Действия после установки НАЙС.ОС

НАЙС.ОС: действия после установки системы

Документ устанавливает порядок первичной настройки после установки НАЙС.ОС (минимальный профиль). Термины «должен», «следует», «допускается», «не допускается» применяются в нормативном смысле.
Примечание. Базовая установка НАЙС.ОС ориентирована на минимальную атакуемую поверхность: создаётся только пользователь `root`, вход `root` по SSH запрещён, политика `iptables` по умолчанию — `DROP` для `INPUT`/`FORWARD`/`OUTPUT` с явными разрешающими правилами, профиль усиления ядра задан через `sysctl`, Docker установлен по умолчанию и является частью платформы (удаление Docker не допускается).
Содержание

---

1. Общие положения и исходное состояние (baseline)

После установки НАЙС.ОС по умолчанию существует только пользователь `root`. Вход

root по SSH запрещён настройкой сервиса SSH. Администратор должен создать обычного пользователя и выполнять администрирование от его имени, повышая привилегии только по необходимости.

Примечание. Принцип НАЙС.ОС: «запрещено всё, что не разрешено явно». Открытие портов и включение маршрутизации выполняется только по назначению узла и фиксируется воспроизводимыми настройками.

---

2. Создание обычного пользователя (обязательно)

Администратор должен создать обычного пользователя, задать пароль (при необходимости), подготовить SSH-ключи и обеспечить возможность входа по SSH под этим пользователем.

Предупреждение. При удалённом доступе не изменяйте правила firewall до подтверждения, что вход по SSH под обычным пользователем работает. В противном случае допускается потеря удалённого доступа до восстановления через консоль гипервизора/serial/VNC.

2.1. Создание пользователя

```
# Пример: имя пользователя admin
USER="admin"

# Создать пользователя с домашним каталогом и оболочкой bash
useradd -m -s /bin/bash "$USER"

# Задать пароль (если парольная аутентификация используется политикой)
passwd "$USER"
```

2.2. Настройка SSH-ключей (рекомендуется)

```
USER="admin"
HOME_DIR="/home/$USER"

# Каталог .ssh
install -d -m 700 -o "$USER" -g "$USER" "$HOME_DIR/.ssh"

# Добавьте ТОЛЬКО публичный ключ (пример формата ed25519)
cat >> "$HOME_DIR/.ssh/authorized_keys" <<'EOF'
```

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAI... user@laptop
EOF

chown "$USER:$USER" "$HOME_DIR/.ssh/authorized_keys"
chmod 600 "$HOME_DIR/.ssh/authorized_keys"
```

Примечание. Предпочтительно использовать ключи Ed25519 и, при допустимости политикой, отключать парольную аутентификацию по SSH.

---

3. Назначение административных прав (sudo / wheel)

Следует применять модель «обычный пользователь + повышение привилегий через `sudo`». Постоянная работа в `root` не допускается, кроме случаев аварийного восстановления.

3.1. Добавление пользователя в группу администраторов

```
USER="admin"

# Типовой вариант для RPM-подобных систем: группа wheel
usermod -aG wheel "$USER"
```

3.2. Разрешение sudo для wheel (через drop-in файл)

Настройку следует выполнять через отдельный файл в `/etc/sudoers.d` или через `visudo`. Редактирование `/etc/sudoers` напрямую без проверки синтаксиса не допускается.

```
cat > /etc/sudoers.d/10-wheel <<'EOF'
%wheel ALL=(ALL) ALL
EOF

chmod 0440 /etc/sudoers.d/10-wheel
```

---

4. SSH: доступ только под обычным пользователем

В НАЙС.ОС вход `root` по SSH запрещён. Администрирование выполняется по схеме: вход под обычным пользователем выполнение команд через `sudo`.

4.1. Проверка эффективных параметров SSH

```
sshd -T | egrep -i
'permitrootlogin|passwordauthentication|pubkeyauthentication|allowusers|allowgroups'
```

4.2. Пример целевой конфигурации (не копировать без оценки рисков)

```
# /etc/ssh/sshd_config (пример ключевых параметров)

PermitRootLogin no
PubkeyAuthentication yes

# При готовых ключах и допустимости политикой:
PasswordAuthentication no

# Ограничить доступ (пример)
AllowUsers admin

LoginGraceTime 30
MaxAuthTries 3
```

4.3. Применение изменений

```
# Проверка конфигурации
sshd -t

# Применение
systemctl restart sshd

# Контроль статуса
systemctl --no-pager --full status sshd
```

Примечание. Запрет root-логина снижает эффективность массовых атак (перебор паролей/credential stuffing) и уменьшает последствия компрометации учётных

данных.

---

5. Firewall: iptables (default deny)

В НАЙС.ОС политики цепочек `INPUT`, `FORWARD`, `OUTPUT` по умолчанию установлены в `DROP`, после чего добавляются явные разрешающие правила. Администратор должен открывать порты строго по назначению сервиса.

5.1. Просмотр текущих правил

```
iptables -L -n -v --line-numbers
iptables -S
```

5.2. Смысл baseline (кратко)

- `RELATED,ESTABLISHED` — разрешение возвратного трафика для уже установленных соединений.
- `tcp dpt:22` — разрешение SSH для административного доступа.

```
# Пример фрагмента вывода (счётчики зависят от реальной нагрузки)
# INPUT:
# 6371K  28G  ACCEPT  all  --  * *  0.0.0.0/0  0.0.0.0/0  ctstate RELATED,ESTABLISHED
# 3530   393K ACCEPT  tcp  --  * *  0.0.0.0/0  0.0.0.0/0  tcp dpt:22
#
# OUTPUT:
# 5748K  788M ACCEPT  all  --  * *  0.0.0.0/0  0.0.0.0/0
```

5.3. Пример: открытие порта 443/tcp (HTTPS)

Перед открытием порта следует убедиться, что сервис действительно слушает порт, и что открытие соответствует политике доступа. Правило следует добавлять до финальных запрещающих правил (порядок правил критичен).

```
# Разрешить входящий HTTPS (пример вставки на позицию 3)
iptables -I INPUT 3 -p tcp --dport 443 -m conntrack --ctstate NEW -j ACCEPT

# Проверить порядок
```

```
iptables -L INPUT -n -v --line-numbers
```

5.4. Персистентность правил

После изменения правил администратор должен обеспечить их сохранение при перезагрузке (способ зависит от сборки/профиля НАЙС.ОС). Ниже приведён типовой подход для систем, где используется сохранение правил в файл:

```
iptables-save > /etc/systemd/scripts/ip4save

# Если в системе предусмотрен сервис восстановления правил:
systemctl enable --now iptables || true
```

Предупреждение. Docker управляет частью цепочек и таблиц iptables (NAT/форвардинг). «Очистка» iptables без учёта docker-цепочек может нарушить работу контейнеров и сетевых политик. При настройках следует учитывать цепочки `DOCKER` и `DOCKER-USER`.

---

6. Docker: контроль состояния и модель доступа

Docker установлен по умолчанию и является частью платформы. Удаление Docker не допускается. Администратор должен контролировать состояние сервиса и осознанно управлять доступом пользователей к Docker.

6.1. Контроль состояния

```
systemctl --no-pager --full status docker
docker version
docker info
```

6.2. Доступ пользователя к Docker (опционально)

```
USER="admin"
usermod -aG docker "$USER"

# Перелогиниться для применения групп
```

```
id "$USER"
```

Предупреждение. Доступ к Docker (членство в группе `docker`) фактически эквивалентен административным возможностям на хосте (через запуск привилегированных контейнеров, монтирование файловой системы хоста и т.п.). В защищённых контурах такой доступ должен быть ограничен и аудируем.

---

7. Sysctl hardening: контроль и назначение параметров

В НАЙС.ОС используется профиль усиления безопасности через файл `/etc/sysctl.d/50-security-hardening.conf`. Администратор должен контролировать фактические значения и изменять их только по назначению роли узла.

7.1. Просмотр конфигурации и значений

```
sed -n '1,200p' /etc/sysctl.d/50-security-hardening.conf

sysctl kernel.randomize_va_space kernel.kptr_restrict kernel.dmesg_restrict kernel.sysrq
sysctl net.ipv4.ip_forward net.ipv6.conf.all.forwarding
sysctl fs.suid_dumpable kernel.yama.ptrace_scope
sysctl net.ipv4.conf.all.rp_filter net.ipv4.conf.default.rp_filter
sysctl net.ipv4.tcp_syncookies net.ipv4.tcp_max_syn_backlog net.ipv4.tcp_tw_reuse
```

7.2. Применение изменений

```
sysctl --system
```

Примечание. Цель hardening-профиля: ограничение утечек информации, усложнение эксплуатации уязвимостей, отключение редко используемых и потенциально опасных возможностей (redirect/source routing/маршрутизация по умолчанию), повышение устойчивости сетевого стека.

---

8. Контроль выполнения (минимальный чек-лист)

```
# 1) Пользователь существует и в нужных группах
id admin

# 2) sudo работает (если включали)
sudo -n true || sudo -v

# 3) SSH: root запрещён, вход под admin разрешён
sshd -T | grep -i permitrootlogin

# 4) Firewall соответствует baseline
iptables -L -n -v --line-numbers

# 5) sysctl применён
sysctl kernel.randomize_va_space kernel.kptr_restrict kernel.dmesg_restrict kernel.sysrq
sysctl net.ipv4.ip_forward net.ipv6.conf.all.forwarding
sysctl fs.suid_dumpable kernel.yama.ptrace_scope

# 6) Docker активен (если используется)
systemctl is-active docker
docker ps
```

---

9. НА реальном железе: обнаружение и установка прошивок

При установке НАЙС.ОС на физическое оборудование администратор должен убедиться, что система распознала устройства и при необходимости установлены требуемые прошивки (firmware). Для выборочной установки прошивок применяется инструмент `niceos-detect-firmware`.

Примечание. Минимальный образ содержит базовый набор прошивок, достаточный для запуска на широком наборе оборудования. Полный набор прошивок намеренно не включается в базовую поставку для контроля объёма и состава системы.

9.1. Порядок выполнения

1. Откройте терминал под обычным пользователем.
2. Выполните команду запуска инструмента.
3. Проверьте список обнаруженного оборудования и рекомендованных пакетов

прошивок.

4. Подтвердите установку (при необходимости) и выполните перезагрузку.

```
sudo niceos-detect-firmware

# После установки рекомендованных пакетов:
sudo reboot
```

9.2. Пример вывода (иллюстративно)

```
Detected hardware inventory:
--------------------------------
PCI Devices:
 - 00:14.3 Network controller: Intel Corporation Wireless-AC 9560
 - 01:00.0 VGA compatible controller: NVIDIA Corporation TU106M [GeForce RTX 2070 Mobile]

USB Devices:
 - Intel Corp. Bluetooth wireless interface
 - Logitech USB Optical Mouse

Kernel Modules Loaded:
 - iwlwifi
 - snd_hda_intel
 - nvidia

Recommended firmware packages:
 - linux-firmware-intel
 - linux-firmware-nvidia
 - linux-firmware-whence

Proceed with installation? [y/N]: y
Installing: linux-firmware-intel linux-firmware-nvidia linux-firmware-whence
```

---

10. Docker и доступ контейнеров в интернет при FORWARD=DROP

В НАЙС.ОС политика цепочки `FORWARD` по умолчанию — `DROP`. При таком baseline контейнеры могут не иметь доступа в интернет, поскольку транзитный трафик между `docker bridge` (например, `docker0`) и внешним интерфейсом блокируется. Для обеспечения доступа контейнеров в сеть администратор должен либо разрешить `FORWARD`, либо добавить точечные правила для Docker, сохранив модель `default deny`.

10.1. Диагностика

```
iptables -L -n -v --line-numbers
iptables -S FORWARD
iptables -S | egrep 'DOCKER|DOCKER-USER|FORWARD' || true
```

10.2. Вариант A (упрощённый): разрешить весь FORWARD

```
iptables -P FORWARD ACCEPT
```

Предупреждение. Политика `FORWARD ACCEPT` снижает строгость сегментации и делает хост «проходным» для широкого класса пересылок пакетов. В средах с повышенными требованиями следует применять вариант B.

10.3. Вариант B (рекомендуемый): оставить FORWARD=DROP и добавить точечные правила

Примечание. В примере внешний интерфейс указан как `eth0`. Фактическое имя следует определить по умолчальному маршруту: `ip route | grep default`.

```
# Определить внешний интерфейс
ip route | grep default

# Пример: внешний интерфейс eth0, docker bridge docker0

# 1) Базовая политика сохраняется
iptables -P FORWARD DROP

# 2) Разрешить возвратный трафик установленных соединений
iptables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

# 3) Разрешить выход контейнеров наружу
iptables -A FORWARD -i docker0 -o eth0 -m conntrack --ctstate NEW -j ACCEPT

# 4) Разрешить обратный трафик к docker0 (наглядность; ответы обычно покрываются RELATED,ESTABLISHED)
iptables -A FORWARD -i eth0 -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

10.4. Проверка результата

```
docker run --rm -it alpine:latest sh -lc 'ip a; nslookup example.com 2>/dev/null || true; wget -qO- https://example.com | head'

iptables -L FORWARD -n -v --line-numbers
```

10.5. Персистентность

После настройки администратор должен обеспечить сохранение правил при перезагрузке тем способом, который принят в текущей сборке НАЙС.ОС (например, через `iptables-save/restore` или systemd-юнит восстановления правил).

---

11. Типовые ошибки и корректирующие действия

11.1. Ошибка: «Не могу войти по SSH после установки»

Причина типовая: попытка входа по SSH под `root` при запрете root-логина. Корректирующее действие: создать пользователя через консоль доступа, настроить ключи и входить по SSH под обычным пользователем.

```
# На консоли под root:
useradd -m -s /bin/bash admin
passwd admin
usermod -aG wheel admin

# Добавить authorized_keys как в разделе 2.2
# Далее подключаться по SSH как admin
```

11.2. Ошибка: «Открыл порт, но сервис недоступен»

Следует проверить три уровня: (1) сервис слушает порт, (2) firewall разрешает, (3) внешний периметр (SG/ACL/маршрутизация в облаке) не блокирует доступ.

```
# 1) Сервис слушает порт?
```

```
ss -lntup | grep -E ':(80|443)\b' || true

# 2) Firewall: есть разрешающее правило и оно выше DROP?
iptables -L INPUT -n -v --line-numbers | sed -n '1,140p'

# 3) Локальная проверка на хосте
curl -vk https://127.0.0.1/ || true
```

11.3. Ошибка: «Отключил hardening, чтобы заработало»

Предупреждение. Ослабление hardening «для устранения симптомов» без оформления профиля роли узла не допускается. Если сценарию требуется маршрутизация/форвардинг/изменение сетевых ограничений, следует оформлять отдельный профиль (например, «шлюз/VPN/NAT») с минимально необходимыми отклонениями от baseline и с воспроизводимой фиксацией в конфигурации.

```
# Пример: включить IPv4 forwarding ТОЛЬКО для роли маршрутизатора (отдельным файлом профиля)
cat > /etc/sysctl.d/60-routing.conf <<'EOF'
net.ipv4.ip_forward = 1
EOF

sysctl --system
```

Примечание. Итоговая целевая модель после первичной настройки: вход по SSH — только под обычным пользователем; `root` — только локально или через `sudo`; входящие соединения — только явно разрешённые; Docker — часть платформы; hardening-профиль — baseline и изменяется только по назначению роли узла.