

RU. КНРШ.00001-01 90 01

УТВЕРЖДЕН

RU. КНРШ.00001-01 90 01-ЛУ

СРЕДСТВО ВИРТУАЛИЗАЦИИ «ЗВЕЗДА»

Руководство администратора

RU.КНРШ.000014-01 90 01

Листов 95

Инв. № подл.	Подп. и дата	Взам. Инв №	Инв. № дубл.	Подп. и дата

## АННОТАЦИЯ

Средство виртуализации (СВ) “Звезда” RU.КНРШ.00014-01 (далее по тексту - СВ “Звезда” или изделие) является гипервизором первого типа, устанавливаемым непосредственно на аппаратное обеспечение в качестве системного программного обеспечения.

## СОДЕРЖАНИЕ

<b>1</b>	<b>Введение</b> .....	<b>6</b>
1.1	Краткое описание возможностей.....	6
<b>2</b>	<b>Назначение и условия применения</b> .....	<b>8</b>
2.1	Назначение изделия.....	8
2.2	Условия применения изделия .....	8
<b>3</b>	<b>Подготовка к работе</b> .....	<b>12</b>
3.1	Действия по приемке поставленного СВ “Звезда” .....	12
3.2	Установка СВ “Звезда” .....	12
3.3	Первоначальные настройки в средстве виртуализации “Звезда” .....	25
3.3.1	Смена пароля пользователя root .....	25
3.3.2	Парольная политика .....	26
3.3.3	Подключение хранилища для резервных копий и снимков.....	27
3.4	Обновление СВ “Звезда” .....	28
3.4.1	Откат к предыдущей версии .....	31
3.5	Изменение сетевых настроек .....	34
<b>4</b>	<b>Защита среды виртуализации</b> .....	<b>35</b>
4.1	Контроль целостности .....	35
4.2	Идентификация и аутентификация пользователей .....	35
4.2.1	Создание и удаление пользователей .....	36
4.2.2	Создание групп пользователей, добавление пользователя в группу.....	37
4.3	Управление ролевым доступом .....	37
4.3.1	Создание роли администратора средств виртуализации .....	38
4.3.2	Создание роли разработчика ВМ.....	39
4.3.3	Создание роли администратора безопасности средства виртуализации .....	40
4.3.4	Создание роли администратора виртуальной машины .....	41
4.3.5	Создание роли разработчика образов контейнеров .....	41
4.3.6	Создание роли администратора информационной системы .....	42
4.3.7	Создание роли администратора безопасности средства контейнеризации .....	43
4.3.8	Журналирование фактов изменения ролевой модели .....	44
4.3.9	Журналирование фактов создания, модификации и удаления образов контейнеров .....	44

4.3.10 Журналирование неуспешных попыток аутентификации пользователей средства контейнеризации .....	45
4.3.11 Журналирование запуска и остановки контейнеров .....	45
4.3.12 Журналирование модификаций запускаемых контейнеров .....	45
4.3.13 Журналирование успешных и неуспешных попыток аутентификации .....	45
4.3.14 Журналирование доступа пользователей средства виртуализации к виртуальным машинам .....	46
4.3.15 Журналирование создания и удаления виртуальных машин .....	46
4.3.16 Журналирование запуска и остановки средства виртуализации .....	46
4.3.17 Журналирование запуска и остановки VM .....	46
4.3.18 Журналирование изменения конфигурации средства виртуализации .....	46
4.3.19 Журналирование изменения конфигурации VM .....	47
4.3.20 Журналирование фактов нарушения целостности объектов контроля .....	47
4.3.21 Журналирование доступа к образам контейнеров .....	47
4.4 Управление потоками информации .....	47
4.5 Регистрация событий безопасности .....	48
4.5.1 Использование auditd .....	48
4.5.2 Использование Logrotate .....	54
4.5.3 Работа с журналом journald .....	57
4.5.4 Система мониторинга системного журнала journald-monitor .....	60
4.5.5 Использование protector.log .....	65
4.6 Резервное копирование с использованием файловой системы Overlayfs .....	65
4.6.1 Подробное описание слоев .....	65
4.7 Резервные копии виртуальных машин .....	66
4.7.1 Создание резервной копии .....	66
4.7.2 Восстановление виртуальной машины .....	67
4.8 Защита памяти .....	68
<b>5 Управление и защита контейнеров .....</b>	<b>69</b>
5.1 Подпись контейнера .....	69
5.2 Создание образов контейнеров .....	69
5.3 Централизованное управление образами контейнеров и контейнерами .....	71
5.3.1 Ограничение прав на использование вычислительных ресурсов контейнера ....	71
5.4 Регистрация событий безопасности в контейнере .....	72
5.5 Изоляция контейнеров .....	72

5.5.1	Использование Docker для настройки Namespaces и Cgroups .....	72
5.5.2	Аппаратная изоляция контейнеров .....	75
5.5.3	Монтирование корневой файловой системы хоста в контейнер в режиме “только для чтения” .....	77
5.6	Выявление уязвимостей в образах контейнеров .....	77
5.7	Контроль целостности контейнеров и их образов.....	77
5.8	Идентификация и аутентификация пользователей .....	78
5.9	Монтирование USB-устройства в контейнер .....	78
5.10	Просмотр журнала событий контейнера.....	79
<b>6</b>	<b>Работа с виртуальными машинами .....</b>	<b>80</b>
6.1	Создание виртуальной машины .....	80
6.1.1	Добавление пароля для подключения по протоколу spice .....	81
6.1.2	Добавление диска к VM .....	81
6.1.3	Подключение VM к сети .....	82
6.1.4	Редактирование VM .....	84
6.2	Снимки состояния VM.....	84
6.2.1	Создание снимка состояния VM .....	84
6.2.2	Восстановление VM с помощью снимка состояния VM .....	85
6.2.3	Удаление снимка VM .....	85
6.3	Миграция VM с хоста на хост.....	85
<b>7</b>	<b>Создание снимков системы .....</b>	<b>87</b>
<b>8</b>	<b>Установка приложений .....</b>	<b>91</b>

## 1 ВВЕДЕНИЕ

### 1.1 Краткое описание возможностей

Изделие реализует функции безопасности в соответствии Требованиями по безопасности информации к средствам виртуализации (утверждены приказом ФСТЭК России от 27 октября 2022 г. № 187):

- доверенную загрузку виртуальных машин;
- контроль целостности;
- регистрацию событий безопасности;
- управление доступом;
- резервное копирование;
- управление потоками информации;
- защиту памяти;
- ограничение программной среды;
- идентификацию и аутентификацию пользователей;
- централизованное управление образами виртуальных машин и виртуальными машинами.

Изделие реализует функции безопасности в соответствии Требованиями по безопасности информации к средствам контейнеризации (утверждены приказом ФСТЭК России от 4 июля 2022 г. № 118):

- изоляцию контейнеров;
- выявление уязвимостей в образах контейнеров;
- контроль целостности контейнеров и их образов;
- регистрацию событий безопасности;
- идентификацию и аутентификацию пользователей;
- проверку корректности конфигурации контейнеров;
- централизованное управление образами контейнеров и контейнерами

СВ “Звезда” предоставляет пользователям следующие возможности:

- Поддержка графического установщика;

- установка непосредственно на аппаратное обеспечение без использования хостовой операционной системы (гипервизор 1 типа);
- создание и редактирование ВМ;
- обеспечение возможности использования в качестве гостевой ОС операционных систем семейств Linux, Windows;
- поддержка 32- и 64-битных гостевых ОС, работающих на серверах стандартной архитектуры x86;
- создание «снимка» работающей системы (контрольной точки состояния операционной системы. В любое время можно вернуться к состоянию на момент создания контрольной точки);
- миграция (перенос исполнения) виртуальных машин между серверами виртуализации;
- создание и хранение образов ВМ для развертывания ВМ;
- создание виртуальных сетевых мостов, а также использование следующих протоколов VLAN (IEEE 802.1Q) и VXLAN (RFC-7348) для изоляции и/или объединения в виртуальные сети сетевого трафика виртуальных машин;
- предоставление доступа к хранилищу данных через протоколы iSCSI, NFS, CIFS/SMB;
- поддержка подключения с помощью Fibre Channel (FC);
- поддержка функции Multipathing.

## 2 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

### 2.1 Назначение изделия

Изделие предназначено для использования в клиент-серверных системах. СВ “Звезда” является гипервизором, устанавливаемым непосредственно на аппаратное обеспечение в качестве системного программного обеспечения, и предназначенным для организации исполнения виртуальных машин (ВМ).

### 2.2 Условия применения изделия

Средство виртуализации “Звезда” включает в себя применение встроенных средств защиты в виртуальной среде. Все функции безопасности инициализируются при загрузке изделия. Изделие устанавливается на аппаратную платформу без ОС общего назначения. Среда функционирования - BIOS. Согласно формуляру RU.КНРШ.000014-01 30 01 (п.п. 6.7, 6.8), администратор средства виртуализации обязан установить пароль на BIOS. Замена батареи питания BIOS, а также его обновление осуществляется администратором средства виртуализации.

После получения изделия необходимо сверить контрольные суммы носителей с указанными в формуляре RU.КНРШ.000014-01 30 01.

В изделии поддерживаются следующие роли безопасности:

- разработчик виртуальной машины;
- администратор безопасности средства виртуализации;
- администратор средства виртуализации;
- администратор виртуальной машины;
- разработчик образов контейнеров;
- администратор безопасности средства контейнеризации;
- администратор информационной (автоматизированной) системы.

Разработчик образов контейнеров должен не реже одного раза в неделю осуществлять проверку уязвимостей в образах контейнеров сертифицированным ФСТЭК России средством (например, программным изделием «Средство обеспечения безопасности ин-

формационных систем MaxPatrol»). При появлении информации о новых уязвимостях в банке данных угроз безопасности информации производитель контейнера проводит с помощью сертифицированных средств проверку незамедлительно. Запрещено создание образов контейнеров, содержащих известные уязвимости критического и высокого уровня опасности.

С целью предотвращения нарушения целостности замкнутой среды СВ “Звезда” на действия суперпользователя root накладываются ограничения. Администратор средства виртуализации с правами суперпользователя может выполнять следующие действия:

- 1) Вносить изменения в конфигурационные файлы, расположенные в директориях:
  - /etc/auditd/;
  - /etc/libvirt/.
- 2) Просматривать все файлы системы с использованием всех доступных системных утилит и средств фильтрации.
- 3) Использовать любые системные утилиты без внесения изменений в системные файлы и файлы конфигураций.

Таблица 2.1 – Характеристики СВ “Звезда”

Характеристика	Показатель
Количество процессоров сервера виртуализации, поддерживаемых гипервизором	от 2 до 4096
Объем оперативной памяти сервера виртуализации, поддерживаемый гипервизором	от 4 Гб до 256 Тб
Объем жесткого диска сервера виртуализации	не менее 100 Гб
Количество процессорных сокетов	не менее 2
Суммарное количество физических ядер сервера виртуализации	от 4 до 2048

Количество виртуальных ЦПУ, поддерживаемых одной VM	от 2 до 256
Количество памяти, поддерживаемой VM	от 1 Гб до 32 Тб
Поддержка виртуальных накопителей в VM объемом (максимальное значение ограничено аппаратными возможностями сервера виртуализации)	от 4 Гб
Количество виртуальных процессоров, поддерживаемых VM	от 2 до 2048
Возможность организации виртуальных сетевых интерфейсов со скоростями	до 100 Гбит/с
Основные поддерживаемые ОС семейства Windows	Windows Server 2016; Windows Server 2012 R2; Windows Server 2012; Windows Server 2016; Windows Server 2019 ;Windows Server 2008 R2 with Service Pack 1; Windows XP/7/10/11
Основные поддерживаемые ОС семейства Linux	AltLinux 8; AstraLinux 2.12 ;AstraLinux 1.5; AstraLinux 1.6;CentOS 8.x; CentOS 7.x; CentOS6.x; Debian 10.x; Debian 9.x; Debian 8.x; Debian 7.x; Ubuntu 17.10; Ubuntu 16.04 LTS; Ubuntu 14.04 LTS; openSUSE 42.x; SLES 11; SLES 12; SLES 15; Oracle Linux 8.x; Oracle Linux 7.x; Oracle Linux 6.x; Oracle Linux 5.x; Oracle Enterprise Linux 4.x; Red Hat Enterprise Linux (RHEL), Oracle DB
Совместимое серверное оборудование	Hewlett Packard Enterprise, Huawei, Lenovo, Cisco, DellEMC, Fujitsu, IBM, Depo Computers, Аквариус, Булат, Т-Платформы.
Объем поддержки томов	Более 2 Тб
Поддержка ПО SAP	SAP, SAP ASE, SAP MaxDB
Поддержка систем управления реляционным базам данных	MS SQL, IBM DB2, PostgreSQL
Поддержка протокола HTML	есть

Таблица 2.2 – Свойства, поддерживаемые в одной виртуальной среде

<b>Поддерживаемое свойство</b>	<b>Показатель</b>
Виртуальные центральные процессоры устройств (далее – ЦПУ)	64
Оперативная память	512 ГБ
Объем дисков в виртуальных машинах и контейнерах	16 ТБ

### 3 ПОДГОТОВКА К РАБОТЕ

СВ “Звезда” поставляется на USB-носителе для установки на ПЭВМ, выполняющей функции сервера виртуализации.

СВ “Звезда” может быть установлено на следующие устройства хранения:

- локальные HDD, SDD/Flash/NVMe;
- SAN LUN (загрузка с SAN);
- требуемая минимальная емкость устройства – 40 ГБ.

#### 3.1 Действия по приемке поставленного СВ “Звезда”

- 1) Необходимо установить USB-накопитель с образом средства виртуализации “Звезда” в ПЭВМ с установленным программным обеспечением фиксации контрольных сумм (например, «ФИКС»).
- 2) Произвести вычисление контрольных сумм по алгоритму ГОСТ 34.11.
- 3) Открыть формуляр RU.КНРШ.000014-01 30 01, сверить полученные контрольные суммы с указанными в Приложении А. В случае совпадения контрольных сумм произвести установку изделия согласно п. 3.2.
- 4) В случае несовпадения контрольных сумм использование СВ “Звезда” запрещено.

#### 3.2 Установка СВ “Звезда”

Для установки СВ “Звезда” необходимо выполнить следующие действия:

- 1) Выбрать пункт **Install** в появившемся окне инсталлятора системы, нажать клавишу **Enter**.



Рисунок 3.1 – Меню установки СВ “Звезда”

*Примечание. Переход по кнопкам осуществляется клавишей **Tab**.*

2) Будет запущен процесс установки.

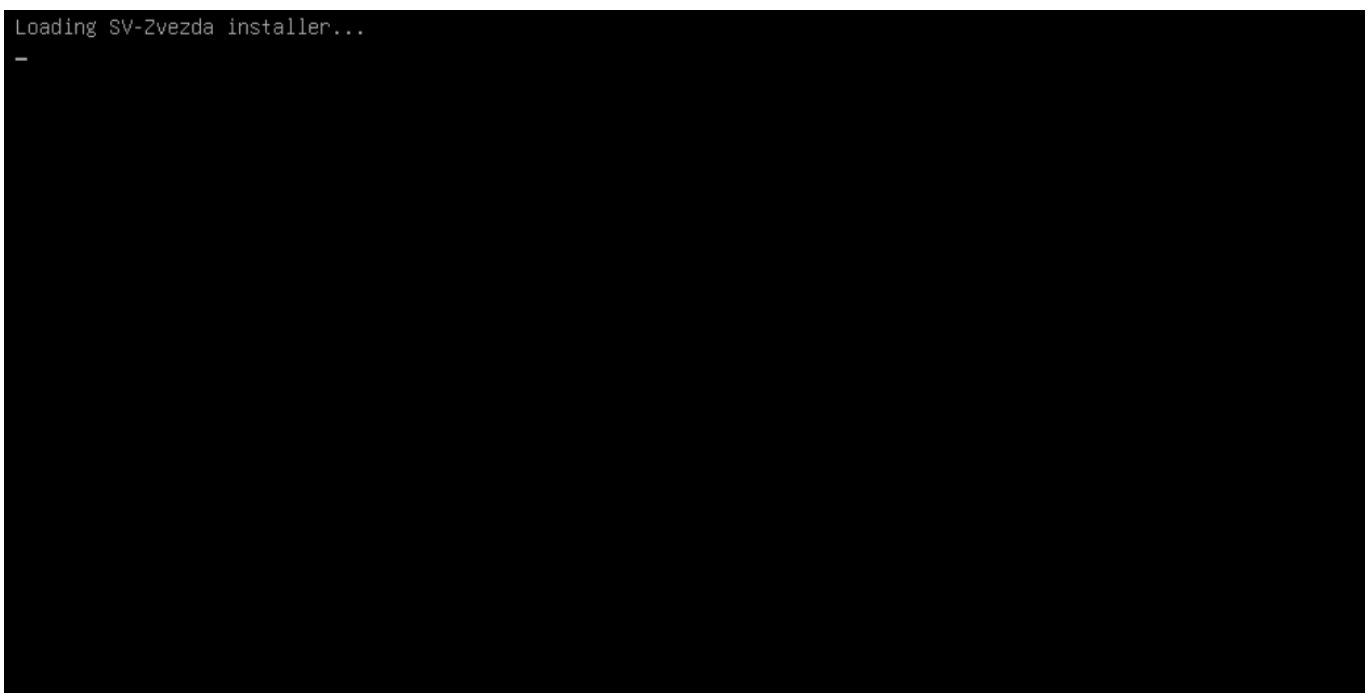


Рисунок 3.2 – Процесс установки запущен

3) В открывшемся окне выбрать русский язык, затем нажать кнопку **Далее**.

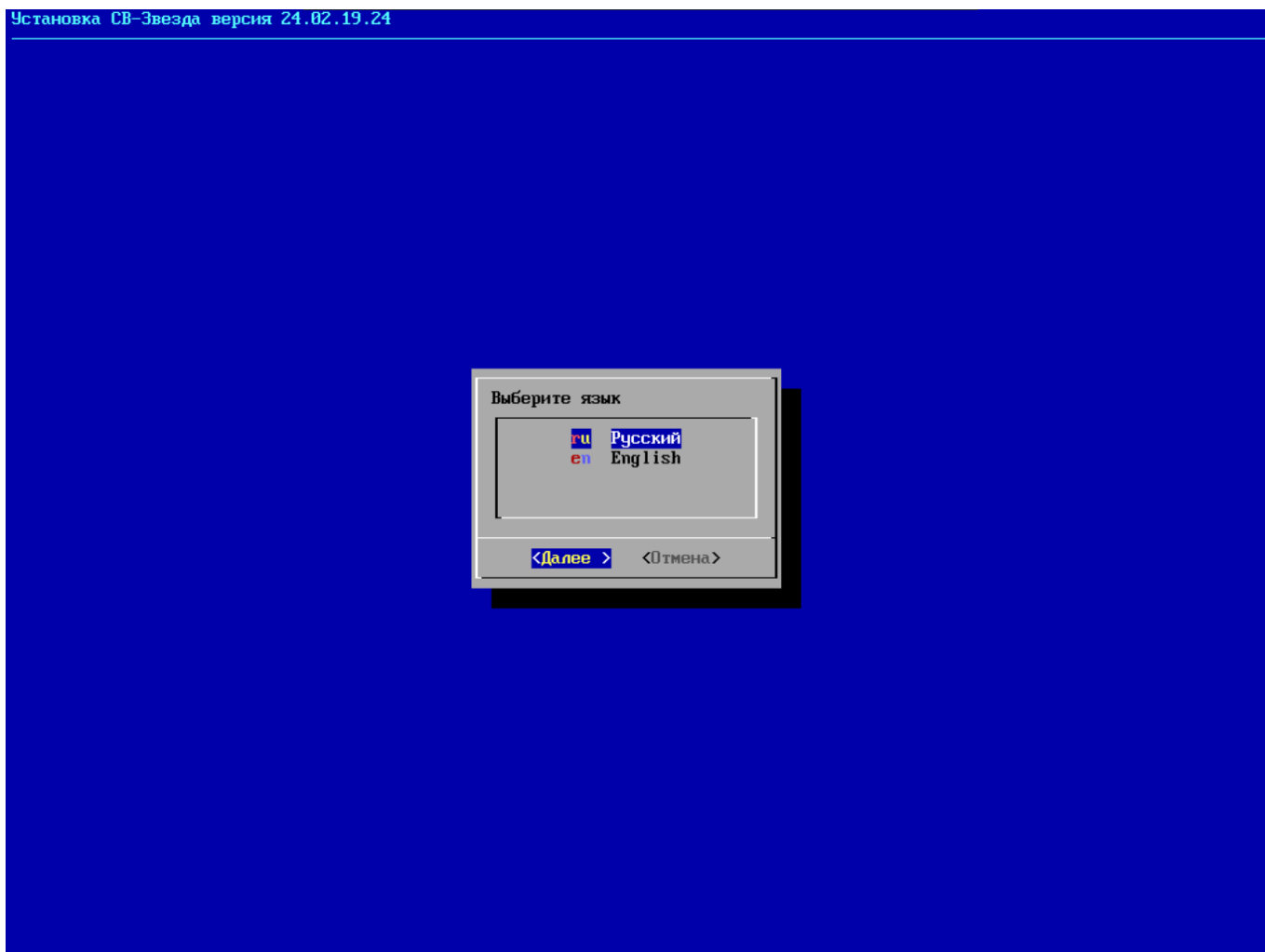


Рисунок 3.3 – Выбор языка

4) На следующем шаге необходимо выбрать диск для установки СВ “Звезда”, затем нажать кнопку **Далее**. На данном этапе установщик сканирует блочные устройства на предмет наличия уже установленного СВ “Звезда” и возможности его обновления. В случае обнаружения установленного СВ “Звезда” далее будет предоставлено меню с возможностью выбора альтернативы для обновления системы.

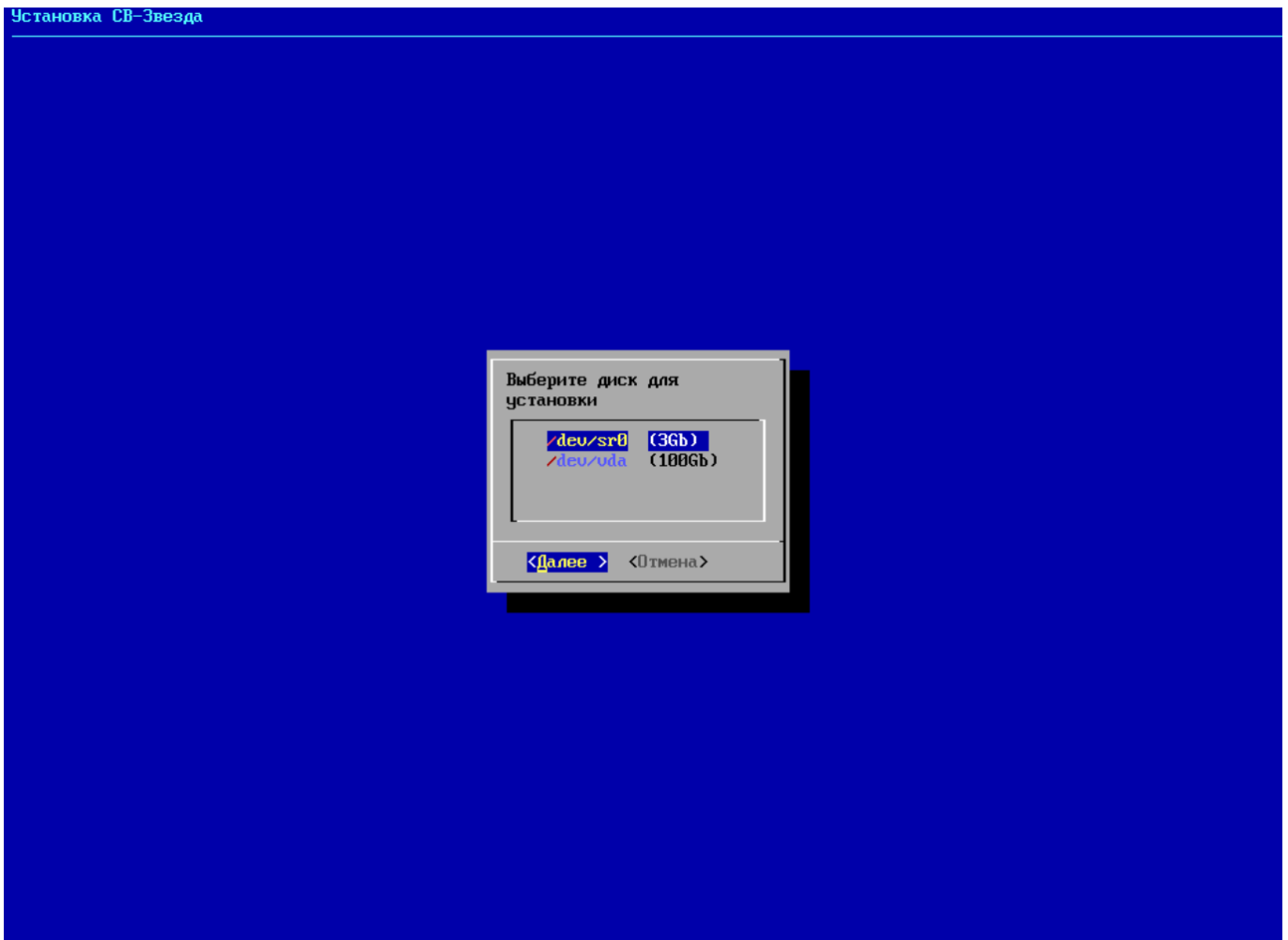


Рисунок 3.4 – Выбор диска

- 5) Далее необходимо выбрать часовой пояс, затем нажать **Ок**.

*Примечание: для перемещения между позициями следует использовать клавиши ←↑↓→ и **Tab**, для выбора позиции – клавишу **Пробел**, для выбора кнопок перемещения между страницами – клавишу **Enter**.*

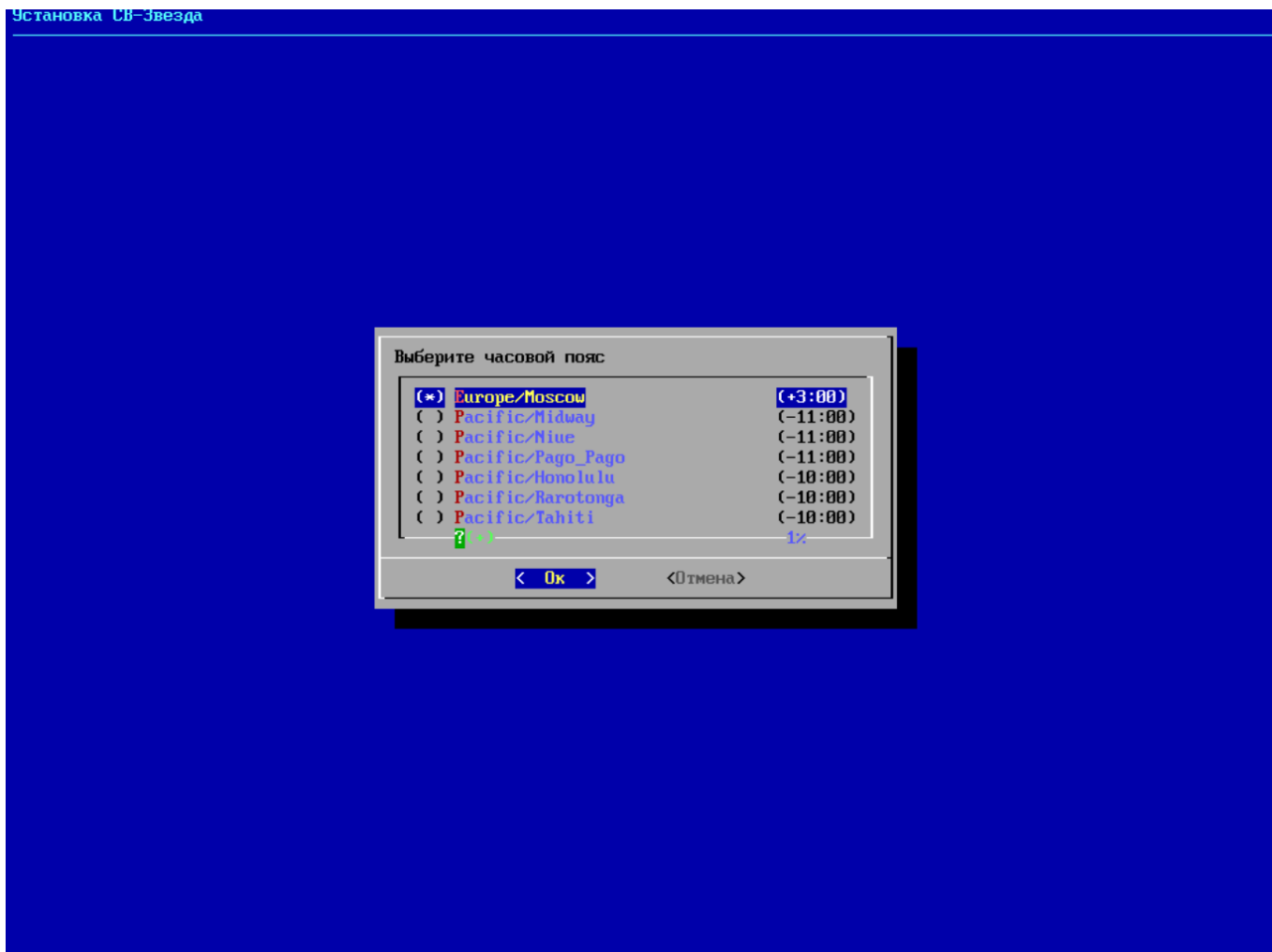


Рисунок 3.5 – Выбор часового пояса

б) Затем необходимо ввести имя хоста, нажать **ОК**.

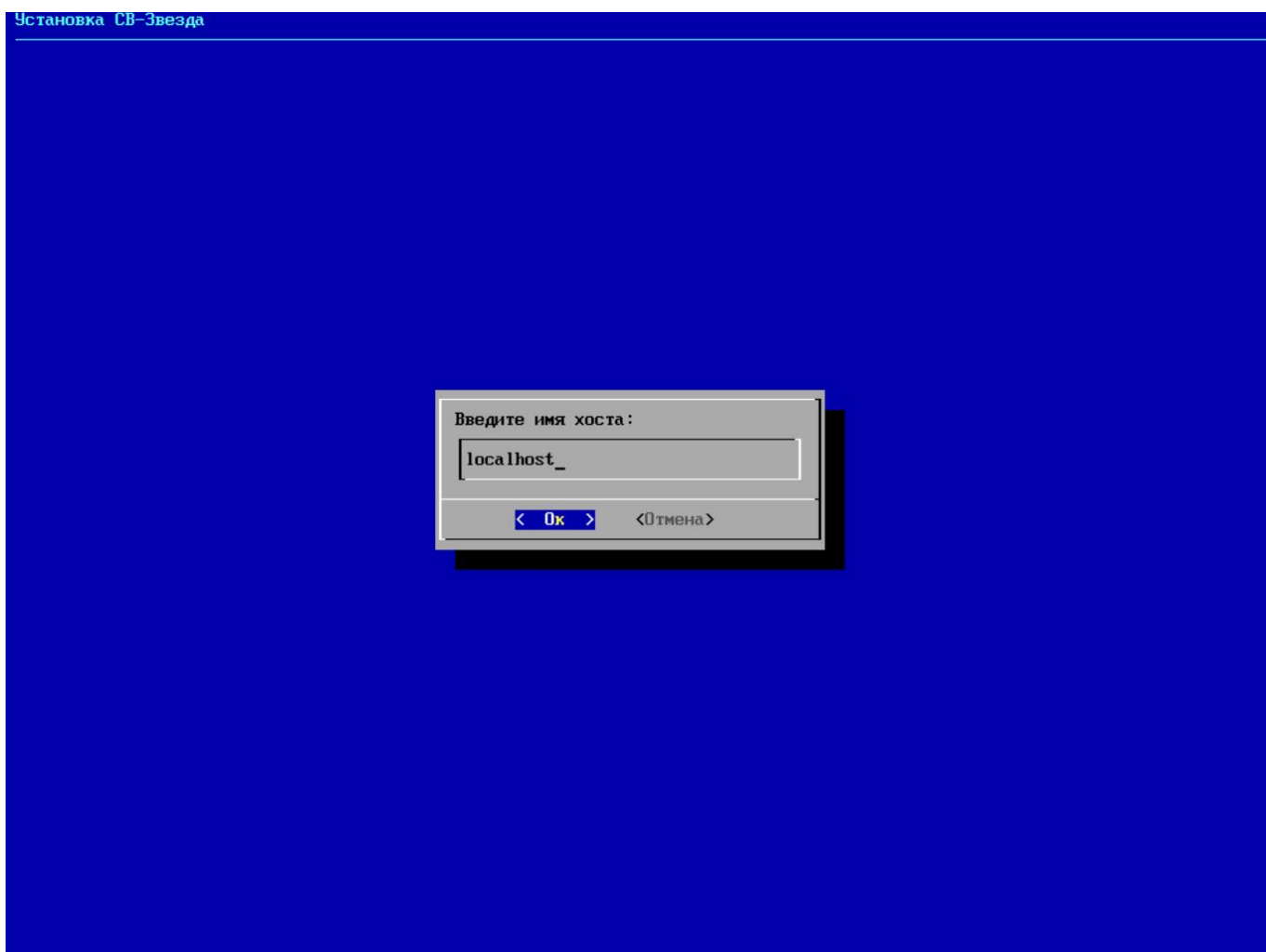


Рисунок 3.6 – Ввод имени хоста

7) Выбрать управляющий интерфейс, нажать **Выбор**.

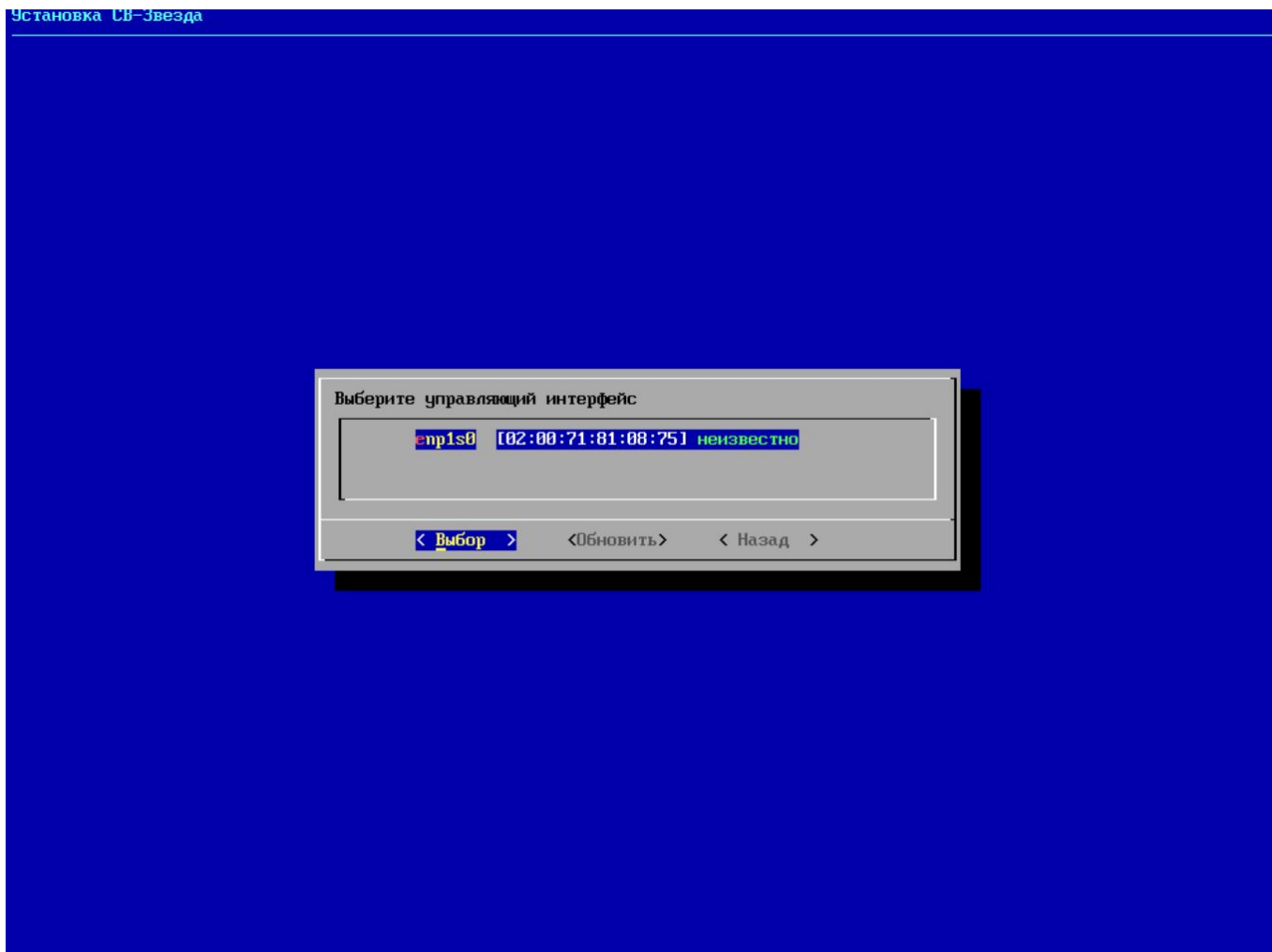


Рисунок 3.7 – Выбор управляющего интерфейса

8) В шаге **Настройка управляющего интерфейса** необходимо заполнить все поля.

Справа от MAC-адреса отображена скорость интерфейса, если это возможно определить. Если скорость интерфейса отображена зеленым цветом, то сетевой кабель подключен. Если красным, то либо определение состояние подключения невозможно, либо сетевой кабель не подключен.

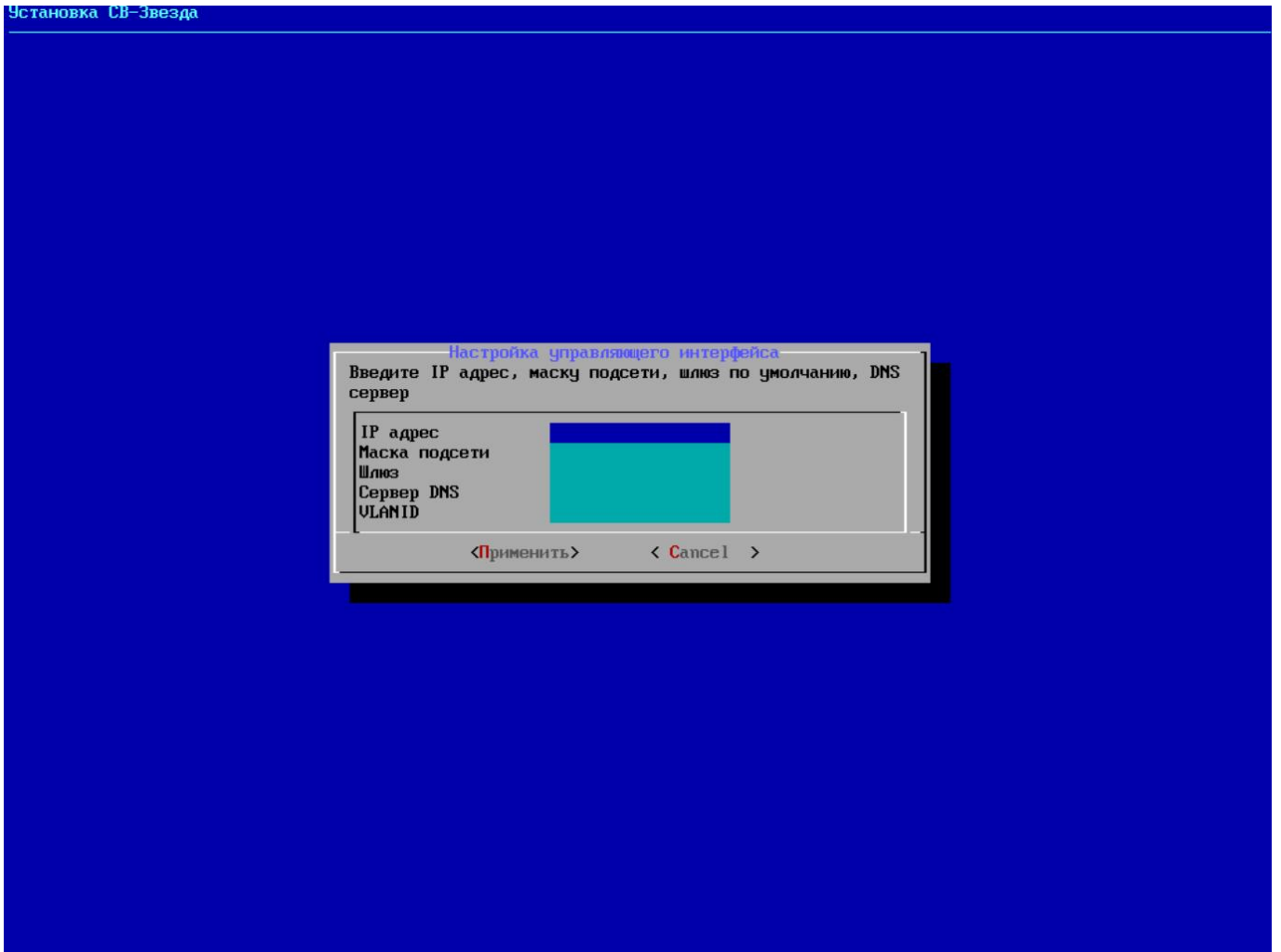


Рисунок 3.8 – Настройка управляющего интерфейса

В СВ “Звезда” при установке поддерживается только статический тип назначения IP адресов. Необходимо заполнить поля.

9) Появится окно подтверждения установки. Для подтверждения нажать кнопку **Да**.

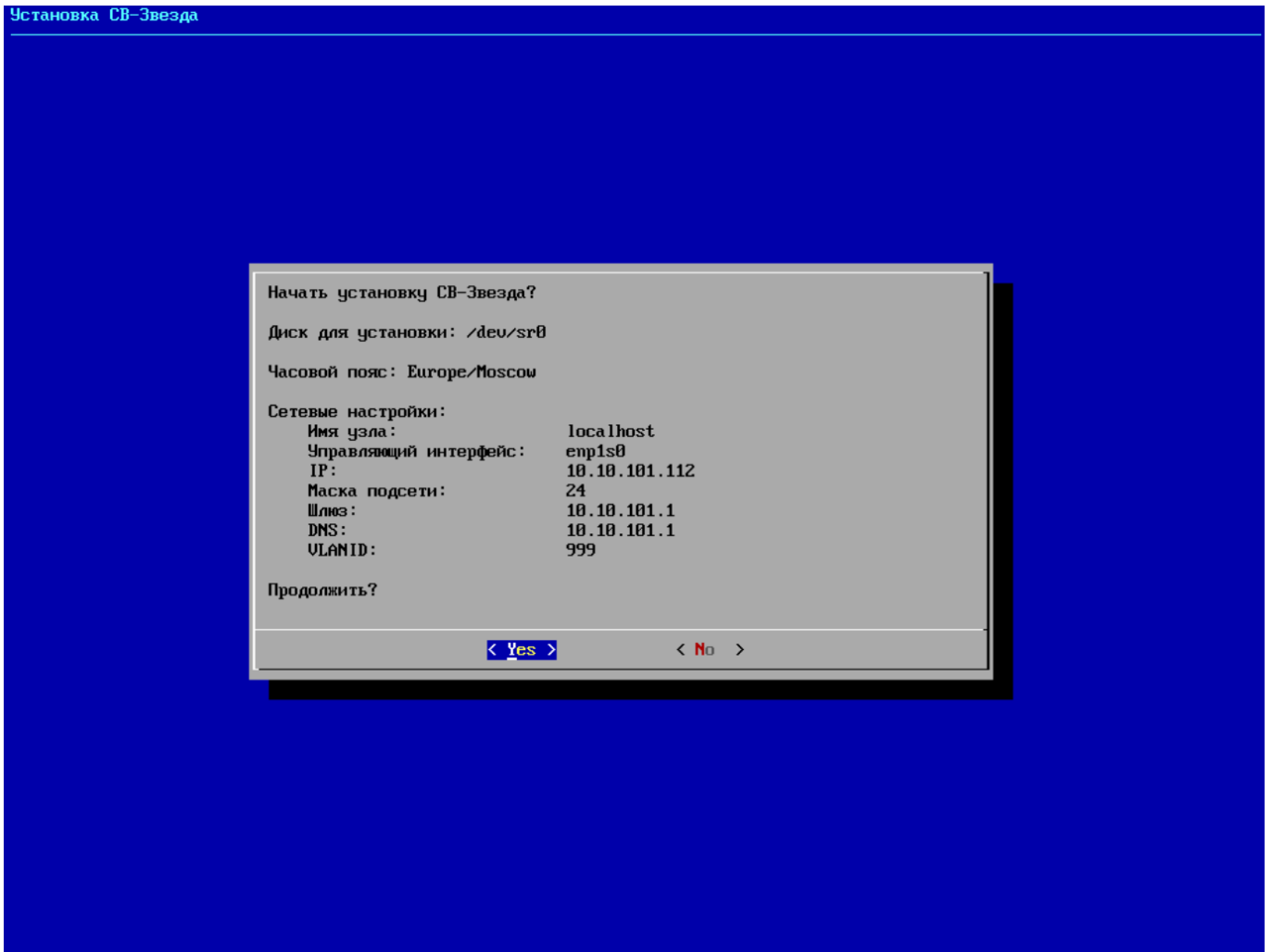


Рисунок 3.9 – Подтверждение установки СВ “Звезда”

10) Запустится процесс установки.

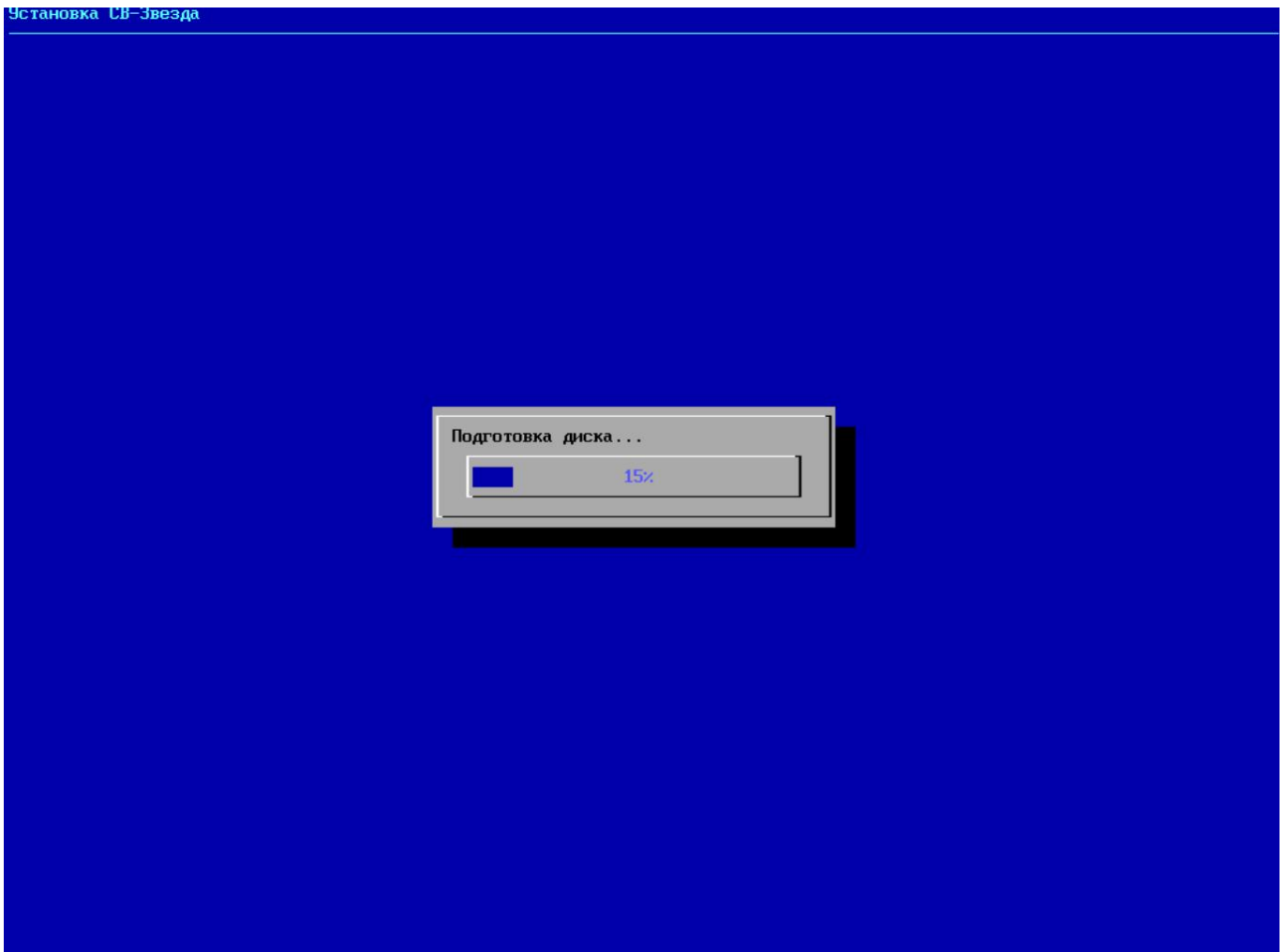


Рисунок 3.10 – Процесс установки

- 11) После окончания установки необходимо перезагрузить систему. Для этого нужно выбрать **Перезагрузить**, нажать **ОК**.

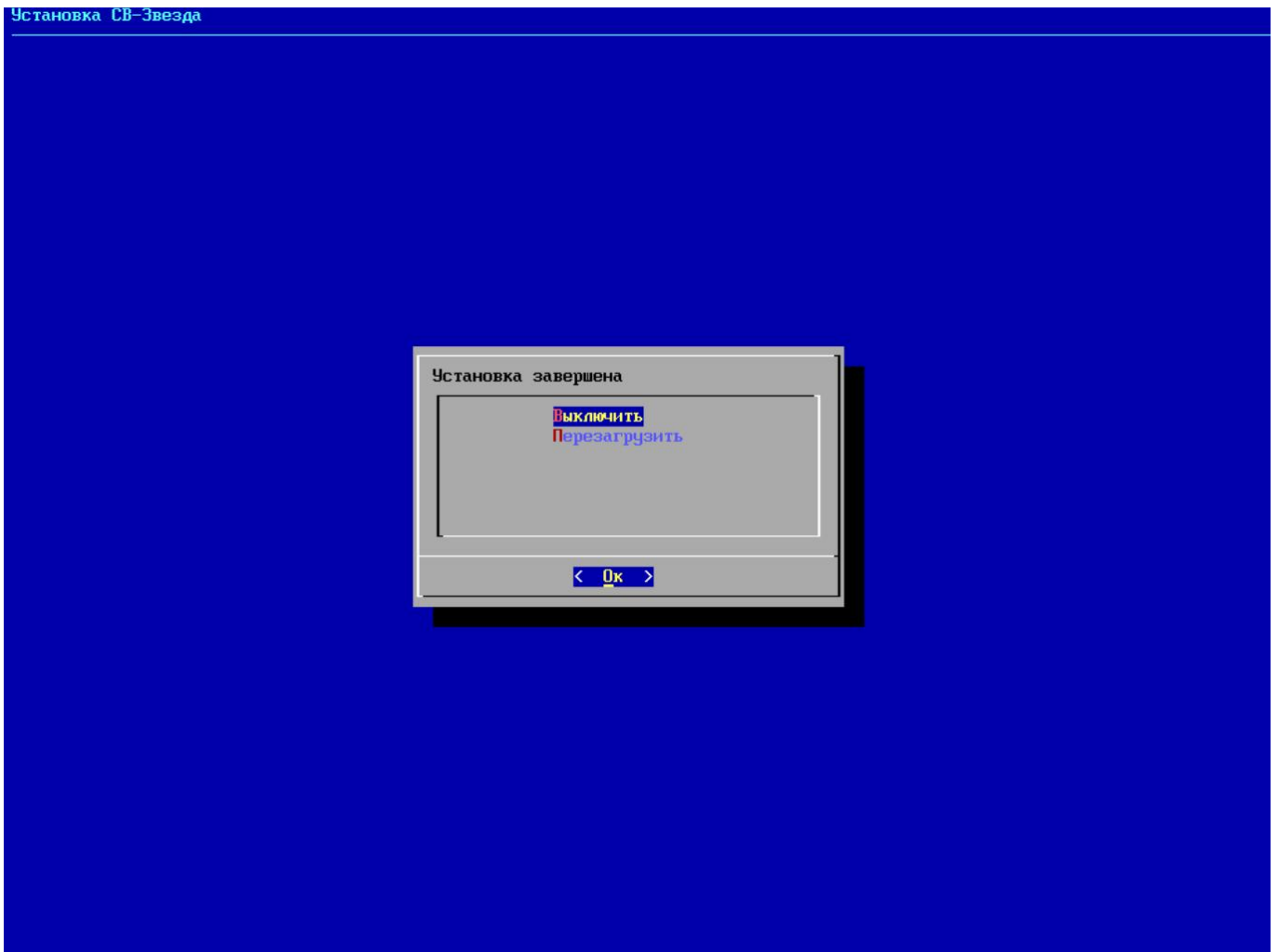


Рисунок 3.11 – Установка завершена

12) После перезагрузки появится следующая информация.

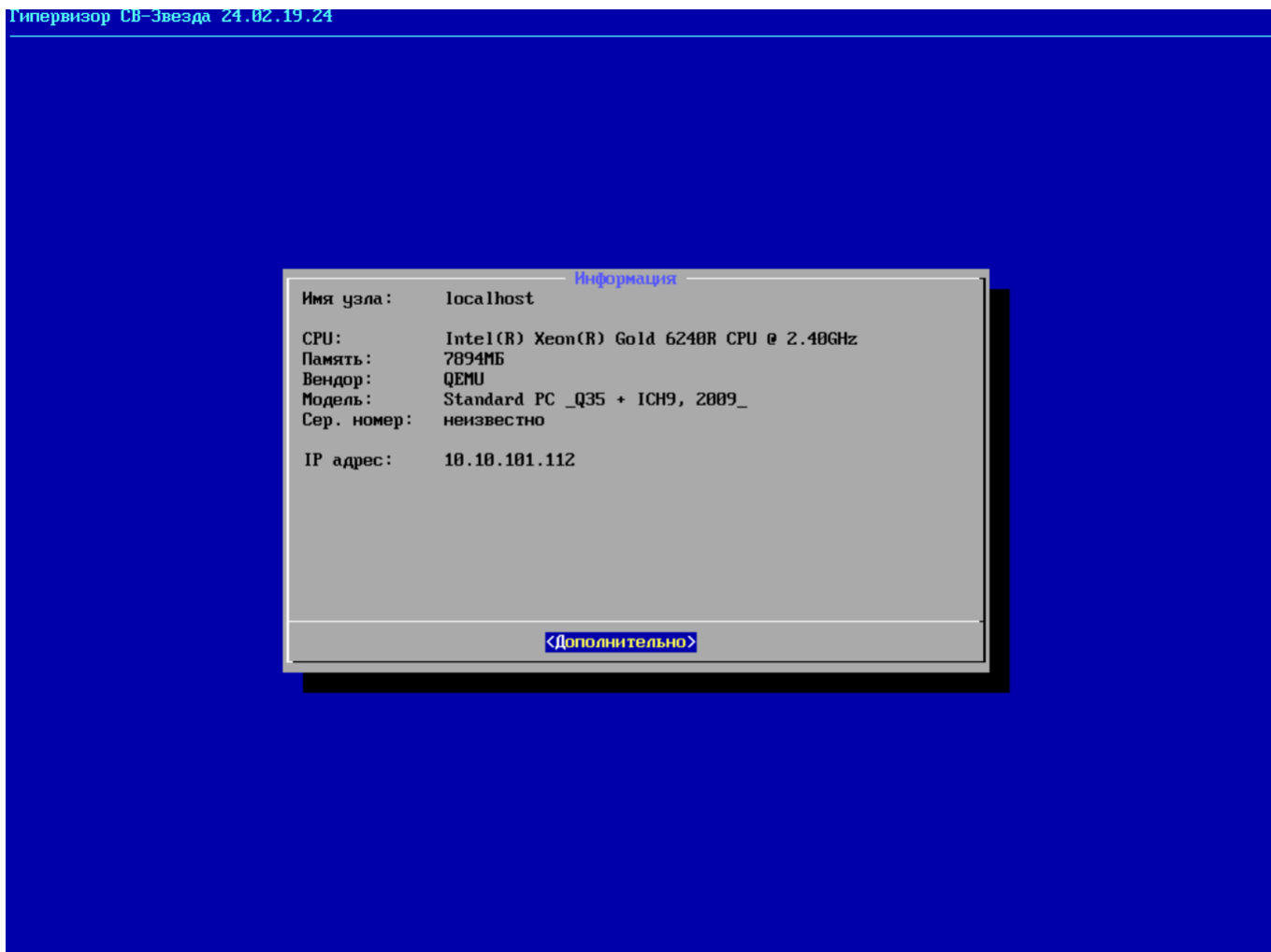


Рисунок 3.12 – Информация

- 13) СВ “Звезда” установлено. При необходимости войти в консоль управления, необходимо нажать сочетание клавиш **Alt+F2**. Для первого входа необходимо использовать логин и пароль root / P@ssw0rd.

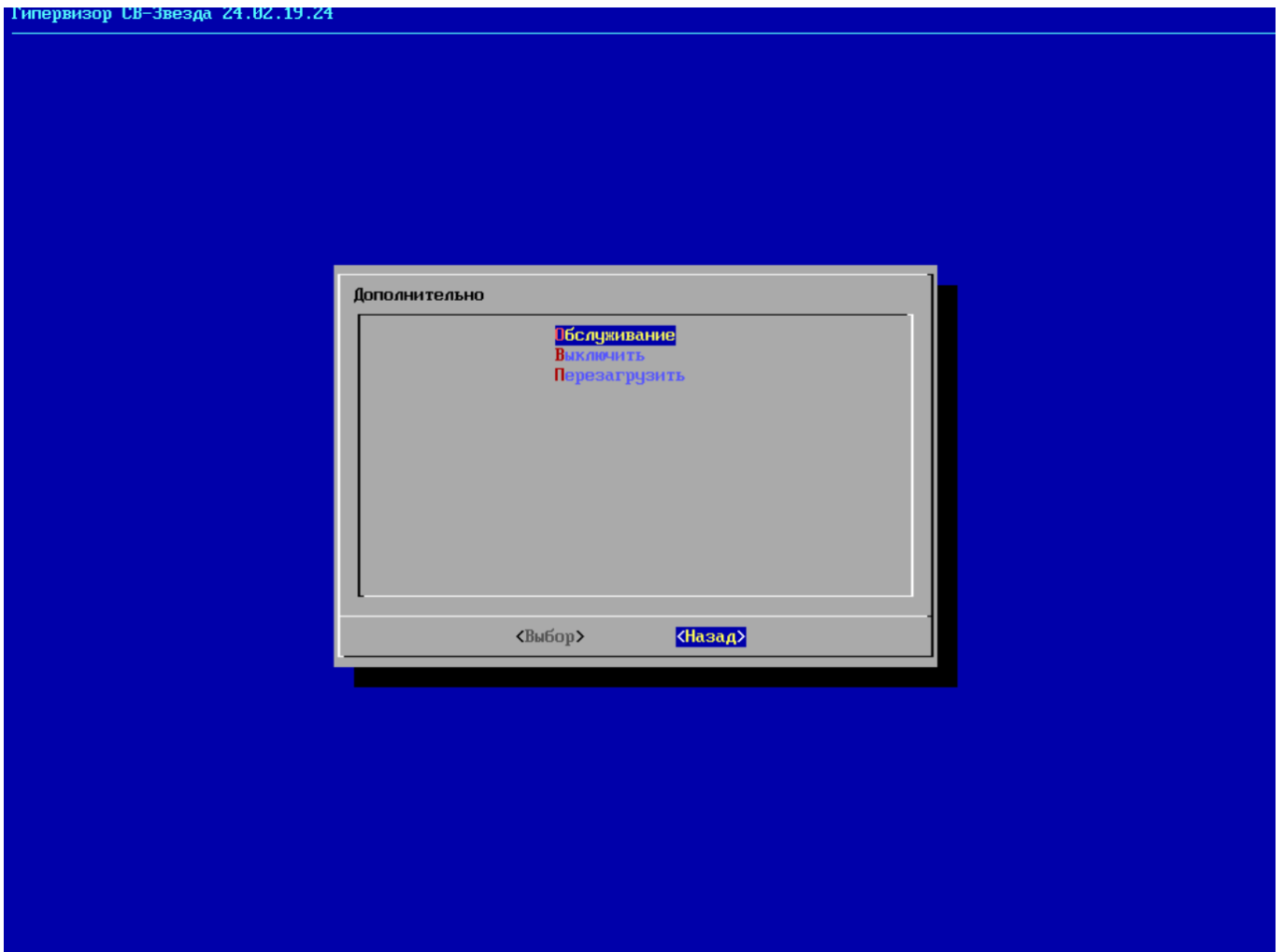


Рисунок 3.13 – Дополнительные функции

14) Консоль управления СВ “Звезда”.

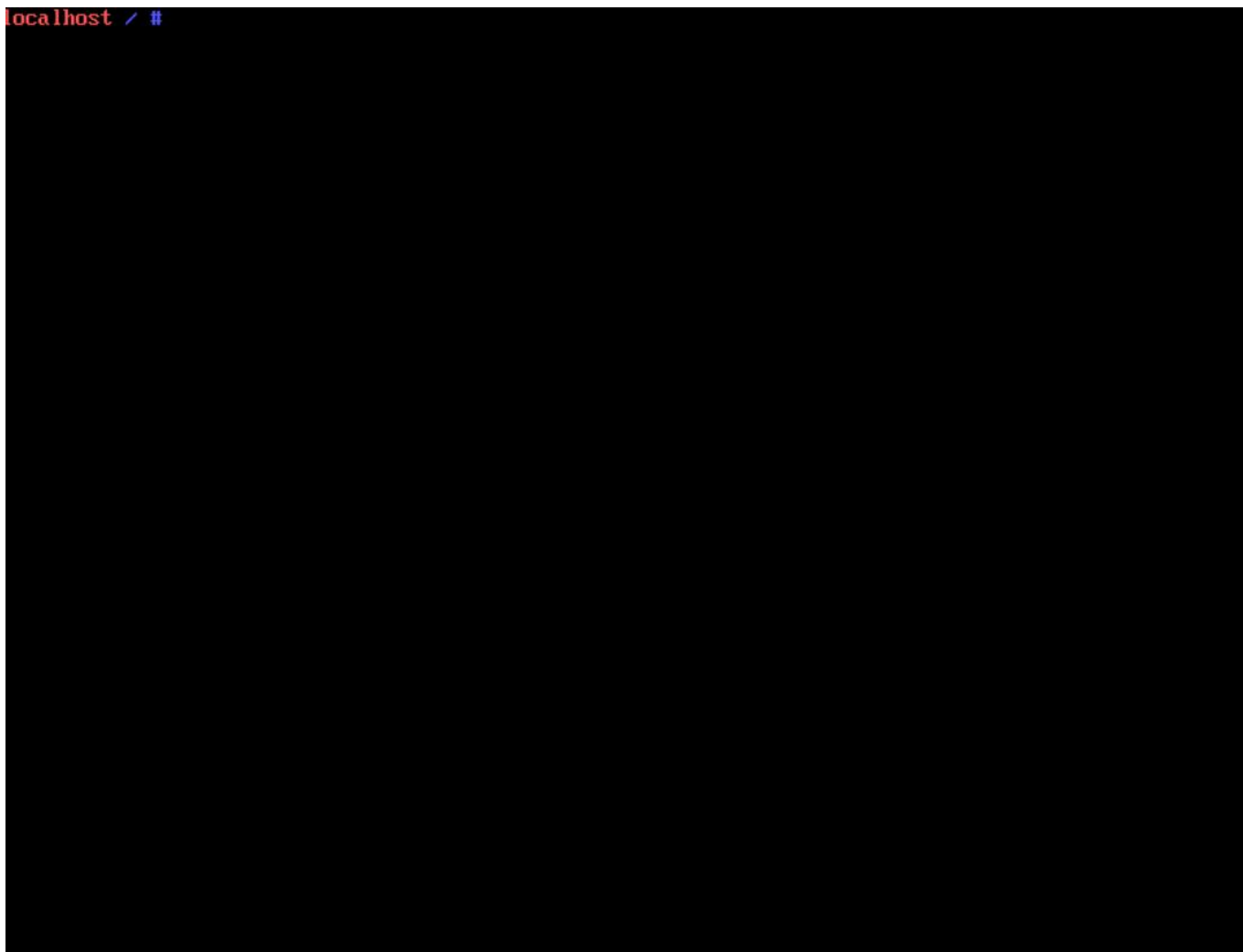


Рисунок 3.14 – Консоль управления СВ “Звезда”

### 3.3 Первоначальные настройки в средстве виртуализации “Звезда”

После установки СВ “Звезда” необходимо:

- Сменить пароль от имени пользователя root (п. 3.3.1);
- подключить дополнительное хранилище для резервных копий и снимков (п.3.3.3);
- создать пользователей системы в соответствии с ролевым доступом (п.4.3)

#### 3.3.1 Смена пароля пользователя root

Для смены пароля:

- 1) Выполнить команду в консоли:

```
passwd <имя_пользователя>
```

- 2) Дважды корректно ввести новый пароль.

3) Перезапустить сервер виртуализации.

### 3.3.2 Парольная политика

Для настройки парольной политики необходимо выполнить следующее:

1) Открыть файл `passwdqc.conf` для редактирования:

```
sudo nano /etc/security/passwdqc.conf
```

Настройки по умолчанию включают в себя следующие параметры:

**min=disabled,24,11,8,7**

- Задаёт минимальные требования к длине пароля: - `disabled` – запрещает использование односимвольных паролей. - 24 – минимальная длина пароля для самого слабого пароля (который не соответствует никаким требованиям сложности). - 11 – минимальная длина для пароля, который соответствует минимальным требованиям сложности. - 8 – минимальная длина для более сложного пароля. - 7 – минимальная длина для самого сложного пароля.

**max=72**

- Ограничивает максимальную длину пароля до 72 символов.

**passphrase=3**

- Требуется минимум 3 уникальных символа в пароле.

**match=4**

- Проверяет наличие 4 совпадающих символов в строках с ранее использованными паролями. Если совпадение есть, пароль считается слабым.

**similar=deny**

- Запрещает использование паролей, которые похожи на текущее или предыдущее значение.

**random=47**

- Определяет минимальную длину для автоматически сгенерированных паролей. Это полезно, если используются случайные пароли.

**enforce=none**

- Не требует обязательного применения политики паролей ко всем пользователям.

**retry=3**

- Задаёт количество попыток ввода пароля при смене или установке нового.

2) Настроить параметры в файле:

```
min=disabled,8,8,8,8  
passphrase=70
```

- `min=disabled,8,8,8,8` – устанавливает минимальную длину пароля в 8 символов для разных категорий паролей (от простых до сложных). Значение `disabled` означает, что односимвольные пароли не принимаются.
- `passphrase=70` – задает минимальное количество уникальных символов (алфавита) в пароле.

3) Сохранить изменения и закрыть файл.

4) Применить изменения. Перезагрузка сервера или перезапуск сервисов не требуется, так как настройки применяются при каждом изменении пароля.

### 3.3.3 Подключение хранилища для резервных копий и снимков

После “чистой” установки СВ “Звезда” рекомендуется подключить дополнительное хранилище для резервных копий и снимков. В роли такого хранилища может выступать любое отформатированное в поддерживаемой файловой системе блочное устройство. Список поддерживаемых файловых систем можно получить командой:

```
grep -v nodev /proc/filesystems
```

Хранилище для резервных копий и снимков определяется специальной точкой монтирования:

```
/var/backups
```

Для подключения блочного устройства необходимо выполнить следующие действия:

1) Отформатировать блочное устройство в поддерживаемой файловой системе:

```
mkfs.ext4 /dev/sdb
```

2) Добавить строку в `/etc/fstab`:

```
/dev/sdb /var/backups auto defaults 0 0
```

3) Выполнить команду:

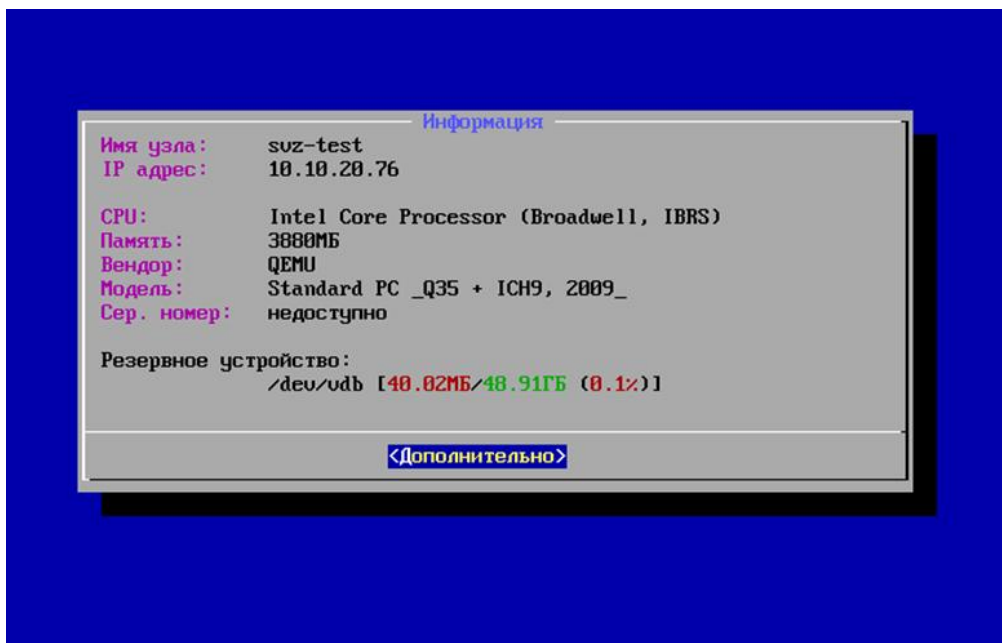
```
systemctl daemon-reload
```

4) Далее необходимо выполнить монтирование с созданием точки монтирования:

```
mount --mkdir /var/backups/
```

5) Перейти в консоль TUI, затем выбрать **Обслуживание**.

6) После подключения резервного копирования в TUI будет отображено следующее



сообщение:

### 3.4 Обновление СВ “Звезда”

Необходимо выполнить пункты 1-4 установки СВ “Звезда”. Если была обнаружена ранее установленная версия СВ “Звезда”, то меню выбора целевого диска будет выглядеть следующим образом:

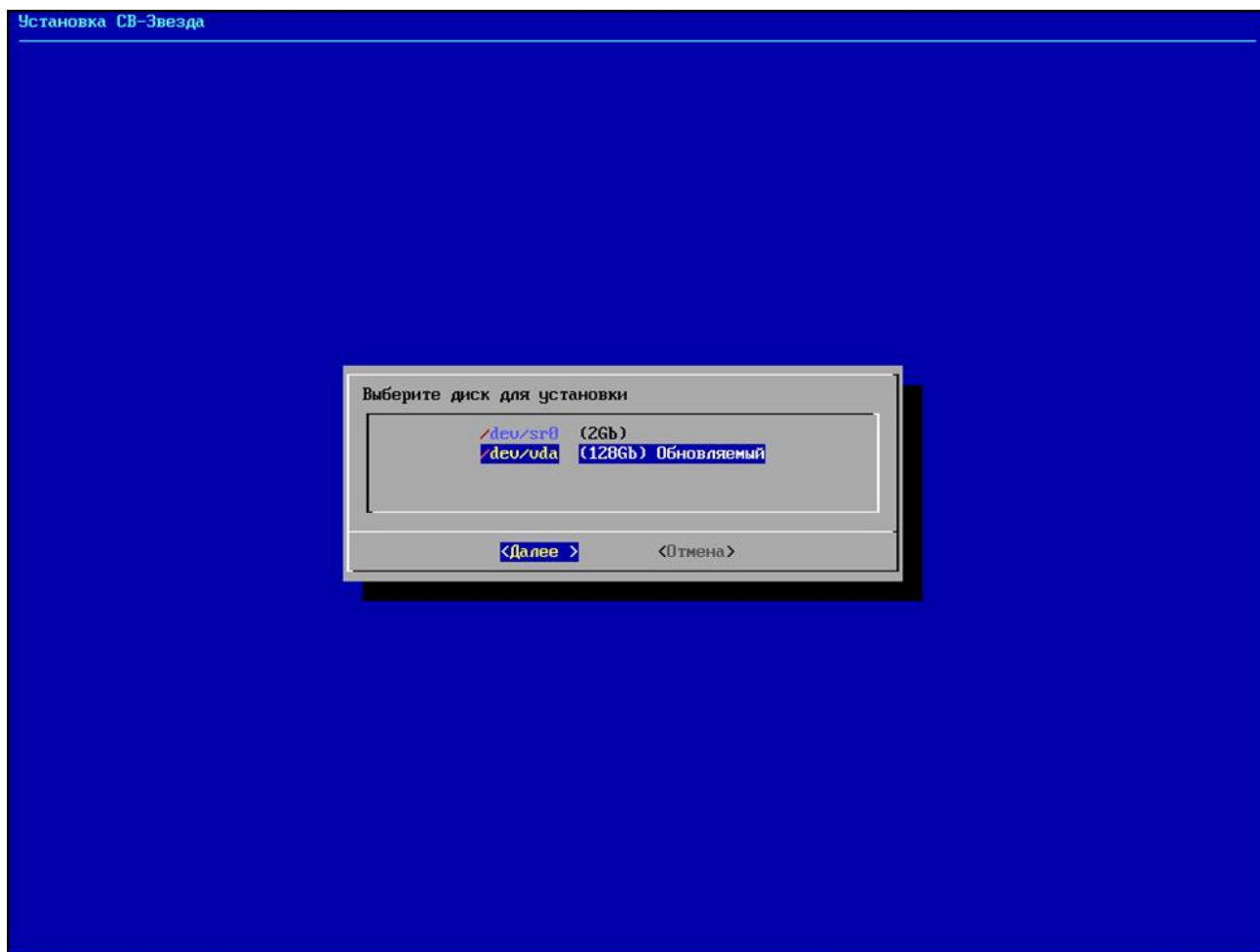


Рисунок 3.15 – Выбор диска

Отметка **Обновляемый** означает, что на данном целевом диске уже есть установленное средство виртуализации “Звезда”, ее можно обновить. После нажатия на данный диск будет доступен выбор:

- **Обновить.** Осуществляет безопасное обновление (с сохранением состояния системы);
- **Чистая установка.** Форматирует блочное устройство с установленным СВ “Звезда”, что приводит к полному удалению системы и произведенных изменений.

При выборе первого варианта система запросит подтверждения выполняемых действий.

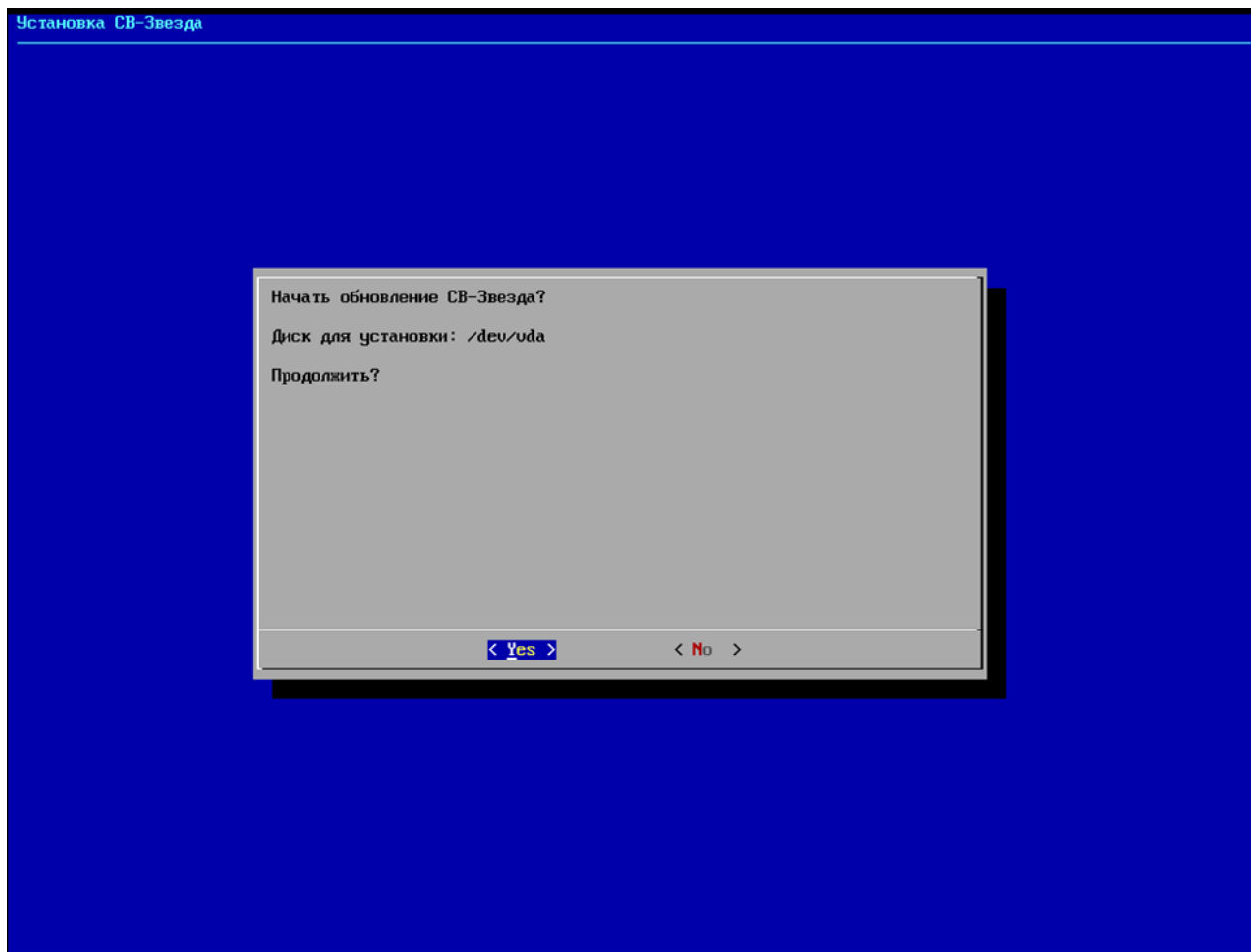


Рисунок 3.16 – Подтверждение обновления

Необходимо дождаться обновления системы.

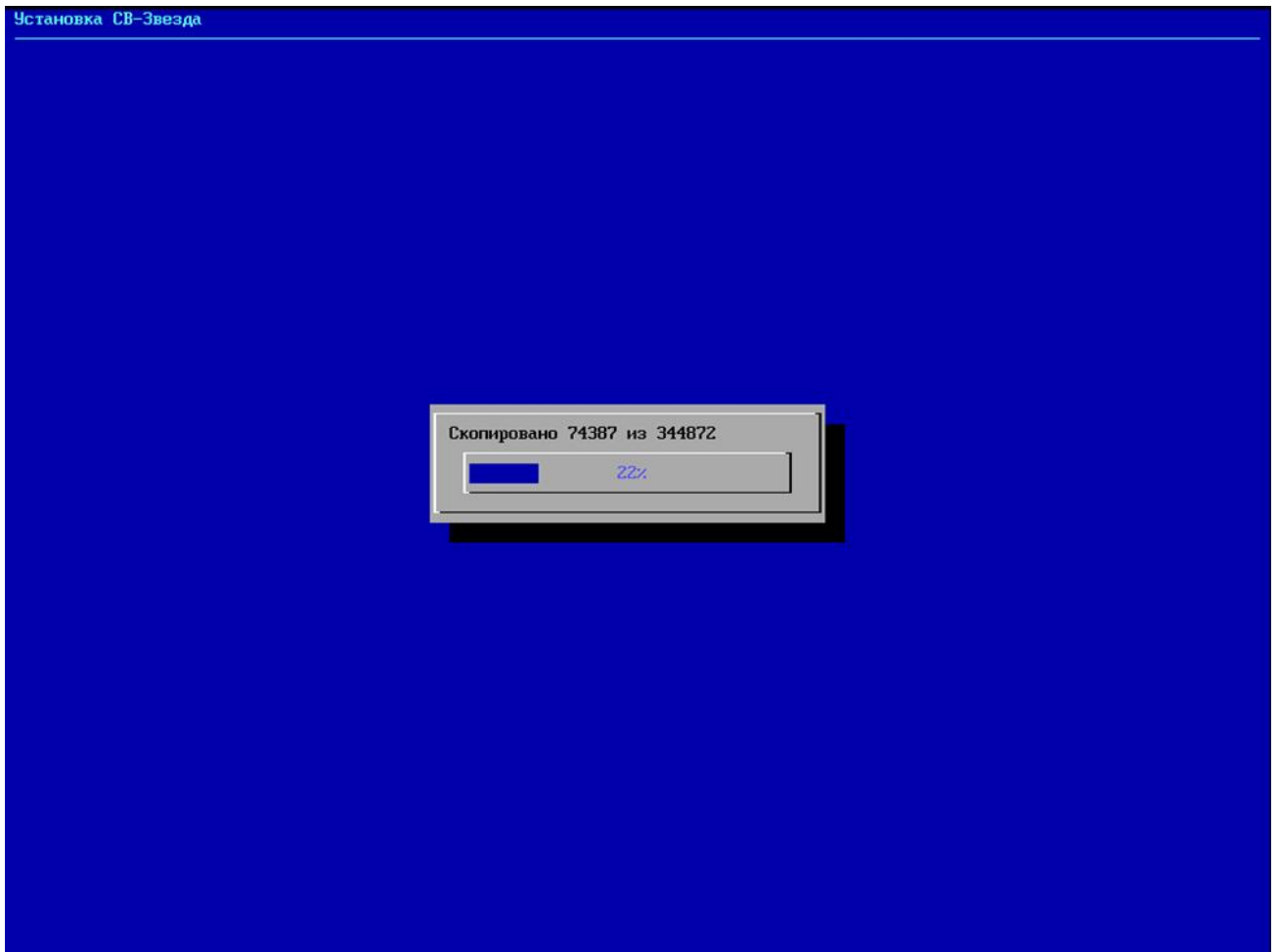


Рисунок 3.17 – Обновление системы

После окончания обновления, необходимо перезагрузить систему.

#### 3.4.1 Откат к предыдущей версии

Для отката к предыдущей версии и состоянию до обновления необходимо войти в TUI СВ “Звезда”, после чего выбрать **Дополнительно > Обслуживание > Обновления**.

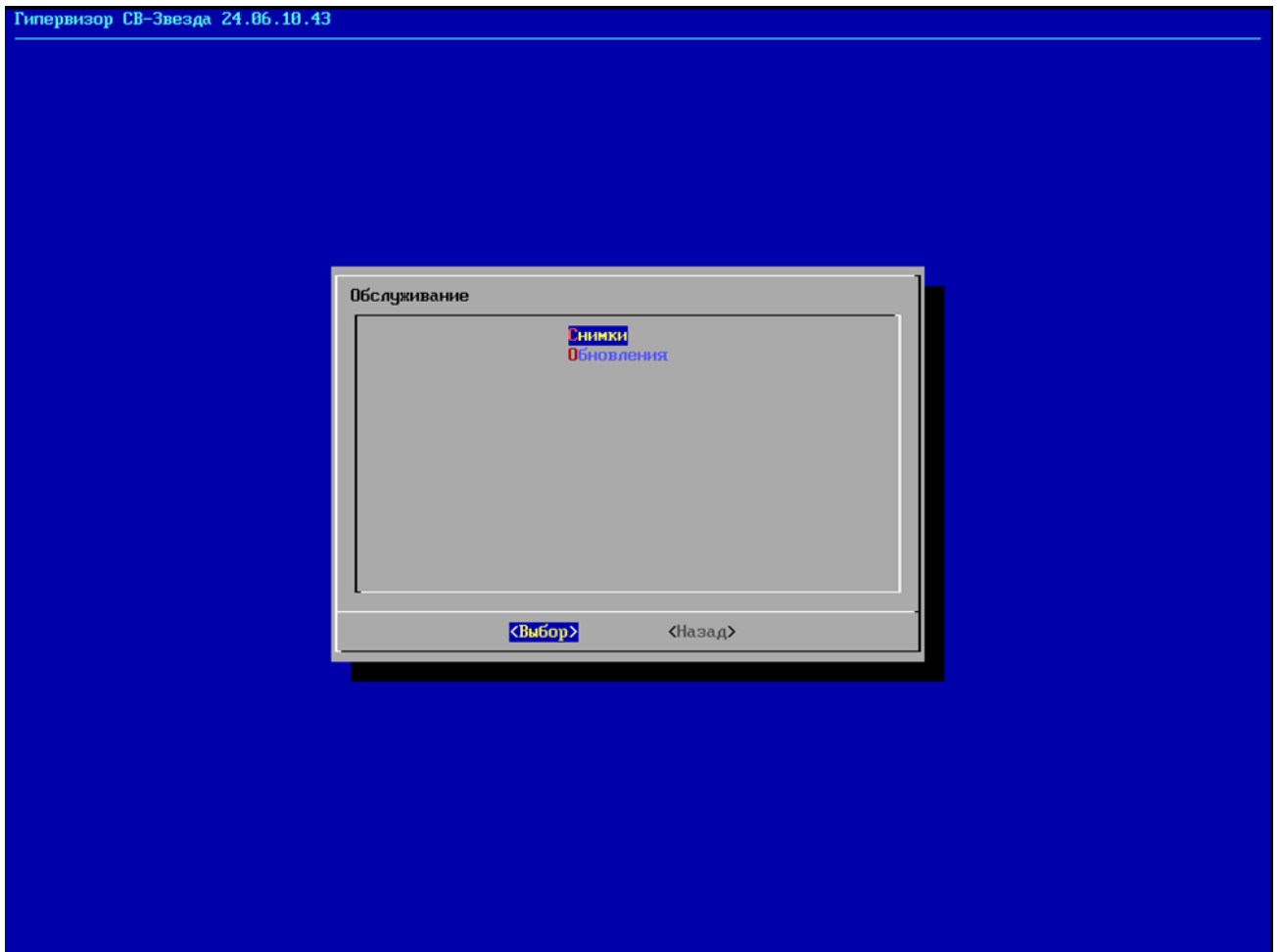


Рисунок 3.18 – Обслуживание

Выбрать версию, к которой нужно совершить откат.

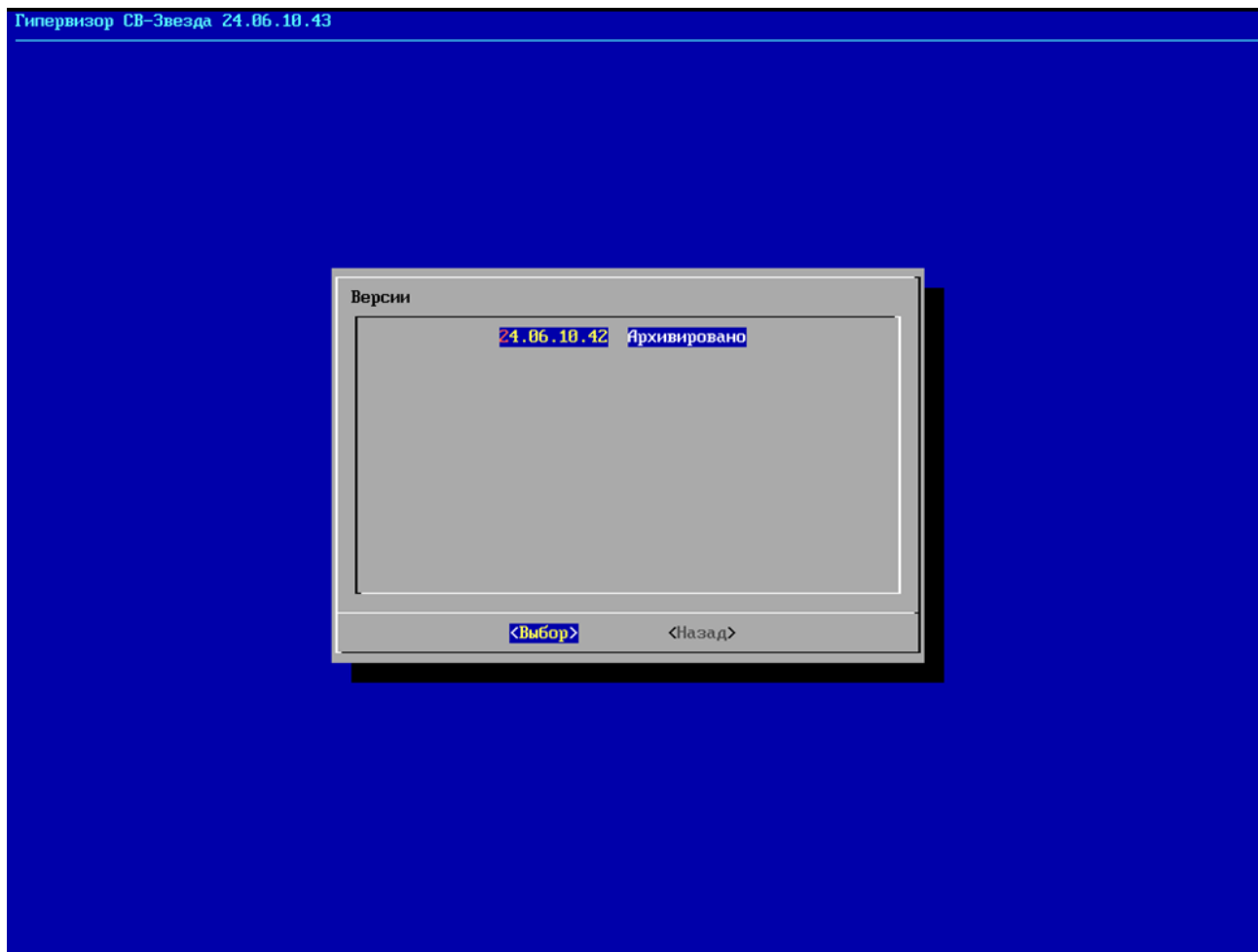


Рисунок 3.19 – Выбор версии

Для отката необходимо нажать **Выполнить откат версии**, для удаления версии - **Удалить версию**.

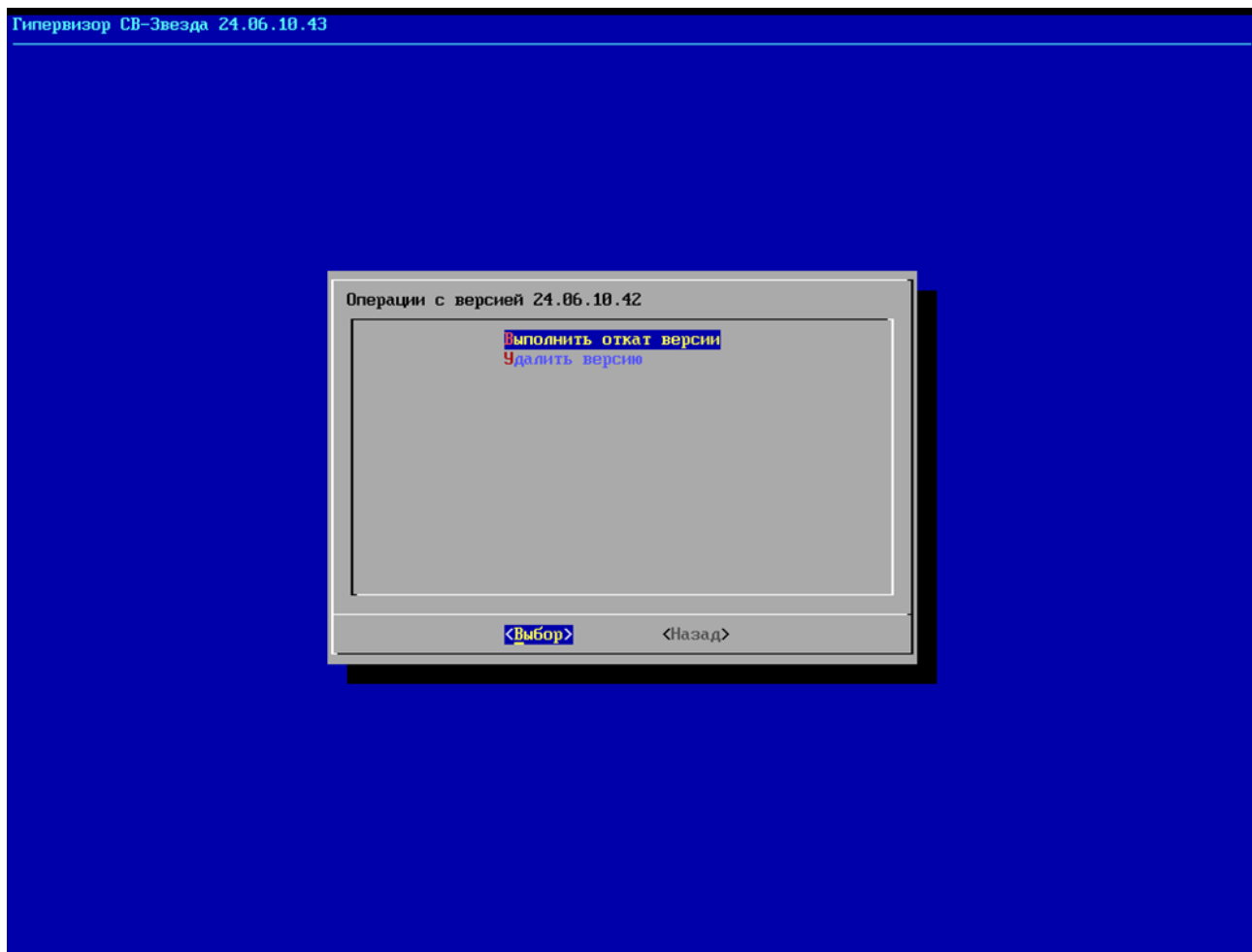


Рисунок 3.20 – Операции с версией

### 3.5 Изменение сетевых настроек

Для изменения IP-адреса необходимо выполнить следующие действия:

- 1) Открыть файл:

```
vim /etc/systemd/network/[наименование_интерфейса]
```

- 2) Отредактировать необходимые строчки. Пример содержания файла:

```
[Match]  
Name=sys0  
[Network]  
Address=10.10.105.20/24  
Gateway=10.10.105.1  
DNS=10.10.105.1
```

- 3) Выполнить перезагрузку СВ "Звезда".

## 4 ЗАЩИТА СРЕДЫ ВИРТУАЛИЗАЦИИ

### 4.1 Контроль целостности

В СВ “Звезда” реализован механизм изоляции программной среды. Все исполняемые файлы подписываются, запуск измененных файлов невозможен. Система производит динамический контроль целостности, при обнаружении изменений в директории /usr производится автоматическое восстановление файлов к исходной версии, созданной при инсталляции системы.

Для вычисления контрольной суммы директории (или директорий) необходимо выполнить команду:

```
gostsum-dir <путь к директории>
```

### 4.2 Идентификация и аутентификация пользователей

Создание пользователей и паролей для них осуществляется администратором. Администратор уполномочен устанавливать уникальное имя и идентификатор для пользователя. При вводе неправильного логина или пароля, система отобразит информацию о том, что вход не выполнен, будет предложено ввести правильное значение повторно. Учетная запись пользователя автоматически блокируется при достижении заданного количества неудачных попыток аутентификации с возможностью разблокировки администратором средства виртуализации или с возможностью автоматической разблокировки по истечении временного интервала, устанавливаемого администратором средства виртуализации (настройки приведены в п. 4.3.1). Максимальное количество неуспешных попыток аутентификации (ввода неправильного пароля) до блокировки – 4.

Аутентификация осуществляется с помощью локальной базы данных пользователей.

#### 4.2.1 Создание и удаление пользователей

Запрещено создавать пользователей, не принадлежащих ни одной роли (п.4.3). 1. Для добавления пользователя необходимо войти в СВ “Звезда” с использованием прав администратора (root). 2. Ввести команду:

```
useradd [имя_пользователя]
```

Далее необходимо настроить пароль созданного пользователя, введя команду `passwd [имя_пользователя]`.

Запрещено устанавливать одинаковые идентификаторы и пароли для разных пользователей. Пароль пользователя средства виртуализации должен содержать не менее 8 символов:

- прописные буквы английского алфавита от А до Z;
- строчные буквы английского алфавита от а до z;
- десятичные цифры от 0 до 9;
- специальные символы, не принадлежащие алфавитно-цифровому набору (например, \*\_@);
- в пароле должны отсутствовать повторяющиеся символы;
- пароль не должен иметь смысловой нагрузки.
- Для создания пользователя с уникальным идентификационным номером (ID) необходимо выполнить команду:

```
useradd -u 1000 [имя_пользователя]
```

где 1000 - это ID, который необходимо добавить пользователю.

- Если необходимо добавить учетную запись с истекающим сроком действия, необходимо выполнить команду:

```
useradd -e 2020-05-30 [имя_пользователя]
```

- Для добавления пользователя с комментарием, необходимо выполнить следующую команду:

```
useradd -c "комментарий" [имя_пользователя]
```

3) Для удаления пользователя, необходимо ввести команду:

```
userdel -r имя_пользователя
```

удалить домашнюю директорию и все связанные с пользователем файлы:

```
rm -rf /home/имя_пользователя
rm -f /var/spool/mail/имя_пользователя
```

Проверить, удалены ли данные пользователя:

```
cat /etc/passwd | grep имя_пользователя
ls /home | grep имя_пользователя
```

#### 4.2.2 Создание групп пользователей, добавление пользователя в группу

1) Для создания групп пользователей используется команда `groupadd [опции] [имя группы]`.

Доступные опции:

- `-f` - если группа уже существует, то утилита возвращает положительный результат операции;
- `-g` - установить значение идентификатора группы GID вручную;
- `-K` - изменить параметры по умолчанию автоматической генерации GID;
- `-o` - разрешить добавление группы с неunikальным GID;
- `-p` - задаёт пароль для группы;
- `-r` - указывает, что группа системная;
- `-R` - позволяет изменить корневой каталог

2) Для того чтобы добавить пользователя в группу, необходимо ввести команду `usermod` с ключами `-a` и `-G` группа.

```
usermod -a -G groupname username.
```

3) Для удаления пользователя из группы, необходимо ввести команду `gpasswd -d username groupname`.

#### 4.3 Управление ролевым доступом

В изделии поддерживаются следующие роли безопасности:

- разработчик виртуальной машины;

- администратор безопасности средства виртуализации;
- администратор средства виртуализации;
- администратор виртуальной машины;
- разработчик образов контейнеров;
- администратор безопасности средства контейнеризации;
- администратор информационной (автоматизированной) системы.

Порядок настройки ролей пользователей приведен в подразделах ниже.

#### **4.3.1 Создание роли администратора средств виртуализации**

Администратор безопасности средства виртуализации имеет права на следующие действия:

- создавать учетные записи пользователей средства виртуализации;
- управлять учетными записями пользователей средства виртуализации;
- назначать права доступа пользователям средства виртуализации к виртуальным машинам;
- создавать и удалять виртуальное оборудование средства виртуализации;
- изменять конфигурации виртуального оборудования средства виртуализации;
- управлять доступом виртуальных машин к физическому и виртуальному оборудованию;
- управлять квотами доступа виртуальных машин к физическому и виртуальному оборудованию;
- управлять перемещением виртуальных машин; Добавить команды миграции в конфиг файл для пользователя;
- удалять виртуальные машины;
- запускать и останавливать виртуальные машины;
- создавать снимки состояния виртуальных машин, включающих файл конфигурации виртуальной машины, образа виртуальной машины и образа памяти виртуальной машины.

Для создания роли администратора средства виртуализации необходимо выполнить следующие шаги:

1) Войти в СВ “Звезда” под учетной записью root.

2) Создать пользователя:

```
useradd adm_sv
```

3) Создать пароль для пользователя:

```
passwd adm_sv  
<ввести пароль для пользователя>
```

4) Открыть файл protector\_rgate.conf:

```
nano /etc/protector_rgate.conf
```

5) Добавить строчки:

```
adm_sv:useradd *  
adm_sv:userdel *  
adm_sv:scp *  
adm_sv:virsh *  
adm_sv:nano /etc/protector_rgate.conf  
adm_sv:virsh destroy *  
adm_sv:virsh undefine *  
adm_sv:virsh edit *  
adm_sv:passwd *
```

6) Сохранить изменения и закрыть файл.

### 4.3.2 Создание роли разработчика VM

Разработчик виртуальной машины имеет права на следующие действия:

- создавать виртуальные машины;
- изменять конфигурации виртуальных машин.

Для создания роли разработчика виртуальной машины необходимо выполнить следующие шаги:

- 1) Войти в СВ “Звезда” под учетной записью администратора средства виртуализации.
- 2) Создать пользователя командой:

```
rgate useradd dev_vm
```

- 3) Создать пароль для пользователя. Необходимо, чтобы пароль содержал не менее 8 символов при алфавите пароля не менее 70 символов. Выполнить команду:

```
rgate passwd dev_vm  
<ввести пароль для пользователя>
```

3) Открыть файл `protector_rgate.conf`:

```
rgate nano /etc/protector_rgate.conf
```

4) Добавить строчки:

```
dev_vm:virsh create *
dev_vm:virsh define *
dev_vm:virsh undefine *
dev_vm:nano *
dev_vm:qemu-img create *
```

5) Сохранить изменения и закрыть файл.

### 4.3.3 Создание роли администратора безопасности средства виртуализации

Администратор безопасности имеет права на следующие действия:

- иметь доступ на чтение к журналу событий безопасности средства виртуализации;
- формировать отчеты с учетом заданных критериев отбора, выгрузку (экспорт) данных из журнала событий безопасности средства виртуализации.

Для создания роли администратора безопасности необходимо выполнить следующие шаги:

- 1) Войти в СВ “Звезда” под учетной записью администратора средства виртуализации.
- 2) Создать пользователя командой:

```
rgate useradd adm_sec
```

- 3) Создать пароль для пользователя. Необходимо, чтобы пароль содержал не менее 8 символов при алфавите пароля не менее 70 символов. Выполнить команду:

```
rgate passwd adm_sec
<ввести пароль для пользователя>
```

3) Открыть файл `protector_rgate.conf`:

```
rgate nano /etc/protector_rgate.conf
```

4) Добавить строчки:

```
sec_adm:aureport
sec_adm:cat /var/log/protector/protector.log
sec_adm:cat /var/log/audit/audit.log
sec_adm:cat /var/log/protector/rgate.log
sec_adm:tail -f /var/log/protector/protector.log
```

```
sec_admin:tail -f /var/log/protector/rgate.log
sec_admin:journalctl *
sec_admin:journalctl
sec_admin:cat /var/log/auth.log
sec_admin:cat /var/log/boot.log
```

5) Сохранить изменения и закрыть файл.

#### 4.3.4 Создание роли администратора виртуальной машины

Роль администратора виртуальной машины позволяет осуществлять доступ к виртуальной машине. Администратор виртуальной машины является единственным пользователем, который имеет доступ к подключению ВМ по протоколу spice. Любая передача сведений о способах подключения к ВМ сторонним лицам запрещена.

Создание виртуальной машины описано в разделе 6.1.

Для создания роли администратора виртуальной машины необходимо выполнить следующие действия:

- 1) Войти в СВ “Звезда” под учетной записью администратора средства виртуализации.
- 2) Создать пользователя командой:

```
rgate useradd adm_vm
```

- 3) Создать пароль для пользователя. Необходимо, чтобы пароль содержал не менее 8 символов при алфавите пароля не менее 70 символов. Выполнить команду:

```
rgate passwd adm_vm
<ввести пароль для пользователя>
```

- 3) Предоставить администратору виртуальной машины необходимой информацией о ВМ для обеспечения административных функций (IP-адрес, порт, способ подключения, созданная учетная запись для администратора внутри ВМ).

#### 4.3.5 Создание роли разработчика образов контейнеров

Разработчик образов контейнеров имеет права на следующие действия:

- менять установленный администратором безопасности средства контейнеризации для разработчика пароль;
- создавать, модифицировать и удалять образы контейнеров.

Для создания роли администратора безопасности необходимо выполнить следующие шаги:

- 1) Войти в СВ “Звезда” под учетной записью администратора средства виртуализации.
- 2) Создать пользователя командой:

```
rgate useradd dev_img_docker
```

- 3) Создать пароль для пользователя. Необходимо, чтобы пароль содержал не менее 8 символов при алфавите пароля не менее 70 символов. Выполнить команду:

```
rgate passwd dev_img_docker  
<ввести пароль для пользователя>
```

- 3) Открыть файл `protector_rgate.conf`:

```
rgate nano /etc/protector_rgate.conf
```

- 4) Добавить строчки:

```
dev_img_docker:passwd dev_img_docker  
dev_img_docker:docker import *  
dev_img_docker:nano Dockerfile  
dev_img_docker:docker rmi *  
dev_img_docker:docker rm *  
dev_img_docker:docker images  
dev_img_docker:docker image *  
dev_img_docker:apply-ima  
dev_img_docker:docker build *  
dev_img_docker:docker update *
```

- 5) Сохранить изменения и закрыть файл.

#### 4.3.6 Создание роли администратора информационной системы

Роль администратора информационной (автоматизированной) системы позволяет:

- менять установленный администратором безопасности средства контейнеризации для администратора информационной (автоматизированной) системы пароль;
- запускать и останавливать контейнеры.

Для создания роли администратора безопасности необходимо выполнить следующие шаги:

- 1) Войти в СВ “Звезда” под учетной записью администратора средства виртуализации.

2) Создать пользователя командой:

```
rgate useradd adm_ias
```

3) Создать пароль для пользователя. Необходимо, чтобы пароль содержал не менее 8 символов при алфавите пароля не менее 70 символов. Выполнить команду:

```
rgate passwd adm_ias  
<ввести пароль для пользователя>
```

3) Открыть файл `protector_rgate.conf`:

```
rgate nano /etc/protector_rgate.conf
```

4) Добавить строчки:

```
adm_ias:passwd adm_ias  
adm_ias:docker run *  
adm_ias:docker stop *
```

5) Сохранить изменения и закрыть файл.

### **4.3.7 Создание роли администратора безопасности средства контейнеризации**

Роль администратора безопасности средства контейнеризации позволяет:

- назначать права доступа пользователям средства контейнеризации к образам контейнеров;
- создавать учетные записи пользователей средства контейнеризации;
- управлять учетными записями пользователей средства контейнеризации;
- иметь доступ на чтение к журналу событий безопасности средства контейнеризации (`rgate journalctl --no-pager | awk '/container/'`);
- формировать отчеты с учетом заданных критериев отбора, выгрузку (экспорт) данных из журнала событий безопасности средства контейнеризации.

Для создания роли администратора безопасности средства контейнеризации необходимо выполнить следующие шаги:

- 1) Войти в СВ “Звезда” под учетной записью администратора средства виртуализации.
- 2) Создать пользователя командой:

```
rgate useradd adm_sec_docker
```

- 3) Создать пароль для пользователя. Необходимо, чтобы пароль содержал не менее 8 символов при алфавите пароля не менее 70 символов. Выполнить команду:

```
rgate passwd adm_sec_docker  
<ввести пароль для пользователя>
```

- 3) Открыть файл `protector_rgate.conf`:

```
rgate nano /etc/protector_rgate.conf
```

- 4) Добавить строчки:

```
adm_sec_docker:docker logs *  
adm_sec_docker:journalctl *  
adm_sec_docker:useradd *  
adm_sec_docker:passwd *  
adm_sec_docker:nano /etc/protector_rgate.conf  
adm_sec_docker:docker events  
adm_sec_docker:cat /var/log/protector/protector.log  
adm_sec_docker:cat /var/log/protector/rgate.log *  
adm_sec_docker:docker inspect *  
adm_sec_docker:cat /var/log/protector/rgate.log
```

- 5) Сохранить изменения и закрыть файл.

#### **4.3.8 Журналирование фактов изменения ролевой модели**

Факты изменения ролевой модели пользователя фиксируются в файле `protector.log`. Для просмотра журнала, администратор безопасности средства виртуализации должен ввести следующую команду:

```
rgate cat /var/log/protector/protector.log
```

Администратору доступна информация о пользователе, который изменил ролевую модель, время, дата, какие именно изменения были внесены.

#### **4.3.9 Журналирование фактов создания, модификации и удаления образов контейнеров**

Для просмотра фактов создания, модификации и удаления образов контейнеров, администратору информационной системы необходимо выполнить следующие действия:

```
rgate cat /var/log/protector/rgate.log
```

В журнале будут доступны записи о фактах создания, модификации и удаления образов контейнеров.

#### **4.3.10 Журналирование неуспешных попыток аутентификации пользователей средства контейнеризации**

Для просмотра фактов неуспешных попыток аутентификации пользователей средства контейнеризации, администратору информационной системы необходимо выполнить следующие действия:

```
rgate cat /var/log/protector/rgate.log
```

В журнале будут доступны записи о фактах неуспешных попыток аутентификации пользователей средства контейнеризации.

#### **4.3.11 Журналирование запуска и остановки контейнеров**

Для просмотра фактов запусков и остановок контейнеров, администратору информационной системы необходимо выполнить следующие действия:

```
rgate cat /var/log/protector/rgate.log
```

В журнале будут доступны записи о фактах запусков и остановок контейнеров.

#### **4.3.12 Журналирование модификаций запускаемых контейнеров**

Для просмотра фактов модификаций запускаемых контейнеров, администратору информационной системы необходимо выполнить следующие действия:

```
rgate cat /var/log/protector/rgate.log
```

В журнале будут доступны записи о фактах модификаций запускаемых контейнеров.

#### **4.3.13 Журналирование успешных и неуспешных попыток аутентификации**

Для просмотра журнала успешных и неуспешных попыток аутентификации необходимо ввести команду:

```
rgate cat /var/log/auth.log
```

Администратору безопасности будет доступна информация о всех успешных и неуспешных попытках аутентификации.

#### **4.3.14 Журналирование доступа пользователей средства виртуализации к виртуальным машинам**

Просмотр журнала доступа пользователей к ВМ осуществляется с помощью команды:

```
rgate cat /var/log/libvirt/qemu/<имя_ВМ>.log
```

#### **4.3.15 Журналирование создания и удаления виртуальных машин**

Для просмотра журнала создания и удаления ВМ, администратору безопасности необходимо выполнить команду:

```
rgate cat /var/log/protector/rgate.log
```

Администратору будет доступна информация о том, какой пользователь выполнял команду создания и удаления ВМ.

#### **4.3.16 Журналирование запуска и остановки средства виртуализации**

Для просмотра журнала запуска и остановки средства виртуализации, администратору безопасности необходимо выполнить команду:

```
rgate cat /var/log/boot.log
```

Администратору будет доступна информация о запуске и остановке средства виртуализации.

#### **4.3.17 Журналирование запуска и остановки ВМ**

Для просмотра журнала запуска и остановки ВМ, администратору безопасности необходимо выполнить команду:

```
rgate cat /var/log/protector/rgate.log
```

Администратору будет доступна информация о том, какой пользователь выполнял команду запуска и остановки ВМ.

#### **4.3.18 Журналирование изменения конфигурации средства виртуализации**

Для просмотра журнала изменения конфигурации средства виртуализации выполнить команду:

```
rgate journalctl -k | grep -E 'sd[a-z]|block|disk'
```

Для дополнительной информации выполнить команду:

```
rgate journalctl -k -f
```

#### **4.3.19 Журналирование изменения конфигурации ВМ**

Для просмотра изменений конфигурации ВМ администратор безопасности должен ввести команду:

```
rgate cat /var/log/protector/protector.log
```

Администратору доступна информация о пользователе, который изменил конфигурацию ВМ, время, дата, какие именно изменения были внесены.

#### **4.3.20 Журналирование фактов нарушения целостности объектов контроля**

Для просмотра фактов нарушения целостности объектов контроля администратор безопасности должен ввести команду:

```
rgate cat /var/log/protector/protector.log
```

Администратору доступна информация о том, какие именно изменения были внесены.

#### **4.3.21 Журналирование доступа к образам контейнеров**

Для просмотра журнала доступа к образам контейнеров, администратору безопасности необходимо выполнить команду:

```
rgate cat /var/log/protector/rgate.log
```

Администратору будет доступна информация о доступе к образам контейнеров.

### **4.4 Управление потоками информации**

Средство виртуализации “Звезда” поддерживает управление потоками информации с помощью технологии локальных сетей (VLAN). Используются следующие механизмы:

- 1) Изоляция трафика виртуальной машин.
- 2) Маскировка исходящего трафика ВМ во внешнюю сеть.

### 3) Маршрутизация сети.

Изоляция сетей осуществляется с помощью многоуровневого коммутатора Open vSwitch, который позволяет фильтровать трафик сети. Для настройки изолированной сети необходимо настроить bridge с помощью консоли управления.

СВ “Звезда” поддерживает инструменты iptables для настройки и управления правилами фильтрации пакетов. Основные параметры и команды для настройки правил фильтрации:

#### 1) Просмотр текущих правил:

```
iptables -L
```

#### 2) Добавление правила для блокировки входящих пакетов от определенного IP-адреса:

```
iptables -A INPUT -s 192.168.1.100 -j DROP
```

#### 3) Разрешение входящих SSH соединений:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

#### 4) Удаление правила:

```
iptables -D INPUT -s 192.168.1.100 -j DROP
```

— iptables-save: Сохраняет текущие правила в файл.

— iptables-restore: Восстанавливает правила из файла.

— iptables -F: Очищает все правила в таблице.

## 4.5 Регистрация событий безопасности

### 4.5.1 Использование auditd

В СВ “Звезда” реализована регистрация событий безопасности. В качестве инструмента используется auditd (Linux Audit Daemon).

Для настройки параметров auditd необходимо выполнить следующие действия:

#### 1) Перейти к файлу конфигурации:

```
cd /etc/audit/auditd.conf
```

Файл конфигурации будет иметь следующий вид:

```

local_events = yes
write_logs = yes
log_file = /var/log/audit/audit.log
log_group = root
log_format = ENRICHED
flush = INCREMENTAL_ASYNC
freq = 50
max_log_file = 8
num_logs = 5
priority_boost = 4
name_format = NONE
##name = mydomain
max_log_file_action = ROTATE
space_left = 75
space_left_action = SYSLOG
verify_email = yes
action_mail_acct = root
admin_space_left = 50
admin_space_left_action = SUSPEND
disk_full_action = SUSPEND
disk_error_action = SUSPEND
use_libwrap = yes
##tcp_listen_port = 60
tcp_listen_queue = 5
tcp_max_per_addr = 1
##tcp_client_ports = 1024-65535
tcp_client_max_idle = 0
transport = TCP
krb5_principal = auditd
##krb5_key_file = /etc/audit/audit.key
distribute_network = no
q_depth = 2000
overflow_action = SYSLOG
max_restarts = 10
plugin_dir = /etc/audit/plugins.d
end_of_event_timeout = 2

```

В таблице представлены события аудита безопасности.

Таблица 4.1 – События аудита безопасности

№	Тип сообщения	Описание
1	USER_LOGIN	Запись о входе пользователя в систему
2	USER_LOGOUT	Запись о выходе пользователя из системы
3	LOGIN	Запись о попытке входа пользователя
4	SERVICE_START	Запись о запуске службы
5	SERVICE_STOP	Запись о остановке службы
6	SYSCALL	Запись о вызове системной функции
7	PROCTITLE	Запись о заголовке процесса, содержащая информацию о процессе

8	USER_MODIFY	Запись об изменении информации о пользователе (например, изменение пароля)
9	USER_CREATE	Запись о создании нового пользователя
10	USER_DELETE	Запись о удалении пользователя
11	FILE_ACCESS	Запись о доступе к файлу (чтение или запись)
12	FILE_MODIFY	Запись о модификации файла
13	FILE_CREATE	Запись о создании файла
14	FILE_DELETE	Запись о удалении файла
15	CONFIG_CHANGE	Запись об изменениях в конфигурации системы
16	LOGIN_FAILURE	Запись о неудачной попытке входа
17	NETWORK_ACCESS	Запись о доступе к сети (например, попытка подключения к удаленному хосту)
18	DEVICE_ACCESS	Запись о доступе к устройствам системы
19	MODULE_LOAD	Запись о загрузке модуля ядра
20	MODULE_UNLOAD	Запись об выгрузке модуля ядра
21	SETUID_EXEC	Запись о выполнении программы с установленным битом setuid
22	SETGID_EXEC	Запись о выполнении программы с установленным битом setgid
23	FILE_RENAME	Запись о переименовании файла
24	FILE_RESTORE	Запись о восстановлении файла
25	CAPABILITY_CHANGE	Запись об изменении привилегий (capabilities) процесса
26	SYSTEM_BOOT	Запись о загрузке системы
27	SYSTEM_SHUTDOWN	Запись о завершении работы системы

#### 4.5.1.1 *log\_file*

Для изменения каталога хранения файлов журнала аудита, необходимо изменить строчку `log_file = /var/log/audit/audit.log`.

#### 4.5.1.2 *max\_log\_file*

Параметр **max\_log\_file** указывает максимальный размер одного файла журнала аудита, который должен быть установлен для полного использования доступного пространства на разделе, где хранятся файлы журнала аудита. Данный параметр

определяет максимальный размер файла в мегабайтах. Указанное значение должно быть числовым.

#### 4.5.1.3 *max\_log\_file\_action*

Параметр **max\_log\_file\_action** определяет, какое действие будет выполнено после достижения установленного предела в параметре **max\_log\_file**; его следует установить на **keep\_logs**, чтобы предотвратить перезапись файлов журнала аудита.

#### 4.5.1.4 *space\_left*

Данный параметр определяет объем свободного места на диске. При достижении лимита объема диска срабатывает действие, установленное в параметре **space\_left\_action**. Необходимо указать число, которое дает администратору время для освобождения дискового пространства. Значение **space\_left** зависит от темпа генерации файлов аудита. Если значение **space\_left** указано как целое число, оно интерпретируется как абсолютный размер в мегабайтах (MiB). Если значение указано как число от 1 до 99, за которым следует знак процента (например, 5%), демон аудита вычисляет абсолютный размер в мегабайтах на основе размера файловой системы, содержащей файл журнала.

#### 4.5.1.5 *space\_left\_action*

Рекомендуется установить параметр **space\_left\_action** на значение **email** или **exec** с соответствующим методом уведомления.

#### 4.5.1.6 *admin\_space\_left*

Данный параметр определяет абсолютное минимальное количество свободного места, при достижении которого срабатывает действие, установленное в параметре **admin\_space\_left\_action**. Это значение должно быть установлено так, чтобы осталось достаточно места для регистрации действий, выполняемых администратором. Числовое значение для этого параметра должно быть меньше значения параметра **space\_left**. Вы также можете добавить знак процента (например, 1%) к числу, чтобы демон аудита вычислил число на основе размера раздела диска.

#### 4.5.1.7 *admin\_space\_left\_action*

Должен быть установлен в значение `single`, чтобы перевести систему в однопользовательский режим и позволить администратору освободить диск.

#### 4.5.1.8 *disk\_full\_action*

Параметр определяет действие, которое срабатывает, когда на разделе, где хранятся файлы журнала аудита, нет свободного места. Должен быть установлен в значение `halt` или `single`. Это обеспечивает остановку системы или переход в однопользовательский режим, когда аудит больше не может регистрировать события.

#### 4.5.1.9 *disk\_error\_action*

Параметр определяет действие, которое срабатывает в случае обнаружения ошибки на разделе, где хранятся файлы журнала аудита. Должен быть установлен в значение `syslog`, `single` или `halt` в зависимости от местных политик безопасности относительно обработки неисправностей оборудования.

#### 4.5.1.10 *flush*

Параметр должен быть установлен в значение `incremental_async`. Он работает в сочетании с параметром `freq`, который определяет, сколько записей может быть отправлено на диск перед принудительной синхронизацией с жестким диском. Параметр `freq` должен быть установлен на 100. Эти параметры гарантируют, что данные событий аудита синхронизируются с файлами журнала на диске, сохраняя при этом хорошую производительность для всплесков активности.

#### 4.5.1.11 *Запуск сервиса auditd*

После конфигурации файла необходимо запустить сервис. Для запуска сервиса `auditd` необходимо ввести команду:

```
systemctl start auditd.service
```

Статус сервиса можно проверить, введя команду:

```
systemctl status auditd.service
```

Убедиться, что сервис работает исправно, перейдя к log-файлам. Каталог хранения файлов журнала аудита указан в конфигурационном файле `log_file = /var/log/audit/audit.log`:

```
vim /var/log/audit/audit.log
```

Убедившись, что логи записываются, выйти из файла.

Команды управления сервисом `auditd`:

*Важно! Необходимо просмотреть файл `/usr/lib/systemd/system/auditd.service` и убедиться, что параметр `RefuseManualStop` имеет значение `no`. После изменения параметра необходимо перезапустить демон командой `systemctl daemon-reload`.*

— Для остановки сервиса:

```
systemctl stop auditd.service
```

— Для перезапуска сервиса:

```
systemctl restart auditd.service
```

— Для перезагрузки и принудительной перезагрузки конфигурационного файла:

```
systemctl reload auditd.service и systemctl force-reload auditd.service
```

Формирование отчетов

Для формирования отчета на основе журнала аудита (`audit.log`) используется утилита **aureport**. Она входит в состав пакета **auditd**, который отвечает за ведение журнала аудита безопасности системы.

В СВ “Звезда” реализован контроль доступа к файлу `audit.log`. Попытки доступа и формирования отчета заносятся в файле `protector.log`.

Примеры команд для формирования отчетов из `audit.log`:

1) Общий отчет по событиям аудита:

```
aureport
```

2) Отчет по событиям, связанным с пользователями:

```
aureport --user
```

3) Отчет по событиям входа в систему:

```
aureport --login
```

4) Отчет по событиям файловой системы:

```
aureport --file
```

5) Отчет по командам, выполненным пользователями:

```
aureport --command
```

6) Отчет по результатам попыток аутентификации:

```
aureport --auth
```

7) Отчет по сетевым событиям:

```
aureport --network
```

8) Формирование отчета в определенный период (например, за последние 24 часа):

```
aureport --start today --end now
```

9) Формирование отчета по конкретному типу события (например, события создания файлов):

```
aureport --summary --type file
```

**Дополнительные опции:** - Можно сохранить отчет в файл, используя перенаправление вывода:

```
aureport --user > user_report.txt
```

Утилита **aureport** помогает агрегировать данные из журнала `audit.log` и выводить их в читаемом формате для анализа.

#### 4.5.2 Использование Logrotate

Утилита `Logrotate` позволяет управлять журналами автоматизированно. Таким образом, можно задать необходимые условия и правила для выполнения действий, связанных с журналом. Например, доступна архивация журналов или передача их на другой сервер при достижении определенного размера, возраста или других критериев.

Проверка условий может быть настроена для выполнения ежедневно, еженедельно или ежемесячно.

#### 4.5.2.1 *Настройка logrotate*

Для настройки logrotate необходимо перейти в конфигурационный файл logrotate.conf: `vim /etc/logrotate.conf`. Пример открытого файла конфигурации:

```
# Default logrotate(8) configuration file for Gentoo Linux.
# See "man logrotate" for details.

# rotate log files weekly.
weekly
#daily

# keep 4 weeks worth of backlogs.
rotate 4

# create new (empty) log files after rotating old ones.
create

# use date as a suffix of the rotated file.
dateext

# compress rotated log files.
compress

notifempty
nomain
noolddir

# packages can drop log rotation information into this directory.
include /etc/logrotate.d

# no packages own wtmp and btmp -- we'll rotate them here.
/var/log/wtmp {
    monthly
    create 0664 root utmp
    minsize 1M
    rotate 1
}
/var/log/btmp {
    missingok
    monthly
    create 0600 root utmp
    rotate 1
}

# system-specific logs may also be configured here.
```

Переменные, с помощью которых осуществляется управление логами:

- 1) Периодичность выполнения проверок совпадения условий:
  - hourly - каждый час;
  - daily - каждый день;

- weekly - каждую неделю;
  - monthly - каждый месяц;
  - yearly - каждый год.
- 2) Основные инструменты:
- rotate - указывает сколько старых логов нужно хранить, в параметрах передается количество;
  - create - указывает, что необходимо создать пустой лог файл после перемещения старого;
  - dateext - добавляет дату ротации перед заголовком старого лога;
  - compress - указывает, что лог необходимо сжимать;
  - delaycompress - не сжимать последний и предпоследний журнал;
  - extension - сохранять оригинальный лог файл после ротации, если у него указанное расширение;
  - mail - отправлять Email после завершения ротации;
  - maxage - выполнять ротацию журналов, если они старше, чем указано;
  - missingok - не выдавать ошибки, если лог файла не существует;
  - olddir - перемещать старые логи в отдельную папку;
  - postrotate/endscript - выполнить произвольные команды после ротации;
  - start - номер, с которого будет начата нумерация старых логов;
  - size - размер лога, когда он будет перемещен.

Каждый лог, который подлежит ротации, описывается следующим образом:

```
адрес_файла_лога {  
  директивы  
}
```

#### 4.5.2.2 *Настройки для ротации лога*

Для примера будет использован файл `rsyslog.conf`, который был создан в папке `/etc/logrotate.d/`. Важно создать файл `messages` в директории `log`. Далее необходимо поместить в данный файл настройки ротации данного лога:

```
/var/log/messages {  
  daily  
  rotate 3  
  size 10M
```

```
compress  
delaycompress  
}
```

Данные настройки отображают следующее:

- `daily` - ротация журналов будет производиться ежедневно;
- `rotate 3` - будут храниться три последних журнала, более старые копии будут удалены;
- `size 10M` минимальный размер для ротации - 10 мегабайт, ротация не будет выполнена;
- `compress` - будет использовано сжатие;
- `delaycompress` - сжатие будет использовано для всех журналов, кроме последнего и предпоследнего.

По такому принципу возможно настроить ротацию логов для любого из журналов.

Для проверки необходимо ввести следующую команду:

```
logrotate -d /etc/logrotate.d/rsyslog.conf
```

В выводе команды должен быть отображен файл лога.

### 4.5.3 Работа с журналом `journald`

Настройка `journald` включает редактирование конфигурационных файлов, чтобы изменить параметры ведения журналов. Основной конфигурационный файл для `journald` находится в `/etc/systemd/journald.conf`. Основные шаги и параметры для настройки `journald`:

#### 1) Открыть файл конфигурации:

```
nano /etc/systemd/journald.conf
```

#### 2) Основные параметры конфигурации:

- `Storage` – определяет, где хранятся журналы. Возможные значения:
  - `volatile` – журналы хранятся в оперативной памяти и исчезают при перезагрузке.
  - `persistent` – журналы сохраняются на диске и сохраняются между перезагрузками.

- `auto` – по умолчанию. Если директория `/var/log/journal/` существует, журналы хранятся на диске, иначе – в оперативной памяти.
- `none` – отключает ведение журналов.

`Storage=persistent`

- `Compress` – указывает, должны ли сжиматься журналы. Значение `yes` включает сжатие, `no` – отключает.

`Compress=yes`

- `RateLimitInterval` и `RateLimitBurst` – параметры, контролирующие ограничение скорости записи сообщений. Например, следующие настройки ограничивают до 200 сообщений каждые 10 секунд:

`RateLimitInterval=10s`

`RateLimitBurst=200`

- `SystemMaxUse` – максимальное пространство на диске, которое может использовать `journald` для хранения журналов.

`SystemMaxUse=500M`

- `MaxFileSec` – максимальное время хранения файлов журналов. Например, чтобы хранить журналы не более 1 месяца:

`MaxFileSec=1month`

- `ForwardToSyslog` – отправлять ли журналы в `syslog`. Значение `yes` включает пересылку, `no` – отключает.

`ForwardToSyslog=yes`

### 3) Применение изменений:

После внесения изменений необходимо перезапустить службу `systemd-journald`:

```
systemctl restart systemd-journald
```

### 4) Дополнительные параметры:

Полный список параметров можно найти в руководстве по конфигурации `journald`:

`man journald.conf`

#### 4.5.3.1 Включение ротации в *journald*

В `systemd-journald` ротация журналов осуществляется автоматически в зависимости от параметров конфигурации. Однако, возможна настройка параметров, которые контролируют ротацию и очистку старых журналов. Важно правильно установить такие параметры, как максимальный размер использования дискового пространства, максимальный размер отдельных файлов журналов и максимальное время их хранения. Для этого:

##### 1) Открыть конфигурационный файл:

```
nano /etc/systemd/journald.conf
```

##### 2) Настроить параметры ротации и хранения:

В файле `/etc/systemd/journald.conf` вы можно настроить следующие параметры:

— `SystemMaxUse` – максимальное пространство на диске, которое может использовать `journald` для хранения журналов. Например, чтобы ограничить до 500 MB:

```
SystemMaxUse=500M
```

— `SystemKeepFree` – минимальное свободное пространство на диске, которое должно быть оставлено. Это полезно для предотвращения заполнения диска журналами. Например, чтобы оставлять минимум 100 MB свободного пространства:

```
SystemKeepFree=100M
```

— `SystemMaxFileSize` – максимальный размер одного файла журнала. Например, чтобы ограничить до 50 MB на файл:

```
SystemMaxFileSize=50M
```

— `MaxFileSec` – максимальное время хранения файлов журналов. Например, чтобы хранить журналы не более 1 месяца:

```
MaxFileSec=1month
```

— `SystemMaxFiles` – максимальное количество файлов журналов. Например, чтобы ограничить до 100 файлов:

```
SystemMaxFiles=100
```

3) **Сохранить файл и закрыть редактор.**

4) **Перезапустить службу `systemd-journald` для применения изменений:**

```
systemctl restart systemd-journald
```

Эти параметры будут автоматически регулировать ротацию и удаление старых журналов, основываясь на данных настройках. `journald` автоматически начнет удалять старые файлы журналов, когда суммарный объем будет превышать указанные лимиты или файлы станут старше указанного времени.

Для получения дополнительной информации о параметрах конфигурации `journald`, можно использовать команду:

```
man journald.conf
```

Это откроет руководство, где подробно описаны все доступные параметры и их использование.

#### **4.5.4 Система мониторинга системного журнала `journald-monitor`**

##### **4.5.4.1 Общие сведения**

Название: `journald-monitor`

Назначение: мониторинг системного журнала (`journald`) для обнаружения ошибок и отправки уведомлений администратору.

##### **Зависимости:**

- Утилита `journalctl` для чтения системного журнала.
- Утилита `mail` для отправки электронной почты.
- Файл конфигурации `/etc/sysconfig/journald-monitor`.

##### **4.5.4.2 Основные функции**

- 1) Автоматический мониторинг системного журнала на наличие ошибок.
- 2) Хранение текущего состояния курсора журнала в файле `/var/tmp/journal_cursor`.

- 3) Отправка уведомлений с ошибками по электронной почте.
- 4) Поддержка настройки через файл конфигурации.

#### 4.5.4.3 *Настройка ssmtp*

Используется лёгкий почтовый сервер ssmtp для отправки писем через SMTP.

#### 4.5.4.4 *Конфигурация ssmtp*

Основной файл конфигурации: /etc/ssmtp/ssmtp.conf.

Открыть файл для редактирования:

```
sudo nano /etc/ssmtp/ssmtp.conf
```

Настроить файл для работы с SMTP-сервером:

```
hostname=localhost
FromLineOverride=YES
AuthUser=example@yandex.ru
AuthPass=example-pass
mailhub=smtp.yandex.ru:587
rewriteDomain=yandex.ru
UseTLS=YES
UseSTARTTLS=YES
```

#### 4.5.4.5 *Установка прав доступа к конфигурации*

Установить права доступа:

```
sudo chmod 600 /etc/ssmtp/ssmtp.conf
```

#### 4.5.4.6 *Настройка файла aliases*

Создать или отредактировать файл /etc/ssmtp/revaliases:

```
sudo nano /etc/ssmtp/revaliases
```

Добавить запись:

```
root:example@yandex.ru:smtp.yandex.ru:587
```

#### 4.5.4.7 *Тестирование отправки писем*

Проверить работоспособность:

```
echo "Тестовое сообщение" | mail -s "Тестовая тема" -r example@yandex.ru
example@yandex.ru
```

#### 4.5.4.8 *Настройка journald-monitor*

Убедиться, что параметр MAIL\_FROM в /etc/sysconfig/journald-monitor соответствует настройкам smtp. Пример:

```
MAIL_FROM=example@yandex.ru
```

#### 4.5.4.9 *Файл конфигурации journald-monitor*

Создать или отредактировать файл /etc/sysconfig/journald-monitor:

```
# Email администратора для получения уведомлений
ADMIN_EMAIL=admin@domain.com

# Email отправителя
MAIL_FROM=example@yandex.ru

# Интервал мониторинга (в секундах)
INTERVAL=10
```

#### 4.5.4.10 *Настройка systemd*

Активировать сервис:

```
systemctl enable journald-monitor.service
systemctl start journald-monitor.service
```

#### 4.5.4.11 *Управление сервисом*

Запустить сервис:

```
systemctl start journald-monitor
```

Остановить сервис:

```
systemctl stop journald-monitor
```

Проверить статус:

```
systemctl status journald-monitor
```

#### 4.5.4.12 *Просмотр логов*

Получить логи работы:

```
journalctl -u journald-monitor.service
```

#### 4.5.4.13 *Принцип работы*

1) Загрузка конфигурации из файла /etc/sysconfig/journald-monitor.

- 2) Проверка состояния курсора в файле `/var/tmp/journal_cursor`.
- 3) Получение новых ошибок через `journalctl`.
- 4) Отправка уведомлений с ошибками по электронной почте.
- 5) Обновление состояния курсора.

#### **4.5.4.14 Решение проблем**

Отсутствие файла конфигурации: необходимо проверить наличие файла `/etc/sysconfig/journald-monitor` и корректность его параметров.

Не отправляются уведомления: необходимо убедиться в установке и настройке утилиты `mail`.

Сервис не запускается: проверить логи:

```
journalctl -u journald-monitor.service
```

#### **4.5.4.15 Резервное копирование журналов**

Возможность резервного копирования журналов `systemd-journald` реализуется путем копирования файлов журналов в безопасное место. Файлы журналов хранятся в формате бинарных файлов и обычно находятся в следующих директориях:

- `/var/log/journal/` (если настроено хранение журналов на диске)
- `/run/log/journal/` (если журналы хранятся в оперативной памяти и исчезают при перезагрузке)

Для настройки резервного копирования журналов используется `overlayfs`, которая создает резервную копию всей файловой системы.

##### **4.5.4.15.1 Настройка поля вывода**

В `systemd-journald` можно настроить вывод определенных полей, используя утилиту `journalctl`. Это особенно полезно, когда нужно фильтровать или форматировать журналы для более удобного анализа. Для этого:

- 1) **Фильтрация по определенным полям**

Можно использовать ключ `-o` (или `--output`) в сочетании с форматом `json` или `json-pretty`, чтобы получить вывод в формате JSON. Затем, используя утилиты обработки JSON (например, `jq`), выбрать конкретные поля. Пример:

```
journalctl -o json | jq '.MESSAGE, .PRIORITY'
```

## 2) Использование формата `short-monotonic`

Формат `short-monotonic` показывает только основные поля, включая метки времени, которые будут полезны при анализе.

```
journalctl -o short-monotonic
```

## 3) Использование формата `short-full`

Формат `short-full` показывает основные поля, включая метки времени, приоритет и идентификаторы:

```
journalctl -o short-full
```

## 4) Создание пользовательского формата

`journalctl` позволяет указать, какие поля отображать, используя формат `verbose`, но это может быть слишком подробным. Можно использовать `awk` или другие утилиты для обработки вывода `journalctl` и выбора конкретных полей. Пример:

```
journalctl -o json | jq '. | {timestamp: __REALTIME_TIMESTAMP, message: .MESSAGE}'
```

## 5) Фильтрация по конкретным критериям

Можно фильтровать вывод `journalctl` по конкретным полям прямо в командной строке. Например, чтобы отфильтровать сообщения от конкретного сервиса или с определенным приоритетом:

```
journalctl _SYSTEMD_UNIT=sshd.service
```

Или по приоритету (например, только ошибки и критические сообщения):

```
journalctl -p err
```

## 6) Использование `--output-fields`

С версии `systemd 246`, можно использовать `--output-fields` для вывода только определенных полей:

```
journalctl --output-fields=MESSAGE,PRIORITY,_PID
```

Примеры использования

Пример 1: Выводить только сообщения и приоритеты

```
journalctl -o json | jq '. | {message: .MESSAGE, priority: .PRIORITY}'
```

Пример 2: Выводить сообщения только от определенного сервиса с указанием времени

```
journalctl _SYSTEMD_UNIT=sshd.service -o json | jq '. | {time: .__REALTIME_TIMESTAMP, message: .MESSAGE}'
```

#### 4.5.5 Использование protector.log

Журнал protector.log осуществляет журналирование попыток входа в защищенные директории, входящие в программный модуль libprotector. Ниже представлен формат записи:

```
<unique_id> <timestamp> <username> <process_name> attempted to <action> protected resource <pathname>
```

#### 4.6 Резервное копирование с использованием файловой системы Overlayfs

В основе работы с бэкапами лежит Overlayfs (Overlay Filesystem), которая представляет собой наложенную файловую систему, позволяющую объединять два отдельных каталога (нижнюю и верхнюю директории) в одну единую файловую систему.

После монтирования содержимое нижней и верхней директорий будет доступно через объединенную точку монтирования /mnt/overlay. Любые изменения в этой директории будут записаны в верхнюю директорию, в то время как нижняя директория останется неизменной.

##### 4.6.1 Подробное описание слоев

Overlayfs функционирует посредством объединения двух директорий в слои:

##### 1) Нижний слой (lowerdir):

- Является неизменяемым и содержит исходные данные.
- Этот слой может быть общей директорией, доступной для чтения нескольким файловым системам Overlayfs.

- Может быть использован в различных сценариях, таких как предоставление базовой файловой системы, на которую накладываются изменения.

## 2) Верхний слой (**upperdir**):

- Является изменяемым и содержит все изменения, вносимые в файловую систему.
- Если файл изменяется или создается в объединенной точке монтирования, он будет записан в верхнюю директорию.
- Если файл в нижнем слое изменяется или удаляется, верхний слой содержит соответствующую информацию для обработки этих изменений.

## 3) Рабочая директория (**workdir**):

- Используется для внутренних операций Overlayfs.
- Эта директория необходима для того, чтобы верхний слой функционировал корректно.
- Рабочая директория должна находиться на том же файловом разделе, что и верхняя директория.

## 4) Промежуточные слои (**read-only layers**):

- Overlayfs поддерживает использование нескольких промежуточных слоев, которые являются неизменяемыми.
- Эти слои располагаются между нижней директорией и верхней директорией.
- Промежуточные слои позволяют создавать более сложные структуры файловых систем с множеством уровней неизменяемых данных.
- Пример монтирования с промежуточными слоями:

```
mount -t overlay overlay -o  
lowerdir=/mnt/lower:/mnt/lower2:/mnt/lower3,upperdir=/mnt/upper,workdir=  
mnt/work /mnt/overlay
```

## 4.7 Резервные копии виртуальных машин

### 4.7.1 Создание резервной копии

Для настройки резервного копирования виртуальных машин необходимо выполнить следующие действия:

- 1) Настроить директорию для хранения резервных копий. По умолчанию установлена директория `/var/backups/[имя_ВМ]`
- 2) По умолчанию, таймер для создания копий ВМ установлен на 00 ч., 00 мин., 0 сек. Для изменения расписания необходимо перейти в директорию `/usr/lib/systemd/system/backup-all-vm.timer`, затем после внесенных изменений выполнить команду:

```
systemctl daemon-reload
```

Для создания резервной копии ВМ необходимо выполнить команду:

```
systemctl start backup-all-vm.service &
```

По умолчанию, при отсутствии копии ВМ, создается полная резервная копия.

#### 4.7.2 Восстановление виртуальной машины

Прежде чем начать восстановление, важно убедиться, что были соблюдены все условия:

- была создана резервная копия диска ВМ;
- доступно хранилище для восстановления копий ВМ.

Если диск виртуальной машины, который необходимо восстановить, все еще работает или существует, нужно остановить ВМ и удалить текущий диск:

```
virsh shutdown имя_виртуальной_машины
```

Удаление диска (если необходимо):

```
virsh vol-delete имя_диска --pool имя_пула
```

Для восстановления виртуальной машины необходимо использовать команду `virtnbdrestore`. Эта команда восстановит данные с NBD-устройства на диск виртуальной машины.

Пример команды:

```
virtnbdrestore -i (ПУТЬ_К_БЕКАПУ) -o (КУДА_СОХРАНИТЬ_ОБРАЗЫ)
```

Убедиться, что диск был восстановлен.

#### **4.8 Защита памяти**

Для предотвращения доступа субъекта к остаточной информации в СВ “Звезда” реализована функция очистки памяти.

При создании образа диска на сервере виртуализации согласно условиям по безопасности создается дисковое пространство, заполненное нулями. Очистка осуществляется посредством перезаписи каждого байта создаваемого образа диска VM нулями. Очистка памяти используется при создании и удалении виртуальных машин (при создании образа диска для VM), таким образом предотвращается доступ субъектов к остаточной информации, а также после переполнения журнала событий.

## 5 УПРАВЛЕНИЕ И ЗАЩИТА КОНТЕЙНЕРОВ

### 5.1 Подпись контейнера

Для подписи контейнера необходимо выполнить следующие действия:

- 1) Импортировать образ контейнера в docker с помощью команды:

```
docker import [название_контейнера].tag.xz [имя_контейнера]
```

- 2) Применить IMA-подписи:

```
apply-ima
```

- 3) Запустить образ с помощью команды:

```
docker run --detach --privileged --cap-add=ALL --network=host --name  
[имя_контейнера] --entrypoint "" [имя_контейнера] /bin/bash -c "while true; do  
sleep 60; done"
```

- 4) Войти в контейнер:

```
docker exec -it [имя_контейнера] bash
```

*Внимание! Неподписанные контейнеры не будут запущены.*

Для запуска docker необходимо предварительно загрузить образ для контейнера, затем запустить контейнер с помощью команды `docker run [имя контейнера]`. Для того чтобы убедиться, что контейнер запущен, необходимо ввести команду `docker ps`. С помощью данной команды можно просмотреть список всех запущенных контейнеров.

### 5.2 Создание образов контейнеров

Для создания образа контейнера разработчик образов контейнеров должен выполнить следующие действия:

- 1) Войти в СВ “Звезда”.
- 2) Создать предварительно директорию `/var/tmp/template/cache` с помощью команды `mkdir`.

- 3) Загрузить полученные от разработчика СВ “Звезда” образ с шаблонами контейнеров с помощью утилиты scp в СВ “Звезда”:

```
scp /место/расположения/шаблонов/
dev_img_docker@10.10.102.102:/var/tmp/template/cache
```

В поле после “@” необходимо ввести IP-адрес СВ “Звезда”.

- 4) Создать bash-сценарий. Ниже представлен пример bash-сценария:

```
#!/bin/bash
#### Set Variables ####
hostname="pve01"
container_name="container104"
template_path="/var/tmp/template/cache"
network="custom_bridge"
ip="192.168.0.93"
subnet="192.168.0.0/24"
nameserver="8.8.8.8"
ram="1024m"
rootpw="password"
gateway="192.168.0.1"
bridge="vbr0"

#### Select Template ####
echo "Выберите шаблон:"
options=("alpine" "alt" "debian" "fedora" "ubuntu")
select distro in "${options[@]}; do
    if [[ -n "$distro" ]]; then
        template="${distro}-signed.tar.xz"
        image_name="${distro}-container"
        break
    else
        echo "Некорректный выбор. Попробуйте снова."
    fi
done

#### Create Network if Not Exists ####
if [[ ! "$(docker network ls | grep $network)" ]]; then
    echo "Создается сеть $network..."
    docker network create --subnet=$subnet $network
fi

#### Load from Template ####
if [[ ! "$(docker images -q $image_name 2>/dev/null)" ]]; then
    echo "Импорт шаблона из $template_path/$template..."
    cat "$template_path/$template" | docker import - "$image_name" /bin/sh
fi

#### Sign Container ####
echo "Подписываю контейнер $image_name..."
apply-ima

#### Run Container ####
docker run -d \
    --name $container_name \
    --hostname $hostname \
    --memory $ram \
```

```

--net $network \
--ip $ip \
--dns $nameserver \
--env "ROOT_PASSWORD=$rootpw" \
--restart unless-stopped \
"$image_name" /bin/sh -c "while true; do sleep 3600; done"

```

```
echo "Контейнер $container_name ($distro) создан и подписан успешно."
```

### 5.3 Централизованное управление образами контейнеров и контейнерами

Для просмотра списка всех контейнеров, которые были запущены, можно использовать команду `docker ps -a`.

```

$ docker ps -a
CONTAINER ID        IMAGE               PORTS              COMMAND           NAMES              CREATED
STATUS
305297d7a235      image1             "uptime"          distracted_goldstine 11 minutes ago
Exited (0) 11 minutes ago
ff0a5c3750b9      image2             "sh"              elated_ramanujan   12 minutes ago
Exited (0) 12 minutes ago

```

Для запуска более одной команды в `docker` необходимо запустить контейнер с флагом `-it`, который подключает интерактивный `tty` в контейнер.

Для централизованного хранения образов контейнера используется локальный реестр образов, который позволяет контролировать доступ в него. Данный реестр настраивается с помощью Docker CLI: `docker run -d -p 5000:5000 --name registry registry:2`. Так же можно тегировать образ для локального реестра с помощью команды `docker tag <image_id> localhost:5000/<repository>:<tag>`. Для загрузки образа в локальный реестр необходимо ввести команду `docker push localhost:5000/<repository>:<tag>`. Для загрузки образа из локального реестра необходимо ввести команду `docker pull localhost:5000/<repository>:<tag>`.

#### 5.3.1 Ограничение прав на использование вычислительных ресурсов контейнера

Для ограничения использования объема оперативной памяти в контейнере необходимо запустить контейнер со следующими параметрами:

```
docker run -d --name test_container --memory=512m ubuntu /bin/bash
```

## 5.4 Регистрация событий безопасности в контейнере

Для просмотра журнала контейнера, необходимо ввести команду `docker logs`. Необходимый контейнер должен быть запущен в этот момент. Далее необходимо узнать `container ID` с помощью команды:

```
docker ps
```

После этого можно посмотреть логи данного контейнера: `docker logs -f ff0a5c3750b9`. Журналы содержат данные выходного потока с отметкой времени. Приведенная выше команда не содержит непрерывный вывод журнала. Чтобы посмотреть непрерывный вывод журнала контейнера, нам нужно использовать параметр `-follow` в команде `docker logs`.

По умолчанию Docker хранит файлы журнала в выделенном каталоге на хосте с помощью драйвера журнала `json-file`. Каталог файла журнала — `/var/lib/docker/containers/<container_id>` на хосте, на котором запущен контейнер.

Для очистки журнала Docker необходимо выполнить команду:

```
truncate -s 0 /var/lib/docker/containers/*/*-json.log
```

Данная команда не удалит файл журнала, но удалит его содержимое.

## 5.5 Изоляция контейнеров

Основные механизмы для обеспечения изоляции и управления ресурсами в контейнерах включают `namespaces` и `cgroups`. Технологии `namespaces` и `cgroups` вместе обеспечивают изоляцию и контроль ресурсов для контейнеров, что позволяет запускать приложения в безопасной и управляемой среде.

### 5.5.1 Использование Docker для настройки Namespaces и Cgroups

В Docker `namespaces` и `cgroups` автоматически настраиваются при создании контейнеров, но можно задавать параметры вручную через флаги командной строки. Ниже приведены шаги по проверке корректной изоляции пространств имен контейнеров.

### 5.5.1.1 *Изоляция пространств имен для пользователей и групп контейнеров*

- 1) Создать пользователя dockremap:

```
sudo useradd -r -s /usr/sbin/nologin dockremap
```

Флаг `-r` создаёт системного пользователя. `-s /usr/sbin/nologin` запрещает вход в систему.

- 2) Проверить, что пользователь создан:

```
id dockremap
```

- 3) Добавить диапазоны UID/GID

```
echo "dockremap:100000:65536" | sudo tee -a /etc/subuid  
echo "dockremap:100000:65536" | sudo tee -a /etc/subgid
```

- 4) Проверить изменения:

```
cat /etc/subuid | grep dockremap  
cat /etc/subgid | grep dockremap
```

- 5) Вывод должен содержать:

```
dockremap:100000:65536
```

- 6) Перезапустить Docker:

```
systemctl restart docker
```

- 7) Запустите контейнер:

```
docker run -it --rm alpine sh
```

- 8) Проверьте user namespace:

```
lsns -t user
```

Ожидаемый результат — новый namespace для контейнера.

### 5.5.1.2 *Изоляция пространств имен хостов и доменов контейнеров*

Изоляция имен хостов контейнеров подразумевает существование разных hostname для каждого контейнера:

- 1) Запустить контейнер №1:

```
docker run -it --entrypoint "" ubuntufortest /bin/bash -c "while true; do sleep 60; done"
```

2) Войти в него:

```
docker ps
CONTAINER ID   IMAGE                                COMMAND                                            CREATED        STATUS
PORTS         NAMES
ab3e560b4de4   ubuntufortest                       "/bin/bash -c 'while...'                       6 seconds ago   Up 6
seconds
svz102 ~ # interesting_mccarthy
svz102 ~ # docker exec -it ab3e560b4de4 bash
```

3) Ввести команду hostname:

```
root@ab3e560b4de4:/# hostname
ab3e560b4de4
root@ab3e560b4de4:/#
```

4) Hostname был отображен. Выполнить шаги 1-4 для контейнера №2 и убедиться, что hostname для каждого контейнера уникальны.

### 5.5.1.3 Изоляция сетевых пространств имен контейнеров

1) Войти в контейнера №1 и №2, ввести команду ifconfig.

2) Выводы команд должны отличаться MAC-адреса.

### 5.5.1.4 Изоляция пространств имен для иерархии каталогов контейнеров

Каталоги контейнеров находятся в директории /var/lib/docker/containers/имя\_контейнера. Изоляция обеспечена существованием для каждого контейнера идентификационным номером, тем самым изменения в контейнере №1 никак не влияют на каталог контейнера №2.

Изоляция представлена на изображениях ниже.

Изоляция пространств имен для иерархии каталогов контейнеров

Рисунок 5.1 – Изоляция пространств имен для иерархии каталогов контейнеров

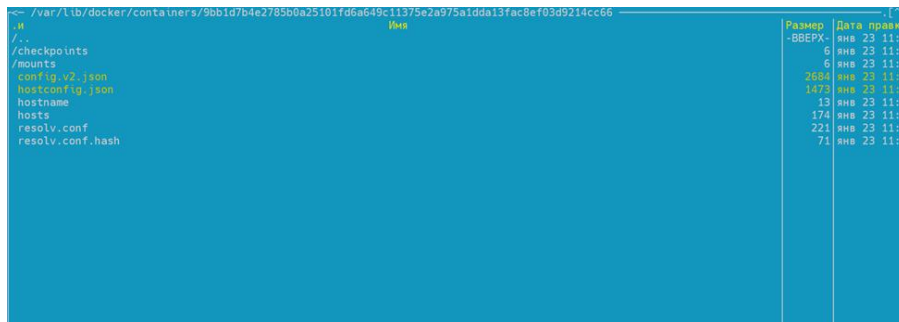


Рисунок 5.2 – Изоляция пространств имен для иерархии каталогов контейнеров

## 5.5.2 Аппаратная изоляция контейнеров

### 1) Пример запуска контейнера с ограничением использования памяти и процессора:

```
docker run -d --name my_container --memory=256m --cpus=0.5 my_image
```

Этот пример создаёт контейнер с лимитом в 256МВ памяти и ограничением использования процессора до 50%.

Технологии namespaces и cgroups вместе обеспечивают изоляцию и контроль ресурсов для контейнеров, что позволяет запускать приложения в безопасной и управляемой среде.

#### — Ограничение I/O:

```
docker run --device-write-bps /dev/sda:1mb nginx
```

#### — Ограничение путем назначения разных VLAN - необходимо настроить и назначить контейнерам разные VLAN:

##### 1) Создать контейнер с VLAN1, предварительно необходимо создать для контейнера сеть:

```
svz ~ # docker network create -d macvlan \
--subnet=10.10.106.0/24 \
--gateway=10.10.106.1 \
-o parent=eno49.10 eno49_10_net
```

##### 2) Запустить контейнер:

```
docker run --rm --detach --name ubuntu-eno49.10 --network eno49_10_net --
entrypoint "" ubuntu /bin/bash -c "while true; do sleep 60; done"
3c152d0f5b3d1467bd9e5def0139f89093fc5309dbf3dff5b192a34304e151e5
```

##### 3) Создать сеть для второго контейнера:

```
docker network create -d macvlan \
--subnet=10.10.107.0/24 \
--gateway=10.10.107.1 \
-o parent=eno49.20 eno49_20_net
```

##### 4) Запустить контейнер №2:

```
docker run --rm --detach --name ubuntu-eno49.20 --network eno49_20_net --
entrypoint "" ubuntu /bin/bash -c "while true; do sleep 60; done"
```

##### 5) Запустить на первом контейнере iperf3 для проверки соединения:

```
Docker exec -it ubuntu-eno49.20 iperf3 -c 10.10.106.2
```

б) Необходимо убедиться, что связи между контейнерами нет.

— Ограничение скорости сети. В примере приводятся ограничения до 100 Мбит/с и 1 Гбит/с:

### 1) Пример ограничения сети до 100 Мбит/с:

Используется команду `tc` для создания ограничения сетевой пропускной способности на уровне Docker. Например, интерфейс контейнера называется `eth0`.

Выполнить следующие команды для ограничения до 100 Мбит/с:

— Необходимо узнать PID процесса контейнера:

```
docker inspect --format '{{.State.Pid}}' limited-memory-ubuntu
```

— Выполнить команду `tc` для ограничения скорости (подставьте PID процесса контейнера в путь):

```
nsenter -t <PID> -n tc qdisc add dev eth0 root tbf rate 100mbit burst 32kbit latency 400ms
```

### 2) Измерение скорости с помощью iperf3:

Теперь можно измерить скорость передачи данных через `iperf3`. На хосте выполнить команду:

— На хосте запустить сервер `iperf3`:

```
iperf3 -s
```

— В контейнере запустить клиент `iperf3`:

```
docker exec -it limited-memory-ubuntu iperf3 -c <ip-хоста>
```

Это измерит скорость передачи данных между контейнером и хостом с учетом ограничения 100 Мбит/с.

### 3) Установка пропускной способности до 1 Гбит/с:

Для установки пропускной способности до 1 Гбит/с повторить команду с новым значением:

```
nsenter -t <PID> -n tc qdisc change dev eth0 root tbf rate 1gbit burst 32kbit latency 400ms
```

### 5.5.3 Монтирование корневой файловой системы хоста в контейнер в режиме “только для чтения”

Выполнить команды:

- Создать папку, например, `testdirectory`:

```
mkdir /root/testdirectory
```

- Выполнить команду:

```
docker run --rm -it --mount  
type=bind,source="$root/testdirectory",target=/app,readonly ubuntu /bin/bash
```

### 5.6 Выявление уязвимостей в образах контейнеров

Выявление уязвимостей в образах контейнеров проводится администратором средства контейнеризации при создании, запуске и перезагрузке контейнера. При обнаружении уязвимостей дальнейшее использование образа запрещено.

Разработчик образов контейнеров должен не реже одного раза в неделю осуществлять проверку уязвимостей в образах контейнеров сертифицированным ФСТЭК России средством (например, программным изделием «Средство обеспечения безопасности информационных систем MaxPatrol»). При появлении информации о новых уязвимостях в банке данных угроз безопасности информации производитель контейнера проводит с помощью сертифицированных средств проверку незамедлительно. Запрещено создание образов контейнеров, содержащих известные уязвимости критического и высокого уровня опасности.

### 5.7 Контроль целостности контейнеров и их образов

В изделии невозможен запуск неподписанных контейнеров. Проверка целостности происходит автоматически при загрузке и запуске образа.

## 5.8 Идентификация и аутентификация пользователей

Процесс идентификации пользователей происходит через встроенную систему аутентификации. Пароль и логин пользователя задает администратор системы, он же уполномочен изменить данные пользователя, а также удалить его.

## 5.9 Монтирование USB-устройства в контейнер

Для монтирования устройства необходимо выполнить следующие действия:

- 1) Подключить USB-устройство к хосту.
- 2) Убедиться, что устройство подключено, выполнив команду:

```
lsblk
```

- 3) Создать директорию, куда будет монтироваться устройство:

```
mkdir -p /mnt/usb
```

- 4) Монтировать устройство:

```
mount /dev/sde1 /mnt/usb0
```

Важно отметить, что вместо sde1 может быть другое название.

- 5) Проверить, смонтировано ли устройство:

```
lsblk
```

Пример вывода команды:

```
sda      8:0    0 745,2G  0 disk
├─sda1   8:1    0  128M  0 part  /boot/grub
├─sda2   8:2    0   32G   0 part  /run/initramfs/container
└─sda3   8:3    0 713,1G  0 part  /var
sdb      8:16   0 745,2G  0 disk
sdc      8:32   0 745,2G  0 disk
sdd      8:48   0 745,2G  0 disk
sde      8:64   1 116,2G  0 disk
└─sde1   8:65   1 116,2G  0 part  /mnt/usb
```

- 6) Запустить контейнер:

```
docker run -v /mnt/usb:/usb-data -it <имя_контейнера> /bin/bash
```

- 7) Убедиться, что устройство было примонтировано, запустив команду **lsblk**, затем перейти в **usb-data** (директория, в которое было смонтировано устройство), убедиться что данные устройства присутствуют.
- 8) Создать текстовый файл в устройстве:

```
touch test.txt
```

- 9) Убедиться, что файл появился в устройстве, выполнив команду:

```
ls
```

## 5.10 Просмотр журнала событий контейнера

Для просмотра журнала событий необходимо использовать команду:

```
docker events
```

Для импорта записей о событиях необходимо дополнить команду:

```
docker events > docker_journal.log
```

## 6 РАБОТА С ВИРТУАЛЬНЫМИ МАШИНАМИ

### 6.1 Создание виртуальной машины

Для работы с виртуальными машинами необходимо убедиться, что сервис `libvirtd.service` запущен. Для этого необходимо ввести команду `systemctl status libvirtd.service`.

Для создания виртуальной машины необходимо выполнить следующие действия:

#### Создание диска

1) Необходимо создать диск формата `qcow2`, для этого необходимо выполнить команду:

```
qemu-img create -f qcow2 ubuntu.qcow2 10G
```

Где 10G - размер диска.

2) Загрузить ISO-образ ОС в СВ “Звезда”.

3) Создать `.xml` файл в директории `/etc/libvirt/qemu` для ВМ, в которой будут указана конфигурация виртуальной машины: `virsh create vm1.xml`.

Пример XML-файла:

```
<domain type='kvm'>
  <name>test_vm</name>
  <memory unit='KiB'>1048575</memory>
  <currentMemory unit='KiB'>1048576</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <os>
    <type arch='x86_64' machine='pc-q35-3.1'>hvm</type>
    <loader type='rom'>/usr/share/OVMF/OVMF.fd</loader>
    <boot dev='hd' />
  </os>
  <features>
    <acpi/>
    <apic/>
    <paе/>
  </features>
  <clock offset='localtime' />
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
  <devices>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2' />
      <source file='/etc/libvirt/qemu/ubuntu.qcow2' />
      <target dev='vda' bus='virtio' />
    </disk>
    <disk type='file' device='cdrom'>
```

```

    <driver name='qemu' type='raw' />
    <source file='/etc/libvirt/qemu/ubuntu-22.04.4-live-server-amd64.iso' />
    <target dev='sda' bus='sata' />
    <readonly />
</disk>

<interface type="bridge">
  <mac address="02:00:45:d6:7b:15" />
  <source bridge="ovs-br0" />
  <virtualport type="openvswitch">
    <parameters interfaceid='63940821-5b8a-4ce6-b32a-9699de8d92bd' />
  </virtualport>
  <target dev="br2" />
  <model type="virtio" />
</interface>

<graphics type='spice' port='5900' autoport='no' listen='0.0.0.0' keymap='en-us'>
  <listen type='address' address='0.0.0.0' />
</graphics>
</devices>
</domain>

```

### 6.1.1 Добавление пароля для подключения по протоколу spice

Для добавления пароля для подключения по протоколу spice необходимо в конфигурации VM изменить строчку `graphics` следующим образом:

```

<graphics type='spice' port='5910' autoport='no' listen='0.0.0.0' keymap='en-us'
passwd='P@ssw0rd'>
  <listen type='address' address='0.0.0.0' />
</graphics>

```

`passwd='P@ssw0rd'` - пароль, который необходимо ввести при подключении по протоколу spice.

### 6.1.2 Добавление диска к VM

Для добавления диска к VM необходимо добавить строчку **disk**:

```

<disk type='file' device='cdrom'>
<source file='/home/createvm/image.iso' />
<target dev='sda' bus='ide' />

```

В строчке **type** необходимо указать тип диска, **device** - тип устройства, в строчке **source file** расположение диска, строчка **bus** - тип шины. В примере указано добавление CD-ROM диска.

- 1) Далее необходимо создать виртуальную машину на основе ранее созданного шаблона `.xml`:

```
virsh define vm1.xml
```

Где `setup.xml` - название шаблона `.xml`.

2) Удостовериться в успешном создании машины с помощью команды `virsh list --all`.

3) Далее необходимо запустить виртуальную машину:

```
virsh start test_vm
```

Где **test\_ubuntu** - имя ВМ, которое было указано в шаблоне `.xml`.

Операционная система была установлена.

Далее необходимо осуществить запуск и вход в операционную систему. Перед этим необходимо выполнить следующие действия:

4) Закрывать и удалять XML-файл, созданный для установки ВМ, перед этим необходимо выключить ВМ:

```
virsh shutdown test_vm
virsh undefine test_vm
```

5) Создать новый XML-файл для изменения запускаемого диска в файле с CD-ROM на жесткий диск. Остальные параметры можно оставить прежними.

6) Необходимо использовать подготовленный образ и конфигурационный XML-файл для создания и запуска виртуальной машины:

```
virsh define vm1.xml
virsh start test_vm
```

### 6.1.3 Подключение ВМ к сети

Для подключения ВМ к сети необходимо отредактировать параметр **interface**:

```
<interface type="bridge">
<mac address="02:00:45:d6:7b:15"/>
<source bridge="ovs-br0"/>
<virtualport type="openvswitch">
<parameters interfaceid="6154375c-a99b-4bc7-b528-9a082c1ab565"/>
</virtualport>
<target dev="test_ubuntu2"/>
<model type="virtio"/>
<address type="pci" domain="0x0000" bus="0x0c" slot="0x01" function="0x0"/>
</interface>
```

А именно:

— `interface type` - тип подключения (в данном случае представлен `bridge`);

- mac address - создается пользователем;
- source bridge - здесь необходимо ввести имя сети. Можно узнать с помощью команды `ovs-vsctl show`;
- virtualport type - тип виртуального порта;
- parameters interfaceid - уникальный номер интерфейса, который необходимо узнать с помощью команды `ovs-vsctl show`;
- target dev - имя сети VM, которое будет отображаться при выводе команды `ovs-vsctl show`;
- model type - тип шины;
- address type - тип устройства.

Необходимо добавить созданный порт VM в bridge. Для этого:

- 1) Выполнить команду `ip a`.
- 2) Найти в списке интерфейс VM.

```
svz1406 ~ # ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UP group default qlen 1000
   link/ether 02:00:71:81:08:75 brd ff:ff:ff:ff:ff:ff
   inet6 fe80::71ff:fe81:875/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
3: ovs-system: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether 92:ef:28:85:0d:d1 brd ff:ff:ff:ff:ff:ff
4: sys0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
   link/ether 0e:4d:bb:ea:b4:31 brd ff:ff:ff:ff:ff:ff
   inet 10.10.105.20/24 brd 10.10.105.255 scope global sys0
       valid_lft forever preferred_lft forever
   inet6 fe80::c4d:bbff:feea:b431/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
5: ovs-br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
   link/ether 0e:2e:ce:13:ff:49 brd ff:ff:ff:ff:ff:ff
   inet6 fe80::c2e:ceff:fe13:ff49/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
6: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
   link/ether 52:54:00:2c:2e:86 brd ff:ff:ff:ff:ff:ff
   inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
11: alpine: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UNKNOWN group default qlen 1000
   link/ether fe:00:45:d6:7b:15 brd ff:ff:ff:ff:ff:ff
   inet6 fe80::fc00:45ff:fed6:7b15/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
```

Рисунок 6.1 – Интерфейс VM

- 3) Убедиться, что bridge существует. При необходимости - создать его. По умолчанию, bridge создан в СВ “Звезда”(ovs-br0). Для создания bridge выполнить следующие действия:

```
ovs-vsctl show
ovs-vsctl add-br br1
ovs-vsctl add-br br2
```

- 4) Выполнить команду `ovs-vsctl add-port br0 test_ubuntu2`, где **test\_ubuntu2** - название интерфейса VM, br0 - bridge.
- 5) Необходимо настроить сеть в VM, после чего будет получен доступ в интернет.

#### 6.1.4 Редактирование VM

Для изменения параметров виртуальной машины необходимо отредактировать исходный файл формата .xml.

- 1) Для изменения названия VM отредактировать строку `<name>test_ubunt3</name>`.
- 2) Для изменения настроек сети отредактировать строки:

```
<interface type='bridge'>
  <virtualport type='openvswitch' />
  <source bridge='br1' />
  <mac address='02:00:71:81:18:01' />
  <target dev='br1-1' />
  <model type='virtio' />
</interface>
```

- 3) Для изменения дисковых параметров необходимо отредактировать следующие строки:

```
<disk type='file' device='disk'>
  <driver name='qemu' type='qcow2' />
  <source file='/fed1.qcow2' />
  <target dev='sda' bus='sata' />
  <address type='drive' controller='0' bus='0' target='0' unit='0' />
</disk>
```

Для помещения VM в 2000 VLAN необходимо выполнить команду `ovs-vsctl set port br1-1 tag=2000`. Таким образом, виртуальная машина не будет получать сетевые пакеты.

## 6.2 Снимки состояния VM

### 6.2.1 Создание снимка состояния VM

Для создания снимка состояния VM необходимо выполнить следующие действия:

- 1) Выполнить команду:

```
virsh snapshot-create-as <VM_name> <snapshot_name> --description
"<snapshot_description>"
```

Где:

— это имя вашей виртуальной машины. — это имя создаваемого снимка. — это описание снимка (необязательное поле).

2) Снимок VM был создан. Убедиться, что снимок был сделан, ввести команду:

```
virsh snapshot-list <VM_name>
```

3) Созданный снимок отобразен в списке.

Если необходимо сохранить не только диск, но и текущее состояние памяти VM, нужно добавить флаг `--live`:

```
virsh snapshot-create-as ubuntu-vm snapshot1 --description "Snapshot before update" --live
```

### 6.2.2 Восстановление VM с помощью снимка состояния VM

Для восстановления VM выполнить следующие действия:

1) Выполнить команду:

```
virsh snapshot-revert <VM_name> <snapshot_name>
```

2) Убедиться, что VM восстановилась, просмотрев список VM с помощью команды:

```
virsh list --all
```

### 6.2.3 Удаление снимка VM

Для удаления снимка VM ввести команду:

```
virsh snapshot-delete <VM_name> <snapshot_name>
```

## 6.3 Миграция VM с хоста на хост

Для перемещения виртуальных машин (VM) и их образов с сохранением конфигурации и настроек используется инструмент `virsh`.

Для выполнения миграции необходимо выполнить следующие действия:

1) Остановить виртуальную машину (если она запущена):

```
virsh shutdown имя_виртуальной_машины
```

— Проверить статус VM:

```
virsh list --all
```

2) Экспортировать конфигурацию VM в файл XML:

```
virsh dumpxml имя_виртуальной_машины > /path/to/имя_виртуальной_машины.xml
```

— Этот файл сохранит всю конфигурацию виртуальной машины.

3) Скопировать образ диска VM на новый сервер:

— Найти расположение дискового файла образа:

```
virsh domblklist имя_виртуальной_машины
```

— Скопировать образ на новый сервер с помощью `scp` или `rsync`:

```
scp /path/to/диск.qcow2 user@новый_сервер:/path/to/
```

Или:

```
rsync -av /path/to/диск.qcow2 user@новый_сервер:/path/to/
```

4) Скопировать XML-файл конфигурации на новый сервер:

```
scp /path/to/имя_виртуальной_машины.xml user@новый_сервер:/path/to/
```

5) Импортировать конфигурацию VM на новом сервере:

— На новом сервере выполнить команду для определения VM:

```
virsh define /path/to/имя_виртуальной_машины.xml
```

— Проверить, что VM определена:

```
virsh list --all
```

6) Запустить виртуальную машину на новом сервере:

```
virsh start имя_виртуальной_машины
```

7) VM перенесена на новый сервер с сохранением всех настроек и конфигураций.

## 7 СОЗДАНИЕ СНИМКОВ СИСТЕМЫ

Снимки в СВ “Звезда” - это резервные копии текущего состояния системы. Текущее состояние системы — это изменения, внесенные пользователем в процессе работы и сохраненные в системе. Эти изменения в силу архитектуры overlayfs сохраняются в отдельной директории, и при необходимости могут быть безопасно восстановлены. Безопасность архивирования/восстановления гарантируется тем, что происходит до момента сборки/монтажа системы, на этапе initramfs.

Необходимо убедиться, что дополнительное хранилище для снимков системы было подключено.

Для создания снимка необходимо выполнить следующие действия:

- 1) Перейти в TUI СВ “Звезда”.
- 2) Выбрать **Обслуживание**.
- 3) Выбрать **Снимки**.

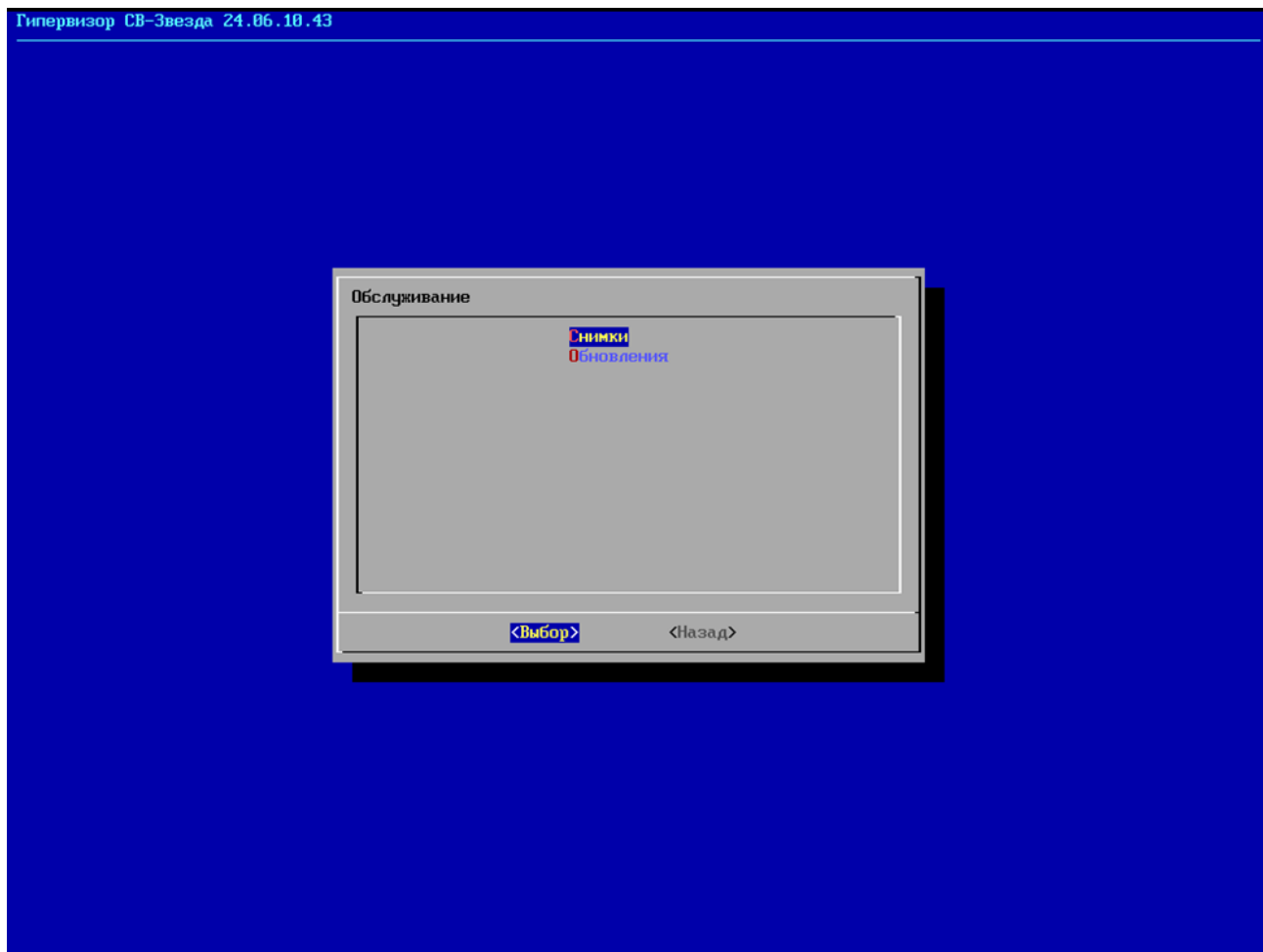


Рисунок 7.1 – Обслуживание

Будет доступен выбор: - Создать снимок; - Применить снимок; - Удалить снимок.

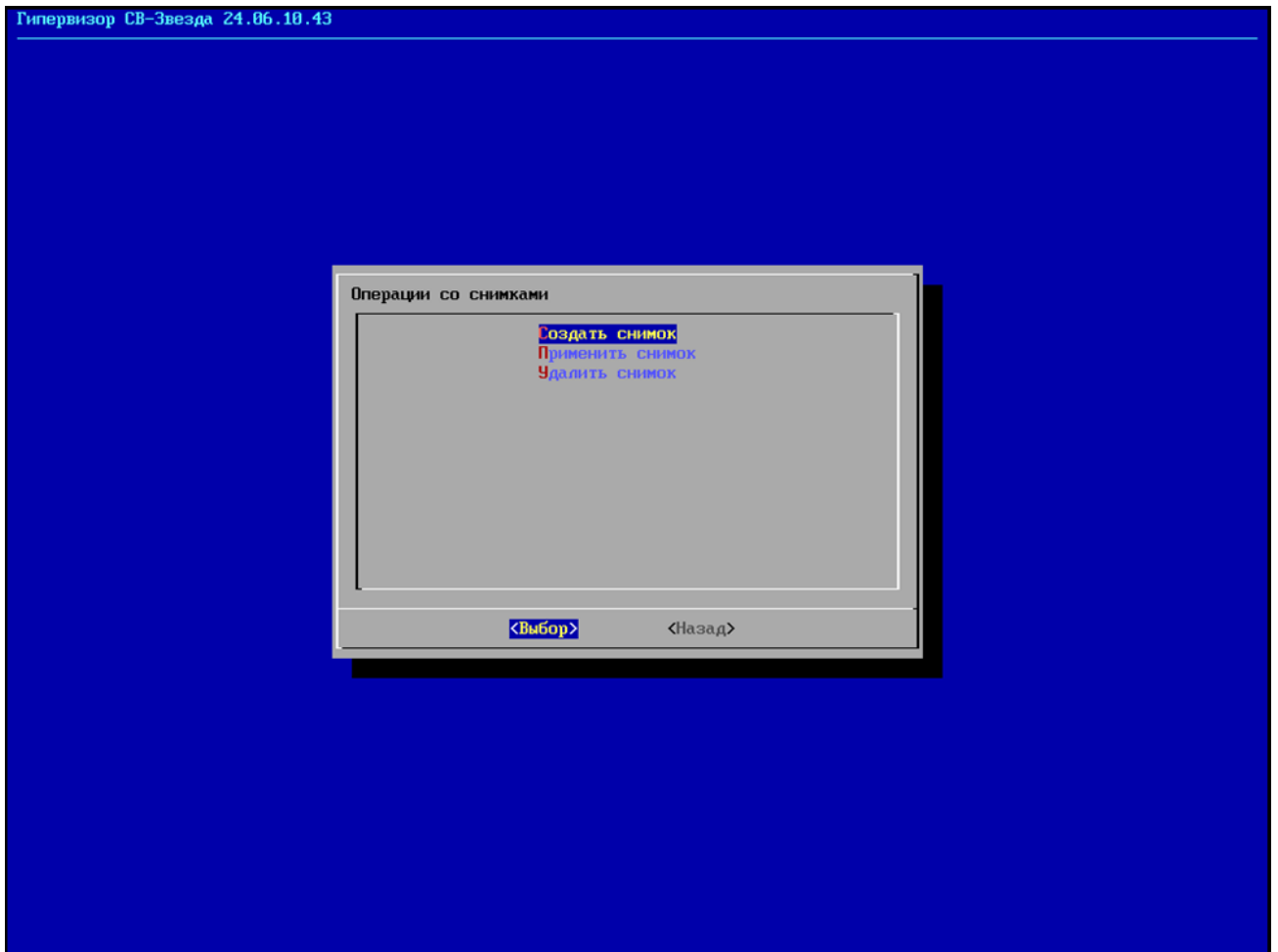


Рисунок 7.2 – Операции со снимками

При выборе **Создать снимок** создается скрипт, который выполняется при следующей загрузке и архивирует текущее состояние системы. После выбора **Создать снимок** средство виртуализации можно либо выключить, либо перезагрузить. После перезагрузки в меню **Обслуживание - Снимки - Применить снимок** будет доступен список снимков. При выборе нужного снимка TUI создаст скрипт, который при следующей загрузке восстановит состояние системы. Также после выбора снимка система спросит, нужно ли сохранить текущее состояние системы. Если подтвердить, то в снимках после перезагрузки появится соответствующий пункт.

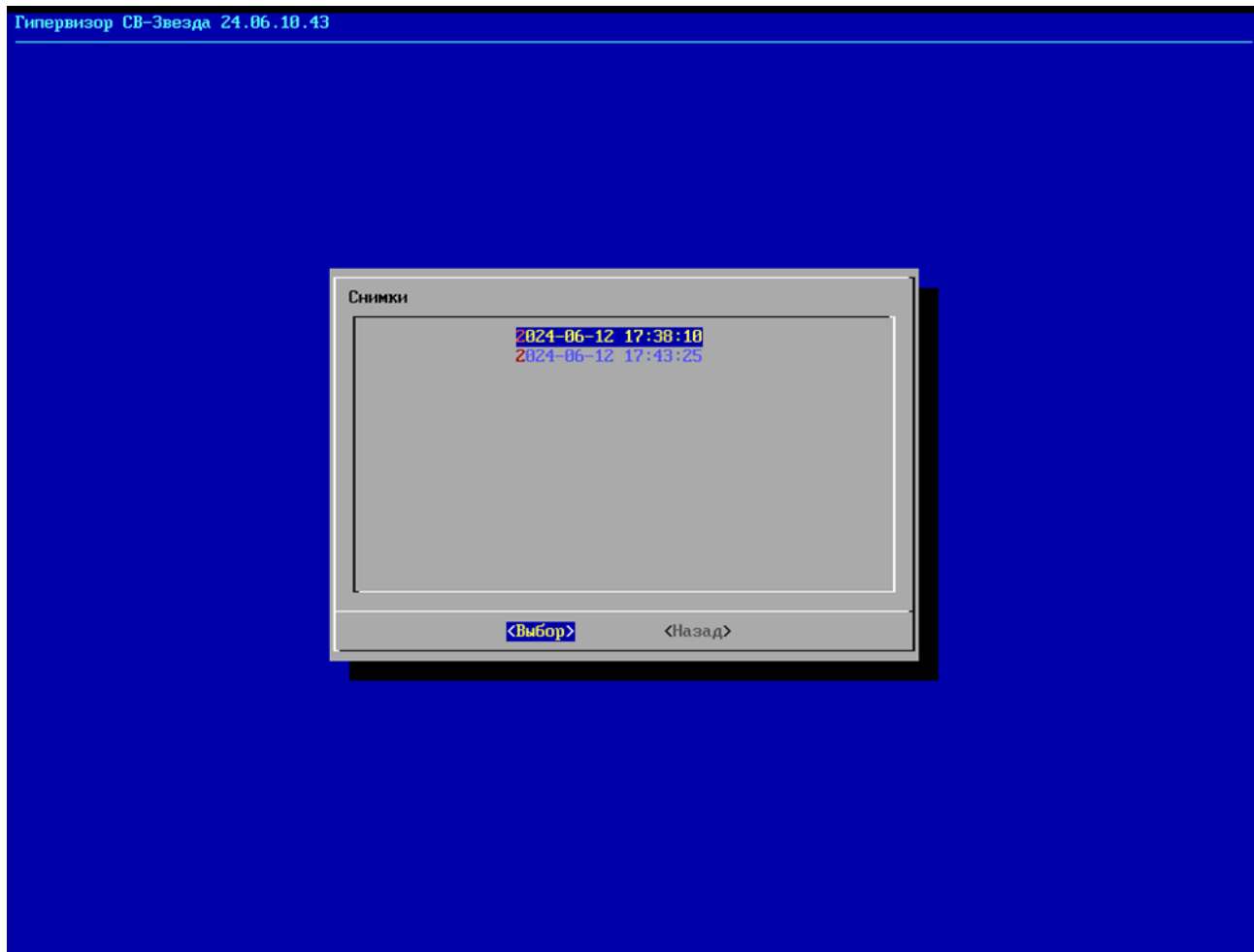


Рисунок 7.3 – Выбор снимка

## 8 УСТАНОВКА ПРИЛОЖЕНИЙ

Для установки приложений необходимо выполнить следующие действия:

- 1) Загрузить приложение в необходимую директорию.
- 2) Перейти в TUI СВ “Звезда”.
- 3) Выбрать **Дополнительно**.
- 4) Выбрать **Обслуживание**.
- 5) Выбрать раздел **Приложения**.



Рисунок 8.1 – Приложения

- 6) в разделе **Приложения** выбрать директорию, куда были загружены предварительно приложения.

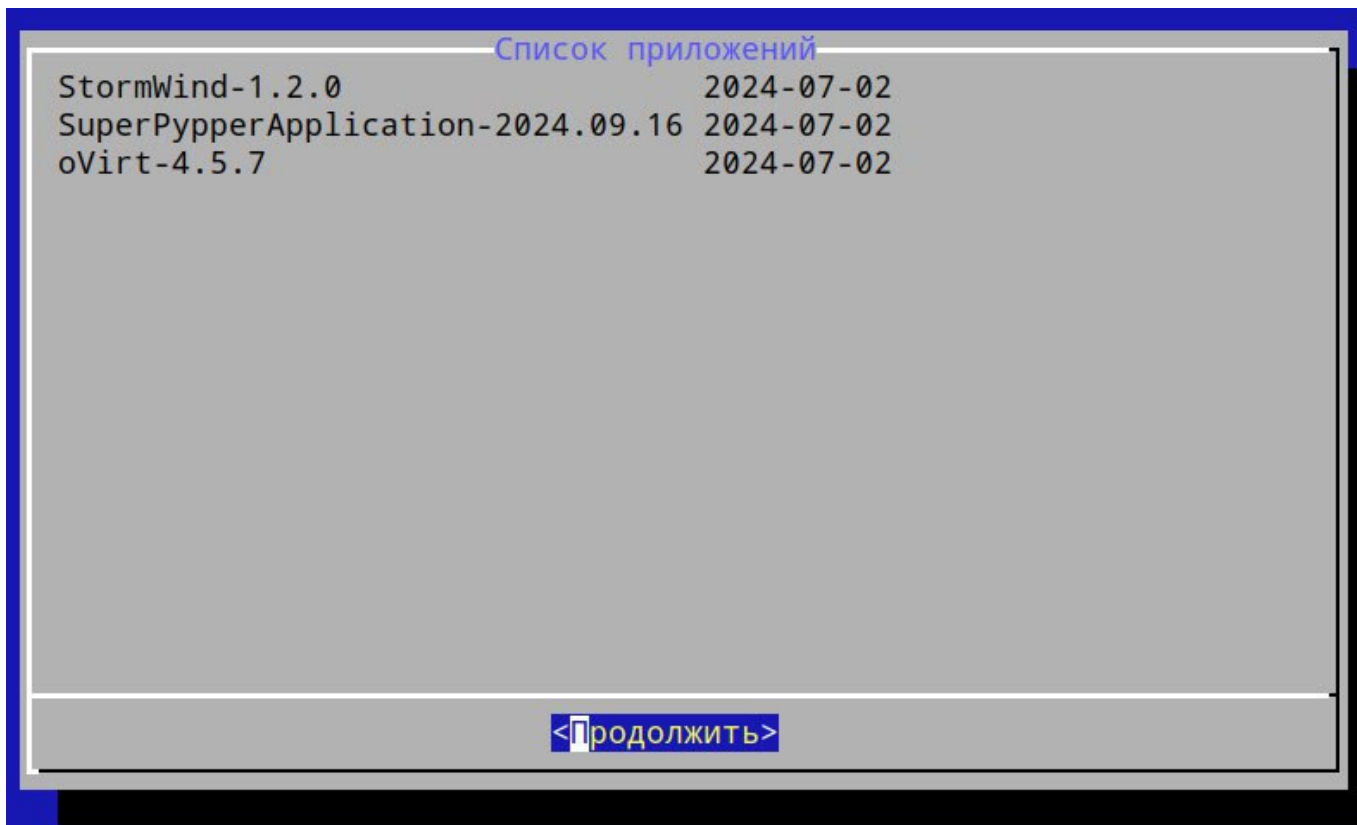


Рисунок 8.2 – Список доступных приложений

- 7) Нажать на приложение.
- 8) Начнется расшифровка приложения.

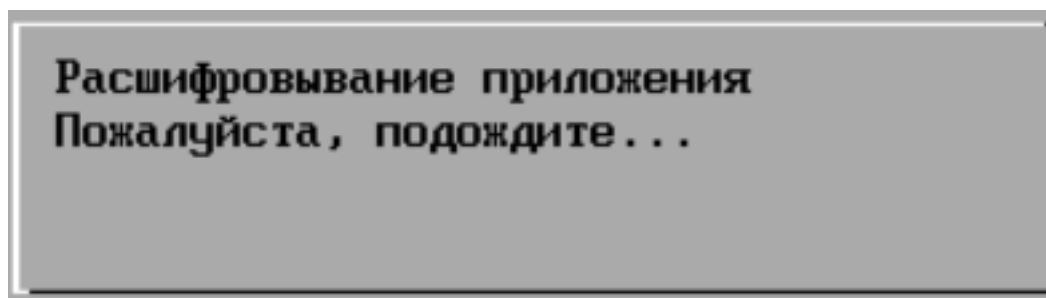


Рисунок 8.3 – Расшифровка приложения

- 9) Далее система предложит установить приложение.

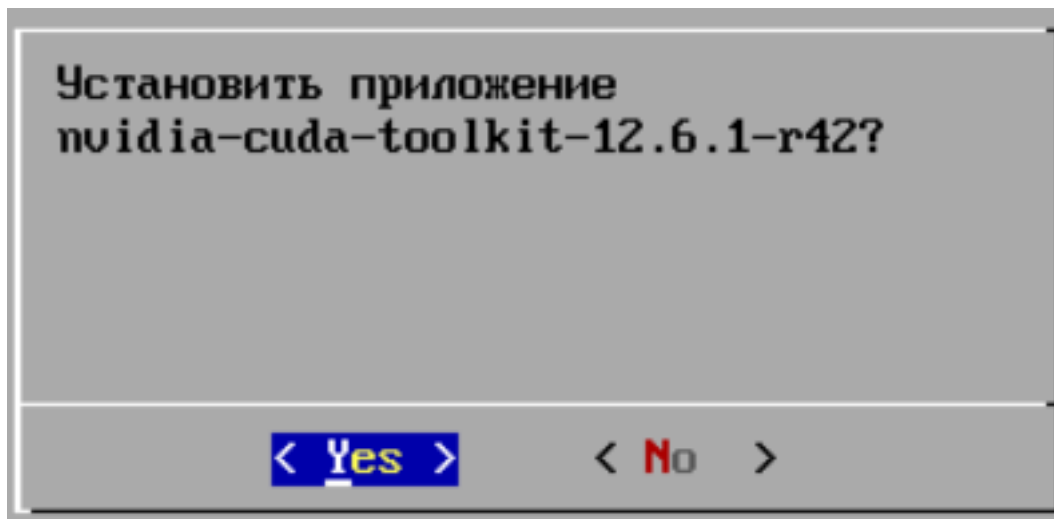


Рисунок 8.4 – Установка приложений

- 10) Дождаться распаковки и установки приложения.
- 11) После установки необходимо перезагрузить систему.

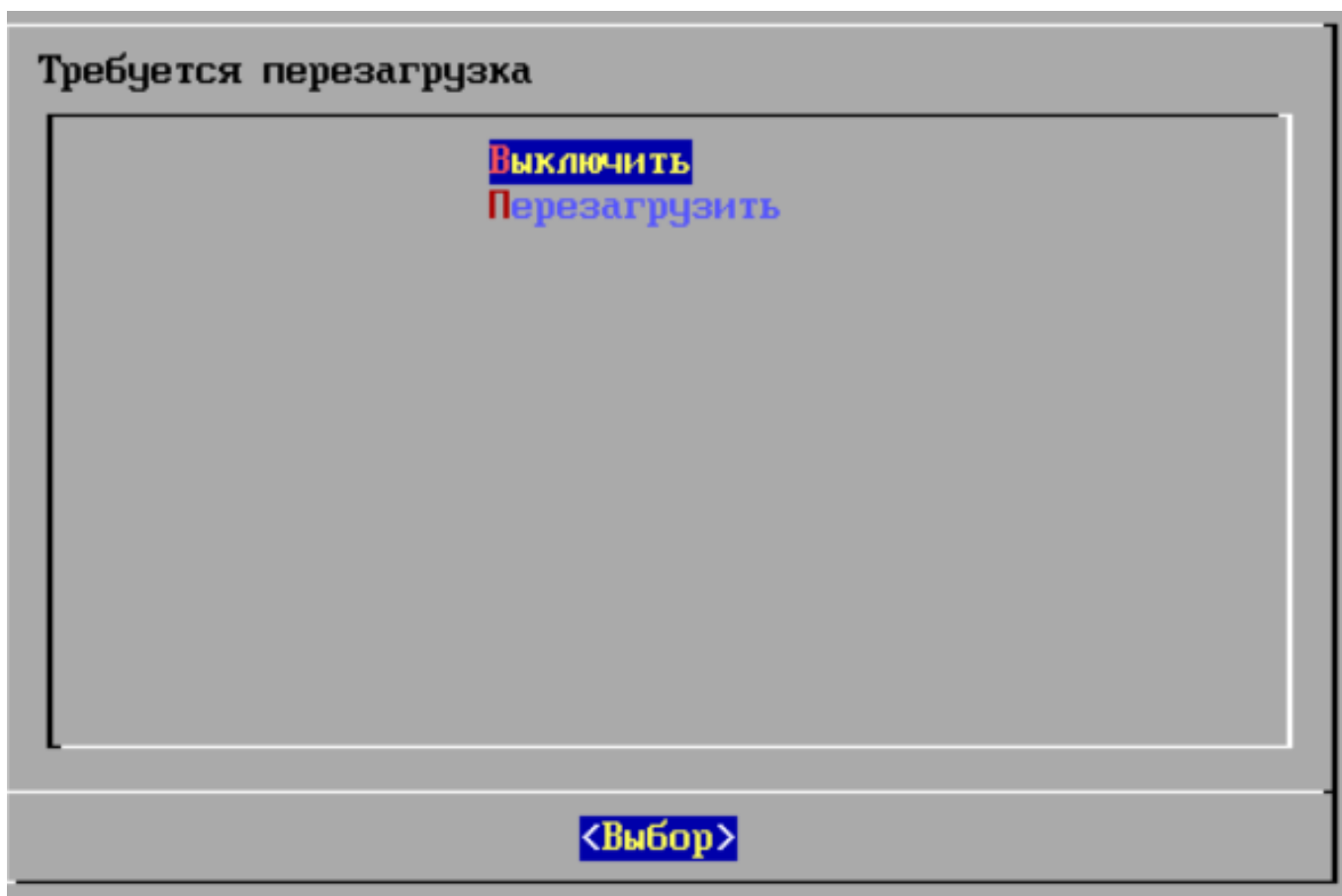


Рисунок 8.5 – Приложение установлено

Для удаления приложений необходимо выбрать **Удалить** в менеджере приложений. Появится список приложений, для удаления нажать кнопку **Выбор**.

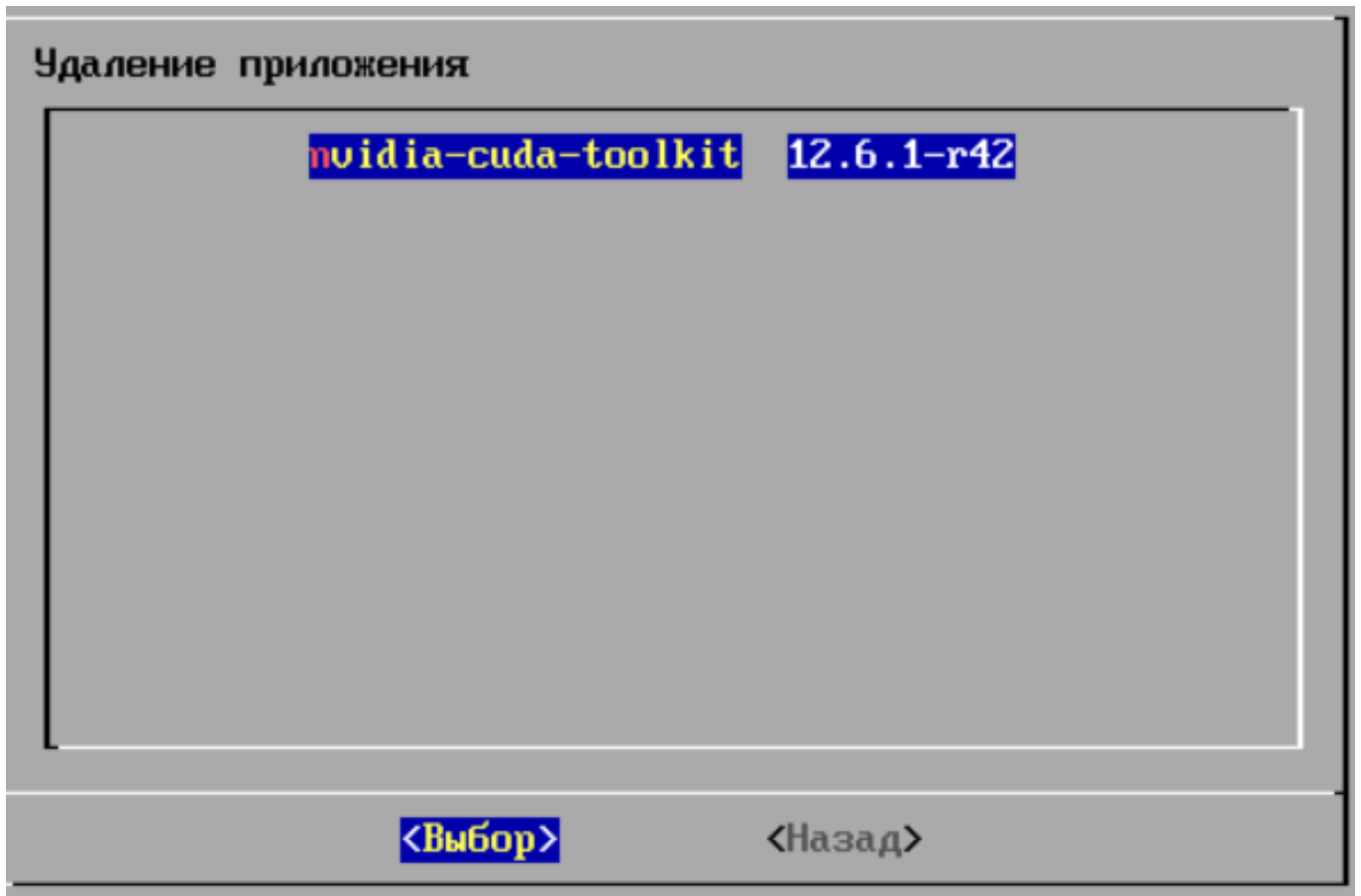


Рисунок 8.6 – Удаление приложения

После удаления приложения необходимо выполнить перезагрузку системы.

