CheckpointX

UserGuide v 3.3.0

Оглавление

Установка

Для CheckpointX необходима установка приложения **smilartos-install** версии **1855.4.0_149** и выше на сервер с установленной операционной системой SmilartOS, которая основана на CoreOS версии **1855.4.0**.

Установка smilartos-install

`smilartos-install` — специальный docker-образ, содержащий информацию о всех продуктах **Smilart** и другие необходимые компоненты для работы.

Чтобы его установить, необходимо выполнить следующие действия:

1. Получить список всех версий `smilartos-install` и выбрать необходимую для установки (обычно, самую последнюю):

   ```
   $ sam se smilartos-install
     List image versions:
     smilartos-install:1855.4.0_149
   ```

2. Установить выбранную версию:

   ```
   $ sam in smilartos-install:1855.4.0_149
   ```

После установки образ `smilartos-install` должен появиться в списке образов:

```
$ sam list | grep smilartos
  smilartos-install  |  1855.4.0_149  |  int
```

После этого можно приступить к установке продуктов **Smilart**.

Установка приложения CheckpointX

1. В консоли выполняем команду

   ```
   $ installproduct
   ```

2. В списке приложений выбираем **checkpointx**.
3. Выбираем самую новую версию.
4. По окончанию установки необходимо обратиться в службу поддержки для получения лицензии и установить её.
5. Перезагружаем сервер.

Управление запуском приложения

После установки приложения оно автоматически запускается и будет автоматически перезапускаться в случае аварийного завершения. Также оно будет автоматически запускаться при перезагрузке сервера.

Для остановки приложения необходимо выполнить команду:

```
$ systemctl stop checkpointx
```

Для запуска приложения необходимо выполнить команду:

```
$ systemctl start checkpointx
```

Для перезапуска приложения необходимо выполнить команду:

```
$ systemctl restart checkpointx
```

Для просмотра статуса приложения (запущено, остановлено, завершилось с ошибкой) необходимо выполнить команду:

```
$ systemctl status checkpointx
```

Для выключения автозапуска приложения выполните команду:

```
$ systemctl disable checkpointx
```

Для включения автозапуска приложения выполните команду (включено по умолчанию):

```
$ systemctl enable checkpointx
```

Настройка приложения

Конфигурационный файл CheckpointX

Конфигурационные файлы приложения расположен по адресу `/etc/checkpointx/config`, но, как правило, приложение способно запуститься без необходимости его изменения.

current.config — конфигурационный файл текущей установленной версии приложения. Он не изменяется при переустановке приложения и предназначен для внесения изменений в поведение приложения.

default.config — полная версия файла конфигурации со значениями по-умолчанию, которые будут использоваться, если они не переопределены в файле **current.config**.

После изменения файла **current.config** приложение необходимо перезапустить.

Логирование

Все запущенные приложения размещают свои логи по адресу `/var/log/<название приложения>` и самостоятельно производят их ротацию. По умолчанию, каждый лог может состоять из 5-и файлов по 100 МБ каждый. В тексте логов используется **локальное время сервера**.

Управление камерами

Для управления камерами необходимо зайти с помощью браузера в web UI приложения **Camera Server**, который доступен на порту **8082**.

В нём можно добавить камеры в приложение, чтобы в дальнейшем можно было обрабатывать с них видео-поток. Исключение составляют веб-камеры, которые добавлять в **Camera Server** не нужно. Управление веб-камерами осуществляется на локальном компьютере в браузере. Веб-камеры можно использовать только для добавления одиночных фотографий персон.

CheckpointX Web Panel

Web UI приложения **CheckpointX** доступно по адресу `http://<checkpointx_server>`.

Для работы пользователей с приложением по зашифрованному каналу (HTTPS и WSS) и/или для возможности захвата изображения с веб-камеры в браузере необходимо настроить на сервере TLS. В этом случае доступ к web UI приложению **CheckpointX** осуществляется по адресу `https://<checkpointx_server>`.

По умолчанию, в системе присутствует пользователь с логином **smilart** и паролем **smilart** с ролью "Служба поддержки", имеющий ограниченные привилегии.

При первом заходе в приложение, этот пользователь должен сменить пароль на собственный, чтобы продолжить работу в системе.

После смены пароля рекомендуется создать нового пользователя с ролью "Администратор" для старшего персонала, который уже сможет управлять базой пользователей и персон.

Управление пользователями

1. Заходим в **Smilart CheckpointX. Web Panel**.
2. В левом меню переходим в пункт **Пользователи**.

3. В списке пользователей можно осуществить поиск по имени пользователя или логину. Вывести всех или только активных пользователей.
4. Чтобы создать пользователя нужно нажать кнопку **Создать пользователя** и заполнить отмеченные звездочкой поля (Логин, Пароль, Имя и Роль)

Роль	**Описание**
Служба поддержки	**Не может менять базу персон**. Роль для человека, который является членом группы технической поддержки для данной системы. В основном, его задачи сводятся к техническим манипуляциям с системой (работа с лицензией, управление настройками), аудитом журнала.
Администратор	Роль для человека, который является главным ответственным лицом за управление базой персон. Имеет полные права по управлению базой персон, журналом и аккаунтами рядовых сотрудников. Как правило, не является технически грамотным специалистом.
Персонал	Роль для рядового сотрудника, который заносит персон в систему. Имеет ограниченные права на удаление фотографий во избежание мошеннических действий.
Скрипт автоматизации	Роль для автоматических программ, которые настраивают базу данных системы. Действия пользователя с данной ролью не записываются в журнал. Не может заходить в web интерфейс, имеет ограниченный набор прав (на данный момент только операции с базой персон).

Пользователь может быть деактивирован, если необходимо запретить ему вход и любые другие операции в приложении без удаления его из системы (в том числе из журнала). Деактивированного пользователя можно активировать.

Для удаления пользователя из системы его необходимо сначала деактивировать.

Не рекомендуется удалять пользователя без особой необходимости. После удаления пользователя будет невозможно найти события этого пользователя в журнале, а также невозможно определить кто именно осуществил любое событие, сделанное удаленным пользователем.

При отсутствии активности пользователя в интерфейсе web UI приложения **CheckpointX** более 5 минут осуществляется принудительный выход из системы. Для продолжения работы пользователю необходимо авторизоваться повторно. Время до принудительного выхода из системы может быть скорректированно под требование заказчика. Не рекомендуется устанавливать время более 15 минут, во избежании неавторизованного

использования приложения.

Журнал CheckpointX

Журнал **CheckpointX** представляет собой два ротирующихся хранилища событий (одно для текстового описания события и второе для изображений) и индекс для поиска событий.

Его содержимое можно просмотреть пользователи с ролями "Служба поддержки" и "Администратор" в CheckpointX UI в пункте меню **Журнал**.

По нашим оценкам, размера хранилища должно хватитить на сохранение данных за пол года работы. В случае более интенсивного использования старые данные будут постепенно удаляться.

Так как хранилище изображений может начать перезаписываться раньше, чем хранилище событий, то вполне нормальна ситуация, когда для связанные с событиями изображениям со временем могут стать недоступны.

Журнал хранит свои данные в `/data/chx_journal/storage/` (настраивается в конфигурационном файле в блоке **journal.storage_path**).

Чтобы изменить размер журнала необходимо:

1. Остановить приложение **CheckpointX**.
2. Вручную удалить файлы журнала.
3. Изменить настройки в файле конфигурации **CheckpointX** в блоке **journal.blob_storage_size_gb** и/или **journal.event_storage_size_gb**.
4. Запустить приложение **CheckpointX**.

Изменение размера журнала после начала работы потребует удаление всех уже записанных в него данных!

Мониторинг

По умолчанию, вместе с данным приложением идёт приложение для сбора и визуализации метрик работы железа и приложений. Его web UI находится на порту **30000**.

Резервное копирование и восстановление

Получить список версий приложения для резервного копирования и восстановления можно с помощью команды:

```
$ sam se checkpointx_backup
```

Установить приложение для резервного копирования можно с помощью команды:

```
$ sam in checkpointx_backup:<version>
```

Запуск процедуры резервного копирования выполняется с помощью команды:

```
$ backup
```

Запуск процедуры восстановления выполняется с помощью команды:

```
$ restore file_to_restore.tar.gz
```

, например:

```
$ restore /data/share/backups/2019-04-18_11-11-54.tar.gz
```

Регулярная процедура резервного копирования настраивается в файле systemd-сервиса **regular_backup** по адресу `/etc/systemd/system/regular_backup.timer`.

Настройка межсетевого экрана

Данные инструкции требуют от выполняющего их человека соответствующей квалификации и понимания рабочего окружения. В случае ошибочных изменений возможны проблемы с удалённым доступом к серверу (в том числе и обычными пользователями), для исправления которых может потребоваться физический доступ (не удалённый) к машине системному администратору или сотруднику службы поддержки.

Для предотвращения несанционированно доступа к серверу рекомендуется настроить правила фильтрации пакетов с помощью iptables.

Правила iptables после перезагрузки сервера сохраняются благодаря сервису iptables-restore.service

Пример **systemd.unit**

```
# /etc/systemd/system/iptables-restore.service
[Unit]
Description=Restore iptables firewall rules
After=multi-user.target
Conflicts=shutdown.target

[Service]
Type=oneshot
ExecStart=/opt/bin/restore-iptables.sh

[Install]
WantedBy=multi-user.target
```

Данный Unit file использует следующий скрипт `/opt/bin/restore-iptables.sh`:

```
#!/bin/bash

iptables -F DOCKER-USER
iptables -A DOCKER-USER -i eno1 -p tcp --match multiport --dports 22,80,443 -j ACCEPT
iptables -A DOCKER-USER -i eno1 -j DROP
iptables -A DOCKER-USER -j RETURN
```

Где **eno1** внешний интерфейс (у вас может быть иной) и **22,80,443** открытые порты. Не забудьте выполнить `chmod +x /opt/bin/restore-iptables.sh`

Чтобы разрешить сервис выполните

`systemctl enable iptables-restore.service`

Для запуска сервиса выполните

`systemctl start iptables-restore.service`

Для изменения правил отредактируйте **/opt/bin/restore-iptables.sh**

Настройка TLS

Для для работы пользователей с UI по протоколам HTTPS и WSS необходимо разместить все необходимые файлы, связанные с сертификатом, по адресу `/etc/checkpointx/certs` и произвести изменения в файле конфигурации в соответствии с типом сертификата и перезапустить приложение.

Сертификат от центра сертификации

В конфигурационном файле приложения необходимо внести следующие изменения в блоке **admin_panel.tls.credentials**:

```
{ca_signed_certificate, #{certfile => "fullchain.pem", keyfile => "privkey.pem"}}
```

, где

fullchain.pem — полученный файл chain.pem и cert.pem.

privkey.pem — полученный приватный ключ для сертификата.

Сертификаты могут быть сгенерированы с помощью сервиса Let's Encrypt или более автоматизировано с помощью acme.sh.

Самоподписанный сертификат

В конфигурационном файле приложения необходимо внести следующие изменения в блоке **admin_panel.tls.credentials**:

```
{self_signed_certificate, #{cacertfile => "root.crt", certfile => "server.crt",
keyfile => "server.key"}}
```

, где

root.crt — полученный файл корневого сертификата.

server.crt — полученный файл сертификат сервера.

server.key — полученный файл приватного ключа сервера.

Reverse proxy с SSL

Если по соображениям безопасности вы не хотите размещать свои сертификаты на сервере с продуктом, то вы можете использовать отдельный сервер в качестве реверс-прокси: разместить на нем сертификаты и проксировать запросы от клиентов до сервера с продуктом делая одновременно TLS termination. Тогда на сервере с продуктом не обязательно производить настройку работы с TLS.

Пример конфигурации для сервера Nginx:

```
server {
        listen 192.168.0.xx:443;
        server_name ui.example.com;

        ssl_certificate /etc/nginx/ssl/fullchain.pem;
        ssl_certificate_key /etc/nginx/ssl/privkey.pem;
        ssl_trusted_certificate /etc/nginx/ssl/chain.pem;

        ssl on;
        ssl_session_cache builtin:1000 shared:SSL:10m;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;
        ssl_prefer_server_ciphers on;

        access_log /var/log/nginx/ui.access.log;
        error_log /var/log/nginx/ui.error.log;

        location / {
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto https;
                proxy_pass http://real-ui.example.com;
        }

        location /api/csi {
                proxy_pass http://real-ui.example.com/api/csi;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
        }
}
```

,где

listen — адрес сетевого интерфейса и порт, на котором Nginx будет принимать запросы от клиентов.

server_name — задаёт имена виртуального сервера. Эту директиву можно удалить, если у вас в конфигурации Nginx нет других виртуальных серверов.

ssl_certificate — файл с сертификатом в формате PEM для данного виртуального сервера.

ssl_certificate_key — файл с секретным ключом в формате PEM для данного виртуального сервера.

ssl_trusted_certificate — файл с доверенными сертификатами CA в формате PEM.

proxy_pass — замените `hostname` в URL на адрес сервера с приложением в обоих объявлениях этой директивы.

Checkpoint HTTP Verification API

Для поддержания обратной совместимости с API продукта "КПП", в CheckpointX реализован HTTP endpoint по адресу `http://<checkpointx_server>/api/checkpoint/verify`, который реализует Checkpoint HTTP Verification API.

Параметры запроса

cameraPid — обязательный. Идентификатор камеры, с которой необходимо запустить верификацию.

personKey — обязательный. Предоставленный идентификатор (номер) персоны, которая пытается пройти верификацию.

Лишние параметры игнорируются.

Формат ответов

Успешная верификация

Возвращается при условии того, что персона по указаному ключу соответствует персоне представленной на кадре с камеры.

Status code

`200 OK`

Content-Type

`text/json; charset=UTF-8`

Body

```
{
  "result": {
    "best correlation": 0.70300865,
    "correlation template id": "photo_id",
    "verification threshold": 0.7
  },
  "person id": "person_id",
  "person name": "person_id",
  "person description": "",
  "person key": "person_id",
  "camera pid": "camera_pid",
  "scheme pid": "camera_pid"
}
```

Неуспешная верификация

Возвращается при условии того, что время выполнения верификации истекло, либо во время выполнения текущего запроса на сервер пришел запрос на верификацию по той же самой камере.

Status code

`504 Gateway Timeout`

Content-Type

`text/json; charset=UTF-8`

Body

```
{
  "person id": "person_id",
  "person name": "person_id",
  "person description": "",
  "person key": "person_id",
  "camera pid": "camera_pid",
  "scheme pid": "camera_pid"
}
```

Другие ответы

Ключ персоны отсутствует в базе или у персоны нет фотографий

Status code

`404 Not Found`

Content-Type

`text/json; charset=UTF-8`

Body

```
{"person key":"person_id","camera pid":"camera_pid","scheme pid":"camera_pid"}
```

Камера с таким идентификатором отсутствует в системе

Status code

`504 Gateway Timeout`

Content-Type

`text/json; charset=UTF-8`

Body

```
{"camera pid":"camera_pid","person key":"perosn_id"}
```

Ключ персоны отсутствует в базе и камера с таким идентификатором отсутствует в системе

Status code

`404 Not Found`

Content-Type

`text/json; charset=UTF-8`

Body

```
{"person key":"person_id","camera pid":"camera_pid"}
```

Один или несколько обязательных параметров отсутствуют в запросе

Status code

`400 Bad Request`

Content-Type

`отсутствует`

Body

```
Incorrect request
```