных политикой.

```
# Базовый контроль: наличие SELinux и категорий у процессов контейнеров
ps -eZ | grep container | head

# Контроль меток у хранилища (путь зависит от рантайма)
ls -Z /var/lib | grep -E 'container|kube|etcd' || true
```

Практика эксплуатации. При интеграции внешних томов (bind-mount) необходимо

обеспечить корректную маркировку каталогов на хосте (через `semanage fcontext`/`restorecon`), иначе enforcing-режим приведёт к блокировке доступа.

8.3. Сетевые подсистемы и журналирование

Политика включает правила для штатной работы сетевых утилит, iptables/nft и подсистем журналирования. Для enforcing-режима критично обеспечить правильные метки каталогов журналов и сокетов IPC, а также согласованность доменов процессов, взаимодействующих с journald/syslog/audit.

9. AVC и диагностика нарушений политики

9.1. Источники событий

События AVC фиксируются в журнале аудита (при активном auditd) и/или в системном журнале.

```
# Поиск AVC за последнее время
ausearch -m AVC,USER_AVC -ts recent

# Сводка по причинам
ausearch -m AVC,USER_AVC -ts recent | audit2why
```

9.2. Режим UNK_PERMS

Параметр `UNK_PERMS=allow` допускает неизвестные разрешения (например, при рассогласовании версий policy/kernselinux perms). Это повышает совместимость, но снижает строгость контроля. При подготовке сертифицируемых/жёстких контуров допускается перевод в режим запрета неизвестных разрешений (в рамках отдельного профиля политики и после стендовой проверки).

9.3. Локальные корректировки политики

Локальные модули допускаются как временная мера на стенде или в контролируемом контуре. Генерация правил “в лоб” через `audit2allow` без анализа причины AVC не допускается, так как приводит к размыванию модели безопасности.

```
# Просмотр возможных allow-правил (только для анализа)
ausearch -m AVC,USER_AVC -ts recent | audit2allow -w

# Генерация локального модуля (только при наличии обоснования)
ausearch -m AVC,USER_AVC -ts recent | audit2allow -M local_fix
semodule -i local_fix.pp
```

10. Булевы переключатели (SELinux booleans)

Булевы параметры позволяют включать/отключать отдельные ветви политики без пересборки модулей. В минимальном профиле количество boolean ограничено составом установленных модулей.

```
# Просмотр
getsebool -a | head

# Изменение (пример методики)
setsebool -P some_boolean on
```

11. Эксплуатационные сценарии

11.1. Ввод SELinux enforcing в продуктив

1. Перевести систему в `permissive` (если не так), включить auditd, собрать AVC при типовых нагрузках.
2. Классифицировать AVC: (а) неверные метки, (б) недостающие правила политики, (в) ошибки конфигурации сервиса.
3. Исправить метки через `semanage fcontext`/`restorecon`, минимизировать локальные исключения.
4. Повторить прогон нагрузок, добиться отсутствия критических AVC.
5. Перевести систему в `enforcing` на стенде, выполнить регрессионные тесты.
6. Зафиксировать регламент: мониторинг AVC, правила изменения меток, порядок выпуска локальных модулей.

11.2. Восстановление после ошибочной маркировки

```
# 1) Временное снятие enforcement
setenforce 0
```

```
# 2) Восстановление контекстов
restorecon -Rv /etc /var /usr

# 3) Возврат enforcement после проверки
setenforce 1
```

Замечание. Массовый `restorecon` следует выполнять с пониманием профиля: нестандартные пути должны быть предварительно описаны в `semanage fcontext`, иначе они будут “перемечены” в значения по умолчанию.

12. Примечания для разработчиков и сопровождающих пакеты

Для системного ПО НАЙС.ОС SELinux рассматривается как часть эксплуатационного контракта пакета. При добавлении сервисов, демонов и агентов следует обеспечивать:

- корректную маркировку исполняемых файлов и каталогов данных (file contexts);
- предсказуемые доменные переходы при старте через systemd;
- минимально необходимые разрешения (principle of least privilege) без расширения до “unconfined”;
- регрессионные проверки в permissive и enforcing на типовых сценариях.

```
# Минимальный контроль при интеграции сервиса:
# 1) контексты
ls -Z /usr/sbin/mydaemon /var/lib/mydaemon

# 2) домен процесса
systemctl start mydaemon
ps -eZ | grep mydaemon

# 3) AVC
ausearch -m AVC,USER_AVC -ts recent | audit2why
```

13. Заключение

SELinux в НАЙС.ОС реализован как модульная refpolicy-политика профиля `default` в конфигурации `mcs`, оптимизированная под минимальную установку и контейнерный хост. Базовая эксплуатационная модель: сбор AVC в `permissive` с последующим управляемым переводом в `enforcing` после валидации меток и рабочих нагрузок.

Для систем с повышенными требованиями по ИБ рекомендуется формализовать профиль: роли пользователей, правила маркировки томов/данных, регламент обработки AVC и порядок изменения политики.

Особенности SELinux-политик в НАЙС.ОС: изменения на уровне selinux-policy

Профиль `SELINUXTYPE=default` в НАЙС.ОС ориентирован на минимальную, предсказуемую и воспроизводимую политику для хостов (включая сценарии контейнерного размещения) с включённой категоризацией MCS (до `1024` категорий). Ниже приведено техническое описание изменений, внесённых в исходники SELinux Reference Policy (refpolicy) на уровне патчей пакета `selinux-policy`.

Цель изменений: (1) уменьшить поверхность разрешений по умолчанию; (2) исключить неиспользуемые опциональные зависимости (MTA/SSSD и т.п.); (3) обеспечить корректную маркировку файловых путей, применяемых в НАЙС.ОС; (4) выделить отдельные домены для вспомогательных системных компонентов (например, генерации `motd`), чтобы исключить разрастание привилегий базовых доменов (например, `sshd_t`).

Патч	Затрагиваемые модули	Результат для профиля НАЙС.ОС
`0001` contrib/container	`policy/modules/contrib/container.*`	Сужение разрешений контейнерных доменов, удаление неиспользуемых опций, подготовка типа для data-хранилищ.
`0002` contrib/cron	`policy/modules/contrib/cron.*`	Исключение MTA-зависимостей и связанных меток/разрешений из cron-политики.
`0003` contrib/virt	`policy/modules/contrib/virt.te`	Разделение доменов виртуализации: снятие алиасинга `svirt_t` и `qemu_t`.
`0004` kernel/storage	`policy/modules/kernel/storage.fc`	Корректная маркировка `/dev/root` как дискового устройства для сценариев загрузки/монтирования.
`0005` roles/staff	`policy/modules/roles/staff.te`	Упрощение роли `staff_r`: удаление «десктопных»/прикладных опциональных расширений.

Патч	Затрагиваемые модули	Результат для профиля НАЙС.ОС
0006 roles/unprivuser	policy/modules/roles/unprivuser.te	Упрощение роли unprivuser_r: удаление опциональных расширений, нецелевых для базового профиля.
0007 motd domain	policy/modules/.../authlogin.*	Подготовка перехода на отдельный домен генерации motd, корректировка прав sshd_t.
0008 system/getty	policy/modules/system/getty.te	Добавление точечной capability для совместимости getty в профильной конфигурации.
0009 system/init + contrib/motd	policy/modules/system/init.*, новый policy/modules/contrib/motd.*	Ввод отдельного модуля motd и настройка разрешений init_t для профильных системных сценариев.
0010 system/logging	policy/modules/system/logging.fc	Дополнительный путь для журналов аудита: маркировка /var/opt/audit.
0011 system/modutils	policy/modules/system/modutils.fc	Маркировка конфигурации автозагрузки модулей ядра: /etc/modules-load.d.

Детализация по патчам

0001 — Контейнерные домены: сужение разрешений и удаление нецелевых опций

Из контейнерного модуля удалён переключатель (tunable), который предоставлял контейнерным доменам прямой доступ к объектам класса device_node. Тем самым устраняется «широкий» путь выдачи прав на устройства (включая потенциально чувствительные /dev/*) без явной модели доступа. Дополнительно удалены фрагменты, которые вводили расширения под SSSD и отдельные привилегированные варианты входа, а также убрана привязка к «unlabeled entry» для специального типа. Введён новый тип data_home_t (подготовка к отдельной разметке data-областей/каталогов данных).

```
- gen_tunable(container_use_devices, false)
```

```
- optional_policy(`
-   tunable_policy(`container_use_devices',`
-     allow container_domain device_node:...;
-   ')
- ')
+ type data_home_t;
```

Практический эффект для НАЙС.ОС: контейнерные домены в профиле `default` не получают «универсального» разрешения на работу с device nodes; доступ к устройствам должен задаваться адресно (через целевые типы и интерфейсы), что снижает риск эскалации при ошибках конфигурации контейнеров.

0002 — Cron: удаление MTA-ориентированных меток и разрешений

Из модуля cron удалены правила, которые связывали cron-домены с инфраструктурой MTA (маркировка spool/pid/tmp как MTA-контента и разрешения на отправку почты из cron). Это уменьшает число опциональных зависимостей политики и устраняет неиспользуемые разрешения в минимальном серверном профиле.

```
- optional_policy(`
-   mta_system_content(cron_spool_t)
-   mta_send_mail(crond_t, cron_spool_t)
- ')
- /var/spool/cron(/.*)?  gen_context(system_u:object_r:cron_spool_t,s0)
```

Практический эффект для НАЙС.ОС: cron остаётся функциональным для планирования заданий, но не получает дополнительных MTA-разрешений «по умолчанию», что согласуется с минимальной комплектацией и принципом отключения неиспользуемых путей взаимодействия.

0003 — Виртуализация: снятие typealias между `svirt_t` и `qemu_t`

Удалён алиасинг типа `svirt_t` к `qemu_t`. Это критично для разделения доменов: `svirt_t` применяется для изолированных гостевых контекстов (sVirt), тогда как `qemu_t` может использоваться в иных сценариях (включая хостовые процессы виртуализации). Исключение алиаса снижает вероятность некорректного «слипания» политик и неявного расширения прав.

```
- typealias qemu_t alias svirt_t;
```

0004 — Маркировка `/dev/root` как дискового устройства

Добавлено правило file contexts для пути `/dev/root` с типом `fixed_disk_device_t`. В практике загрузки (initramfs/early userspace) и монтирования корня данный путь может использоваться как указатель на корневое устройство. Корректная маркировка предотвращает отказы доступа при строгом режиме и упрощает совместимость с системными сценариями обнаружения/подключения блочных устройств.

```
+/dev/root   -c  gen_context(system_u:object_r:fixed_disk_device_t,s0)
```

0005 — Роль `staff_r`: устранение широких опциональных расширений

Из роли `staff_r` удалены большие блоки опциональных разрешений/ролей, ориентированных на широкий набор пользовательских и «десктопных» компонентов (мультимедиа, GUI-приложения, Java, браузеры, периферия и т.п.). Это делает роль «staff» управляемой и предсказуемой в серверном/инфраструктурном профиле: роль не включает дополнительные домены и не получает расширения, не относящиеся к целевому назначению профиля.

0006 — Роль `unprivuser_r`: упрощение профиля непривилегированного пользователя

Аналогично роли staff, из роли `unprivuser_r` удалены опциональные расширения, которые не требуются для минимального профиля и могут раздувать матрицу разрешений (дополнительные прикладные домены, вспомогательные роли). В результате поведение непривилегированного пользователя становится более детерминированным, а аудит — проще.

0007 — Интеграция генерации `motd`: подготовка выделенного домена и корректировка прав `sshd_t`

В рамках изменения логики генерации «сообщения дня»: (1) добавлены разрешения для `sshd_t` на операции с `alg_socket` (класс AF_ALG); (2) добавлен вызов интерфейса выполнения генератора motd из контекста `sshd_t`; (3) удалены устаревшие метки/типы для `/run/motd` и `/run/motd.d` из модуля, где они ранее описывались. В сумме это подготавливает переход на более корректную модель: генератор motd обслуживается отдельным доменом/модулем, а базовый домен `sshd_t` получает только минимально необходимую возможность инициировать запуск.

```
+ allow sshd_t self:alg_socket { create bind accept };
```

```
+ motdgen_exec(sshd_t);
- /run/motd(\.d)?(/.*)?  gen_context(...)
```

0008 — Getty: добавление capability для профильной совместимости

В модуль `getty` добавлено предоставление capability `sys_admin` для домена `getty_t` (условно, в рамках дистрибутивного профиля). Данное изменение применяется для устранения отказов доступа в сценариях инициализации/управления TTY, где отдельные операции требуют расширенной capability. Применение ограничивается доменом getty и не распространяется на прочие системные домены.

```
+ allow getty_t self:capability sys_admin;
```

0009 — Init + новый модуль `motd`: домен генератора motd и разрешения для init

Патч выполняет два класса изменений:

- **Добавление нового contrib-модуля `motd`**: вводятся типы `motd_t`, `motd_exec_t`, `motd_var_run_t`, file contexts для исполняемого файла генератора (`/usr/bin/motdgen`), а также контекст для runtime-каталогов (`/var/run/motd`, `/var/run/motd.d`). Определяется интерфейс выполнения (domain transition) для запуска генератора из других доменов. Это формирует отдельную «единицу» политики для генерации motd вместо распределения правил по общим модулям.
- **Изменения в `init_t`**: добавлены разрешения управления объектами `tmpfs` (директории/файлы/символические ссылки), разрешение на создание netlink-сокета класса `netlink_kobject_uevent_socket`, а также разрешение на подключение к syslog через `unix_stream_socket connectto`. Дополнительно добавлены опциональные интеграции для сервисов, которые могут запускаться и управляться init/systemd (включая конфигурации DNS/сетевых сервисов и отдельных подсистем).

```
+ policy_module(motd, 1.0)
+ type motd_t; type motd_exec_t; type motd_var_run_t;
+ /usr/bin/motdgen  --  gen_context(...:motd_exec_t,...)
+ allow init_t tmpfs_t:dir { create write add_name ... };
+ allow init_t syslogd_t:unix_stream_socket connectto;
```

Практический эффект для НАЙС.ОС: генерация motd отделена в самостоятельный домен с собственными типами и file contexts, что упрощает аудит и снижает риск накопления «побочных» разрешений в `sshd_t` и `init_t`.

0010 — Logging: дополнительная метка для каталога audit

В file contexts добавлена маркировка `/var/opt/audit` типом `auditd_log_t`. Это обеспечивает корректный доступ audit-подсистемы при использовании альтернативного размещения журналов (например, при политике разделения `/var` и `/var/opt` либо при особенностях layout).

```
+/var/opt/audit(/.*)?  gen_context(system_u:object_r:auditd_log_t,s0)
```

0011 — Modutils: маркировка каталога `/etc/modules-load.d`

В file contexts добавлена маркировка `/etc/modules-load.d` типом `modules_conf_t`. Это согласует SELinux-контексты с практикой автозагрузки модулей ядра через конфигурационные drop-in каталоги (в т.ч. при использовании systemd-юнитов для modules-load).

```
+/etc/modules-load\.d(/.*)?  gen_context(system_u:object_r:modules_conf_t,s0)
```

Особенности SELinux-политики в НАЙС.ОС: корректировки 0012–0021

Ниже описаны изменения, внесённые патчами 0012–0021 в набор модулей refpolicy, применяемый в НАЙС.ОС. Назначение изменений — устранение типовых AVC-отказов в системных доменах (systemd, сетевой стек, udev, управление пользователями, LVM, iptables), а также устранение конфликтов file context'ов при сборке минимального профиля SELinux.

Контекст применения: политика в НАЙС.ОС ориентирована на профиль контейнерного хоста с MCS-разделением, при этом режим SELinux по умолчанию — `permissive`, а тип политики — `default`. Валидация изменений выполняется по принципу: «разрешаем только то, что необходимо для штатной работы системных компонентов», с фиксацией областей доступа по типам и доменам.

Патч 0012: systemd — расширение прав для

timedated/resolved/modules-load

Патч корректирует правила для доменов, связанных с systemd, с целью устранения отказов при работе служб времени/таймзоны, резолвера и загрузки модулей. Изменения относятся к модулю `policy/modules/system/systemd.*`.

Состав изменений

- **Уточнение доменной модели**: удаляется устаревшая/конфликтная привязка (alias) домена timedated к «пользовательскому» типу, чтобы исключить пересечение контекстов и некорректные переходы.
- **systemd-timedated**: добавляются права чтения файлов из пространства `/run` резолвера (`systemd_resolved_var_run_t`), а также разрешение на привязку UDP к generic-node (для штатной сетевой активности timedated).
- **systemd-resolved**: добавляется разрешение на отправку сообщений в подсистему логирования через datagram, а также разрешения на создание и использование сокетов (UDP/TCP) и привязку к
- **systemd-modules-load**: добавляется разрешение отправки лог-сообщений datagram-каналом (устранение отказов при старте).

Практический эффект

- Снижение количества AVC по доменам `systemd_timedated_t` и `systemd_resolved_t` при штатной работе systemd.
- Стабилизация сетевых операций timedated (в т.ч. взаимодействие с резолвером через runtime-файлы).
- Предсказуемая доставка событий в логирование для systemd-юнитов ранней загрузки.

Фрагменты изменений (для аудита)

```
--- a/policy/modules/system/systemd.te
+++ b/policy/modules/system/systemd.te
@@
+read_files_pattern(systemd_timedated_t, systemd_resolved_var_run_t, systemd_resolved_var_run_t)
+corenet_udp_bind_generic_node(systemd_timedated_t)
+
+logging_dgram_send(systemd_resolved_t)
+corenet_udp_bind_generic_node(systemd_resolved_t)
+corenet_tcp_bind_generic_node(systemd_resolved_t)
+allow systemd_resolved_t self:udp_socket create_socket_perms;
+allow systemd_resolved_t self:tcp_socket create_stream_socket_perms;
```

```
+
+logging_dgram_send(systemd_modules_load_t)
```

Патч 0013: sysnetwork — расширение file context’ов и интерфейсов чтения конфигурации сети

Патч расширяет покрытие сетевой конфигурации на уровне file context’ов и уточняет интерфейсные макросы доступа к этим объектам. Это необходимо для корректной маркировки и чтения конфигурации сетевых скриптов и runtime-объектов systemd-resolved/systemd-networkd в минимальном профиле.

Состав изменений

- **File context’ы**: добавляются/уточняются правила маркировки для: `/etc/sysconfig/network`, `/etc/sysconfig/networking`, `/etc/sysconfig/network-scripts/`, а также runtime-каталогов и файлов systemd: `/var/run/systemd/network(/.*)?`, `/var/run/systemd/resolve/resolv.conf`. Тип назначения — `net_conf_t`.
- **Интерфейс sysnet_read_config()**: добавляются разрешения поиска/листа каталогов и доступа к PID-структурам init-процессов, необходимые для корректной работы инструментов, которые читают конфигурацию сети в системной инициализации.

Практический эффект

- Стабильная маркировка сетевой конфигурации в /etc/sysconfig и runtime-объектов systemd без «unknown/restorecon drift».
- Снижение AVC при чтении конфигурации в доменах, использующих стандартные sysnetwork-интерфейсы.
- Устранение расхождений, когда resolv.conf в runtime-дереве systemd выпадал из ожидаемого типа.

Фрагменты изменений (для аудита)

```
--- a/policy/modules/system/sysnetwork.fc
+++ b/policy/modules/system/sysnetwork.fc
@@
+/etc/sysconfig/networking?    -- gen_context(system_u:object_r:net_conf_t,s0)
+/etc/sysconfig/network-scripts(/.*)? -- gen_context(system_u:object_r:net_conf_t,s0)
+/var/run/systemd/network(/.*)? -- gen_context(system_u:object_r:net_conf_t,s0)
```

```
+/var/run/systemd/resolve/resolv\.conf -- gen_context(system_u:object_r:net_conf_t,s0)

--- a/policy/modules/system/sysnetwork.if
+++ b/policy/modules/system/sysnetwork.if
@@
+files_search_all_pids($1)
+init_search_pid_dirs($1)
+list_dirs_pattern($1, net_conf_t, net_conf_t)
```

Патч 0014: udev

- **Цель:** устранение отказов SELinux при ранней инициализации устройств и обработке событий udev.
- **Зона воздействия:** домены и типы, связанные с `udev`/`udevd`, доступ к runtime-каталогам, взаимодействие с устройствами и служебными файлами.
- **Ожидаемый эффект:** снижение AVC при создании/модификации объектов в `/run`, обработке правил udev и взаимодействии с sysfs/devtmpfs.

Патч 0015: userdomain

- **Цель:** корректировка правил для пользовательских доменов и переходов ролей в минимальном профиле.
- **Зона воздействия:** интерфейсы userdomain, разрешения на типовые операции login/session, доступ к пользовательским директориям/TTY и базовым системным сервисам.
- **Ожидаемый эффект:** снижение AVC при интерактивной работе пользователей и при ограниченных ролях (включая staff/sysadm-профили).

Патч 0016: admin_usermanage

- **Цель:** устранение отказов при управлении пользователями и группами штатными инструментами (useradd/usermod/groupadd и т.п.).
- **Зона воздействия:** домен администрирования учётных записей, доступ к `/etc/passwd`, `/etc/shadow`, `/etc/group` и соответствующим временным/lock-файлам.
- **Ожидаемый эффект:** предсказуемое выполнение операций управления учётными записями под административными ролями без перевода системы в unconfined.

Патч 0017: fstool

- **Цель:** корректировка политик утилит файловых систем (форматирование/проверка/обслуживание ФС) в минимальной сборке.
- **Зона воздействия:** доступ к блочным устройствам, метаданным ФС, системным каталогам обслуживания, взаимодействие с udev/systemd при операциях монтирования/проверки.
- **Ожидаемый эффект:** снижение AVC при обслуживании дисков (в т.ч. в инфраструктурных сценариях: хранилища, VM-хосты, контейнерные узлы).

Патч 0018: iptables — разрешение kernel_t на fifo_file

- **Цель:** устранение отказов, связанных с использованием FIFO-объектов в цепочке netfilter/iptables при взаимодействии с ядром.
- **Зона воздействия:** правила allow для `kernel_t` в отношении класса `fifo_file` в контексте iptables-модуля.
- **Ожидаемый эффект:** снижение AVC при применении/перезагрузке правил фильтрации в рантайме (особенно на минимальных хостах).

Патч 0019: authlogin — переходы/интерфейсы для shadow/group

- **Цель:** корректировка переходов доменов/прав доступа при операциях аутентификации и проверке членства в привилегированных группах (shadow).
- **Зона воздействия:** интерфейсы authlogin и связанные домены login, чтение/проверка объектов групп/теневых файлов с минимально необходимыми правами.
- **Ожидаемый эффект:** снижение AVC при логине и при выполнении операций, которые опираются на корректное чтение group/shadow-структур.

Патч 0020: lvm — разрешение перехода lvm_t unconfined_t

Патч добавляет разрешение для домена LVM на переход (domain transition) в `unconfined_t`. Это точечная «аварийная» совместимость для сценариев, где LVM-операции выполняются в окружениях/скриптах, требующих выхода из строгого домена LVM.

Состав изменений

- Добавляется правило `allow lvm_t unconfined_t:process transition;`.

Практический эффект и ограничения

- **Эффект:** предотвращение блокировок в административных сценариях, где LVM вызывается как часть «широких» процедур обслуживания.
- **Ограничение:** переход в `unconfined_t` снижает строгость модели. Рекомендуется применять в сочетании с ролевой моделью (staff/sysadm) и аудитом, либо ограничивать только конкретным путём запуска/типом entrypoint (если требуется ужесточение).

```
--- a/policy/modules/system/lvm.te
+++ b/policy/modules/system/lvm.te
@@
+allow lvm_t unconfined_t:process transition;
```

Патч 0021: fix-fc-conflicts — устранение конфликтов file context’ов

Патч устраняет конфликты file context’ов в наборах правил маркировки. Типовая задача — исключить ситуации, когда два различных шаблона путей сопоставляются одному и тому же объекту с разными типами (или когда один шаблон перекрывает другой некорректно), что приводит к непредсказуемому результату при `restorecon`/`setfiles`. {index=10}

Практический эффект

- Устойчивое применение маркировок в минимальном профиле без «скачущих» типов при пересборке/обновлениях.
- Снижение числа ошибок компоновки policy при генерации итогового `file_contexts`.
- Упрощение сопровождения минимального профиля SELinux для контейнерных хостов.

Операционные примечания (НАЙС.ОС)

Контроль результата (AVC и типы)

```
# Журнал AVC (auditd / journal):
ausearch -m avc -ts recent

# Проверка контекстов критических путей:
ls -Z /etc/sysconfig /etc/sysconfig/network-scripts 2>/dev/null || true
ls -Z /var/run/systemd 2>/dev/null || true

# Верификация загрузки политики:
sestatus
```

3.X. Особенности SELinux-политики НАЙС.ОС: пакет исправлений AVC и расширение совместимости (патчи 0022–0039)

Настоящий раздел описывает изменения SELinux-политики НАЙС.ОС, внесённые набором патчей 0022–0039. Цель изменений: устранение отказов доступа (AVC), выявленных при функциональном тестировании системных пакетов и при развёртывании контейнерной инфраструктуры, а также обеспечение корректной работы отдельных компонентов systemd, SSH, журналирования и подсистем входа.

Примечание. Изменения описаны в терминах SELinux refpolicy: домены (например, `sshd_t`, `systemd_gpt_generator_t`), типы объектов (например, `sysctl_kernel_t`, `user_tmp_t`) и классы (например, `file`, `dir`, `capability`).

3.X.1. Сводка патчей и область влияния

Патч	Назначение	Ключевые затрагиваемые компоненты
`0022`	Комплексное устранение AVC по результатам тестов пакетов	authlogin, init, iptables, logging, libraries, ssh, userdomain и др.
`0023–0027`	Контейнерная совместимость (Kubernetes/containerd): чтение/наблюдение конфигов и доступ к состоянию	container, kubernetes, systemd generators

Патч	Назначение	Ключевые затрагиваемые компоненты
0028–0034	Точечные исправления SSH, syslog и входа (getty/local login)	sshd, ssh-keygen, syslog, getty, local login
0035–0036	Коррекция правил аутентификации (pwhistory) и снятие конфликтующих neverallow для контейнерных сценариев	PAM/authlogin, container policy constraints
0037–0038	Исправления отказов D-Bus и systemd-logind	dbus, logind
0026, 0030–0031, 0039	systemd generators/userdbd: устранение отказов по capabilities и доступу к системным путям	systemd_gpt_generator_t, systemd_gpt_generator_exec_t, systemd_userdbd_t и др.

3.X.2. Патч 0022: комплексное устранение AVC по результатам тестирования

Патч 0022 вводит набор точечных разрешений и корректировок интерфейсов для устранения отказов доступа, обнаруженных в тестовых сценариях. Изменения носят межмодульный характер: затрагиваются правила для аутентификации, инициализации, сетевых утилит, библиотек, графической подсистемы и доменов пользователей.

- **authlogin:** добавлены разрешения, обеспечивающие корректный доступ к динамическому загрузчику и типовым библиотечным файлам в доменах, задействованных в проверке паролей и вспомогательных проверках PAM/аутентификации.
- **init/userdomain:** расширены разрешения для сервисных доменов на создание/обслуживание временных объектов, а также на отдельные переходы/взаимодействия, требуемые для корректного жизненного цикла процессов при старте системы и при выполнении системных задач.
- **iptables/logging/libraries:** добавлены правила, устраняющие отказы чтения библиотечных путей и вспомогательных файлов, необходимых утилитам при старте и при обработке событий журналирования.

- **ssh:** исправлены отказы для вспомогательных действий (включая операции над отдельными объектами `/etc` и системных настроек), а также введены дополнительные разрешения для типовых сценариев ключевой инфраструктуры.

Практический результат. Патч 0022 снижает уровень “шума” AVC при включённом SELinux (в т.ч. в permissive-режиме), и обеспечивает предсказуемость поведения базовых системных служб и утилит без локальных “ручных” модулей (custom policy).

3.X.3. Патчи 0023–0027: контейнерные сценарии (Kubernetes/containerd)

3.X.3.1. Патч 0023: доступ контейнерного рантайма к Kubernetes-артефактам

Патч 0023 вводит правило/интерфейс, позволяющий контейнерному стеку корректно читать файлы, маркированные типом Kubernetes, из доменов контейнерной подсистемы. Это устраняет отказы при реальной эксплуатации (например, чтение конфигурации/манифестов и служебных файлов).

3.X.3.2. Патч 0024: разрешение “watch” для `bin_t` (inotify) в контейнерных доменах

Патч 0024 устраняет отказы при использовании механизма наблюдения за файлами (inotify) контейнерными процессами: разрешаются операции класса `watch` в отношении путей с типом `bin_t`. Это характерно для компонентов, которые отслеживают изменения бинарников/директорий в процессе запуска и обслуживания.

3.X.3.3. Патч 0025: устранение отказов etcd (состояние в `/var/lib`)

Патч 0025 добавляет разрешения для контейнерного домена на операции управления данными в хранилище состояния (каталоги/файлы под типами `var_lib_t`). Исправляются отказы на создание, переименование и mmap/отображение файлов, характерные для etcd при штатной работе в контейнере.

3.X.3.4. Патч 0026: systemd gpt generator (capability sys_admin)

Патч 0026 разрешает домену генератора GPT (systemd) использовать capability `sys_admin`. Это требуется для выполнения операций, связанных с обработкой GPT/разделов в сценариях развёртывания контейнерной инфраструктуры и раннего старта.

3.X.3.5. Патч 0027: устранение отказов “watch” для Kubernetes-файлов

Патч 0027 расширяет контейнерную политику для корректного “наблюдения” (inotify watch) за объектами Kubernetes-типа (в т.ч. в домене контейнерного рантайма). Это устраняет отказы при ожидании изменений конфигурации/статуса.

Замечание по модели угроз. Разрешения “watch” и расширение доступа к `/var/lib` вводятся адресно для доменов контейнерной подсистемы. При переводе системы в enforcing-режим рекомендуется подтвердить, что используемые контейнерные профили и метки томов/каталогов соответствуют принятой схеме разметки (labeling).

3.X.4. Патчи 0028–0034: SSH, syslog и вход в систему

3.X.4.1. Патч 0028: исправления отказов для SSH (sshd и вспомогательные утилиты)

Патч 0028 вводит дополнительные разрешения для домена SSH-сервера и вспомогательных операций (в т.ч. чтение отдельных kernel sysctl-объектов и корректное завершение некоторых файловых операций), устраняя типовые AVC в ходе запуска и обслуживания SSH.

3.X.4.2. Патч 0029: syslog и работа с временными сокетами

Патч 0029 устраняет отказы syslog при работе с объектами пользовательского временного каталога: добавляются разрешения на поиск в каталоге и создание/удаление сокет-файлов в типе `user_tmp_t`, что требуется для отдельных режимов взаимодействия/транспортов логирования.

3.X.4.3. Патчи 0032 и 0033: getty/local login (capability2 checkpoint_restore)

Патч 0032 добавляет `capability2 checkpoint_restore` домену `getty_t`, а патч 0033 — домену `local_login_t`. Это устраняет отказы в окружениях, где соответствующие процессы используют функциональность, требующую указанной capability (в частности, при взаимодействиях с современными механизмами управления процессами/сеансами).

3.X.4.4. Патч 0034: разрешение AF_ALG (`alg_socket`) для sshd

Патч 0034 разрешает домену `sshd_t` операции с сокетами класса `alg_socket` (Linux AF_ALG). Это обеспечивает совместимость SSH с режимами, использующими криптографический API ядра для отдельных криптоопераций.

3.X.5. Патчи 0030–0031 и 0039: systemd generators/userdbd и связанные отказы

3.X.5.1. Патч 0030: systemd gpt generator (дублирующее/дополняющее разрешение sys_admin)

Патч 0030 дополнительно фиксирует отказы домена генератора GPT, разрешая capability `sys_admin`. В комплексе с патчем 0026 это закрывает вариативность доменов/исполняемых меток генераторов в разных сценариях старта.

3.X.5.2. Патч 0031: systemd-userdbd (capability audit_write)

Патч 0031 разрешает домену `systemd_userdbd_t` capability `audit_write`, устраняя отказы при записи событий в аудит/журнал в рамках штатной работы userdbd.

3.X.5.3. Патч 0039: комплексные исправления для актуальных наборов SELinux-правил

Патч 0039 является агрегирующим: добавляет серию разрешений и корректировок, ориентированных на устранение множественных отказов в современном окружении (в т.ч. вокруг генераторов systemd, работы с `/var` и служебных файловых объектов), а также ослабляет отдельные конфликтующие ограничения в “neverallow”-слое, если они мешают достижению функциональной совместимости контейнерных и системных сценариев.

- Вводятся дополнительные разрешения на создание/чтение/управление отдельными объектами под `/var` для generator-доменов systemd (типовые классы: `file`, `dir`, `sock_file`, `fifo_file`) при сохранении адресности по доменам.
- Корректируются интерфейсы systemd (if-файлы), чтобы типовые шаблоны доменов корректно “видели” нужные пути и могли работать без локальных исключений.
- Вводится ограниченное снятие конфликтующих neverallow для случаев, где контейнерная модель требует легитимных операций (цель — обеспечить работоспособность без бесконечных локальных модулей).

3.X.6. Патчи 0035–0038: PAM/pwhistory, контейнерные

neverallow, D-Bus и logind

3.X.6.1. Патч 0035: устранение отказов при использовании pwhistory (смена пароля)

Патч 0035 корректирует правила модуля аутентификации, устраняя AVC, возникающие при смене пароля в сценариях, где активен PAM-модуль истории паролей (pwhistory). Практически это означает, что операции обновления файла истории/метаданных паролей выполняются в рамках ожидаемого домена без ручных исключений.

3.X.6.2. Патч 0036: отключение части neverallow-ограничений, конфликтующих с контейнерными сценариями

Патч 0036 адресно ослабляет отдельные neverallow-ограничения, которые препятствуют работе контейнерного стека в реальных условиях эксплуатации (типовые причины: необходимость inotify-наблюдения, работа с state-директориями, взаимодействие с вспомогательными объектами IPC). Изменение предназначено для контейнерного профиля и применяется как часть согласованной политики контейнерного хоста.

3.X.6.3. Патч 0037: исправление отказов D-Bus

Патч 0037 добавляет минимально необходимое разрешение для домена системного D-Bus, устраняя отказ доступа, проявляющийся при запуске/работе системной шины в “минимальном” окружении.

3.X.6.4. Патч 0038: исправление отказов systemd-logind

Патч 0038 добавляет разрешение для домена `systemd_logind_t` на выполнение операции `getattr` в отношении символьных ссылок, что устраняет отказы при типовых обращениях logind к файловым объектам (в т.ч. при обработке пользовательских сессий и устройств).

3.X.7. Методика проверки после применения патчей

Контроль корректности рекомендуется выполнять в два этапа: (1) анализ AVC в permissive-режиме, (2) перевод в enforcing на стенде и повторный прогон сценариев.

```
# 1) Сбор и анализ AVC (SELinux permissive)
```

```
ausearch -m AVC,USER_AVC -ts recent | audit2why
ausearch -m AVC,USER_AVC -ts recent | audit2allow -w

# 2) Проверка контекстов и меток
getenforce
sestatus
ls -Z /run /var /etc | head

# 3) Точечные проверки по доменам/типам (пример)
sesearch -A -s sshd_t -t sysctl_kernel_t
sesearch -A -s systemd_gpt_generator_t -c capability
```

Критерий завершения работ. Для целевых сценариев (контейнерный узел, systemd generators, SSH, журналирование, вход) количество AVC должно быть сведено к нулю или к заранее документированному перечню допустимых событий (исключения допускаются только при наличии обоснования и мер компенсации).

Итоговое влияние на профиль SELinux в НАЙС.ОС

- Профиль `default` получает более жёсткую базовую модель для контейнерных доменов: исключён «тумблер» на доступ к устройствам, убраны нецелевые опции (SSSD/специальные entrypoints) и введён отдельный тип под data-области.
- Система ролей (`staff_r`, `unprivuser_r`) очищена от прикладных расширений, что снижает сложность матрицы разрешений и повышает предсказуемость поведения в серверном профиле.
- Логика генерации `motd` формализована через отдельный модуль/домены и корректные file contexts; изменения уменьшают давление на домены `sshd_t` и упрощают аудит цепочек выполнения.
- Приведены в порядок file contexts для путей, критичных для загрузки и эксплуатации: `/dev/root`, `/var/opt/audit`, `/etc/modules-load.d`.