R-Vision

Программный комплекс «Р-Вижн ЭВО». Руководство по установке.

Версия 1.01

ОГЛАВЛЕНИЕ

Общие замечания по установке

Программный комплекс «Р-Вижн ЭВО» включает в себя набор функциональных блоков (компонентов) отвечающих за решение разнообразных задач, связанных с процессами управления ИБ. Функциональные блоки могут устанавливаться как совместно на одном техническом средстве (сервер или виртуальная машина), так и на разных в зависимости от их типа согласно данной инструкции.

1. УСТАНОВКА ФУНКЦИОНАЛЬНЫХ БЛОКОВ УПРАВЛЕНИЯ: АКТИВАМИ, УЯЗВИМОСТЯМИ, ИНЦИДЕНТАМИ, АУДИТАМИ И РИСКАМИ ИБ

1.1. Технические требования

Ниже представлены рекомендуемые требования к аппаратному обеспечению для размещения модулей в зависимости от масштабов ИТ-инфраструктуры.

Система предполагает возможность установки на любые серверы с архитектурой x86 независимо от производителя. Возможна установка компонентов системы как на виртуальную (таблица 1), так и на физическую инфраструктуру.

Таблица 1. Общие требования к программному обеспечению

Характеристика	**Поддерживаемое ПО**
Среда виртуализации	VMware, MS Hyper-V, Xen, Parallels, VirtualBox
Тип диска	SSD
Операционная система сервера	Astra SE 1.6, Astra SE 1.7, Astra CE 2.12, RED OS 7.3, RED OS 7.3c ALT Server 10 (p 10.x), ALT 8 SP Server (c 8.2)CentOS 7.5 - 7.9, RHEL 7.7 – 9.2, Debian 10.x - 11.1,
СУБД	PostgreSQL 11.14, Jatoba J1
Браузер клиента	Google Chrome (97 и выше), Firefox (96 и выше), Edge (97 и выше)

Корректная работа системы гарантируется при масштабе страницы браузера и экрана в операционной системе, установленном по умолчанию (100%).

Таблица 2. Рекомендации по распределению дискового пространства в операционной системе по разделам

Раздел	Назначение раздела	Рекомендуемое свободное пространство раздела
/	Раздел под ОС	Более 15 ГБ (зависит от требований ОС)
/opt/	Раздел под файлы, скрипты, docker-image, docker-volume	Зависит от конфигурации. См. таблицы 3, 4, 5, 6.
/tmp/	Раздел для установки/обновления, docker-compose	Более 15 ГБ

Таблица 3. Требования к вычислительной части (при размещении БД на той же виртуальной машине, в которой функционирует сервер R-Vision).

Требования			Процессор	Память	Свободное место в разделе /opt
Активов	Сценариев реагирования, макс.	Пользователей, макс.			
Минимальные требования	1	10	4vCPU 2 ГГц	Не менее 12 ГБ	Не менее 100 ГБ (при новой установке), или (если больше 100 ГБ) 41 + 1,3*S (но не более 200 Гб), где S - размер базы данных при обновлении с бездокерной версии
до 5 000	10	20	12vCPU 2 ГГц	32 ГБ	250 ГБ
10000	20	30	22vCPU 2 ГГц	48 ГБ	
20000	50	50	40vCPU 2 ГГц	64 ГБ	
50000	100	100	54vCPU 2 ГГц	80 ГБ	450 ГБ
100000 и более	200	150	64vCPU 2 ГГц	96 ГБ	

Таблица 4. Требования к вычислительной части для сервера R-Vision (БД размещается на выделенном сервере).

Требования			Процессор	Память	Свободное место в разделе /opt
Активов	Сценариев реагирования, макс.	Пользователей, макс.			
Минимальные требования	1	10	2vCPU 2 ГГц	Не менее 8 ГБ	Не менее 100 ГБ
до 5 000	10	20	2vCPU 2 ГГц	10 ГБ	
10000	20	30	4vCPU 2 ГГц	12 ГБ	
20000	50	50	6vCPU 2 ГГц	12 ГБ	
50000	100	100	8vCPU 2 ГГц	16 ГБ	200 ГБ
100000 и более	200	150	10vCPU 2 ГГц	16 ГБ	

Таблица 5. Требования к серверу, на котором размещается коллектор

Характеристика	**Минимальные требования**	**Рекомендуемые требования**
Процессор	2vCPU 2 ГГц	4vCPU 2 ГГц
Память	Не менее 4 ГБ	8 ГБ
Свободное место в разделе /opt	Не менее 30 ГБ	50 ГБ

Таблица 6. Требования к серверу БД

Требования			Процессор	Память	Свободное место в разделе /opt	Поддерживаемая ОС
Активов	Сценариев реагирования, макс.	Пользователей, макс.				
Минимальные требования	1	10	4vCPU 2 ГГц	Не менее 8 ГБ	Не менее 100 ГБ	• Ubuntu 14,16 • CentOS 7 • RHEL 7 • Windows Server (2012/2016) • FreeBSD 11
до 5 000	10	20	10vCPU 2 ГГц	24 ГБ	200 ГБ	
10000	20	30	16vCPU 2 ГГц	32 ГБ		
20000	50	50	32vCPU 2 ГГц	48 ГБ		
50000	100	100	42vCPU 2 ГГц	56 ГБ	400 ГБ	
100000 и более	200	150	48vCPU 2 ГГц	64 ГБ		

1.2. Требования к внешним системам для работы с системой

Система поддерживает инвентаризацию узлов со следующими ОС:

Наименование	Версия
Microsoft Windows	XP, Vista, 7, 8, 8.1, 10,11
Microsoft Windows Server	2003, 2008, 2008 R2, 2012, 2012 R2, 2016, 2019, 2022
RedHat Enterprise Linux	5 - 8
CentOS	5 - 8
Ubuntu	12 - 22
Oracle Linux	5 - 9
Debian	6 - 11
Fedora	7 - 36
SUSE Linux Enterprise Server	11 - 15
Astra Linux Common Edition релиз "Орел"	2.12
Astra Linux Special Edition релиз "Смоленск"	1.6

1.3. Установка и обновление отдельных компонентов

В системе поддерживается возможность установки и обновления отдельных компонентов с помощью переменной COMPONENTS_TO_DEPLOY.

> Скрипт установки поддерживает автоматическое определение установленных компонентов. Если нет необходимости добавлять новые компоненты при обновлении, указывать переменную COMPONENTS_TO_DEPLOY необязательно.

1.3.1. Компоненты

Все компоненты подразделяются на два типа: основные и опциональные.

Основные компоненты:

- SMP - основные сервисы;

- COL - коллектор;
- DB - база данных;

Опциональные компоненты:

1. MON - службы мониторинга;
2. EXP - экспортеры службы мониторинга;
3. COCKPIT - портал администратора.

Пример использования переменной:

```
COMPONENTS_TO_DEPLOY="COL" ./install.sh # установка/обновление только коллектора
COMPONENTS_TO_DEPLOY="SMP COL" ./install.sh # установка/обновление только коллектора
и основных сервисов
```

1.3.2. Особенности использования переменной

Поведение переменной COMPONENTS_TO_DEPLOY при обновлении различается для каждого типа компонентов:

При работе с основными компонентами в переменной COMPONENTS_TO_DEPLOY нужно указать все компоненты, установленные на компьютере.
Например, если на компьютере уже установлен компонент SMP и нужно добавить компонент COL, команда запуска будет следующей:

```
COMPONENTS_TO_DEPLOY="SMP COL" ./install.sh
```

Установку или обновление опциональных компонентов можно производить без основных компонентов.
Например, если нужно добавить компонент MON, команда запуска будет следующей:

```
COMPONENTS_TO_DEPLOY="MON" ./install.sh
```

Если вы обновляете основные компоненты, поведение опциональных компонентов при обновлении может поменяться. Так, при установленных компонентах SMP, COL и MON нельзя обновить выборочно только компоненты SMP и COL, а компонент MON не обновлять. Это действие расценивается скриптом инсталляции как отключение компонента. Такое поведение не поддерживается в системе.

Отключить уже установленные компоненты нельзя.

Используйте переменную COMPONENTS_TO_DEPLOY с осторожностью и только если вы полностью уверены в необходимости ее применения.
Если вы хотите установить базу данных (DB) и основные сервисы (SMP) на разные серверы, для лучшего контроля конфигурации базы данных и возможности поддержки кластера рекомендуем самостоятельно или с помощью службы поддержки R-Vision поставить PostgreSQL 11.14, а не устанавливать базу данных с помощью команды COMPONENTS_TO_DEPLOY=DB.

1.4. Установка

Процесс установки выполняется в два этапа:

4. Установка системы из образа ISO.
5. Обновление системы до актуальной версии.

1.4.1. Установка из образа ISO

Чтобы установить систему из образа ISO, выполните следующие действия:

1. Скачайте образ системы по ссылке. Для получения ссылки обратитесь в службу клиентской поддержки ООО “Р-Вижн” по адресу: support@rvision.ru.
2. На вашем гипервизоре создайте и настройте новую виртуальную машину со следующими параметрами:
   a. Тип: Linux CentOS7.
   b. Задайте параметры виртуальной машины в соответствии с информацией, приведенной в разделе Технические требования.
3. Установите для виртуальной машины ISO-образ системы R-Vision в качестве установочного диска.
4. Запустите виртуальную машину. После запуска виртуальной машины на экране отобразится меню выбора типа установки CentOS7.
5. В меню выберите пункт **Install CentOS 7.**
6. Подтвердите выбор, нажав на клавишу **Enter**.

Установка операционной системы займет некоторое время. После завершения установки на экране отобразится сообщение **Installation complete. Press return to quit**.

7. Нажмите на клавишу **Enter** для перезагрузки сервера и продолжения установки.

8. После перезагрузки сервера, в консоли отобразится запрос данных для авторизации. Введите следующие данные:

    a. login: root

    b. password: pxtm0222

    После входа в систему, автоматически запустится скрипт установки системы.

9. Укажите сетевые интерфейсы, которые будут использоваться при работе с системой. По умолчанию используется интерфейс **ens**.
10. Укажите режим работы интерфейса: DHCP или статический адрес.
11. В появившемся окне проверьте и подтвердите заданные вами настройки сетевого интерфейса.
12. Подтвердите начало установки.
13. Выберите тип устанавливаемого продукта:

    a. **AiO** (All-in-One) поставляется как сервер системы R-Vision, со встроенным коллектором.

    b. **Application** – сервер системы R-Vision, со встроенным коллектором, но без БД.

    c. **Database** – База данных.

    d. **Collector** – коллектор R-Vision, вынесенный для отдельной установки, для использования вместе с сервером R-Vision.

14. Введите пароль базы данных (по умолчанию **pxtm0222**).
15. Подтвердите обновление серверных компонентов.
16. На экране отобразится сообщение об успешном продолжении установки.
17. После окончания процесса отобразится окно с сообщением об успешной установке и напоминанием о перезапуске системы. Нажмите **Enter**.

> По соображениям безопасности рекомендуем после установки системы сменить пароль root-пользователя системы.

Когда система установлена, обновите ее до актуальной версии.

1.4.2. Обновление системы до актуальной версии

Чтобы установить обновление:

1. Скачайте zip-архив с обновлением (rvision_<номер версии>.zip) по ссылке. Чтобы получить ссылку на скачивание zip-архива с обновлением, обратитесь в службу поддержки по адресу support@rvision.ru.
2. Загрузите файл rvision_<номер версии>.zip во временную директорию /tmp на сервер, на котором установлена система R-Vision (в примере показано обновление до версии компонета 5.0.0-rc11.

```
scp ./rvision_5.0.0-rc11.zip user@rvnserver:/tmp/
```

3. Если вам необходимо убедиться в подлинности устанавливаемого дистрибутива, сделайте это с помощью электронной подписи.
4. Подключитесь по протоколу SSH к серверу, на котором установлена система R-Vision.

```
ssh user@rvnserver
```

5. Для дальнейших действий потребуются права суперпользователя (root). Выполните следующую команду:

```
su -
```

6. Распакуйте файл с обновлением командой

```
unzip -o /tmp/rvision_5.0.0-rc11.zip -d /tmp/rvn
```

7. Запустите скрипт обновления командой

```
/tmp/rvn/install.sh
```

Для подробного вывода информации можно задать переменную VERBOSE.

```
sudo VERBOSE=yes /tmp/rvn/install.sh
```

8. Подтвердите продолжение установки, нажав клавишу **Enter**:
9. Система отобразит на экране полное доменное имя сервера, на который производится установка. При необходимости отредактируйте его.
10. Укажите параметры базы данных.
11. После появления сообщения о готовности к началу установки нажмите **Enter**.
12. В процессе установки на экране отобразится предложение переопределить сетевой диапазон, используемый под docker-контейнеры. Переопределение диапазона помогает избежать пересечения с уже используемым диапазоном. Введите адрес с маской для интерфейса docker0 (определяет адресацию контейнеров, созданных

без указания пользовательской docker-сети). Если вы не хотите переопределять сетевой диапазон, нажмите **Enter**, чтобы пропустить этот шаг.

13. Если вы переопределяете сетевой диапазон, укажите адрес подсети, в которой будут создаваться пользовательские сети. Если вы не хотите переопределять сетевой диапазон, нажмите **Enter**, чтобы пропустить этот шаг.

Система предлагает переопределить сетевой диапазон и указать адрес подсети только при первичной установке системы. Чтобы выполнить эти настройки вручную в любое время, используйте файл daemon.json в директории /etc/docker/.

14. Следуйте инструкциям установщика. Если установка прошла успешно, на экране отобразится сообщение об успешном завершении установки. Также система отобразит ссылку, с помощью которой можно перейти в веб-интерфейс R-Vision. Также это можно сделать через IP-адрес сервера на следующем шаге.
15. Если вы хотите перейти в веб-интерфейс R-Vision через IP-адрес сервера, дальнейшие действия зависят от режима работы интерфейса, выбранного на этапе инсталляции (шаг 10):

    a. Статический адрес: запустите браузер на рабочей станции и введите IP-адрес сервера R-Vision в адресную строку браузера, чтобы перейти в веб-интерфейс R-Vision.

    b. DHCP: определите IP-адрес сервера R-Vision. Для этого в консоли сервера выполните команду **ip addr** и посмотрите IP-адрес интерфейса, выбранного на этапе инсталляции (шаг 9). Затем введите этот IP-адрес в адресную строку браузера, чтобы перейти в веб-интерфейс системы.

16. Система перенаправит вас на страницу авторизации. По умолчанию используются следующие учетные данные:

    a. логин: **admin**

    b. пароль: **admin**

17. Введите учетные данные и нажмите на кнопку **Войти**. В окне браузера отобразится стартовая страница системы.

Функционал системы ограничен, так как не указан файл лицензии. Для получения файла лицензии отправьте уникальный код, содержащийся в поле **SERVER-ID** в разделе **Настройки→Общие→Лицензия**, на электронный адрес support@rvision.com.

- Электронная подпись
- Установка и обновление отдельных компонентов

- Установка системы с вынесенной базой данных

1.5. Электронная подпись

Убедиться в подлинности устанавливаемого дистрибутива можно с помощью электронной подписи (в примере рассматривается подтверждение электронной подписи для дистрибутива функционального блока управления: активами, уязвимостями, инцидентами, аудитами и рисками ИБ версии 5.2.0-a):

1.5.1. Проверка электронной подписи

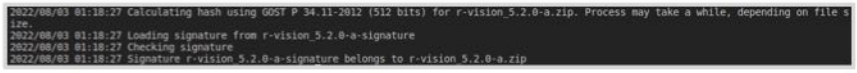
1. Выполните следующую команду.

```
unzip checksums_5.2.0-a.zip
verify-signature verify --file r-vision_5.2.0-a.zip --signature
   signature --verification-key public.key
```

Система отобразит одно из следующих сообщений:

- Если электронная подпись не подтверждена:

- Если электронная подпись подтверждена:



1.5.2. Проверка контрольных сумм исполняемых файлов во время работы системы (производится вручную)

1. Для проверки основных сервисов системы выполните следующую команду:

```
/opt/r-vision/utils/verify-checksums.sh
```

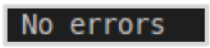
2. Для проверки коллектора выполните следующую команду:

```
/opt/r-vision/utils/verify-checksums-col.sh
```

Система отобразит одно из следующих сообщений:

- Если проверка не пройдена:

- Если проверка пройдена успешно:



1.5.3. Подсчет и сохранение контрольных сумм

Чтобы система рассчитала и сохранила контрольные суммы файлов, находящихся в контейнерах сервисов, после установки системы необходимо вручную запустить следующие скрипты:

- Скрипт для работы с контрольными суммами SMP:

```
/opt/r-vision/utils/calculate-checksums.sh /opt/r-vision/data/checksums_new.csv
```

- Скрипт для работы с контрольными суммами коллектора:

```
/opt/r-vision/utils/calculate-checksums-col.sh /opt/r-
vision/data/collectors/checksums_new.csv
```

Если вы производите установку с системной переменной SMP_CHECKSUMS=yes, эти скрипты запустятся автоматически.

1.6. Установка системы с вынесенной базой данных

Для установки системы на вынесенную базу данных необходимо выполнить подготовку базы данных и установить APP сервер.

В этом примере описана установка на ОС Astra SE 1.7.

1.6.1. Особенности установки на ОС Astra SE 1.7

Пользователь, учетная запись которого используется для первоначального входа в систему, должен:

- входить в группу **astra-admin.**
- иметь максимальный уровень целостности 63.
- входить в систему под этим уровнем.

Пользователь, который создается при установке системы, обладает этими правами, однако при создании новой учетной записи пользователя нужно:

- добавить ее в группу с помощью команды:

```
usermod -a -G astra-admin USERNAME
```

- повысить ее уровень целостности с помощью команды:

```
pdpl-user -i 63 USERNAME
```

В результате создаваемую учетную запись можно будет использовать для установки системы.

1.6.2. Подготовка сервера баз данных

- Подключите репозиторий postgres для установки базы данных.
- Добавьте ключ:

```
wget --quiet -O -
    https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo
    apt-key add -
```

- Добавьте репозиторий:

```
echo "deb http://apt.postgresql.org/pub/repos/apt/ buster-pgdg
    main" > /etc/apt/sources.list.d/pgdg.list
```

- Установите необходимые зависимости:

```
curl -o ./libllvm7_7.0.1-8+deb10u2_arm64.deb
    "http://ftp.de.debian.org/debian/pool/main/l/llvm-toolchain-
    7/libllvm7_7.0.1-8+deb10u2_amd64.deb"
apt-get install ./libllvm7_7.0.1-8+deb10u2_arm64.deb
```

- Установите postgres. По умолчанию репозитории ОС Astra имеют более высокий приоритет, чем вновь подключенные репозитории. Необходимо использовать ключ -t.

```
apt-get install -t buster-pgdg postgresql-11
```

- Настройте сервер на прием подключения клиентских приложений по всем IP адресам и задайте максимальное количество подключений:

```
/etc/postgresql/11/main/postgresql.conf
  listen_addresses = '*'
  max_connections = 500
```

- Разрешите установку подключений с APP сервера:

```
/etc/postgresql/11/main/pg_hba.conf
  host all all IP_APP_СЕРВЕРА/32 md5
```

Если БД полностью вынесена на другой хост, указанной команды будет достаточно для установки подключений. Если БД вынесена из контейнера, но находится на том же хосте, что и сервер приложения, рекомендуется отредактировать файл pg_hba.conf. Это обеспечит корректное резервное копирование данных посредством скрипта backup-db.sh.

Добавьте в файл pg_hba.conf следующие строки:

```
1. host all rvision xxx.xxx.xxx.xxx/xx md5
2. host all rvision yyy.yyy.yyy.yyy/yy md5
```

где xxx.xxx.xxx.xxx/xx - адрес сети docker0, а yyy.yyy.yyy.yyy/yy - адрес сети для подключения контейнеров.
Пример:

```
3. host all rvision 172.27.4.0/22 md5
4. host all rvision 172.27.0.0/22 md5
```

После добавления строк нужно перезапустить сервер БД:

```
5. systemctl restart postgresql-11.service
```

- Запустите postgres:

```
systemctl restart postgresql
systemctl enable postgresql
```

- Создайте базу данных, пользователя, предоставьте права для базы данных и создайте расширения:

```
sudo -u postgres psql -c "CREATE DATABASE rvision;"
sudo -u postgres psql -c "CREATE USER rvision WITH PASSWORD
    'pxtm0222';"
sudo -u postgres psql -d rvision -c "ALTER USER rvision WITH
    SUPERUSER;"
sudo -u postgres psql -d rvision -c "CREATE EXTENSION pg_trgm;"
sudo -u postgres psql -d rvision -c "CREATE EXTENSION
    intarray;"
sudo -u postgres psql -d rvision -c "CREATE EXTENSION
    pgcrypto;"
```

- Отключите мандатный контроль для postgres. Дополнительный параметр **zero_if_notfound** определяет, является ли ошибкой отсутствие мандатных атрибутов пользователя в базах данных.

```
sed -i 's/zero_if_notfound:.*/zero_if_notfound: yes/g'
    /etc/parsec/mswitch.conf
```

1.6.3. Установка APP сервера

1. Распакуйте архив (например, в папку /tmp).

```
COMPONENTS_TO_DEPLOY="SMP COL" /tmp/install.sh
```

2. После запуска установщика введите параметры подключения к базе данных.

2. УСТАНОВКА ФУНКЦИОНАЛЬНОГО БЛОКА ИМИТАЦИИ ИНФРАСТРУКТУРЫ

2.1. Технические требования

2.1.1. Управляющий сервер

Характеристика	Минимальные значения	Рекомендуемые значения
Процессор	4 ядра 2.4 и более ГГц	6 ядер 2.4 и более ГГц
Оперативная память	8-16 ГБ	32 и более ГБ
Дисковое пространство (SSD-накопитель)	150 ГБ	500 ГБ
Сетевые интерфейсы	1 сетевая карта - интерфейс управления	1 сетевая карта - интерфейс управления
Операционная система	2. CentOS 7.9 3. Astra CE 2.12 Orel 4. Astra SE 1.7 Smolensk 5. RockyLinux 8.5+ (начиная со сборки 2.0) 6. RedOS 7.3(c) (начиная со сборки 2.2)	

2.1.2. Сервер управления ловушками

Характеристика	Минимальные значения
Процессор	4 ядра 2.4 и более ГГц
Оперативная память	16 ГБ
Дисковое пространство (SSD-накопитель)	150 ГБ
Сетевые интерфейсы	1 сетевая карта - интерфейс управления 1 и более сетевая карта - порт доступа/магистральный порт
Операционная система	• Astra CE 2.12 Orel • Astra SE 1.7 Smolensk • RedOS 7.3(c) (начиная со сборки 2.2) • RockyLinux 8.5+ (начиная со сборки 2.0) • CentOS 7.9

При работе на Astra Linux 1.7 создание ловушек типа FullOS поддерживается только для ОС Linux.

2.1.3. Сетевое взаимодействие между компонентами функционального блока имитации инфраструктуры

Источник	Назначение	Протокол	Порт
Управляющий сервер	Сервер управления ловушками	SSH	22 TCP
Управляющий сервер	Сервер управления ловушками	HTTP(S)	80/443 TCP
Сервер управления ловушками	Управляющий сервер	lumberjack	5044 TCP
FullOS Traps (Windows, Linux)	Управляющий сервер	lumberjack	5044 TCP
АРМ Пользователя (Консоль управления R-Vision Deception)	Управляющий сервер	HTTP(S)	80/443 TCP

2.1.4. Интеграция с внешними сервисами

Взаимодействие между хостом и сервисом размещения приманок:

Источник	Назначение	Протокол	Порт
Управляющий сервер	Хост Windows / Linux	TCP	Service Port
Хост Windows / Linux	Управляющий сервер	TCP	5555

Внешние сервисы:

Источник	Назначение	Протокол	Порт
Управляющий сервер	Active Directory	LDAP(S)	Service Port
Управляющий сервер	Syslog	TCP / UDP	Service Port

2.2. Установка функционального блока имитации инфраструктуры

2.2.1. Установка функционального блока имитации инфраструктуры без предустановленной ОС

> Компонент требует две виртуальные машины для установки.
> Если вы устанавливаете сервер Trap Manager на виртуальный сервер, включите возможность вложенной виртуализации, а также разрешите активировать режимы Promiscuous mode и Forged transmits.

Вы можете установить модуль на пустые виртуальные машины из ISO-образов.

Чтобы получить ссылки на скачивание ISO-образов, обратитесь в компанию R-Vision по адресу support@rvision.ru Вы получите доступ к ISO-образам:

- **tdp-cc-<version>.iso**: образ Control Center.
- **tdp-tm-<version>.iso**: образ Trap Manager.

2.3. Процедура установки

Для установки компонента:

2. На виртуальной машине Control Center настройте сетевой интерфейс управления. Сетевой сегмент этого интерфейса не должен пересекаться с сетью для размещения ловушек Trap Manager, описанной на шаге 12.
3. Задайте ISO-образ **tdp-cc-<version>.iso** в качестве установочного диска на первой виртуальной машине.
4. Запустите виртуальную машину.
5. Настройте сетевой адаптер и задайте пароль суперпользователя. Для взаимодействия Control Center и Trap Manager достаточно настроить один сетевой интерфейс.
6. Для настройки HTTPS утвердительно ответьте на вопрос **Do you want to enable HTTPS mode support for the gateway?** В этом случае при развертывании системы будут автоматически сгенерированы самоподписанные ключ и сертификат сроком на 10 лет в каталоге `/opt/deception-cc/packages/envoy/certs/`. Если у вас уже есть собственные ключ и сертификат, этот шаг можно пропустить отрицательным ответом на вопрос и после установки поместить ваши файлы в каталог. Переключение режимов HTTP и HTTPS описано в этом разделе.
7. Дождитесь завершения автоматической установки CentOS 7.
8. Демонтируйте ISO-образ **tdp-cc-<version>.iso**.
9. Запустите виртуальную машину повторно.
10. Введите данные авторизации в консоли:
    - login: **root**
    - password: заданный пароль суперпользователя
11. Дождитесь автоматической установки компонентов Control Center. Панель администратора будет доступна по адресу **http://your-cc-ip**. Логин для входа: **admin**. Пароль: **admin**.
12. Повторите шаги 1-6 для установки сервера Trap Manager из образа **tdp-tm-<version>.iso** на вторую виртуальную машину.
13. При первой попытке входа в систему терминал ОС виртуальной машины с Trap Manager запросит IP-адрес для отправки логов в Control Center. Введите IP-адрес вашей виртуальной машины с Control Center, оставив порт по умолчанию.
14. Настройте две виртуальные сети на виртуальной машине с Trap Manager:
    a. Управляющую сеть, в которой размещены Trap Manager и Control Center.
    b. Сеть для размещения ловушек.

На интерфейсе, который ведет в сеть для размещения ловушек, не должны быть назначены IP-адрес и шлюз.

2.3.1. Установка функционального блока имитации инфраструктуры на предустановленную ОС

Вы можете установить данный компонент из `.tar.gz` архивов на следующие операционные системы:

- Astra CE 2.12 Orel (начиная со сборки 1.7)
- Astra SE 1.7 Smolensk (начиная со сборки 1.8)
- RedOS 7.3(c) (начиная со сборки 2.2)
- CentOS 7.9
- RockyLinux 8.5+ (начиная со сборки 2.0)

2.3.1.1. Системные пакеты зависимостей

Системные пакеты зависимостей выделены из состава инсталлятора. Перед началом развертывания инсталлятор проверяет наличие на хосте пакетов, необходимых для работы Control Center и Trap Manager. Если хотя бы один пакет из списка (docker, docker-compose, python3 и т.д..) отсутствует, инсталлятор пытается найти архив offline-репозитория, чтобы поставить из него все необходимые пакеты, но если архив репозитория не найден, предлагает установить пакеты самостоятельно.

Алгоритм действий инсталлятора:

1. Поиск offline-репозитория с набором необходимых пакетов в родительском каталоге инсталлятора
2. Установка необходимых пакетов из offline-репозитория, расположенного в указанном пользователем месте.
3. Отображение подсказки о необходимости установить отсутствующие пакеты самостоятельно, а затем - прерывание установки.

Offline-репозитории с пакетами зависимостей для поддерживаемых ОС можно скачать с сервера хранения релизов R-Vision. Для получения ссылок обратитесь в службу клиентской поддержки ООО “Р-Вижн” по адресу: support@rvision.ru.

2.3.1.2. Процедура установки

Чтобы получить ссылки на скачивание дистрибутива обратитесь в службу клиентской поддержки ООО “Р-Вижн” по адресу: support@rvision.ru.

Вы получите доступ к архивам:

2. r-tdp-cc_install_<version>.tar.gz: архив Control Center.
3. r-tdp-tm_install_<version>.tar.gz: архив Trap Manager.

Для установки модуля:

1. На виртуальной машине Control Center создайте каталог для распаковки и распакуйте архив **r-tdp-cc_install_<version>.tar.gz** в директорию `/tmp`. Если системные пакеты зависимостей для компонента не были установлены ранее, распакуйте offline-репозиторий под свою ОС (**sys-update-<os_id>.tar.gz**) в директорию `/tmp` или в домашнюю директорию пользователя первой ВМ.

2. Создайте каталог для установки пакетов зависимостей, распакуйте архив и запустите установку пакетов. При этом нужно указать компонент, для которого ставятся пакеты (доступны интерактивный и неинтерактивный режимы).

   Все действия на шагах 2 и 3 выполняются из-под ограниченной учетной записи, которая в дальнейшем будет использоваться для подключения Trap Manager к Control Center. Учетная запись должна быть членом sudoers-группы в зависимости от OS: **wheel**, **sudo** или **astra-admin**:

```
mkdir -p ~/sysprep
tar zxvf sys-update-<os_id>.tar.gz -C ~/sysprep
sudo ~/sysprep/update-system.sh tdp-cc
```

```
mkdir -p ~/sysprep
tar zxvf sys-update-<os_id>.tar.gz -C ~/sysprep
sudo ~/sysprep/update-system.sh tdp-tm
```

3. Создайте каталог для распаковки инсталлятора, предварительно убедившись, что на выбранном разделе есть достаточно места, распакуйте и запустите скрипт установки **install.sh.** Если это первая установка, перед развертыванием система дополнительно запросит FQDN- или IP-адрес CC-сервера.

```
mkdir -p ~/tdp
tar zxvf /tmp/r-tdp-cc_install_<version>.tar.gz -C ~/tdp/
sudo ~/tdp/install.sh
```

4. Дождитесь сообщения об окончании установки.

5. Повторите пункты 1-4 на виртуальной машине Trap Manager с архивами **r-tdp-tm_install_<version>.tar.gz** и **sys-update-<os_id>.tar.gz.** Система также дополнительно запросит IP-адрес CC-сервера. Скрипт установки автоматически определит отключенные сетевые интерфейсы и запустит их в статическом режиме без IP-адреса.
6. Перезагрузите виртуальную машину Trap Manager.

Логи журналирования процесса установки доступны в директории /tmp/deception-logs.

Неинтерактивный режим установки (**CI=true**) позволяет разворачивать компоненты в автоматическом режиме. При первой установке необходимо передать в инсталлятор переменную **TDP_CC_HOST,** чтобы избежать ошибки при развертывании.

Пример команды:

```
sudo CI=true TDP_CC_HOST=tdp-cc.local ./install.sh # инсталлятор CC принимает
TDP_CC_HOST в формате FQDN или IP-адреса
sudo CI=true TDP_CC_HOST=10.10.10.99 ./install.sh # инсталлятор TM принимает
TDP_CC_HOST только в формате IP-адреса
```

2.3.1.3. Особенности установки на Astra Linux 1.7

1. При установке Trap Manager на Astra Linux 1.7 могут возникнуть проблемы с виртуализацией Kernel-based Virtual Machine (KVM) по причине устаревшей версии ядра (**5.4.0-54-generic**). Для устранения проблем обновите ядро до актуальной версии **5.4.0-110-generic** или выше. Убедитесь, что проблемы устранены, с помощью команды:

   ```
   libguestfs-test-tool
   ```

   Проверить что с виртуализацией не возникнет никаких проблем можно командой `libguestfs-test-tool.`
2. При установке на Astra Linux каталог /tmp очищается после каждой перезагрузки. В связи с этим файлы и директории, необходимые для дальнейшей работы, рекомендуется перенести перед перезагрузкой в домашний каталог пользователя.

2.3.2. Установка пользовательского сертификата (CA)

Для установки пользовательского сертификата:

1. Поместите свои корневые сертификаты в каталог **/opt/deception/packages/backend/ca-trusted/**.

В этом каталоге может уже находиться файл **ca-bundle.crt**, который является сборкой корневых сертификатов (CA) доверенных центров сертификации. Если файл отсутствует, система сформирует его после перезапуска службы deception.

2. Перезапустите службу deception:

```
systemctl restart deception
```

Если система отобразит ошибку **Failed to restart deception.service: Unit not found**, выполните команды:

```
cd /opt/deception/docker
docker-compose stop
docker-compose up -d
```

Пользовательский сертификат (CA) установлен.

2.3.3. Переключение режимов работы веб-сервера (HTTP/HTTPS)

После развертывания из ISO или архива в каталоге **/opt/deception/scripts** появляется скрипт **set-proxy-http-mode.sh**, позволяющий менять режимы работы с HTTP на HTTPS и обратно. Данный скрипт не создает ключ (**serverkey.pem**) и сертификат (**servercert.pem**), а использует для этой цели файлы, имеющиеся в директории **/opt/deception/packages/envoy/certs/**. Если эти файлы не будут найдены в указанной директории, переключить режим с HTTP на HTTPS не удастся.

1. Перейдите в каталог, в котором находится скрипт:

```
cd /opt/deception/scripts
```

2. Запустите приведенный ниже скрипт, указав режим, в котором должен работать веб-сервер:

```
./set-proxy-http-mode.sh http          # для переключения на
   http
./set-proxy-http-mode.sh https         # для переключения на
   https
```

Система переключится в указанный режим.

2.3.4. Установка системных пакетов зависимостей

Системные пакеты зависимостей можно установить как из offline-репозиториев, так и из репозиториев разработчиков ОС.

2.3.4.1. Установка из offline-репозитория

В отсутствие доступа в Интернет вы можете воспользоваться имеющимися offline-репозиториями для каждой из поддерживаемых ОС. Скачать offline-репозитории можно по следующим ссылкам:

- AstraLinux
- CentOS
- RED OS
- RockyLinux

2.3.4.1.1 Развертывание Control Center из offline-репозитория

Воспользуйтесь командами:

```
mkdir -p ~/sysprep
tar zxvf sys-update-<os_id>.tar.gz -C ~/sysprep
sudo ~/sysprep/update-system.sh tdp-cc
```

2.3.4.1.2 Развертывание Trap Manager из offline-репозитория

Воспользуйтесь командами:

```
mkdir -p ~/sysprep
tar zxvf sys-update-<os_id>.tar.gz -C ~/sysprep
sudo ~/sysprep/update-system.sh tdp-tm
```

2.3.4.2. Установка из репозиториев разработчиков ОС

Если у вас есть доступ в Интернет, вы можете поставить пакеты, необходимые для подготовки ОС к развертыванию модуля, самостоятельно.

2.3.4.2.1 AstraLinux CE 2.12.X

Control Center:

```
cat <<EOF | sudo tee -a /etc/apt/sources.list
deb http://deb.debian.org/debian stretch main contrib non-free
deb-src http://deb.debian.org/debian stretch main contrib non-free

deb http://deb.debian.org/debian stretch-updates main contrib non-free
deb-src http://deb.debian.org/debian stretch-updates main contrib non-free

deb http://security.debian.org/debian-security/ stretch/updates main contrib non-
free
deb-src http://security.debian.org/debian-security/ stretch/updates main contrib
non-free
EOF
sudo apt update
sudo apt install -y traceroute wget nano unzip telnet nmap tcpdump gcc ca-
certificates curl rsync screen htop
sudo apt install -y chrony firewalld python3 python3-dev python3-pip python3-docker
python3-lxml python3-setuptools python3-requests
sudo apt install -y ca-certificates gnupg2
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
echo -e "\ndeb [arch=amd64] https://download.docker.com/linux/debian stretch stable"
| sudo tee -a /etc/apt/sources.list
sudo apt update
sudo apt install -y docker.io docker-compose
```

Code Block 1 cc

Trap Manager:

```
cat <<EOF | sudo tee -a /etc/apt/sources.list
deb http://deb.debian.org/debian stretch main contrib non-free
deb-src http://deb.debian.org/debian stretch main contrib non-free

deb http://deb.debian.org/debian stretch-updates main contrib non-free
deb-src http://deb.debian.org/debian stretch-updates main contrib non-free

deb http://security.debian.org/debian-security/ stretch/updates main contrib non-
free
deb-src http://security.debian.org/debian-security/ stretch/updates main contrib
non-free
EOF
sudo apt update
sudo apt install -y traceroute wget nano unzip telnet nmap tcpdump gcc ca-
certificates curl rsync screen htop
sudo apt install -y chrony firewalld python3 python3-dev python3-pip python3-docker
python3-lxml python3-setuptools python3-libvirt pkg-config libvirt-clients libvirt-
daemon-system libvirt-dev astra-kvm libguestfs-tools openvswitch-switch openvswitch-
common
sudo apt install -y ca-certificates gnupg2
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
echo -e "\ndeb [arch=amd64] https://download.docker.com/linux/debian stretch stable"
| sudo tee -a /etc/apt/sources.list
sudo apt update
sudo apt install -y docker.io docker-compose
sudo apt install -y linux-5.10-generic
sudo reboot
```

Code Block 2 tm

2.3.4.2.2 AstraLinux SE 1.7.X

Control Center:

```
sudo apt update
sudo apt install -y traceroute wget nano unzip telnet nmap tcpdump gcc ca-
certificates curl rsync screen htop
sudo apt install -y chrony firewalld python3 python3-dev python3-pip python3-docker
python3-lxml python3-setuptools python3-requests
export DOCKER_VERSION="20.10.2+dfsg1-2astra.se9"
export DOCKER_COMPOSE_VERSION="1.29.2-1astra.se1+ci1"
sudo -E apt install -y docker.io=$DOCKER_VERSION docker-
compose=$DOCKER_COMPOSE_VERSION
```

Code Block 3 cc

Trap Manager:

```
sudo apt update
sudo apt install -y traceroute wget nano unzip telnet nmap tcpdump gcc ca-
certificates curl rsync screen htop
sudo apt install -y chrony firewalld python3 python3-dev python3-pip python3-docker
python3-lxml python3-setuptools python3-libvirt pkg-config libvirt-clients libvirt-
daemon-system libvirt-dev astra-kvm libguestfs-tools openvswitch-switch openvswitch-
common
export DOCKER_VERSION="20.10.2+dfsg1-2astra.se9"
export DOCKER_COMPOSE_VERSION="1.29.2-1astra.se1+ci1"
sudo -E apt install -y docker.io=$DOCKER_VERSION docker-
compose=$DOCKER_COMPOSE_VERSION
sudo apt install -y linux-5.10-generic
sudo reboot
```

Code Block 4 tm

2.3.4.2.3 CentOS 7.X

Control Center:

```
sudo yum install -y epel-release
sudo yum install -y traceroute wget nano unzip telnet nmap tcpdump gcc ca-
certificates curl rsync screen htop
sudo yum install -y yum-utils audit chrony python3 python3-devel libselinux-python3
sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-
ce.repo
export DOCKER_VERSION="20.10.23"
sudo -E yum install -y docker-ce-$DOCKER_VERSION docker-ce-cli-$DOCKER_VERSION
docker-ce-rootless-extras-$DOCKER_VERSION docker-scan-plugin
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/bin/docker-compose
sudo chmod +x /usr/bin/docker-compose
sudo python3 -m pip install docker lxml setuptools requests
```

Code Block 5 cc

Trap Manager:

```
sudo yum install -y epel-release centos-release-openstack-train libmodulemd
sudo yum install -y traceroute wget nano unzip telnet nmap tcpdump gcc ca-
certificates curl rsync screen htop
sudo yum install -y yum-utils audit chrony python3 python3-devel libselinux-python3
libvirt libvirt-devel qemu-kvm libguestfs-tools libguestfs-winsupport kernel-devel
openvswitch
sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-
ce.repo
export DOCKER_VERSION="20.10.23"
sudo -E yum install -y docker-ce-$DOCKER_VERSION docker-ce-cli-$DOCKER_VERSION
docker-ce-rootless-extras-$DOCKER_VERSION docker-scan-plugin
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/bin/docker-compose
sudo chmod +x /usr/bin/docker-compose
sudo python3 -m pip install docker lxml setuptools libvirt-python
sudo yum install -y elrepo-release
sudo yum remove -y kernel-tools kernel-tools-libs kernel-headers
sudo yum --disablerepo="*" --enablerepo="elrepo-kernel" install -y kernel-lt kernel-
lt-{devel,headers,tool} gcc glibc
sudo grubby --set-default-index=0
sudo reboot
```

Code Block 6 tm

2.3.4.2.4 RED OS 7.3.X

Control Center:

```
sudo dnf install -y traceroute wget nano unzip telnet nmap tcpdump gcc ca-
certificates curl rsync screen htop
sudo dnf install -y yum-utils audit chrony python3 python3-devel libselinux-python3
export DOCKER_VERSION='20.10.10-1.el7'
export DOCKER_COMPOSE_VERSION='1.29.2-1.el7'
sudo -E dnf install -y docker-ce-$DOCKER_VERSION docker-ce-cli-$DOCKER_VERSION
docker-ce-rootless-extras-$DOCKER_VERSION docker-compose-$DOCKER_COMPOSE_VERSION
sudo python3 -m pip install docker lxml setuptools requests
```

Code Block 7 cc

Trap Manager:

```
sudo dnf install -y traceroute wget nano unzip telnet nmap tcpdump gcc ca-
certificates curl rsync screen htop
sudo dnf install -y yum-utils audit chrony python3 python3-devel libselinux-python3
libvirt libvirt-devel qemu-kvm libguestfs-tools libguestfs-winsupport kernel-devel
openvswitch
export DOCKER_VERSION='20.10.10-1.el7'
export DOCKER_COMPOSE_VERSION='1.29.2-1.el7'
sudo -E dnf install -y docker-ce-$DOCKER_VERSION docker-ce-cli-$DOCKER_VERSION
docker-ce-rootless-extras-$DOCKER_VERSION docker-compose-$DOCKER_COMPOSE_VERSION
sudo python3 -m pip install docker lxml setuptools libvirt-python
```

Code Block 8 tm

2.3.4.2.5 RockyLinux 8.X

Control Center:

```
sudo dnf install -y epel-release centos-release-openstack-train libmodulemd
sudo dnf install -y traceroute wget nano unzip telnet nmap tcpdump gcc ca-
certificates curl rsync screen htop
sudo dnf install -y yum-utils audit chrony python3 python3-devel libselinux-python3
sudo dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-
ce.repo
export DOCKER_VERSION="20.10.23"
sudo -E dnf install -y docker-ce-$DOCKER_VERSION docker-ce-cli-$DOCKER_VERSION
docker-ce-rootless-extras-$DOCKER_VERSION docker-scan-plugin
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/bin/docker-compose
sudo chmod +x /usr/bin/docker-compose
sudo python3 -m pip install docker lxml setuptools requests
```

Code Block 9 cc

Trap Manager:

```
sudo dnf install -y epel-release
sudo dnf install -y traceroute wget nano unzip telnet nmap tcpdump gcc ca-
certificates curl rsync screen htop
sudo dnf install -y yum-utils findutils dnf-plugins-core
sudo dnf install -y https://www.rdoproject.org/repos/rdo-release.el8.rpm
sudo dnf config-manager --set-disabled centos-rabbitmq-38
sudo dnf install -y audit chrony python3 python3-devel libselinux-python3 libvirt
libvirt-devel qemu-kvm libguestfs-tools libguestfs-winsupport kernel-devel
openvswitch
sudo dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-
ce.repo
export DOCKER_VERSION="20.10.23"
sudo -E dnf install -y docker-ce-$DOCKER_VERSION docker-ce-cli-$DOCKER_VERSION
docker-ce-rootless-extras-$DOCKER_VERSION docker-scan-plugin
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/bin/docker-compose
sudo chmod +x /usr/bin/docker-compose
sudo python3 -m pip install docker lxml setuptools libvirt-python
sudo dnf install -y elrepo-release
sudo dnf remove -y kernel-tools kernel-tools-libs kernel-headers
sudo dnf --enablerepo=elrepo-kernel install -y kernel-lt kernel-lt-
{devel,headers,tool} gcc glibc
sudo grubby --set-default-index=0
sudo reboot
```

Code Block 10 tm

3. УСТАНОВКА ФУНКЦИОНАЛЬНОГО БЛОКА МОНИТОРИНГА И АНАЛИЗА СОБЫТИЙ БЕЗОПАСНОСТИ

3.1. Технические требования

3.1.1. Требования к аппаратному обеспечению

Платформа работает на компьютерах со следующими характеристиками:

Характеристика	Требования	
	Минимальные	Рекомендуемые
Процессор	8 ядер, 2 ГГц и выше	10 ядер, 2,4 ГГц и выше
Оперативная память	16 ГБ	32 ГБ и более
Дисковое пространство (только SSD)	150 ГБ	256 - 512 ГБ
Сетевые интерфейсы	1 для управления, 100+ Мбит/сек	1 для управления, 100+ Мбит/сек

3.1.2. Требования к сетевому оборудованию внешних систем для интеграции

Название	Версия
Cisco ASA	9 и выше
PaloAlto	9 и выше
Checkpoint	R80.10 и выше
UserGate	6.X и выше

3.1.3. Требования к программному обеспечению

Платформа работает в операционных системах:

Название	Версия
CentOS	7 и выше
Astra Linux Special Edition релиз "Смоленск"	1.7 и выше
Astra Linux Common Edition релиз "Орел"	2.11.3 и выше
Ред ОС	7.2 и выше
Red Hat Enterprise Linux	8 и выше

Платформу можно запускать в среде виртуализации VMWare.

Поддерживаются следующие базы данных:

- Сертифицированная СУБД «Postgres Pro» (сертификат № 3637 от 05.10.2016 г. действителен до 05.10.2024 г.).
- PostgreSQL из состава сертифицированной РЕД ОС.

Платформа работает в следующих веб-браузерах:

Название	Версия
Google Chrome	50 и выше
Mozilla Firefox	50 и выше
Safari	

Для работы платформы требуется доступ к сети Интернет (напрямую или через прокси - сервер) и к DNS-серверу (внешнему или внутреннему).
Платформа должна быть доступна по следующим протоколам:

- WEB - для пользователей.
- WEB и SSH - для администраторов.

3.1.4. Требования для развертывания SIEM-ноды

Для развертывания SIEM-ноды рекомендуется использовать отдельный сервер.

SIEM-нода может быть развернута на компьютерах со следующими характеристиками:

Характеристика	Требования	
	Минимальные	Рекомендуемые
Процессор	8 ядер, 2 ГГц и выше	8 ядер, 2,4 ГГц и выше
Оперативная память	8 ГБ и более	32 ГБ и более
Дисковое пространство (только SSD)	250 ГБ[1]	400 ГБ[1]

[1] точный объем необходимого дискового пространства будет зависеть от потока событий, размера событий и иных факторов.

Практика показывает, что для определенных значений EPS достаточно использовать систему со следующими характеристиками:

EPS	Процессор	Оперативная память	Дисковое пространство
1k	2-4 CPU	8 ГБ RAM	250 ГБ SSD
6-8k	4 CPU	10-12 ГБ RAM	300 Gb SSD
18k	8 CPU	32 ГБ RAM	600+ Gb SSD (при условии настройки периода ротации событий = 7 дней)

3.1.5. Рекомендации по установке

- Модуль может быть установлен как на физической (bare metal), так и на виртуальной машине.
- В случае установки на виртуальной машине, следует предоставить модулю дисковое хранилище в монопольное использование. Это позволит обеспечить максимальное быстродействие.
- В промышленной эксплуатации строго рекомендуется использовать накопители типа SSD для обеспечения максимального быстродействия. Использование HDD строго не рекомендуется: при использовании HDD нет никаких гарантий нормальной производительности системы.
- Не рекомендуется использование RAID 5 и его комбинаций. В качестве RAID рекомендуется использовать RAID 0 для обеспечения быстродействия, RAID 1 для

обеспечения отказоустойчивости, комбинацию RAID 10 для обеспечения обоих свойств, либо RAID 6. Следует убедиться, что RAID организован с помощью аппаратного контроллера. Использование программных контроллеров RAID не рекомендуется.

3.2. Установка

3.2.1. Установка компонента

Если ваша машина имеет RAID-конфигурацию дискового пространства, вы не сможете установить CentOS и модуль из образа ISO. В этом случае установите чистую ОС, а затем разверните на ней модуль.

Чтобы установить систему из образа ISO:

1. Скачайте образ системы по ссылке. Для получения ссылки обратитесь в службу клиентской поддержки ООО “Р-Вижн” по адресу: support@rvision.ru.
2. Установите для виртуальной машины ISO-образ системы R-Vision в качестве установочного диска.
3. Запустите виртуальную машину. Операционная система CentOS 7 будет установлена и настроена автоматически. Установка операционной системы займет некоторое время. После установки инсталлятор автоматически отключит виртуальную машину.
4. Демонтируйте образ, из которого выполнялась установка.
5. Повторно запустите виртуальную машину.
6. В консоли отобразится запрос данных для авторизации. Введите следующие данные:
    a. login: **root**
    b. password: **pxtm0222**
7. Укажите сетевые интерфейсы, которые будут использоваться при работе с системой и задайте их параметры.
8. Укажите режим работы интерфейса: DHCP или статический адрес.

- В появившемся окне проверьте и подтвердите заданные вами настройки сетевого интерфейса.
- Кнопка **OK** запускает установку.
- Установка выполнена успешно, если на экране не отобразилось сообщение об ошибке и доступен веб-интерфейс Threat Intelligence Platform.

- Веб-интерфейс продукта доступен по адресу `http://<ip-address>` или `https://<dns-name>`, если есть внутреннее доменное имя.
- По умолчанию заданы следующие данные для входа:

> По соображениям безопасности рекомендуется после установки сменить пароли.

a. Для веб-интерфейса:

Логин: **admin**; пароль: **admin**

b. Для SSH:

Логин: **root**; пароль: **pxtm0222**

3.2.2. Установка веб-интерфейса

Для установки веб-интерфейса:

1. Скачайте .tgz-архив с файлами установки по ссылке. Для получения ссылки обратитесь в компанию R-Vision по адресу support@rvision.ru.
2. Распакуйте полученный архив командой

```
tar -xvf update-tip-v<номер_версии>.tgz
```

3. Запустите скрипт `update.sh` в папке `update` с правами администратора:

```
cd update
sudo ./update.sh
```

4. Введите IP-адрес или доменное имя сервера, на котором будет доступен веб-интерфейс.
5. После ввода URL веб-интерфейса, дождитесь окончания автоматической установки.

После окончания установки веб-интерфейс будет доступен по введенному вами IP-адресу или доменному имени.

Для остановки сервиса веб-интерфейса остановите и удалите все запущенные контейнеры:

```
docker rm -f $(docker ps -q)
```

Для запуска веб-интерфейса после остановки перезапустите сервис tip-supervisor:

```
service tip-supervisor restart
```

3.2.2.1. Настройка HTTPS

Для настройки доступа к веб-интерфейсу по протоколу HTTPS:

1. Разместите ваши файлы сертификата и ключа `cert.pem` и `cert.key` в директории `/opt/tip/certs` или сгенерируйте самоподписанный сертификат командой:

```
openssl req -new -x509 -days 9999 -nodes -newkey rsa:2048 -subj
    '/C=RU/ST=./L=./O=Company name /OU=.' -out
    /opt/tip/certs/cert.pem -keyout /opt/tip/certs/cert.key
```

2. Перезапустите сервис реверс-прокси командой:

```
cd /opt/tip
docker container restart reverse-proxy
```

Чтобы отключить доступ к веб-интерфейсу по протоколу HTTPS, удалите сертификат, ключ и перезагрузите сервис реверс-прокси командой:

```
rm -rf cert.pem || cert.key && docker container restart reverse-proxy
```

3.2.3. Сетевое взаимодействие

Ниже приведена таблица портов и протоколов для обеспечения сетевого взаимодействия между сервисами R-Vision.

Источник	Назначение	Протокол	Порт	Комментарий
Сервер R-Vision	Почтовый сервер	SMTP	25/TCP, 587/TCP 465/TCP	Подготовьте адрес почтового сервера и учетную запись для доступа к нему — для настройки уведомлений от сервера R-Vision
Сервер R-Vision	АРМ Пользователя R-Vision (Консоль управления R-Vision	HTTP(S)	80,443 TCP	
АРМ Администратора R-Vision	Сервер R-Vision	SSH	22 TCP	
Сервер R-Vision	Провайдеры данных киберразведки (threat intelligence)	HTTP(S)	80, 443 TCP	Убедитесь, что система имеет свободный доступ в сеть Интернет. В случае использования поставщика «ФинЦЕРТ: АСОИ», передача данных до личного кабинета предоставляется заказчиком.

Источник	Назначение	Протокол	Порт	Комментарий
Потребители данных threat intelligence	Сервер R-Vision	HTTP(S)	80, 443 TCP	СЗИ, межсетевые экраны, аналитические системы, и др.
Сервер R-Vision	IBM qRadar	HTTP(S)	API: 80, 443 TCP syslog: any TCP (user-defined)	Зависит от конфигурации SIEM
Сервер R-Vision	Micro Focus ArcSight	HTTPS, syslog	API: 80, 443 TCP syslog: any UDP (user-defined)	Зависит от конфигурации SIEM
Сервер R-Vision	Max Patrol SIEM	HTTPS, syslog	API: 80, 443, IAM: 3333, 3334 AMQP: 5672 TCP (default)	Зависит от конфигурации SIEM. Опубликуйте порт RabbitMQ для корректной работы интеграции.
Сервер R-Vision	Cisco	SSH	22 TCP	Зависит от конфигурации устройства. Убедитесь, что присутствует учетная запись с привилегиями на выполнение команды **shun**.
Сервер R-Vision	SIEM-ноды R-Vision	HTTP(S)	10000 TCP (or user-defined)	GRPC
Сервер R-Vision	Apache Kafka	HTTPS, syslog	9092, TCP	Подготовьте следующие данные: 1. **username** — логин пользователя для авторизации на сервере Kafka. 2. **password** — пароль пользователя для авторизации на сервере Kafka. 3. **bootstrapservers** — список пар хост:порт, разделенных пробелами, которые следует использовать для установления соединения с кластером Kafka. 4. **topic name** — имя топика Kafka для подписки. Достаточно указать только **одно** имя темы. Длина имени может составлять до 255 символов и включать следующие символы: **a-z**, **A-Z**, **0-9**, **.** (точка), **_**

Источник	Назначение	Протокол	Порт	Комментарий
				(подчеркивание) и **-** (минус). 5. **consumer group id** — строка, идентифицирующая группу потребителей, к которой принадлежит данный сервис. 6. **default message offset** — индекс начала выдачи сообщений.

4. УСТАНОВКА ФУНКЦИОНАЛЬНОГО БЛОКА ПОВЕДЕНЧЕСКОГО АНАЛИЗА ОБЪЕКТОВ ЗАЩИТЫ

4.1. Технические требования

В этом разделе представлены технические требования к аппаратной и программной части оборудования.

4.1.1. О вариантах конфигурации

Компонент может быть установлен в двух вариантах:

- **Моно-инсталляция** предполагает установку всех компонентов системы на одной машине.
- **Мульти-инсталляция** предполагает установку хранилищ (ClickHouse, PostgreSQL) на отдельной машине.

Тип конфигурации зависит от параметров используемых машин, а также от планируемого количества событий в секунду (Events Per Second, EPS). Рекомендуется устанавливать ClickHouse и PostgreSQL на отдельном сервере при планировании нагрузки от **10000 EPS**.

В разделе Аппаратные требования приведена таблица соответствия EPS разным типам инсталляции с требованиями к оборудованию.

Для повышения производительности вы можете вынести сервис ClickHouse на отдельную машину в ходе эксплуатации.

4.1.2. Аппаратные требования

Ниже приведена таблица типов установки и требований к аппаратной части в зависимости от планируемого количества передаваемых событий в секунду (EPS):

Событий в секунду	Тип установки	Объем оперативной памяти, ГБ	Количество ядер процессора	Объем SSD (на 6 месяцев)
1000	Моно	26	17	3 ТБ
5000	Моно	34	25	15 ТБ
10000	Узел основных компонентов	14	21	500 ГБ (без срока)
	Узел сервисов хранения данных (ClickHouse, PostgreSQL)	40	19	45 ТБ
15000	Узел основных компонентов	16	25	500 ГБ (без срока)
	Узел сервисов хранения данных (ClickHouse, PostgreSQL)	50	25	60 ТБ
20000	Узел основных компонентов	20	248	500 ГБ (без срока)

Событий в секунду	Тип установки	Объем оперативной памяти, ГБ	Количество ядер процессора	Объем SSD (на 6 месяцев)
	Узел сервисов хранения данных (ClickHouse, PostgreSQL)	62	31	80 ТБ
25000	Узел основных компонентов	22	33	500 ГБ (без срока)
	Узел сервисов хранения данных (ClickHouse, PostgreSQL)	72	38	105 ТБ
30000	Узел основных компонентов	24	37	500 ГБ (без срока)
	Узел сервисов хранения данных (ClickHouse, PostgreSQL)	82	44	120 ТБ
35000	Узел основных компонентов	26	41	500 ГБ (без срока)
	Узел сервисов хранения данных (ClickHouse, PostgreSQL)	94	50	120 ТБ
40000	Узел основных компонентов	28	46	500 ГБ (без срока)
	Узел сервисов хранения данных (ClickHouse, PostgreSQL)	104	57	165 ТБ
45000	Узел основных компонентов	32	50	500 ГБ (без срока)
	Узел сервисов хранения данных (ClickHouse, PostgreSQL)	114	63	180 ТБ
50000	Узел основных компонентов	34	53	500 ГБ (без срока)
	Узел сервисов хранения данных (ClickHouse, PostgreSQL)	124	69	205 ТБ

Для нагрузок выше **5000 EPS** рекомендуется использовать процессоры со следующими характеристиками:

- AMD, ~3.6 GHz, не старше 2 лет.
- Intel, ~3.6 GHz, не старше 3 лет.

4.1.3. Требования к операционной системе

Копмонент поддерживает следующие операционные системы:

- Astra Linux SE 1.7.
- RED OS 7.2.
- CentOS 7 и 7.9.2009.
- Debian 11.
- Ubuntu 18.04.5 LTS.
- ALT Linux Server 8.2SP и Server 10.

Требования к ОС:

- 64-битная инсталляция.
- Версия ядра Linux: 4.11 или выше.
- Iptables: 1.4 или выше.
- Docker: 17.05.0 или выше.
- Docker-compose: 1.29.2 или выше.
- Git: 1.7 или выше.
- Исполняемый файл `ps`, обычно предоставляемый `procps` или аналогичным пакетом.
- XZ Utils 4.9 или выше.
- Правильно смонтированная иерархия `cgroupfs`: одной общей точки монтирования `cgroup` недостаточно для гарантированной работы.

4.1.4. Требования к системе хранения данных

Количество циклов чтения-записи в секунду (IOPS) на системе хранения данных должно составлять не менее 6000.

Проведите тест для проверки вашей системы хранения данных на производительность:

- Откройте сессию терминала и запустите команду:

```
fio --randrepeat=1 --ioengine=libaio --direct=1 --
gtod_reduce=1
--name=fiotest --filename=testfio --bs=4k --iodepth=16 -
-size=1G
--readwrite=randrw --rwmixread=40
```

- Проверьте количество IOPS на одной сессии, затем запустите еще две сессии с той же командой.
- Проверьте количество IOPS на трех сессиях, затем запустите еще две сессии с той же командой.

Если при переходах от одной сессии к трем, а затем к пяти сессиям происходит резкое снижение IOPS (в разы), такая система хранения данных может серьезно снизить производительность всей платформы.

4.1.5. Требования к Docker

Модуль требует следующие параметры Docker для корректной работы:

- Логирование в формате JSON (`json-file`).
- Ограничение по количеству (`log-opts.max-file`) и размеру логов (`log-opts.max-size`): максимум 3 файла по 10 мегабайт каждый.

- Директория Docker (`data-root`) по умолчанию: `/opt/var/docker`.
- Пользователь системы, работающий с Docker, принадлежит к группе Docker, а также имеет полные права на папку с файлами установки как sudoer.

4.2. Установка компонента

Для установки системы вам могут потребоваться права суперпользователя (`sudo`).

4.2.1. Подготовка сервера к установке

Ознакомьтесь с технических требованиями.

- Проверьте версию ядра Linux:

```
uname -r
```

- Убедитесь, что установлены все необходимые обновления безопасности и исправления.
- Убедитесь, что установлен Docker версии не ниже 17.05 со всеми зависимостями:

```
docker -v
```

- Выделите три точки монтирования:
  a. ОС: /
  b. Docker: `/opt/var/docker`

  - Предварительно перенесите данные Docker в новую директорию `/opt/var/docker`, чтобы избежать проблем с работой ОС, если на выделенной точке монтирования закончится место.

  a. R-Vision: `/opt/sense`
- Убедитесь, что у пользователя есть полные права на Docker (т. е. он состоит в группе `docker`), а также на папку с R-Vision (т. е. он состоит в группе `sudoers`).

Возможен запуск скрипта установки в режиме проверки системных требований, без непосредственной установки. Для этого нужно запустить скрипт **install.sh** (см. шаг 8) и использовать ключ `--check`.

4.2.2. Шаг 1. Скачивание и распаковка установочных файлов

- Скачайте установочные файлы. Для получения ссылки обратитесь в службу клиентской поддержки ООО “Р-Вижн” по адресу: support@rvision.ru.
- Перейдите в директорию `/opt`. По умолчанию компонент устанавливается в эту директорию.
- Загрузите архив и переместите его в директорию `/opt` любым удобным способом.
- Распакуйте установочные файлы командой:

```
tar -xzvf /opt/sense-<TAG>.tar.gz -C /opt/
```

где <TAG> — это номер версии, который необходимо подставить в команду без пробелов.

> В архиве на верхнем уровне находится директория `sense`. Если указана другая директория для установки продукта, перенесите файлы из директории `sense` в нужную директорию с заменой файлов.

```
sense
├── config          # конфигурационные файлы от разработчика
├── data            # каталог данных (пустой)
├── install.sh      # скрипт установки
├── migrate.sh      # скрипт обновления структур БД (миграции)
├── sense.img       # архив Docker-образов
└── update          # каталог файлов обновления, удаляется по
завершении установки/обновления
    ├── docker-compose.prod.balancer.overrides.yml  # новая
конфигурация docker-compose для балансировщика
    ├── docker-compose.prod.yml                     # новая
конфигурация docker-compose основных сервисов
    ├── .env                                        # общий для
всех сервисов env-файл, используется командой docker-compose
автоматически
    └── release.info                                # служебная
информация о релизе (версия, хеш коммита, дата сборки и др.)
```

Code Block 11 Структура архива

3. Для экономии места на жестком диске вы можете удалить архив из директории `/opt/` после распаковки:

```
rm /opt/sense-<TAG>.tar.gz
```

4.2.3. Шаг 2. Инициализация настроек

Шаги 2-7 скрипт установки **install.sh** выполняет автоматически, инициализируя настройки по умолчанию. Если вам нужно задать индивидуальные настройки, воспользуйтесь инструкцией ниже. Чтобы применить настройки, используйте команду: `docker-compose -f docker-compose.prod.yml up -d`.

Все внешние настройки и учетные записи содержатся в файлах конфигурации в директориях с постфиксом `_default`.

7. Общие параметры конфигурации и переменные окружения: `/opt/sense/config/envs_default`.
8. Общие настройки прокси-сервера: `/opt/sense/config/envoy/config`.
9. Настройки внутреннего прокси: `/opt/sense/config/envs/balancer`.
10. Демонстрационные SSL-сертификаты: /opt/sense/config/envoy/certs_default.
11. Файлы конфигурации ClickHouse (при использовании): `/opt/sense/config/clickhouse`.

При первичной установке:

- Скопируйте директорию `/opt/sense/config/envs_default` как `/opt/sense/config/envs`:

```
cp -R /opt/sense/config/envs_default /opt/sense/config/envs
```

- Скопируйте директорию `/opt/sense/config/config_default` как `/opt/sense/config/config`:

```
cp -R  /opt/sense/config/envoy/config_default
   /opt/sense/config/envoy/config
```

- Скопируйте директорию `/opt/sense/config/envoy/certs_default` как `/opt/sense/config/envoy/certs`:

```
cp -R  /opt/sense/config/envoy/certs_default
   /opt/sense/config/envoy/certs
```

- При использовании ClickHouse скопируйте в директорию /opt/sense/config/clickhouse одну из стандартных директорий, в зависимости от вашего вида установки:

  a. /opt/sense/config/clickhouse_mono_default для монолитной установки.

  b. /opt/sense/config/clickhouse_cluster_default для кластерной установки.

После копирования стандартных файлов конфигурации приступите к их настройке.

4.2.4. Шаг 3. Изменение конфигурационных файлов окружения

При необходимости измените конфигурационные файлы в директории opt/sense/config/envs. Здесь вы можете ввести имена хостов, порты, названия БД и пароли к ним, ключи шифрования, порты коннекторов и т. д.

Переменные, обязательные для изменения:

1. В файле /opt/sense/config/envs/backend:
   - **FRONTEND_URL**: адрес, на котором будет располагаться UI.
   - **JWT_ACCESS_TOKEN_SECRET**: приватный ключ для выпуска JWT access-токена.
   - **JWT_REFRESH_TOKEN_SECRET**: приватный ключ для выпуска JWT refresh-токена.
   - **CREDENTIAL_KEY**: мастер-ключ для шифрования паролей.
2. В файле /opt/sense/config/envs/gateway:
   c. **SERVER_NAME_LIST**: список имен хостов, совпадающий с сертификатом SSL. Указывается через пробел.
   d. **SERVER_NAME**: имя хоста из сертификата SSL.
3. В файле /opt/sense/config/envs/warden/warden.production.env:
   1. **ACCESS_TOKEN**: токен для доступа к сервису Warden.

4.2.5. Шаг 4. Установка ограничений по ресурсам и масштабирование сервисов

Типы сервисов модуля, их ограничения для RAM и CPU, а также возможности масштабирования описаны в инструкции по установке ограничений ресурсов.

4.2.6. Шаг 5. Организация работы с envoy-gateway прокси

Параметры точек входа в систему (например, коннекторы) указаны в файлах директории `/opt/sense/config/envoy/config`. Для внесения изменений в параметры точек входа отредактируйте файлы в поддиректориях:

2. /opt/sense/config/envoy/config/clusters
3. /opt/sense/config/envoy/config/listeners

> Например, если вы не используете QRadar, удалите соответствующие файлы из обеих директорий. При необходимости добавьте новые точки входа в обе директории.

После любого изменения точек входа перезагрузите контейнер `gateway` при помощи docker-compose.

4.2.7. Шаг 6. Подключение источников событий

После удаления неиспользуемых коннекторов подключите источники событий, которые собираетесь использовать. Ниже приведен список инструкций для подключения стандартных источников событий:

Руководства по конфигурации соответствующих внешних систем для источников событий приведены в разделе Настройка внешних систем.

4.2.8. Шаг 7. Настройка SSL

В процессе установки распаковываются демо-сертификаты SSL. Для дальнейшей работы с сертификатами воспользуйтесь инструкцией по настройке SSL.

4.2.9. Шаг 8. Установка компонента

Установка системы после инициализации настроек производится в автоматическом режиме.

В директории `/opt/sense` выполните скрипт инсталляции **install.sh**. Скрипт автоматически загрузит все Docker-образы, необходимые для работы модуля, и запустит контейнеры.

4.2.10. Шаг 9. Проверка работоспособности

После установки проверьте наличие ошибок в запущенных сервисах:

2. изучите логи запуска и работы с помощью команды:

```
docker logs -t --tail 1000 <имя_контейнера>
```

3. Проверьте работоспособность контейнеров через сервис Warden, используя следующий URL:
   http://<ip-address>/api/warden/healthrгде <ip-address> - адрес веб-интерфейса.

Если в логах не наблюдается ошибок, перейдите в веб-интерфейс по адресу `http://<ip-address>` или `http://<dns-name>`, если есть внутреннее доменное имя. Данные для входа по умолчанию:

1. Логин: **admin**.
2. Пароль: **admin**.

По соображениям безопасности рекомендуется изменить логин и пароль после установки.

Для перезапуска модуля введите команду

```
6. docker-compose -f /opt/sense/docker-compose.prod.yml restart
```

4.2.11. Описание скрипта установки install.sh

Скрипт `install.sh` используется для установки и обновления модуля.

Лог работы сохраняется в файле **/tmp/sense-update-logs/install_<yyyyMMdd>-<HHmmss>.log**, где <yyyyMMdd> - дата (yyyy - год, MM - месяц, dd - день), а <HHmmss> - время (HH - часы, mm - минуты, ss - секунды).

В лог записывается информация о системе (при включении подробного вывода информация дублируется на консоль):

- ОС,
- версия ядра,
- версии Docker и Docker Compose,
- количество ядер CPU,
- объем памяти,
- свободное место на диске,
- свободные inodes.

Пример записи в логе:

```
--> Write system info to log
11:21:43 [INFO]
  System info:
    OS:             Debian GNU/Linux 11 (bullseye)
    Kernel:         5.10.0-20-amd64
    Docker:         Docker version 20.10.24, build 297e128
    Compose:        docker-compose version 1.29.2, build 5becea4c
    CPU, cores:     2
    RAM, GB:        6

  App directory info:
    Directory:      /opt/sense
    Data size:      1009M
    Free space:     25G
    Free inodes:    2014925

11:21:43 [INFO] check_right: current user id: 0
```

Перед началом установки выводится информация о параметрах и расположении лога, которое при необходимости можно изменить.

4.2.11.1. Параметры

```
Параметры:
      -c, --check             - только проверить возможность установки, не выполнять действий
      -h, --help            - вывести справку
      -v, --verbose         - подробный вывод
      -y, --yes             - не запрашивать подтверждение начала установки у пользователя

Поддерживаемые переменные окружения:
      - CHECK_ONLY - только проверить возможность установки, не выполнять действий
      - ACCEPTED - не запрашивать подтверждение начала установки у пользователя
      - VERBOSE  - подробный вывод
      - CLICKHOUSE_MODE - режим инициализации конфигурации ClickHouse: 'mono' или 'cluster'
      - SENSE_COMPOSE_OVERRIDES_FILE - дополнительный конфигурационный файл docker-compose
```

Code Block 12 Поддерживаемые параметры и переменные

4.2.11.2. Порядок действий

1. Подготовка к работе. Перед началом установки скрипт проверяет, соответствует ли сервер необходимым условиям.

Возможен запуск скрипта в режиме проверки системных требований без непосредственной установки с помощью параметра `--check` или переменной окружения `CHECK_ONLY`.

2. Запрос подтверждения выполнения действий у пользователя.

Запрос можно отключить параметром `--yes` или переменной окружения `ACCEPTED`.

3. Остановка текущей версии и удаление образов (при обновлении).

4. Подготовка каталогов данных (при новой установке). Создаются каталоги томов в каталоге `data`.

5. Подготовка конфигурационных файлов (при новой установке).

1. Создается копия каталога `config/envs_default` как `config/envs`.
2. Создается копия каталога config/envoy/config_default как config/envoy/config.
3. Конфигурация ClickHouse в `config/clickhouse` инициализируется стандартными настройками для монолитной или кластерной установки (определяется переменной окружения `CLICKHOUSE_MODE`).
4. Демонстрационные сертификаты помещаются в каталог `config/envoy/certs`.
5. В переменных `FRONTEND_URL`, `SERVER_NAME_LIST`, `SERVER_NAME` задается имя текущего хоста в конфигурационных файлах каталогов `config/envs/backend` и `config/envs/gateway`.

6. Настройка доступа пользователя к данным и конфигурационным файлам.

7. Загрузка новых образов из архива.

8. Выполнение миграций.

9. Запуск сервисов.

4.2.12. Перемещение корневой директории Docker и настройка лог-файлов

Перед установкой системы необходимо настроить лог-файлы Docker.

- Остановите Docker-демон:

```
service docker stop
```

- Для Docker должны быть установлены следующие ограничения по файлам журналов: формат JSON, размер журнала до 10 МБ.

a. Создайте конфигурационный файл **/etc/docker/daemon.json** со следующим содержимым:

```
{
"data-root": "/opt/var/docker",
"log-driver": "json-file",
"log-opts": {
    a. "max-size": "10m",
    b. "max-file": "3"
}
}
```

- Рекомендуется перенести данные Docker в новую директорию `/opt/var/docker`, чтобы избежать проблем с работой ОС, если на выделенной точке монтирования закончится место.

  a. Скопируйте текущие данные Docker в новую директорию и переименуйте старую Docker-директорию:

```
cp -rp /var/lib/docker/* "/opt/var/docker"
mv /var/lib/docker /var/lib/docker.old
```

- Перезапустите Docker-демон:

```
service docker start
```

- Удалите старую Docker-директорию:

```
rm -rf /var/lib/docker.old
```

4.2.13. Установка ограничений по ресурсам и масштабирование сервисов

Все сервисы подразделяются на четыре категории.

- **Хранилища** могут разворачиваться в виде отдельного кластера для увеличения доступных ресурсов хранения и увеличения производительности:
  - **ClickHouse** (clickhouse) — хранилище данных для инструментов анализа и обучения.
  - **PostgreSQL** (postgres) — хранилище пользовательских конфигураций.

- **Stateful** и **stateless-сервисы** не могут масштабироваться, доступно только ограничение по ресурсам:
    - **Gateway** — фронт-прокси и балансировщик нагрузки внешнего TCP- и HTTP-трафика.
    - **Balancer** - балансировщик нагрузки внутреннего TCP- и HTTP трафика
    - **Backend** — обеспечивает доступ к функционалу сервисов по GraphQL API.
    - **Correlation rules** - система обработки фильтров простых правил.
    - **Jm** — сервис программных экспертов.
    - **NATS** — брокер сообщений.
    - **Notifier** — система обработки уведомлений пользователей.
    - **Mailer** — интеграция с серверами электронной почты.
    - **SOAR Integration** — интеграция с компонентом управления инцидентами.
    - **Dkron** — планировщик задач и запросов.
    - **Frontend** — сервис nginx для вывода статических страниц.
    - **Change Log** — сервис для журналирования изменений в пользовательских данных и конфигурациях.
    - **DGA Domain Checker** — подсервис программных экспертов.
    - **Lookalike Domain Checker** — подсервис программных экспертов.
    - **Analyst** — сервис управления лицензиями.
    - **LDAP** — сервис синхронизации с Active Directory.
    - **Migrate** — сервис миграции при обновлении.
- Для **коннекторов** и **нормализаторов** доступно как ограничение по ресурсам, так и масштабирование.
    - **vector** — основной обработчик всего потока событий.
- Для **сервисов для сбора и отображения метрик и журналов логов элементов инсталляции** доступно только ограничение по ресурсам:
    - **Prometheus** — система мониторинга и хранения метрик в виде временных рядов.

- **Warden** — система проверки состояния сервисов и поэкземплярного восстановления их работы, а также сбора метрик.
- **Vector Container Logs** - система сбора логов контейнеров в единую таблицу в ClickHouse.
- **Vector Raw Log Collector** - система сбора дампов входящих сообщений, отправленных на специальные порты.

NATS может запускаться в нескольких экземплярах как отдельные сервисы, основанные на одном и том же образе. Сформируйте из всех развернутых экземпляров отдельный кластер по инструкции. Используйте имена сервисов в качестве доменных имен. Сконфигурируйте массив имен экземпляров для всех сервисов, использующих NATS, чтобы в случае падения одного из экземпляров произошло переподключение к другому с последующей доставкой сообщений.

4.2.13.1. Масштабирование

Вы можете масштабировать сервисы в разделе `deploy` файла **docker-compose.prod.yml**:

```
vector:
    image: registry.rvision.pro/sap/x-project/vector:1.12.0
    restart: always
    deploy:
      mode: replicated
      replicas: 4
    ...
```

Параметр `replicas` указывает количество экземпляров, запущенных для данного сервиса. **Vector** оснащен мультипоточной моделью обработки данных: пул потоков устанавливается в общее количество доступных ядер CPU для процесса.

4.2.13.2. Ограничение по ресурсам

Ограничение по ресурсам также производится в файле **docker-compose.prod.yml**. В раздел `deploy` добавьте раздел `resources` с двумя параметрами `limits` и `reservations`.

```
vector:
    image: registry.rvision.pro/sap/x-project/vector:1.12.0
    restart: always
    deploy:
      mode: replicated
      replicas: 4
      resources:
        limits:
          cpus: '2.0'
```

```
          memory: '256M'
        reservations:
          memory: '64M'
    ...
```

Раздел `limits` ограничивает ресурсы указанной максимальной величиной, а `reservations` определяет минимальный размер ресурсов, которые сразу будут выделены экземпляру при старте.

Ресурсы выделяются для каждого запущенного экземпляра.
В приведенном выше примере четыре экземпляра могут использовать восемь ядер процессора (по два на каждый экземпляр), для контейнеров будет выделено 256 МБ оперативной памяти (4 x 64) при ограничении в 1 ГБ (4 x 256).

При настройке ограничений по ресурсам для сервисов время выполнения запросов может значительно увеличиться. Чтобы избежать этого, увеличьте таймауты для запросов и соединений в настройках, которые находятся в директории `opt/sense/config/envs/gateway/` в файле **gateway.production.env** и в директории `opt/sense/config/envs/balancer/` в файле **balancer.production.env:**

```
LISTENER_ROUTE_TIMEOUT=120s
CLUSTER_CONNECTION_TIMEOUT=15s
```

Приемлемые значения для таймаутов соединений — от 60 до 120 секунд. Приемлемое значение таймаута для кластеров балансировки — 15 секунд.

Увеличение таймаута увеличит время ответа при отказе соединения.

Примерное распределение ресурсов по CPU и RAM между категориями сервисов:

Категория	CPU (%)	RAM (%)
Хранилища	50%	75%
Stateful- и stateless-сервисы	10%	10%
Коннекторы и нормализаторы	30%	10%
Сборщики метрик и дашборды	10%	5%

Реальное распределение может отличаться в зависимости от аппаратных ресурсов.

4.2.14. Настройка SSL

Для настройки демо-сертификатов SSL выполните следующие шаги:

- Добавьте порт **4000:4000** в раздел `ports` сервиса `gateway` в файле **docker-compose.prod.yml**.
- Перезапустите сервис `gateway`.

- Демо-сертификаты сгенерированы для домена `my-sense.local`. Вы можете изменить переменные `SERVER_NAME_LIST` и `SERVER_NAME`, чтобы установить другое доменное имя.

При необходимости использования собственных сертификатов добавьте сертификаты, подписанные доверенным центром сертификации, либо создайте самоподписанные сертификаты при помощи инструкции ниже. В шаге **7** приведены директории, в которые вы можете поместить сертификаты, подписанные доверенным центром сертификации: при их наличии, пропустите шаги **1-6**.

- Сгенерируйте ключ RSA-2048:

```
openssl genrsa -des3 -out rootCA.key 2048
```

- Сгенерируйте корневой сертификат:

```
openssl req -x509 -new -nodes -key rootCA.key -sha256 -days
   1024 -out rootCA.pem
```

При генерации сертификата заполните следующие поля:

Country Name: ISO-код страны.

State or Province Name: область, где была проведена официальная регистрация компании.

Locality Name: город, где была проведена официальная регистрация компании.

Organization Name: полное название организации (без сокращений).

Organizational Unit Name: название отдела организации.

Common Name: полное доменное имя центра сертификации.

Email Address: электронная почта организации.

- Добавьте корневой сертификат в список доверенных:

```
sudo cp rootCA.pem /usr/local/share/ca-certificates/rootCA.crt
sudo update-ca-certificates
```

- Создайте следующие файлы для генерации сертификата домена:

  server.csr.cnf: `CN` - доменное имя, остальные поля заполняются так же, как на шаге 2.

```
[req]
default_bits=2048
prompt=no
default_md=sha256
distinguished_name=req_distinguished_name

[req_distinguished_name]
C=RU
ST=Saint-Petersburg
L=Saint-Petersburg
O=RandomOrganization
OU=RandomOrganizationUnit
emailAddress=hello@example.com
CN=sensehost.local
```

v3.ext: `DNS.1` - доменное имя, идентичное `CN` в файле **server.csr.cnf** .

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment,
   dataEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = sensehost.local
```

- Сгенерируйте доменный ключ на основе файла **server.csr.cnf**:

```
openssl req -new -sha256 -nodes -out server.csr -newkey
   rsa:2048 -keyout key.pem -config server.csr.cnf
```

- Сгенерируйте доменный сертификат на основе файла **v3.ext**:

```
openssl x509 -req -in server.csr -CA rootCA.pem -CAkey
   rootCA.key -CAcreateserial -out cert.pem -days 500 -sha256 -
   extfile v3.ext
```

- Переместите сгенерированные ключи и сертификат в директорию `certs` :

```
/opt/sense/config/envoy/certs/rootCA.pem
/opt/sense/config/envoy/certs/cert.pem
/opt/sense/config/envoy/certs/key.pem
```

- Откройте файл **/opt/sense/config/envs/gateway/gateway.production.env.**
  - Добавьте ваши доменные имена в переменную `SERVER_NAME_LIST;`
  - Убедитесь, что переменная `SERVER_NAME` содержит имя хоста из списка в пункте выше. Если вы используете wildcard-сертификат, имя хоста должно подпадать под маску адреса.
- Определите GUID пользователя `envoy` (101 по умолчанию):

```
docker exec $(docker ps -qf name=gateway) id envoy
```

- Измените владельцев файлов сертификатов и ключа на GUID, определенный на шаге 9 (в примере - 101):

```
chown 101:101 /opt/sense/config/envoy/certs/rootCA.pem
chown 101:101 /opt/sense/config/envoy/certs/cert.pem
chown 101:101 /opt/sense/config/envoy/certs/key.pem
```

4.2.15. Настройка функционального блока управления инцидентами

- Перейдите в директорию **/opt/sense**.
- Откройте файл **docker-compose.prod.yml**.
- В разделе `gateway.ports` добавьте внешний порт **4049**, соединив его с портом контейнера **9009**:

```
gateway:
  ports:
    - '4049:9009'
```

- Перейдите в директорию **/opt/sense/config/envoy/config_default/listeners**.
- Откройте файл **vector_soar_listener.yaml**.
- Убедитесь, что порт `port_value` в разделе `address.socket_address` соответствует внутреннему порту, указанному на шаге 3:

```
address:
  socket_address:
    address: 0.0.0.0
    port_value: 9009
```

- Перенесите файл **vector_soar_listener.yaml** в директорию **/opt/sense/config/envoy/config/listeners**.
- Перейдите в директорию **/opt/sense/config/envoy/config_default/clusters**.
- Откройте файл **vector_soar.yaml**.
- Убедитесь, что порт `port_value` в разделе `load_assignment.endpoints.lb_endpoints.endpoint.address.socket_address` соответствует внутреннему порту, указанному на шаге 3:

```
connect_timeout: ${CLUSTER_CONNECTION_TIMEOUT_PLACEHOLDER}
type: strict_dns
lb_policy: round_robin
load_assignment:
  cluster_name: vector_soar
  endpoints:
    - lb_endpoints:
        - endpoint:
            address:
              socket_address:
                address: vector
                port_value: 9009
```

- Перенесите файл **vector_soar.yaml** в директорию **/opt/sense/config/envoy/config/clusters**.

При необходимости, удалите неиспользуемые коннекторы и кластеры из директорий **/opt/sense/config/envoy/config/listeners** и **/opt/sense/config/envoy/config/clusters**. Для этого удалите соответствующие файлы **.yaml**.

Запустите контейнеры:

1. Перейдите в директорию `/opt/sense`.
2. Перезапустите сервисы с пересозданием контейнеров, используя команду:

```
docker-compose -f ./docker-compose.prod.yml up --force-recreate
   --no-deps -d gateway vector
```

4.2.16. Настройка PostgreSQL на выделенном сервере

Настройка PostgreSQL на выделенном сервере включает следующие шаги:

1. Установка PostgreSQL на выделенном сервере.
2. Настройка параметров PostgreSQL.
3. Настройка сервера с основной инсталляцией SENSE для работы с PostgreSQL по сети.

4.2.16.1. Установка PostgreSQL на выделенном сервере

1. Загрузите файлы инсталляции PostgreSQL с официального сайта: https://www.postgresql.org/download/
2. Установите PostgreSQL, следуя инструкциям на сайте.

Со списком поддерживаемых операционных систем также можно ознакомиться на официальном сайте разработчика.

4.2.16.2. Настройка параметров PostgreSQL

1. Откройте конфигурационный файл **postgresql.conf**.

   Расположение файла зависит от используемой операционной системы:
   - В дистрибутивах Linux, основанных на Debian, файл **postgresql.conf** находится в директории `/etc/postgresql/11/main/`.
   - В дистрибутивах Linux, основанных на Red Hat, файл **postgresql.conf** находится в директории `/var/lib/pgsql/data/`.

   Если вы используете иную операционную систему, расположение файла **postgresql.conf** может быть другим.

2. Задайте в параметре `listen_addresses` IP-адрес сервера PostgreSQL, который будет использовать для подключения к базе данных.
3. Задайте в параметре `port` порт, который будет прослушивать сервер PostgreSQL в ожидании подключений.
4. Сохраните и закройте файл **postgresql.conf**.
5. Откройте конфигурационный файл **pg_hba.conf**. Этот файл расположен в той же директории, что и файл **postgresql.conf**.
6. Настройте аутентификацию клиентов по хешам MD5. Для этого строку `host all all 127.0.0.1/32 peer` приведите к виду `host all all 127.0.0.1/32 md5`.
7. Разрешите подключение с сервера компонента. Для этого добавьте строку `host all all IP/32 md5`, где `IP` - IP-адрес сервера компонента.
8. Сохраните и закройте файл **pg_hba.conf**.
9. Перезапустите PostgreSQL, выполнив следующую команду:

   ```
   service postgresql restart
   ```

   или:

   ```
   systemctl restart postgresql
   ```

10. При установке PostgreSQL создает суперпользователя `postgres`. По умолчанию пароль для него не задан. Чтобы задать пароль для пользователя `postgres`:

    1. Переключитесь на пользователя `postgres` в командной строке:

```
41. su postgres
```

2. Запустите утилиту `psql`, выполнив следующую команду:

```
42. psql
```

3. Задайте пароль пользователя `postgres` с помощью команды:

```
43. ALTER USER postgres WITH password 'PASSWORD';
```

где `PASSWORD` – это новый пароль пользователя `postgres`.

4. Выйдите из утилиты `psql` с помощью команды `\q`.

Настройка PostgreSQL завершена.

1. При установке компонента нужен пользователь PostgreSQL с правами на создание новых баз данных и пользователей. Для этого можно использовать пользователя `postgres` или создать нового.
2. Компонент использует следующие БД PostgreSQL:
   a. `postgres;`
   b. `sense_db;`
   c. `mail_service_db.`

   При необходимости имена БД можно изменить в конфигурации компонента.

4.2.16.3. Настройка сервера с основной инсталляцией компонента для работы с PostgreSQL по сети

При новой установке необходимо настроить параметры подключения к PostgreSQL **до запуска скрипта** `install.sh`, т. е. до выполнения шага 8 установки системы.

Если сервисы уже запущены, нужно предварительно остановить их и удалить существующие контейнеры:

```
44. cd /opt/sense
45. docker-compose -f docker-compose.prod.yml down
```

После этого можно перейти к настройке подключения к PostgreSQL:

- Задайте следующие параметры подключения к основным БД компонента в **env**-файлах в каталоге `/opt/sense/config/envs/`:

  1. `PG_DB_HOST`, `DB_HOST` - IP-адрес сервера PostgreSQL;
  2. `PG_DB_PORT`, `DB_PORT` - порт сервера PostgreSQL;
  3. `PG_DB_USERNAME`, `DB_USERNAME` - имя пользователя для подключения к БД;
  4. `PG_DB_PASSWORD`, `DB_PASSWORD` - пароль пользователя для подключения к БД;
  5. `PG_DB_DATABASE`, `DB_DATABASE` - название основной БД, по умолчанию - `sense_db`;
  6. `PG_DB_SYSTEM_DATABASE` - название системной БД, по умолчанию - `postgres`;

В этом же файле задайте параметры подключения к БД сервиса электронной почты компонента:

- `MAIL_SERVICE_DB_DATABASE` - название БД сервиса электронной почты, по умолчанию - `mail_service_db.`;
- `MAIL_SERVICE_DB_HOST` - IP-адрес сервера PostgreSQL;
- `MAIL_SERVICE_DB_PORT` - порт сервера PostgreSQL;
- `MAIL_SERVICE_DB_USERNAME` - имя пользователя для подключения к БД;
- `MAIL_SERVICE_DB_PASSWORD` - пароль пользователя для подключения к БД;
- `MAIL_SERVICE_MIGRATION_DB_PASSWORD` - пароль пользователя для выполнения миграций;
- `MAIL_SERVICE_MIGRATION_DB_USERNAME` - имя пользователя для выполнения миграций.

Распределение параметров по конфигурационным файлам:

```
# файл db-migrate/db.migrate.production.env
PG_DB_HOST=remote_IP
PG_DB_PORT=remote_port
PG_DB_USERNAME=remote_user
PG_DB_PASSWORD=remote_password
PG_DB_SYSTEM_DATABASE=postgres
PG_DB_DATABASE=sense_db
MAIL_SERVICE_DB_DATABASE=mail_service_db

# файлы:
#   backend/backend.production.env
#   jm/jm.production.env
#   notifier/notifier.production.env
#   soar-integration/soar.integration.production.env
DB_HOST=remote_IP
DB_PORT=remote_port
DB_USERNAME=remote_user
DB_PASSWORD=remote_password
DB_DATABASE=sense_db

# файл mail-service/mail.service.production.env
MAIL_SERVICE_DB_HOST=remote_IP
MAIL_SERVICE_DB_PORT=remote_port
MAIL_SERVICE_DB_DATABASE=mail_service_db
MAIL_SERVICE_DB_PASSWORD=remote_password
MAIL_SERVICE_DB_USERNAME=remote_user
MAIL_SERVICE_MIGRATION_DB_PASSWORD=remote_password
MAIL_SERVICE_MIGRATION_DB_USERNAME=remote_user
```

- Настройте синхронизацию с ClickHouse. Для этого добавьте в конфигурацию сервиса `clickhouse` секцию `extra_hosts`:

```
services:
  clickhouse:
    extra_hosts:
      - 'postgres:remote_IP'
```

где `remote_IP` - IP-адрес сервера PostgreSQL.

- Отключите сервис PostgreSQL в файле **docker-compose.prod.yml**. Для этого закомментируйте в конфигурации docker-compose определение сервиса `postgresql`, а также другие строки, помеченные комментарием `# if PostgreSQL is hosted on a separate server, then comment out the entire postgres section:`

```
#  postgres: # if PostgreSQL is hosted on a separate server,
   then comment out the entire postgres section
#    image: harbor.defensys.com/mirror/postgres:11.14-stretch
#    privileged: false
#    restart: always
...
```

5. УСТАНОВКА ФУНКЦИОНАЛЬНОГО БЛОКА ЗАЩИТЫ КОНЕЧНЫХ ТОЧЕК

5.1. Технические требования

Рекомендуемые требования к аппаратному обеспечению для размещения системы (3-5 тысяч агентов на 1 сервер):

Процессор	Память	Свободное место на диске
4vCPU	Не менее 8 ГБ	От 50 до 100 ГБ

Перед началом установки убедитесь, что следующие порты в системе открыты:

6. **1514/TCP** - агент подключается к серверу управления для постоянного взаимодействия.
7. **1515/TCP** - вновь установленный агент подключается к серверу управления для регистрации.
8. **55000/TCP** - интерфейс API, используемый для интеграции с другими решениями.
9. **80/TCP** - веб-интерфейс сервера управления.
10. **443/TCP** - веб-интерфейс сервера управления.

Обмен данными между агентом и сервером проходит с использованием шифрованного соединения.

5.1.1. Доступ к внешним системам

18. Доступ к https://threat-feed.rvision.pro осуществляется по протоколу HTTPS через порт 443/TCP.
19. Доступ к компоненту и компоненту управления инцидентами осуществляется через порты 80/TCP и 443/TCP.
20. Доступ к компоненту поведенческого анализа объектов защиты осуществляется через порт 4123/TCP.

5.2. Установка компонента

Установка системы включает в себя два последовательно выполняемых этапа:

- Установка сервера компонента из образа ISO или из файла rpoint-x.y.z.run (здесь x.y.z - номер текущей версии системы).

  - Для установки компонента на готовую ОС необходимо наличие установленной платформы Docker. Если платформа не установлена, необходимо:

скачать установочный файл по ссылке ниже, подставив вместо **PACKAGE_ID** название используемой ОС (например, **astra**, **ubuntu**) по ссылке предоставленной службой клиентской поддержки ООО “Р-Вижн” по адресу: support@rvision.ru.

- поместить установочный файл в директорию, в которой хранится файл rpoint-x.y.z.run.

- Установка агента на компьютер пользователя.

После установки системы при необходимости синхронизируйте ее с продуктами R-Vision.

5.2.1. Установка сервера из образа ISO

Чтобы установить сервер из образа ISO, выполните следующие действия:

1. Скачайте образ системы по ссылке. Для получения ссылки обратитесь в службу клиентской поддержки ООО “Р-Вижн” по адресу: support@rvision.ru.
2. На вашем гипервизоре создайте и настройте новую виртуальную машину со следующими параметрами:
   a. Тип: Linux CentOS7.
   b. Задайте параметры виртуальной машины в соответствии с информацией, приведенной в разделе **Технические требования**.
3. Установите для виртуальной машины ISO-образ системы в качестве установочного диска.
4. Запустите виртуальную машину. После запуска виртуальной машины на экране отобразится меню выбора типа установки CentOS7.
5. Введите пароль.
6. Подтвердите введенный пароль.
7. Введите полное имя хоста на сервере.
8. Укажите сетевой интерфейс, который будет использоваться при работе с системой. По умолчанию используется интерфейс **eth0**.
9. Укажите режим работы интерфейса: DHCP или статический адрес.
10. В появившемся окне проверьте и подтвердите заданные вами настройки сетевого интерфейса.
11. Когда система предложит настроить еще один сетевой интерфейс, выберите **Нет**.

12. На экране отобразится сообщение об успешном завершении настройки сетевого интерфейса и продолжении установки.
13. Система продолжит процесс установки и произведет перезагрузку.
14. После завершения установки дождитесь появления логина пользователя в командной строке консоли в формате **[login@hostname]** и информации об адресе сервера.
15. Введите в адресную строку браузера адрес сервера, отображенный в командной строке консоли на шаге 14. Система перенаправит вас на страницу авторизации.
16. Укажите следующие учетные данные:
    3. логин: **admin**
    4. пароль: **RPoint-123!**
17. Нажмите на кнопку **Войти**. В окне браузера отобразится стартовая страница системы.

5.3. Установка сервера из установочного файла

Вы можете установить сервер с помощью установочного файла rpoint-x.y.z.run.

Чтобы установить сервер:

3. Скачайте установочный файл rpoint-x.y.z.run (здесь x.y.z - номер текущей версии системы).
4. Скопируйте этот файл на целевой сервер, используя клиент WinSCP или команду scp. В примере используется файл для версии 1.4.1.

```
$ scp rpoint-1.4.1.run user@rpoint-server.example.local:/tmp
```

5. Подключитесь к серверу по протоколу ssh.

```
$ ssh user@rpoint-server.example.local
```

6. Для установки сервера выполните команды:

```
sudo chmod +x rpoint-1.4.1.run
sudo ./rpoint-1.4.1.run
```

7. После завершения установки дождитесь появления в командной строке консоли информации об адресе сервера.

8. Введите в адресную строку браузера адрес сервера компонента, отображенный в командной строке консоли на шаге 5. Система перенаправит вас на страницу авторизации.
9. Укажите следующие учетные данные:
   - логин: **admin**
   - пароль: **RPoint-123!**
10. Нажмите на кнопку **Войти**. В окне браузера отобразится стартовая страница системы.

5.3.1. Управление сервером из консоли

Для управления сервером из консоли воспользуйтесь командой:

```
systemctl COMMAND endpoint-single-node.service
```

Здесь вместо COMMAND необходимо подставить одну из следующих команд:

start - запуск сервера;

stop - остановка сервера;

restart - перезапуск сервера;

status - отображение статуса сервера.

5.3.2. Работа с агентом в различных ОС

- Установка и удаление агента на Windows
- Установка и удаление агента на macOS
- Установка агента в дистрибутивах Linux

5.3.2.1. Установка и удаление агента на Windows

Агент запускается на узле, работу которого необходимо контролировать, и взаимодействует с менеджером. Данные в режиме реального времени отправляются по зашифрованному каналу.

Поддерживаемые версии: от Windows XP до Windows 11 и Windows Server 2022.

> Для выполнения установки вам необходимы права администратора.

1. Чтобы начать процесс установки, загрузите установщик Windows.

2. Установите агент через интерфейс командной строки.

По умолчанию все файлы агента после установки сохраняются в `C:\Program Files (x86)\ossec-agent`.

5.3.2.1.1 Установка через интерфейс командной строки

Чтобы установить агент, отредактируйте переменную `RPOINT_MANAGER` так, чтобы она содержала IP-адрес или имя хоста менеджера.

Команда приведена для типа командной оболочки CMD.

Установочный файл имеет формат rpoint-x.y.z (здесь x.y.z - номер текущей версии системы). В примере используется файл для версии 1.4.1.

```
46.msiexec.exe /i rpoint-agent-1.4.1.msi /q RPOINT_MANAGER="10.10.1.1"
```

На этом процесс установки завершен. Вы можете запустить агент из графического интерфейса или выполнив следующую команду:

```
NET START RpointSvc
```

После запуска агент начнет процесс регистрации и зарегистрируется в менеджере.

5.3.2.1.2 Удаление агента на Windows

Для удаления агента выполните следующую команду:

```
msiexec.exe /x rpoint-agent-4.3.8-1.msi /qn
```

Агент будет полностью удален из Windows-системы.

5.3.2.2. Установка и удаление агента на macOS

Агент запускается на узле, работу которого необходимо контролировать, и взаимодействует с менеджером. Данные в режиме реального времени отправляются по зашифрованному каналу.

2. Чтобы начать процесс установки, загрузите агент для macOS. Он подходит для macOS Sierra или более поздней версии.
3. Выберите метод установки - через интерфейс командной строки или через графический интерфейс пользователя.

По умолчанию все файлы агента после установки сохраняются в `/Library/Ossec/`.

5.3.2.2.1 Метод установки на macOS - графический интерфейс

4. Чтобы установить агент в своей системе, запустите загруженный файл и следуйте шагам мастера установки. В случае затруднений выбирайте ответы по умолчанию.
5. Для завершения процесса установки запустите агент с помощью следующей команды:

```
# sudo /Library/Ossec/bin/wazuh-control start
```

5.3.2.2.2 Метод установки на macOS - интерфейс командной строки

2. Чтобы установить агент, отредактируйте переменную `RPOINT_MANAGER` так, чтобы она содержала IP-адрес или имя хоста менеджера, а затем выполните указанную команду. Числовой компонент названия файла имеет формат x.y.z (здесь x.y.z - номер текущей версии системы). В примере используется файл для версии 1.4.1.

```
# launchctl setenv RPOINT_MANAGER='10.10.1.1' && installer -pkg
   rpoint-agent-1.4.1.pkg -target /
```

3. Чтобы завершить процесс установки, запустите агент с помощью команды:

```
# sudo /Library/Ossec/bin/wazuh-control start
```

5.3.2.2.3 Удаление агента на macOS

Для удаления агента выполните следующие команды:

- Остановите службу агента:

```
# /Library/Ossec/bin/wazuh-control stop
```

- Удалите папку /Library/Ossec/ :

```
# /bin/rm -r /Library/Ossec
```

- Остановите и выгрузите диспетчер:

```
# /bin/launchctl unload
   /Library/LaunchDaemons/com.wazuh.agent.plist
```

- Удалите launchdaemons и StartupItems:

```
# /bin/rm -f /Library/LaunchDaemons/com.wazuh.agent.plist
# /bin/rm -rf /Library/StartupItems/WAZUH
```

- Удалите пользователей и группы:

```
# /usr/bin/dscl . -delete '/Users/wazuh'
# /usr/bin/dscl . -delete '/Groups/wazuh'
```

- Удалите данные из pkgutil:

```
# /usr/sbin/pkgutil --forget com.wazuh.pkg.rpoint-agent
```

5.3.2.3. Установка агента в дистрибутивах Linux

Для установки агента в дистрибутивах Linux используйте команду **RPOINT_MANAGER**, содержащую IP-адрес или имя хоста менеджера. Числовой компонент названия файла имеет формат x.y.z (здесь x.y.z - номер текущей версии системы). В примерах ниже используется файл для версии 1.4.1.

47. Установка агента в CentOS
48. Установка агента в Ubuntu

5.3.2.3.1 Установка агента в CentOS

Пример команды:

```
sudo RPOINT_MANAGER='10.10.1.1' rpm -i rpoint-agent-1.4.1.rpm
```

5.3.2.3.2 Установка агента в Ubuntu

Пример команды:

```
sudo RPOINT_MANAGER='10.10.1.1' dpkg -i rpoint-agent-1.4.1.deb
```