

ПК «SAFEERP EM»
Установка, обновление и удаление

Аннотация

Настоящий документ содержит сведения, необходимые администратору модуля расширения программного комплекса SAFEERP (далее – ПК SAFEERP, комплекс) для установки и обновления ПК SAFEERP.

Настоящее руководство состоит из информационной части (аннотации и листа содержания) и разделов основной части, включающих в себя информацию для следующих вариантов исполнения комплекса:

- вариант 1 (Code Security Extension Module) – ПК «SAFEERP CS EM»;
- вариант 2 (Platform Security Extension Module) – ПК «SAFEERP PS EM»;
- вариант 3 (Extension module) – ПК «SAFEERP EM».

Содержание

1	ПК «SAFEERP EM»	5
1.1	Технические требования к программному обеспечению и программные требования.....	5
1.2	Установка ПК «SAFEERP EM»	6
1.3	Установка ПК «SAFEERP EM» с подключением к внешним СУБД	7
1.4	Обновление ПК «SAFEERP EM»	9
1.5	Удаление ПК «SAFEERP EM»	10
2	ПК «SAFEERP CS EM»	11
2.1	Технические требования к программному обеспечению и программные требования.....	11
2.2	Установка ПК «SAFEERP CS EM»	11
2.3	Установка ПК «SAFEERP CS EM» с подключением к внешней СУБД	12
2.4	Обновление ПК «SAFEERP CS EM»	12
2.5	Удаление ПК «SAFEERP CS EM»	12
3	ПК «SAFEERP PS EM»	13
3.1	Технические требования к программному обеспечению и программные требования.....	13
3.2	Установка ПК «SAFEERP PS EM»	14
3.3	Установка ПК «SAFEERP PS EM» с подключением внешней СУБД Jatoba ...	14
3.4	Обновление ПК «SAFEERP PS EM»	14
3.5	Удаление ПК «SAFEERP PS EM»	15
3.6	Создание пользователя с ролью PentestSystem	15
4	Редактирование .env файла	16
5	Настройка Kerberos.....	23
5.1	Создание пользователя	23
5.2	Создание SPN и keytab-файла	26
5.3	Внесение данных в docker-compose файл	27
6	Сервис HTTPServiceGIS	30
6.1	Установка сервиса	30
6.1.1	Загрузка сервиса	30
6.1.2	Публикация сервиса.....	31
6.2	Обновление сервиса.....	32
7	Установка сервера администрирования кластеров.....	33

7.1 Запуск сервера администрирования в качестве службы Windows	33
8 Необходимые настройки для подключения источников типа «Файловая система» для раздела «Статический анализатор»	36
9 Необходимые настройки для подключения почтового сервера	37
10 Настройки с системой отслеживания задач «Jira»	38
11 Обновление SSL-сертификатов	40
11.1 Проверка сертификатов	40
11.2 Обновление сертификатов	41
11.2.1 Загрузка сертификата в контейнер	41
11.2.2 Загрузка в compose файл	42
11.2.3 Внесение информации в .env-файл	42
11.2.4 Ключ с паролем	42
11.2.5 Выгрузка сертификатов с другим расширением	43
11.3 Возможные ошибки при обновлении сертификата	43
11.3.1 nginx не стартует и выдает ошибку: «cannot load certificate»	43
11.3.2 nginx не стартует и выдает ошибку: «cannot load certificate key»	44
11.3.3 nginx выдает ошибку: «key values mismatch»	44
11.3.4 Браузер показывает старый сертификат	44
11.3.5 При mount /certs пропали сертификаты из образа	44
12 Возможные ошибки и их устранения	45
12.1 Ошибки в ПК «SafeERP PS EM»	45
12.1.1 Не отвечает Агент 1С	45
Приложение 1. docker-compose.yml файл для установки ПК «SAFEERP EM» и подключению к внешней СУБД	46
Приложение 2. docker-compose.yml файл для установки ПК SAFEERP CS EM и подключению к внешней СУБД	57
Приложение 3. docker-compose.yml файл для установки ПК SAFEERP PS EM и подключению к внешней СУБД	63
Приложение 4. Генерация токена для источников типа GIT	74
Перечень сокращений	75

1 ПК «SAFEERP EM»

1.1 Технические требования к программному обеспечению и программные требования

Технические требования к оборудованию

Для работы ПК «SAFEERP Extension Module» (далее – «SAFEERP EM») требуется оборудование со следующими минимальными характеристиками:

- процессор – 32 виртуальных или физических ядер;
- ОЗУ – 64 Гбайт;
- HDD/SSD – 500 Гбайт.

Технические требования к программному обеспечению

В ОС должно быть установлено следующее ПО:

- Docker (docker compose), включая модули: docker-ce, docker-ce-cli, containerd.io и docker-compose-plugin.

ПК «SAFEERP EM» рекомендуется устанавливать на чистую 64-разрядную ОС. Поддерживаются ОС семейства Linux:

- Debian версий 11 и 12;
- Ubuntu 20.04 и 22.04;
- РЕД ОС от 7.3;
- Astra Linux 1.7 («Орел»);
- ALT Сервер от 10.2.

В ОС должно быть установлено следующее ПО:

- Docker (docker compose), включая модули: docker-ce, docker-ce-cli, containerd.io и docker-compose-plugin.

Примечание. На сервере, где разворачивается ПК, должна быть обеспечена поддержка AVX.

Состав ПК «SAFEERP EM»

ПК «SAFEERP EM» состоит из:

- сервера управления;
- сканирующего агента (Pentest);
- агента 1С (Анализ платформы).

ПК представляет собой контейнерное решение, которое включает следующие образы:

- web-security-service (wsc) – предназначен для сканирования веб-ресурсов;
- frontend-react – предназначен для работы веб-интерфейса;
- code-security – предназначен для работы раздела «Статический анализ»;
- sca – предназначен для работы раздела «SCA»;
- sca api server – предназначен для корректной работы раздела «SCA»;
- sca frontend – предназначен для корректной работы раздела «SCA»;
- authentication – предназначен для предоставления пользователям прав доступа пользователей в ПК «SAFEERP EM»;
- target-manager – предназначен для работы раздела «Цели сканирования»;

- mailing – предназначен для работы почтовых уведомлений;
- debezium connect – вспомогательный контейнер для работы почтовых уведомлений;
- redos-postgres – база данных статического анализатора. Является необязательным при установке;
- redos_mongo – база данных для хранения логов ПК «SAFEERP EM», доступных для просмотра в подразделе «Журнал»;
- redos_opensearch – система поиска, сбора, анализа и визуализации данных в подразделе «Журнал»;
- redos_graylog – сервис, предназначенный для настройки сбора и просмотра логов в подразделе «Журнал».
- dast – предназначен для работы раздела динамического анализа веб-приложений;
- infodb-c2server – предназначен для работы сервиса по обновлению БДУ;
- kafka – предназначен для обработки результатов сканирования;
- pentest-webgui-db – база данных для хранения информации о проектах раздела «Pentest»;
- pentest-info-db – база данных уязвимостей. Содержит в себе информацию о CVE, CPE;
- pentest-backend – предназначен для работы раздела «Pentest»;
- pentest-agent – сканирующий агент «Pentest»;
- gateway – предназначен для обработки запросов, корректной работы авторизации и аутентификации в ПК «SAFEERP EM»;
- proxy_auth – предназначен для аутентификации на сканируемых веб-ресурсах
- platform-security – предназначен для работы раздела «Анализ платформы».

1.2 Установка ПК «SAFEERP EM»

Для установки ПК «SAFEERP EM» необходимо выполнить следующие действия:

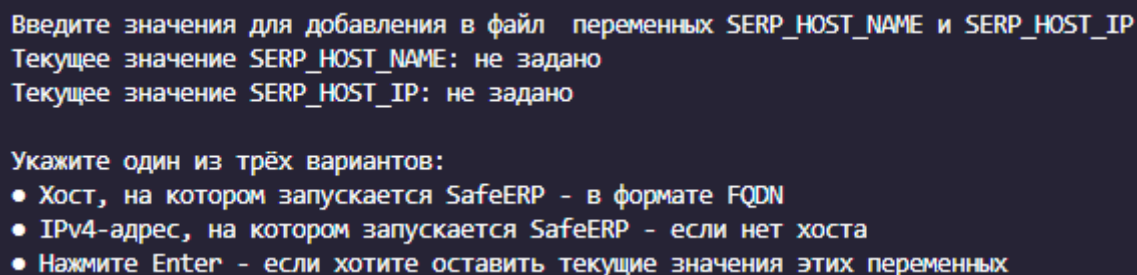
- 1) Создать служебного пользователя 'safeerp_sa'
`sudo adduser Safeerp_sa`
- 2) Установить пароль для данного пользователя
`sudo passwd Safeerp_sa`
- 3) Предоставить пользователю права на управление docker
`sudo usermod -aG docker Safeerp_sa`
- 4) Разархивировать приложение
`sudo unzip SAFEERP-<Version>.zip -d /home/Safeerp_sa`
- 5) Предоставить пользователю права на каталог
`sudo chown -R Safeerp_sa:Safeerp_sa /home/Safeerp_sa`
- 6) Проверить статус службы docker, если не запущена, то запустить
`systemctl start docker`
- 7) Запустить установку приложения
`cd /home/Safeerp_sa`

```
sudo bash install.sh
```

Примечание. После запуска `install.sh` необходимо ввести следующие значения переменных (рис. 1):

- `SERP_HOST_NAME`;
- `SERP_HOST_IP`.

Значения переменных автоматически войдут в `.env` файл.



```
Введите значения для добавления в файл переменных SERP_HOST_NAME и SERP_HOST_IP
Текущее значение SERP_HOST_NAME: не задано
Текущее значение SERP_HOST_IP: не задано

Укажите один из трёх вариантов:
• Хост, на котором запускается SafeERP - в формате FQDN
• IPv4-адрес, на котором запускается SafeERP - если нет хоста
• Нажмите Enter - если хотите оставить текущие значения этих переменных
```

Рисунок 1 – Ввод переменных при установке

При первичной установке ПК «SAFEERP EM» создаются пользователи:

- 1) Пользователь **system** с ролью администратора, с паролем по умолчанию **System__12**, который необходимо сменить при первой аутентификации.
- 2) Пользователь **operator** с ролью оператора с паролем по умолчанию **Operator__12**, который необходимо сменить при первой аутентификации.
- 3) Пользователь **securityAdmin** с ролью администратора информационной безопасности с паролем по умолчанию **SecurityAdmin__12**, который необходимо сменить при первой аутентификации.
- 4) Пользователь **pentestSystemUser** с паролем по умолчанию **PentestSystem__12**. Эта учетная запись необходима для корректной работы раздела «Pentest». При удалении параметров пользователя из `env`-файла раздел «Pentest» станет недоступен. Инструкция по созданию пользователя с ролью `PENTEST_AGENT_CALLBACK` приведено в подразделе 3.7.

Примечания:

1. Если необходимо создать подключение к конфигурации 1С с помощью источника типа 1С Клиент необходимо с помощью источника типа 1С Клиент установить толстый клиент 1С.
2. Толстый клиент 1С (файл `1cv8`) должен быть установлен на хосте, используемом для разворачивания docker контейнеров "SAFEERP CS EM, по пути `/opt/1cv8`". Настройка пути указывается в `volumes` при запуске контейнера (`docker-compose`).

1.3 Установка ПК «SAFEERP EM» с подключением к внешним СУБД

ПК «SAFEERP EM» совместима с СУБД Jatoba 5.5.2.

Перед установкой ПК, необходимо установить СУБД.

Для установки ПК «SAFEERP EM» и настройки подключения к СУБД необходимо выполнить следующие действия:

- 1) Создать служебного пользователя 'safeerp_sa'
`sudo adduser Safeerp_sa`
- 2) Установить пароль для данного пользователя
`sudo passwd Safeerp_sa`
- 3) Предоставить пользователю права на управление docker
`sudo usermod -aG docker Safeerp_sa`
- 4) Разархивировать приложение
`unzip SAFEERP-<Version>.zip -d /home/Safeerp_sa`
- 5) Предоставить пользователю права на каталог
`sudo chown -R Safeerp_sa:Safeerp_sa /home/Safeerp_sa`
- 6) Перейти в УЗ postgres в ОС
`su - postgres`
- 7) *Перейти в БД*
`psql`
- 8) Создать пользователя БД
`create user Safeerp_sa;`
- 9) Установить пароль для данного пользователя
`alter user Safeerp_sa encrypted password 'ввести пароль';`
- 10) Создать БД
`create database SAFEERP_db;`
- 11) Предоставить пользователю Safeerp_sa все полномочия на БД SAFEERP_db
`grant all privileges on database SAFEERP_db to Safeerp_sa;`
- 12) Установить пользователя SAFEERP_psem владельцем базы SAFEERP_db
`alter database SAFEERP_db owner to Safeerp_sa;`
- 13) Проверить корректность настроек (рис. 2)
`\l`

```
postgres=# \l
```

Список баз данных							
Имя	Владелец	Кодировка	LC_COLLATE	LC_CTYPE	локаль ICU	Провайдер локали	Права доступа
ems_db	safeerp_sa	UTF8	ru_RU.UTF-8	ru_RU.UTF-8		libc	=Tc/safeerp_sa + safeerp_sa=CtC/safeerp_sa
postgres	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8		libc	
safeerp_db	safeerp_sa	UTF8	ru_RU.UTF-8	ru_RU.UTF-8		libc	=Tc/safeerp_sa + safeerp_sa=CtC/safeerp_sa
template0	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8		libc	=c/postgres + postgres=CtC/postgres
template1	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8		libc	=c/postgres + postgres=CtC/postgres

(5 строк)

Рисунок 2 – Проверка корректности настроек

- 14) Выйти из консоли postgres
`exit`
- 15) Выйти из УЗ postgres
`exit`
- 16) Убедиться, что в файле /var/lib/jatoba/5/data/postgresql.conf указан следующий

параметр:

```
listen_addresses = '*'
```

- 17) Отредактировать файл (рис. 3). Необходимо настроить разрешение подключения пользователя Safeerp_sa к базе данных SAFEERP_db с хостов в диапазоне 172.0.0.0/8 (контейнерная подсеть, которая будет подключаться к БД) с использованием аутентификации md5.

```
nano /var/lib/jatoba/5/data/pg_hba.conf
```

GNU nano 4.3		/var/lib/jatoba/5/data/pg_hba.conf		Изменён	
#	TYPE	DATABASE	USER	ADDRESS	METHOD
# "local" is for Unix domain socket connections only					
local	all		all		peer
# IPv4 local connections:					
host	all		all	127.0.0.1/32	ident
host		safeerp_db	safeerp_sa	172.0.0.0/8	md5
#host		ems_db	safeerp_sa	172.42.22.0/24	md5
# IPv6 local connections:					
host	all		all	:::1/128	ident
# Allow replication connections from localhost, by a user with the					
# replication privilege.					
local	replication		all		peer
host	replication		all	127.0.0.1/32	ident
host	replication		all	:::1/128	ident

Рисунок 3 – Редактирование файла

- 18) Перезапустить службу

```
systemctl restart jatoba-5.service
```

- 19) Внести изменения в файл docker-compose.yml

```
cd /home/Safeerp_sa
```

```
nano /home/Safeerp_sa/docker-compose.yml
```

Примечание. Пример файла docker-compose.yml представлен в приложении 1.

- 20) Проверить статус службы docker, если не запущена, то запустить

```
systemctl start docker
```

- 21) Запустить установку приложения

```
cd /home/Safeerp_sa
```

```
sudo bash install.sh
```

Примечание. После запуска install.sh необходимо ввести следующие значения переменных (см. рис. 1):

- SERP_HOST_NAME;
- SERP_HOST_IP.

Значения переменных автоматически войдут в .env файл.

1.4 Обновление ПК «SAFEERP EM»

Для обновления ПК «SAFEERP EM» необходимо выполнить следующие действия:

- 1) Остановить контейнер при помощи команды `docker stop container_name`, где `container_name` – название контейнера.
- 2) Удалить контейнер `container_name` при помощи команды `docker rm -f container_name`.
- 3) Удалить образ контейнера `image_name` при помощи команды `docker rmi image_name`.
- 4) Загрузить новый образ контейнера с помощью команды `docker load --input container_name.tar`, где `container_name` – название контейнера.
- 5) Запустить контейнер при помощи команды `docker-compose up -d`.

1.5 Удаление ПК «SAFEERP EM»

Для того чтобы удалить ПК «SAFEERP EM» необходимо выполнить следующие действия:

- 1) Остановить с помощью команды `docker-compose down`.
- 2) Перейти в директорию, в которой содержится файл `docker-compose.yml` `/home/Safeerp_sa/` и удалить контейнеры при помощи команды `docker-compose rm -fsv`.
- 3) Удалить данные, сохраненные в volumes контейнеров, с помощью команд:

```
sudo rm -rf ./pgdata
```

Примечание. Данные в домашней директории где установлен ПК «SAFEERP EM» или в директории, которую указали в переменной `${HOST_HOMEDIR}` в `.env`-файле.

- 4) Удалить образы с помощью команды `docker image prune -a` и подтвердить.
- 5) Удалить неиспользуемые подсети docker с помощью команды `docker network prune` и подтвердить.

2 ПК «SAFEERP CS EM»

2.1 Технические требования к программному обеспечению и программные требования

Технические требования к оборудованию

ПК «SAFEERP Code Security Extension Module» (далее – «SAFEERP CS EM») требуется оборудование со следующими минимальными характеристиками:

- процессор – 16 виртуальных или физических ядер;
- ОЗУ – 32 Гбайт;
- HDD/SSD – 250 Гбайт.

Программные требования

ПК «SAFEERP CS EM» рекомендуется устанавливать на чистую 64-разрядную ОС. Поддерживаются ОС семейства Linux:

- Debian версий 11 и 12;
- Ubuntu 20.04 и 22.04;
- РЕД ОС от 7.3;
- Astra Linux 1.7 («Орел»);
- ALT Сервер от 10.2.

В ОС должно быть установлено следующее ПО:

- Docker (docker compose), включая модули: docker-ce, docker-ce-cli, containerd.io и docker-compose-plugin.

Примечание. На сервере, где разворачивается ПК, должна быть обеспечена поддержка AVX.

Состав ПК «SAFEERP CS EM»

ПК представляет собой контейнерное решение, которое включает следующие образы

- frontend-react;
- code-security;
- authentication;
- mailing;
- sca api server;
- sca frontend;
- target-manager;
- redos-postgres;
- redos_mongo;
- redos_opensearch;
- redos_graylog;
- kafka;
- gateway.

Подробное описание данных образов представлено в подразделе 1.1.

2.2 Установка ПК «SAFEERP CS EM»

Установка выполняется в соответствии с инструкцией, представленной в подразделе

1.2.

При первичной установке ПК «SAFEERP CS EM» создаются пользователи:

- 1) Пользователь **system** с ролью администратора, с паролем по умолчанию **System__12**, который необходимо сменить при первой аутентификации
- 2) Пользователь **operator** с ролью оператора с паролем по умолчанию **Operator__12**, который необходимо сменить при первой аутентификации.
- 3) Пользователь **securityAdmin** с ролью администратора информационной безопасности с паролем по умолчанию **SecurityAdmin__12**, который необходимо сменить при первой аутентификации.
- 4) Пользователь **pentestSystemUser** с паролем по умолчанию **PentestSystem__12**. Эта учетная запись необходима для корректной работы раздела «Pentest». При удалении параметров пользователя из env-файла раздел «Pentest» станет недоступен. Инструкция по созданию пользователя с ролью PENTEST_AGENT_CALLBACK приведено в подразделе 3.7.

Примечания:

1. Если необходимо создать подключение к конфигурации 1C с помощью источника типа 1C Designer необходимо с помощью источника типа 1C Designer установить толстый клиент 1C.
2. Толстый клиент 1C (файл 1cv8) должен быть установлен на хосте, используемом для разворачивания docker контейнеров ПК «SAFEERP CS EM», по пути `"/opt/1cv8"`.
Настройка пути указывается в volumes при запуске контейнера (docker-compose).

2.3 Установка ПК «SAFEERP CS EM» с подключением к внешней СУБД

ПК «SAFEERP CS EM» совместима с СУБД Jatoba 5.5.2.

Перед установкой ПК «SAFEERP CS EM», необходимо установить СУБД.

Для установки ПК «SAFEERP CS EM» и настройки подключения к СУБД необходимо выполнить действия, представленные в подразделе 1.3.

При установке docker-compose.yml файл необходимо привести к виду, представленному в приложении 2.

2.4 Обновление ПК «SAFEERP CS EM»

Для обновления ПК «SAFEERP CS EM» необходимо выполнить действия из подраздела 1.4.

2.5 Удаление ПК «SAFEERP CS EM»

Для того чтобы удалить ПК «SAFEERP CS EM», необходимо выполнить действия из подраздела 1.5.

3 ПК «SAFEERP PS EM»

3.1 Технические требования к программному обеспечению и программные требования

Технические требования к оборудованию

ПК «SAFEERP Platform Security Extension Module» (далее – «SAFEERP PS EM») требуется оборудование со следующими минимальными характеристиками:

- процессор – 16 виртуальных или физических ядер;
- ОЗУ – 32 Гбайт;
- HDD/SSD – 250 Гбайт;

Программные требования

ПК «SAFEERP PS EM» рекомендуется устанавливать на чистую 64-разрядную ОС. Поддерживаются ОС семейства Linux:

- Debian версий 11 и 12;
- Ubuntu 20.04 и 22.04;
- РЕД ОС от 7.3;
- Astra Linux 1.7 («Орел»);
- ALT Сервер от 10.2.

В ОС должно быть установлено следующее ПО:

- Docker (docker compose), включая модули: docker-ce, docker-ce-cli, containerd.io и docker-compose-plugin.

Примечание. На сервере, где разворачивается ПК, должна быть обеспечена поддержка AVX.

Состав ПК «SAFEERP PS EM»

ПК «SAFEERP PS EM» представляет собой контейнерное решение, которое включает следующие образы:

- web-security-service (wsc);
- frontend-react;
- authentication;
- mailing;
- target-manager;
- infodb-c2server redos-postgres;
- redos_mongo redos_opensearch;
- redos_graylog;
- dast;
- kafka;
- pentest-webgui-db;
- pentest-info-db;
- pentest-backend;
- pentest-agent;
- gateway;

- proxy_auth;
- platform-security.

Подробное описание данных образов представлено в подразделе 1.1.

3.2 Установка ПК «SAFEERP PS EM»

Установка выполняется в соответствии с инструкцией, представленной в подразделе 1.2.

При первичной установке ПК «SAFEERP PS EM» создаются пользователи:

- 1) Пользователь **system** с ролью администратора, с паролем по умолчанию **System__12**, который необходимо сменить при первой аутентификации
- 2) Пользователь **operator** с ролью оператора с паролем по умолчанию **Operator__12**, который необходимо сменить при первой аутентификации.
- 3) Пользователь **securityAdmin** с ролью администратора информационной безопасности с паролем по умолчанию **SecurityAdmin__12**, который необходимо сменить при первой аутентификации.
- 4) Пользователь **pentestSystemUser** с паролем по умолчанию **PentestSystem__12**. Эта учетная запись необходима для корректной работы раздела «Pentest». При удалении параметров пользователя из env-файла раздел «Pentest» станет недоступен. Инструкция по созданию пользователя с ролью PENTEST_AGENT_CALLBACK приведено в подразделе 3.6 .

Примечание. Для корректной работы источника типа «Агент» в разделе «Анализ платформы» необходимо установить сервис HTTPServiceGIS. Инструкция по установке сервиса представлена в разделе 6 .

3.3 Установка ПК «SAFEERP PS EM» с подключением внешней СУБД Jatoba

ПК «SAFEERP PS EM» совместима с СУБД Jatoba 5.5.2.

Перед установкой ПК, необходимо установить СУБД.

Для установки «ПК SAFEERP PS EM» и настройки подключения к СУБД необходимо выполнить действия, представленные в подразделе 1.3.

При установке docker-compose.yml файл необходимо привести к виду, представленному в приложении 3.

Примечание. Для корректной работы раздела «Анализ платформы» необходимо установить сервис HTTPServiceGIS. Инструкция по установке сервиса представлена в разделе 6 .

3.4 Обновление ПК «SAFEERP PS EM»

Для обновления ПК «SAFEERP PS EM» необходимо выполнить действия из подраздела 1.4.

3.5 Удаление ПК «SAFEERP PS EM»

Для того чтобы удалить ПК «SAFEERP PS EM», необходимо выполнить действия из подраздела 1.5.

3.6 Создание пользователя с ролью PentestSystem

Для создания пользователя с ролью PentestSystem необходимо выполнить следующие действия:

- 1) Создать нового пользователя. Подробное описание создания нового пользователя приведено в документе «Руководство администратора ПК SAFEERP PS EM».
- 2) Провести первичную авторизацию и смену пароля.
- 3) Указать в .env-файле значения логина и пароля пользователя с ролью "PENTEST_AGENT_CALLBACK" в следующих переменных:

```
{"username":<name>,"password":<password>,"role":"PENTEST_AGENT_CALLBACK"}
```

Подробное описание env файла представлено в разделе 4 .

- 4) Перезапустить сервисы с помощью команды:

```
docker compose up -d --force-recreate pentest-backend pentest-agent authentication
```

4 Редактирование .env файла

Файл .env находится в установочной директории рядом с docker-compose.yml (по умолчанию - /home/Safeerp_sa).

Для редактирования .env файла необходимо выполнить следующие действия:

- 1) cd /home/Safeerp_sa
- 2) nano .env (файл можно открыть также с другого доступного редактора)

Пример описания .env файла представлен ниже.

```
# =====
# Основные настройки
# =====
LANG=ru_RU.UTF-8
LC_ALL=ru_RU.UTF-8
TZ=Europe/Moscow
HOST_HOMEDIR=/home/SafeERP-EM/services

# =====
# Настройки баз данных
# =====
POSTGRES_USER=postgres
POSTGRES_PASSWORD=postgres

# =====
# Настройки баз данных уязвимостей
# =====
INFO_DB_PASSWORD=serp_pentest
C2SERVER_INFODB_USER=postgres
C2SERVER_INFODB_PASSWORD=qLFg6K7hEctKL7CPDT9m5WLkuzYNCn3U

# =====
# Настройки Graylog
# =====
GRAYLOG_PASSWORD_SECRET=pepperandsaltsmellsgoodright
GRAYLOG_ROOT_PASSWORD_SHA2=8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8
a81f6f2ab448a918

# =====
# Настройки Kafka
# =====
KAFKA_CFG_NODE_ID=0
KAFKA_CFG_PROCESS_ROLES=controller,broker
KAFKA_CFG_LISTENERS=PLAINTEXT://:9092,CONTROLLER://:9093,EXTERNAL://0.0.0.0:9094
KAFKA_CFG_ADVERTISED_LISTENERS=PLAINTEXT://kafka:9092,EXTERNAL://localhost:9094
KAFKA_CFG_LISTENER_SECURITY_PROTOCOL_MAP=PLAINTEXT:PLAINTEXT,CONTROLLER:PLAINTEXT,EXTERNAL:PLAINTEXT
KAFKA_CFG_CONTROLLER_QUORUM_VOTERS=0@kafka:9093
```

```
KAFKA_CFG_CONTROLLER_LISTENER_NAMES=CONTROLLER
```

```
# =====
```

```
# Настройки Pentest Agent
```

```
# =====
```

```
PENTEST_AGENT_PORT=8888
```

```
PENTEST_AGENT_PROXY_PORT=10000
```

```
PENTEST_AGENT_LOGS_LIFETIME=30
```

```
PENTEST_AGENT_SHARED_FOLDER=${HOST_HOMEDIR}/pentest/agent
```

```
PENTEST_AGENT_SEND_SCAN_RESULTS_TIMEOUT=3.0
```

```
PENTEST_AGENT_SCAN_NETWORK_OPERATIONS_TIMEOUT=2.0
```

```
PENTEST_AGENT_DNS_TIMEOUT=2.0
```

```
PENTEST_AGENT_TCP_SERVICE_WAIT_PAYLOAD_SENDING_RESULT_TIMEOUT=10.0
```

```
# =====
```

```
# Настройки Pentest Control Server
```

```
# =====
```

```
PENTEST_BACKEND_PROTO=https
```

```
PENTEST_BACKEND_PORT=8443
```

```
PENTEST_IN_SAFE_ERP_CONTOUR=true
```

```
PENTEST_HOSTS_TO_SCAN_AT_ONCE=6
```

```
PENTEST_PORTS_TO_SCAN_AT_ONCE=1000
```

```
# =====
```

```
# Настройки логирования Pentest
```

```
# =====
```

```
PENTEST_LOG_LEVEL=INFO
```

```
DJANGO_LOG_LEVEL=INFO
```

```
PENTEST_ERROR_LOG_FILE_SIZE_IN_MB=10
```

```
PENTEST_INFO_LOG_FILE_SIZE_IN_MB=10
```

```
# =====
```

```
# Настройки аутентификации
```

```
# =====
```

```
JWT_LIFE_TIME=36000
```

```
COOKIE_MAX_AGE=36000
```

```
SESSION_TIMEOUT=36000
```

```
KERBEROS_AUTHENTICATION_ENABLE=false
```

```
USER_INIT_LIST=[{"username":"system","password":"System__12","role":"ADMIN",  
,"email":"incoming01@bas.mail.ru"},
```

```
{"username":"operator","password":"Operator__12","role":"OPERATOR"},
```

```
{"username":"securityAdmin","password":"SecurityAdmin__12","role":"SECURITY  
_ADMIN"},
```

```
{"username":"pentestSystemUser","password":"PentestSystem__12","role":"PENT  
EST_AGENT_CALLBACK"}]
```

```
# =====
```

```
# Настройки почты
```

```
# =====
```

```
MAIL_HOST=smtp.example.com
```

```
MAIL_USERNAME=user@example.com
MAIL_PASSWORD=emailpassword123
MAIL_PORT=25
MAIL_TRANSPORT_PROTOCOL=smtp
MAIL_SMTP_AUTH=true
MAIL_SMTP_STARTTLS_ENABLED=true
MAIL_SMTP_SSL_ENABLED=false
MAIL_DEBUG_ENABLED=false

SCA_DATABASE_MODE="external"
SCA_DATABASE_URL="jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db"
SCA_DATABASE_DRIVER="org.postgresql.Driver"
SCA_DATABASE_USERNAME="postgres"
SCA_DATABASE_PASSWORD="postgres"
SCA_API_KEY=
SCA_BASE_URL=

#ИМЯ ХОСТА И IP
SERP_HOST_NAME=
SERP_HOST_IP=
```

Примечание. Обязательными параметрами для установки агента являются следующие:

- SERP_HOST_NAME;
- PENTEST_AGENT_PORT;
- PENTEST_AGENT_PROXY_PORT;
- PENTEST_AGENT_LOGS_LIFETIME;
- PENTEST_AGENT_SHARED_FOLDER;
- PENTEST_BACKEND_PROTO;
- PENTEST_BACKEND_PORT;
- PENTEST_IN_SAFE_ERP_CONTOUR;
- USER_INIT_LIST;
- INFODB_PASSWORD;
- C2SERVER_INFODB_USER;
- C2SERVER_INFODB_PASSWORD.

В области «Основные настройки» доступны следующие поля:

- «LANG=ru_RU.UTF-8» – установка локалей для контейнеров;
- «LC_ALL=ru_RU.UTF-8» – установка локалей для контейнеров;
- «TZ=Europe/Moscow» – установка часового пояса для контейнеров;
- «HOST_HOMEDIR=/home/SafeERP-EM/services» – директория, в которой будут храниться файлы комплекса.

В области «Настройки базы данных» доступны следующие поля:

- «POSTGRES_USER» – логин пользователя от базы данных;
- «POSTGRES_PASSWORD» – пароль пользователя от базы данных.

В области «Настройки базы данных уязвимостей» доступны следующие поля:

- «INFO_DB_PASSWORD» – пароль для read-only доступа к InfoDB (база данных с CVE);
- «C2SERVER_INFODB_USER» – имя пользователя для доступа к InfoDB, который может писать в базу данных (нужно для сервиса-обновлятора InfoDB);
- «C2SERVER_INFODB_PASSWORD» – пароль пользователя для доступа к InfoDB, который может писать в базу данных (нужно для сервиса-обновлятора InfoDB).

В области «Настройки Graylog» доступны следующие поля:

- «GRAYLOG_PASSWORD_SECRET» – секретная строка используемая Graylog для хеширования/подписывания чувствительных данных. Должна быть случайной и длинной; если меняется после создания пользователей/токенов — существующие токены/сессии могут перестать работать;
- «GRAYLOG_ROOT_PASSWORD» – SHA-256 хеш пароля администратора (root) в формате hex. Graylog сравнивает введенный при первом входе пароль, захешированный тем же алгоритмом, с этим значением.

В области «Настройка Kafka» доступны следующие поля:

- «KAFKA_CFG_NODE_ID» – уникальный идентификатор узла в кластере Kafka . Используется для адресации и согласования ролей контроллера/брокера
- «KAFKA_CFG_PROCESS_ROLES» – указывает, какие процессы выполняет этот инстанс — одновременно контроллер кластера (контролирует метаданные) и брокер (обслуживает продюсеров/консьюмеров);
- «KAFKA_CFG_LISTENERS=PLAINTEXT://:9092,CONTROLLER://:9093,EXTERNAL://0.0.0.0:9094» – список интерфейсов и портов, на которых процесс слушает входящие соединения. Формат: <имя_лисенера>://: ; список интерфейсов и портов, на которых процесс слушает входящие соединения. Формат: <имя_лисенера>://: . Доступны следующие переменные:
 - а) PLAINTEXT://:9092 — стандартный внутренний клиентский listener (принимать prod/cons на порт 9092 на всех интерфейсах по умолчанию).
 - б) CONTROLLER://:9093 — слушатель для внутренней коммуникации контроллеров, порт 9093;
 - в) EXTERNAL://0.0.0.0:9094 — внешний listener, доступный снаружи (bind на все интерфейсы) на порт 9094;
- «KAFKA_CFG_ADVERTISED_LISTENERS=PLAINTEXT://kafka:9092,EXTERNAL://localhost:9094» – адреса/эндпоинты, которые брокер сообщает клиентам/другим узлам как доступные для подключения. Используется, когда bind-адрес (LISTENERS) отличается от адреса, по которому к нему обращаются. Доступны следующие переменные:
 - а) PLAINTEXT://kafka:9092 — внутреннее имя/адрес, по которому другие контейнеры/сервисы могут подключаться;
 - б) EXTERNAL://localhost:9094 — адрес, который будет рекламироваться внешним клиентам;
- «KAFKA_CFG_LISTENER_SECURITY_PROTOCOL_MAP» – предназначено для сопоставления имени listener -> используемый протокол безопасности

(SASL/SSL/PLAINTEXT). Все три listener используют незашифрованный PLAINTEXT протокол;

- «KAFKA_CFG_CONTROLLER_QUORUM_VOTERS» – список участников кворума контроллеров в формате @:<controller_port>. Указывает, какие узлы имеют право голосовать при выборе контроллера/согласовании метаданных KRaft. В данном случае один голосующий контроллер с nodeId=0 по адресу kafka:9093;
- «KAFKA_CFG_CONTROLLER_LISTENER_NAMES» – имя listener, который используется для внутренней controller-коммуникации.

Примечание. Переменные, указанные в области «Настройка Kafka», кроме «KAFKA_CFG_NODE_ID=0» и «KAFKA_CFG_PROCESS_ROLES», конфигурируют Kafka в KRaft (без ZooKeeper) режиме, задавая роль инстанса, на каких интерфейсах и портах слушать клиентские и внутренние соединения, какие адреса рекламировать клиентам, карту безопасности listener и состав кворума контроллеров.

В области «Настройка Pentest Agent» доступны следующие поля:

- «PENTEST_AGENT_PORT» – порт стенда, на котором работает агент. Используется при эксплуатации определенных CVE, или например, при создании локального агента;
- «PENTEST_AGENT_PROXY_PORT» – порт прокси-сервера. Нужен для выполнения пентестов;
- «PENTEST_AGENT_LOGS_LIFETIME» – указывает, сколько будут храниться файлы с логами (в днях);
- «PENTEST_AGENT_SHARED_FOLDER» – директория, где находится код агента. Нужна для корректного проброса docker volumes
- «PENTEST_HOSTS_TO_SCAN_AT_ONCE» – количество хостов, сканируемых за одну итерацию;
- «PENTEST_PORTS_TO_SCAN_AT_ONCE» – количество портов, сканируемых за одну итерацию.

В области «Настройки Pentest Control Server» доступны следующие поля:

- «SEP_CONTROL_SERVER_PROTO» – HTTP-протокол, на котором запущен сервер Pentest Backend. Используется агентом при выполнении callback-запросов к Pentest Backend;
- «SEP_CONTROL_SERVER_PORT» – порт, на котором работает Pentest Backend. Используется агентом при выполнении callback-запросов к Pentest Backend.

В области «Настройки Pentest Backend» доступны следующие поля:

- «PROJECT_LOG_LEVEL» – уровень логирования в коде Pentest Backend и Pentest Agent;
- «DJANGO_LOG_LEVEL» – уровень логирования внутри фреймворка Django, который используется в проекте Pentest Backend;
- «SIZE_OF_ONE_ERROR_LOG_FILE_IN_MB» – максимальный размер одного файла с логами уровня WARNING. Размер файла необходимо указать в Мбайт;

- «SIZE_OF_ONE_INFO_LOG_FILE_IN_MB» – максимальный размер одного файла с логами уровня, заданного в поле «PROJECT_LOG_LEVEL» (если есть такой env). Размер файла необходимо указать и в Мбайтах;
- «IS_PROJECT_IN_SAFE_ERP_CONTOUR» – необходимо для успешной авторизации между Pentest Backend и Pentest Agent.

В области «Настройки SafeERP Integration» доступны следующие поля:

- «SAFEERP_DEV_URL» – адрес, куда агенту необходимо обратиться для аутентификации в Safeerp authentication;
- «CALLBACKS_URL» – адрес для агента, который позволяет выполнить callback-запрос на pentest backend.

В области «Настройки аутентификации» доступны следующие поля:

- «COOKIE_MAX_AGE» – время жизни cookie. Доступно указание времени в следующих форматах: 100s, 30m или 1h. Если внести данные без указания единицы времени, то время будет считываться в секундах;
- «SESSION_TIMEOUT» – время жизни сессии пользователя. Доступно указание времени в следующих форматах: 100s, 30m или 1h. Если внести данные без указания единицы времени, то время будет считываться в секундах;
- «KERBEROS_AUTHENTICATION_ENABLE» – включена ли аутентификация через Kerberos;
- «ADMIN_USERNAME» – имя пользователя по умолчанию с ролью ADMIN;
- «ADMIN_PASSWORD» – пароль пользователя по умолчанию с ролью ADMIN;
- «OPERATOR_USERNAME» – имя пользователя по умолчанию с ролью OPERATOR;
- «OPERATOR_PASSWORD» – пароль пользователя по умолчанию с ролью OPERATOR;
- «SEC_ADMIN_USERNAME» – имя пользователя по умолчанию с ролью SECURITY_ADMIN;
- «SEC_ADMIN_PASSWORD» – пароль пользователя по умолчанию с ролью SECURITY_ADMIN;
- «PENTEST_SYSTEM_USERNAME» – имя пользователя по умолчанию с ролью PENTEST_AGENT_CALLBACK
- «PENTEST_SYSTEM_PASSWORD» – пароль пользователя по умолчанию с ролью PENTEST_AGENT_CALLBACK.

Примечание. Если в .env-файле отсутствует пользователь с ролью «ADMIN», то контейнер «autentication» не запустится.

В области «Настройка почты» доступны следующие поля:

- «MAIL_HOST» – хост почтового сервера, который использует сервис почтовой рассылки;
- «MAIL_USERNAME» – данные для подключения ПК «SAFEERP EM» к почтовому сервису;
- «MAIL_PASSWORD» – пароль от учетной записи пользователя;

- «MAIL_PORT» – порт для подключения к почтовому серверу.

Доступны настройки раздела «Статический анализатор»:

- «SCA_DATABASE_MODE» – использовать ли внешнюю базу для SCA;
- «SCA_DATABASE_URL» – строка подключения к БД SCA;
- «SCA_DATABASE_DRIVER» – имя драйвера для подключения к БД SCA;
- «SCA_DATABASE_USERNAME» – имя пользователя БД SCA;
- «SCA_DATABASE_PASSWORD» – пароль пользователя БД SCA;
- «SCA_API_KEY» – ключ для подключения к SCA API;
- «SCA_BASE_URL» – URL-адрес для подключения к SCA API.

5 Настройка Kerberos

Для аутентификации пользователя через Kerberos необходимо выполнить следующие действия:

- 1) Создать пользователя (см. подраздел 5.1).
- 2) Создать SPN и keytab-файл (см. подраздел 5.2).
- 3) Внести данные в файл docker-compose (см. подраздел 5.3).

5.1 Создание пользователя

Для создания пользователя необходимо выполнить следующие действия:

- 1) Создать пользователя в домене, который будет использоваться как Service Principal Name (SPN) (рис. 4).

Имя	Дата изменения	Тип	Размер
Центр администрирования Active Directory	08.05.2021 11:15	Ярлык	2 КБ
Управление компьютером	08.05.2021 11:14	Ярлык	2 КБ
Управление групповой политикой	08.05.2021 11:15	Ярлык	2 КБ
Средство проверки памяти Windows	08.05.2021 11:15	Ярлык	2 КБ
Службы	08.05.2021 11:14	Ярлык	2 КБ
Службы компонентов	08.05.2021 11:14	Ярлык	2 КБ
Службы Microsoft Azure	08.05.2021 11:15	Ярлык	2 КБ
Системный монитор	08.05.2021 11:14	Ярлык	2 КБ
Система архивации данных Windows Server	08.05.2021 11:16	Ярлык	2 КБ
Сведения о системе	08.05.2021 11:15	Ярлык	2 КБ
Редактор реестра	08.05.2021 11:14	Ярлык	2 КБ
Редактирование ADSI	08.05.2021 11:15	Ярлык	2 КБ
Просмотр событий	08.05.2021 11:14	Ярлык	2 КБ
Пользователи и компьютеры Active Directory	06.12.2024 11:10	Ярлык	2 КБ
Планировщик заданий	08.05.2021 11:14	Ярлык	2 КБ
Очистка диска	08.05.2021 11:14	Ярлык	2 КБ

Рисунок 4 – Выбор домена

- 2) Выбрать в контекстном меню «Создать» → «Пользователь» (рис. 5).

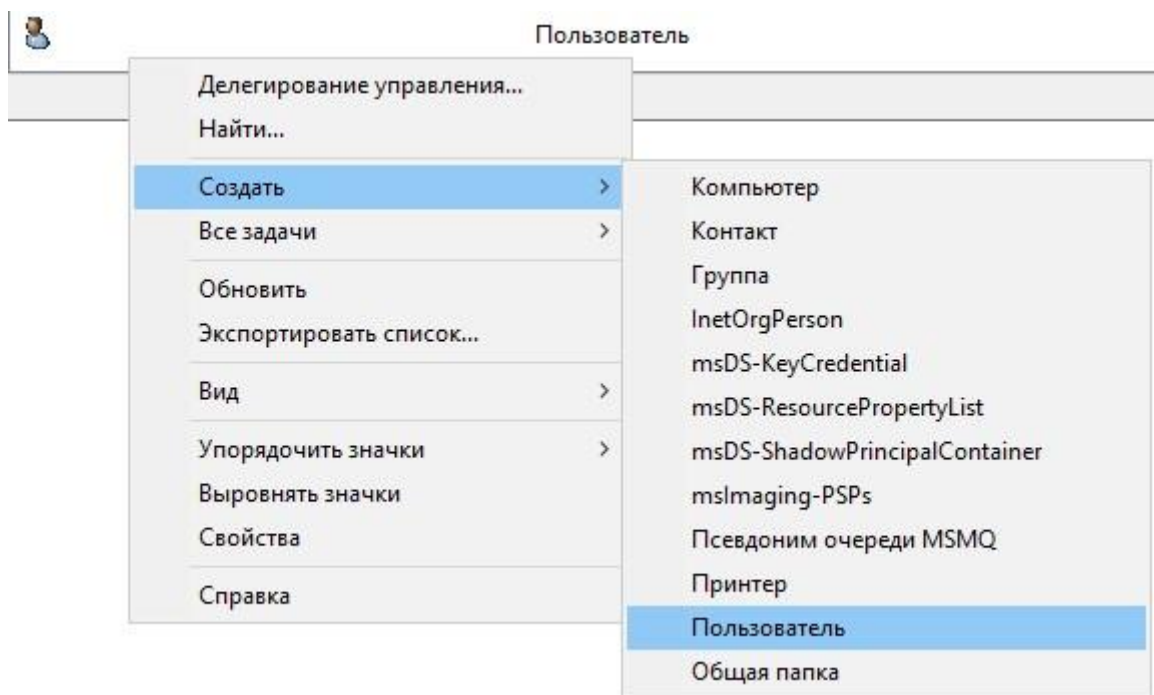


Рисунок 5 – Создание пользователя в выбранном домене

3) Заполнить поле «Имя» (рис. 6).

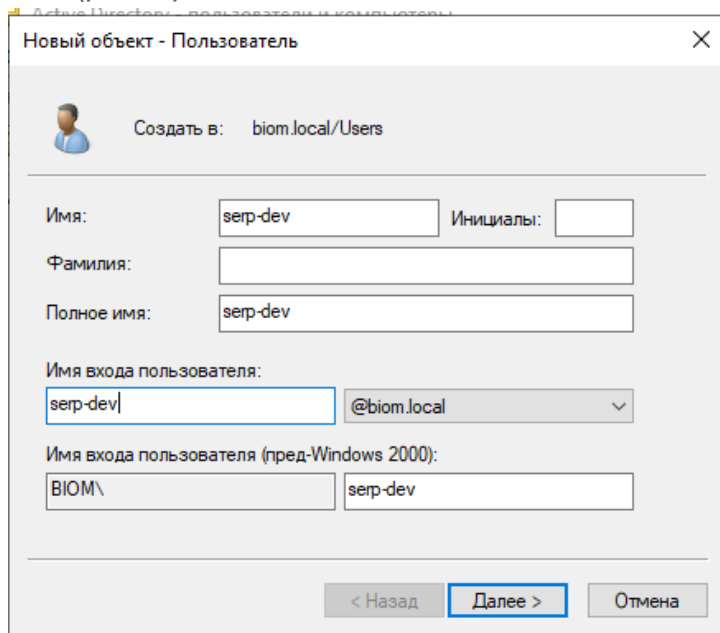


Рисунок 6 – Имя пользователя

4) Установить пароль пользователю в полях «Пароль» и «Подтверждение». Убрать флаг напротив поля «Требовать смены пароля при следующем входе в систему» (рис. 7).

Новый объект - Пользователь

Создать в: biom.local/Users

Пароль:

Подтверждение:

☐ Требовать смены пароля при следующем входе в систему

☒ Запретить смену пароля пользователем

☒ Срок действия пароля не ограничен

☐ Отключить учетную запись

< Назад **Далее >** Отмена

Рисунок 7 – Задание пароля пользователю

- 5) Зайти в свойства пользователя после создания (рис. 8).

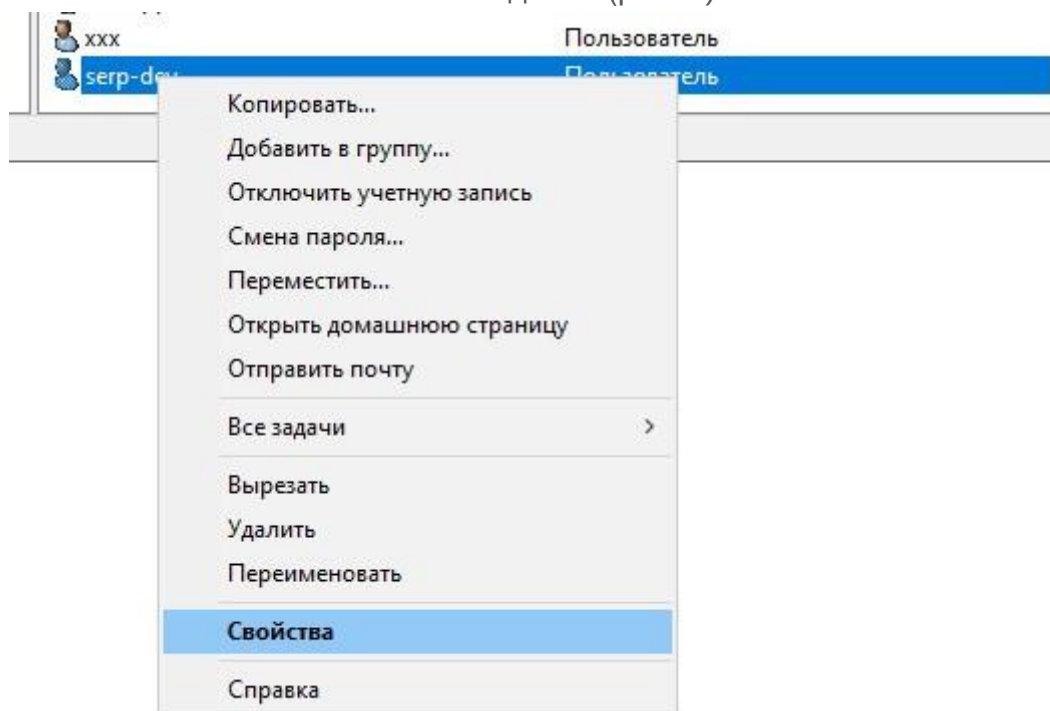


Рисунок 8 – Свойства пользователя

- 6) Переключиться на вкладку «**Учетная запись**». Активировать следующие пункты: «Данная учетная запись поддерживает 128-разрядное шифрование AES», «Данная учетная запись поддерживает 256-разрядное шифрование AES» (рис. 9).

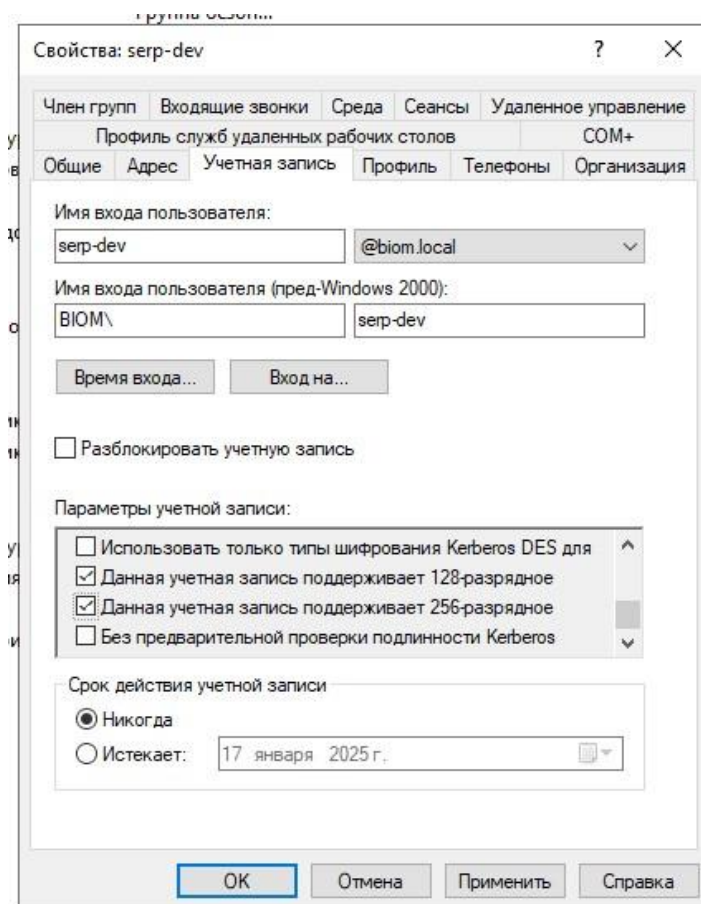


Рисунок 9 – Настройка учетной записи

5.2 Создание SPN и keytab-файла

Для создания SPN и keytab-файла необходимо выполнить следующие действия:

- 1) Запустить оболочку Powershell. Выполнить команду setspn с параметрами:
`setspn -A HTTP/SAFEERP-dev.SAFEERP.local@SAFEERP.LOCAL serp-adm`
где
-A – создание нового SPN;
HTTP/ – тип SPN;
SAFEERP-dev.SAFEERP.local@SAFEERP.LOCAL – адрес хоста приложения с реалмом Kerberos;
serp-adm – пользователь.
- 2) Посмотреть зарегистрированные SPN для пользователя с помощью команды
`setspn -L serp-adm`
где
-L – просмотр SPN;
serp-adm – пользователь.
- 3) Создать keytab-файл для SPN с помощью команды ktpass:

```
ktpass -princ HTTP/SAFEERP-dev.SAFEERP.local@SAFEERP.LOCAL -mapuser  
serp-adm -crypto ALL -ptype KRB5_NT_PRINCIPAL -pass local -target  
DC.SAFEERP.local -out c:\SAFEERP-adm.keytab
```

где

- princ – имя созданного нами принципа SPN;
- mapuser – пользователь, на который мэпится SPN;
- crypto – типы ключей (ALL - все ключи);
- ptype – тип (создается KRB5_NT_PRINCIPAL);
- pass – пароль пользователя, к которому привязываем SPN;
- target – адрес целевого контроллера домена;
- out – Путь файловой системы, где будет сохранен keytab файл.

Примечания:

1. Keytab-файл содержит набор ключей для заданного принципа.
2. Необходимо проделать процедуру создания keytab-файла для каждого хоста в домене, где будет развернут ПК «SAFEERP EM».

5.3 Внесение данных в docker-compose файл

В docker-compose файле необходимо внести следующие данные:

- KERBEROS_AUTHENTICATION_ENABLE - активирует работу и создание компонентов приложения для аутентификации с помощью Kerberos.
- SERVICE_PRINCIPAL - SPN (Service Principal Name, создание описано ниже: Kerberos аутентификация + LDAP авторизация (Сценарий Active Directory). Обычно имеет такой формат: HTTP/cs-em-dev.SAFEERP.local@SAFEERP.LOCAL

Для правильной работы доменное имя хоста, идущее после HTTP/ должно совпадать с доменным именем хоста, где будет исполняться приложение (сервер). При необходимости необходимо будет создать новый SPN.

- KEYTAB_LOCATION – переменная задаёт в файловой системе контейнера расположение keytab-файла (абсолютный путь).
- KRB5_CONFIG_LOCATION – переменная задает в файловой системе контейнера расположение файла конфигурации kerberos (абсолютный путь).

Файл выглядит следующим образом:

```
libdefaults  
default_realm = SAFEERP.LOCAL
```

```
forwardable=true

realms
SAFEERP.LOCAL = {
  kdc = DC.SAFEERP.local:88
}

domain_realm
SAFEERP.local=SAFEERP.LOCAL
.SAFEERP.local=SAFEERP.LOCAL
```

Для авторизации с помощью LDAP-сервера необходимо заполнить docker-compose файл.

В файле необходимо ввести следующие данные:

- LDAP_SERVER_URL – переменная задаёт URL LDAP-сервера Active Directory. Обычно совпадает с адресом хоста контроллера домена.

Например: ldap://DC.SAFEERP.local/

- LDAP_SEARCH_BASE – переменная задаёт базовый уровень просмотра каталогов LDAP-сервера. Для Windows чаще всего это базовый уровень домена. Например: dc=SAFEERP,dc=local

- LDAP_SEARCH_FILTER – переменная задаёт фильтр для поиска пользователя, который однозначно определит на LDAP-сервере DN пользователя, аутентифицированного с помощью Kerberos. По умолчанию для Active Directory поисковый фильтр имеет следующий вид:

(| (userPrincipalName={0}) (sAMAccountName={0}))

Пример описания файла docker-compose.yml выглядит следующим образом:

```
authentication:
  hostname: authentication.SAFEERP-em
  container_name: authentication
  image: authentication:latest
  build:
    context: http://${TOKEN}@10.10.58.101/SAFEERP-
ems/authentication.git#develop
  args:
    DOCKER_BUILDKIT: 1
    DEBUG_PORT: 5009
  environment:
    SERVICE: authentication
    SERVER_PORT: 8449
    KERBEROS_AUTHENTICATION_ENABLE: true
    SERVICE_PRINCIPAL: HTTP/cs-em-dev.SAFEERP.local@SAFEERP.LOCAL
    KEYTAB_LOCATION: /mnt/krb5-share/cs-em-dev.keytab
    KRB5_CONFIG_LOCATION: /mnt/krb5-share/krb5.conf
    LDAP_SERVER_URL: ldap://DC.SAFEERP.local/
    LDAP_SEARCH_BASE: dc=SAFEERP,dc=local
```

```
LDAP_SEARCH_FILTER: (| (userPrincipalName={0}) (sAMAccountName={0}))
ports:
- "8449:8449"
- "5009:5009"
volumes:
- /home/h2data:/app/h2data
- ./krb5-share:/mnt/krb5-share
depends_on:
- prepare
- postgres
<<: *services-common
```

Примечания:

1. В ПК «SAFEERP EM» базовым ролям ADMIN, SECURITY_ADMIN, OPERATOR присвоены группы SAFEERPAdmins, SAFEERPSecurityAdmins, SAFEERPOperators соответственно. Необходимо завести группы в Active Directory (AD) с данными названиями. Чтобы пользователь приобрел права нужной роли ПК «SAFEERP EM», необходимо добавить пользователя в соответствующую группу в AD.
2. При добавлении пользователя AD сразу в несколько групп он получит права сразу всех ролей, в группы которых был добавлен.
3. Пользователь должен иметь права на чтение групп, в которых он состоит.

6 Сервис HTTPServiceGIS

Сервис HTTPServiceGIS необходимо установить для корректной работы источника типа «Агент» в разделе «Анализ платформы». Подробное подключение агента описано в документе «Руководство администратора ПК «SafeERP PS EM».

Сервис HTTPServiceGIS предназначен для:

- выгрузки настроек конфигурации 1С;
- выгрузки списка прав и ролей пользователей.

Для корректной установки сервиса HTTPServiceGIS пользователю необходимо предоставить права на работу со следующими расширениями:

- «Полные права»;
- «Администрирование»;
- «Администрирование расширений конфигурации».

6.1. Установка сервиса

Для установки необходимо выполнить следующие действия:

- 1) Загрузить сервис (см. пункт 6.1.1).
- 2) Опубликовать сервис (см. пункт 6.1.2).

6.1.1 Загрузка сервиса

Для загрузки сервиса необходимо выполнить следующие действия:

- 1) Создать пустое расширение в конфигураторе:
«Конфигурация» → «Расширения конфигурации» (рис.10).

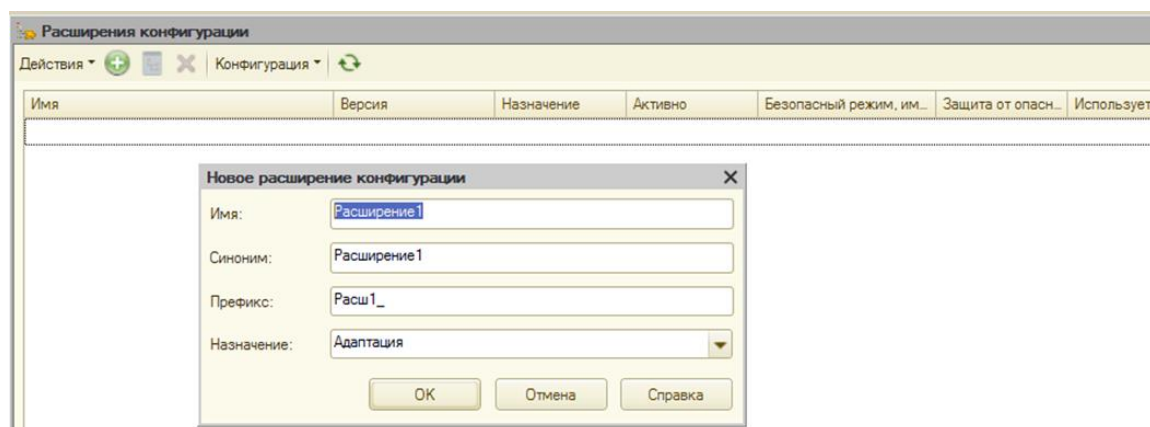


Рисунок 10 – Окно расширения конфигурации

- 2) Загрузить файл с расширением .CFE, предоставленный вместе с дистрибутивом, или загрузить файл с расширением .CFE, предоставленный вендором. Порядок загрузки расширения приведен на рис. 11.

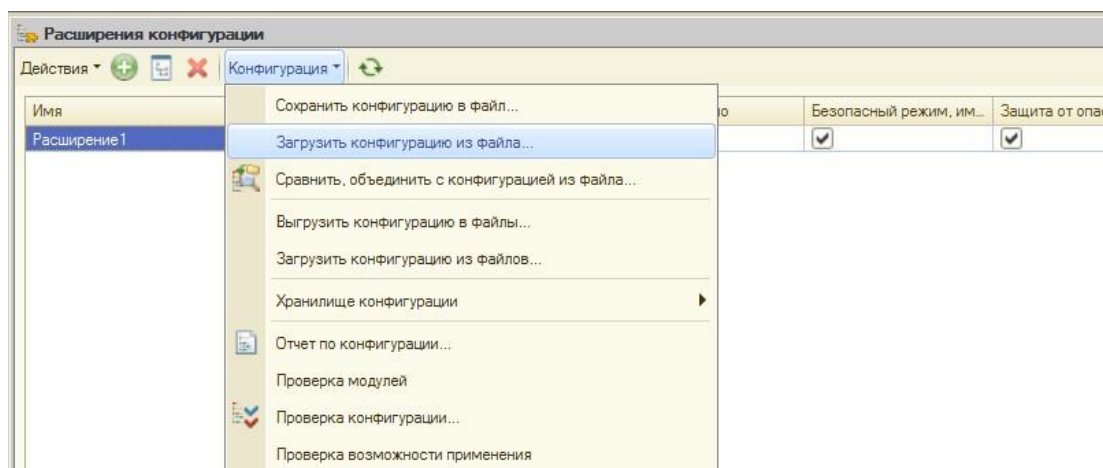


Рисунок 11 – Загрузка расширения из файла cfe

3) Убедиться, что установлен только флаг «Активно».

4) Обновить конфигурацию, нажав кнопку «F7».

Примечание. При повторной загрузке файла с расширением .CFE происходит полное замещение ранее загруженного файла.

6.1.2 Публикация сервиса

Для публикации сервиса необходимо выполнить следующие действия:

1) Перейти по следующему адресу:

«Администрирование» → «Публикация на веб-сервере».

Примечание. Если публикация ИБ уже выполнялась ранее, то необходимо установить флаги напротив «Публиковать HTTP сервисы расширений по умолчанию» и «Публиковать HTTP сервисы по умолчанию» (рис. 12).

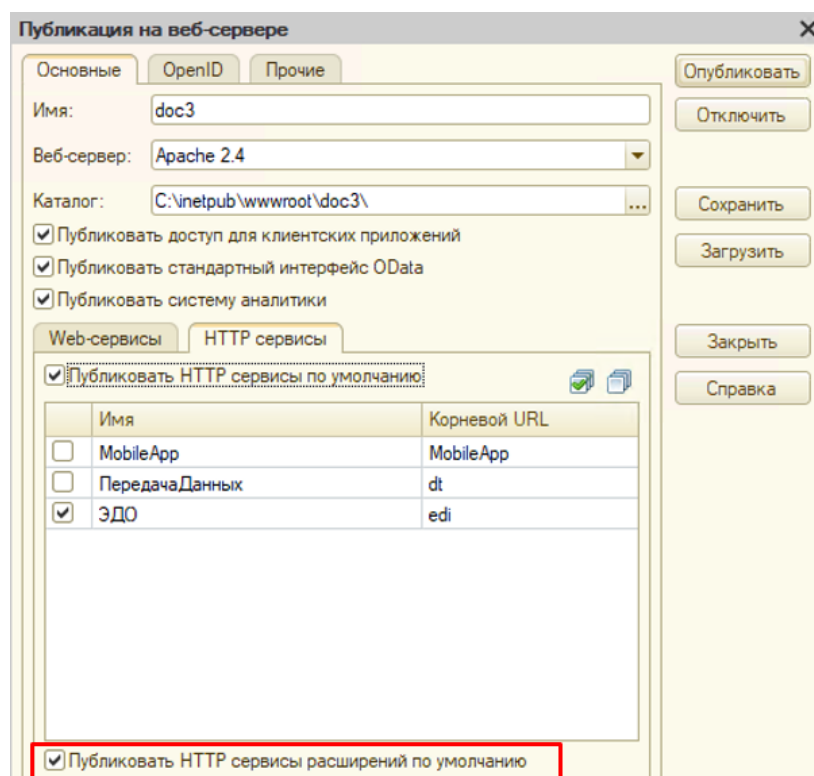


Рисунок 12 – Публикация http-сервиса в расширении

- 2) Создать учетную запись с ролью ServiceGIS (роль с правами на чтение - доступна после установки расширения).
- 3) Передать инженерам, настраивающим ПК «SAFEERP EM», следующее:
 - адрес сервера, порт, имя публикации для обращения к сервису;
 - данные учетной записи для работы с HTTP-сервисом.

Примечание. Работоспособность публикации можно проверить путем выполнения Test запроса через веб-браузер, введя в поисковой строке:

`http://<IP-адрес>/<Имя публикации>/hs/HTTPServiceGIS/Test`

Если выдается ответ True, следовательно, публикация доступна.

6.2. Обновление сервиса

Для обновления сервиса необходимо выполнить следующие действия:

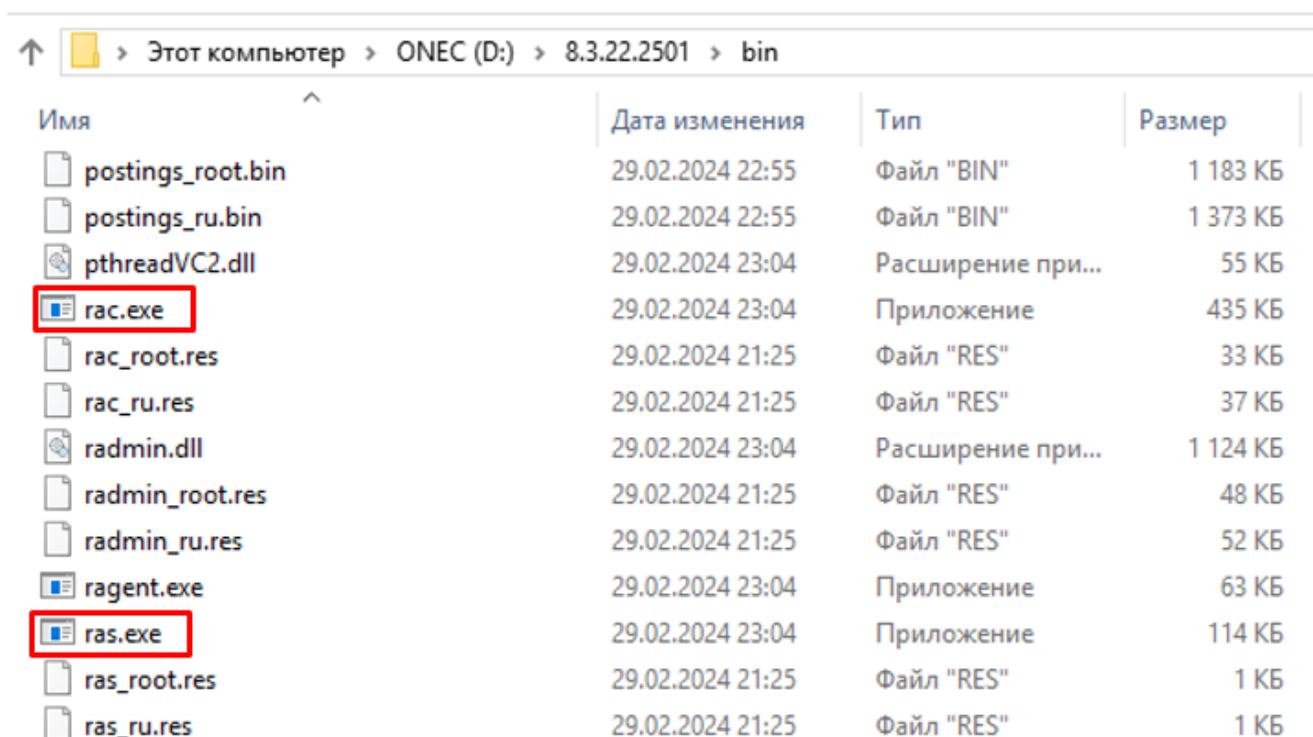
- 1) Загрузить созданное расширение с сервисом и ролью пользователя:
«Конфигурация» → «Загрузить конфигурацию из файла». Порядок загрузки расширения приведен на рис. 11.
- 2) Убрать все флаги, кроме флага «Активно».
- 3) Обновить конфигурацию, нажав кнопку «F7».

7 Установка сервера администрирования кластеров

Север администрирования используется для подключения ПК «SAFEERP EM» к кластеру или информационной базе 1С и последующего анализа изменений параметров профилей безопасности.

Сервер и утилита администрирования входят в состав компонент сервера 1С.

Для проверки данных необходимо перейти в каталог с файлами сервера 1С, найти в нем соответствующие утилиты (рис. 13).



Имя	Дата изменения	Тип	Размер
postings_root.bin	29.02.2024 22:55	Файл "BIN"	1 183 КБ
postings_ru.bin	29.02.2024 22:55	Файл "BIN"	1 373 КБ
pthreadVC2.dll	29.02.2024 23:04	Расширение при...	55 КБ
rac.exe	29.02.2024 23:04	Приложение	435 КБ
rac_root.res	29.02.2024 21:25	Файл "RES"	33 КБ
rac_ru.res	29.02.2024 21:25	Файл "RES"	37 КБ
radmin.dll	29.02.2024 23:04	Расширение при...	1 124 КБ
radmin_root.res	29.02.2024 21:25	Файл "RES"	48 КБ
radmin_ru.res	29.02.2024 21:25	Файл "RES"	52 КБ
ragent.exe	29.02.2024 23:04	Приложение	63 КБ
ras.exe	29.02.2024 23:04	Приложение	114 КБ
ras_root.res	29.02.2024 21:25	Файл "RES"	1 КБ
ras_ru.res	29.02.2024 21:25	Файл "RES"	1 КБ

Рисунок 13 – Утилиты для администрирования кластеров 1С

7.1 Запуск сервера администрирования в качестве службы Windows

Для упрощения процесса запуска сервера администрирования необходимо настроить его как службу Windows, чтобы не выполнять запуск вручную каждый раз. Для создания и регистрации службы следует использовать системную утилиту sc.

Процесс настройки службы необходимо разделить на следующие шаги:

1. Создание пользователя для службы.
Для повышения безопасности рекомендуется создать отдельного пользователя Windows, который будет использоваться для работы службы. Например, создать локального пользователя с именем USR1CV8_RAS и паролем Pass123.
2. Создание скрипта для регистрации службы.
Для регистрации службы необходимо создать и выполнить bat-файл. Содержимое

файла должно включать команды для создания службы через утилиту sc. Пример содержимого bat-файла:

```
@echo off
rem %1 - полный номер версии 1С:Предприятия
set SrvUserName=.\USR1CV8_RAS
set SrvUserPwd="Pass123"
set CtrlPort=1540
set AgentName=localhost
set RASPort=1545
set SrvcName="1С:Enterprise 8.3 Remote Server"
set BinPath="\"C:\Program Files\1cv8\%1\bin\ras.exe\" cluster --service
--port=%RASPort% %AgentName%:%CtrlPort%"
set Description="1С:Enterprise 8.3 Remote Server"
sc stop %SrvcName%
sc delete %SrvcName%
sc create %SrvcName% binPath= %BinPath% start= auto obj= %SrvUserName%
password= %SrvUserPwd% displayname= %Description%
```

Данный процесс позволяет настроить автоматический запуск сервера администрирования при старте системы.

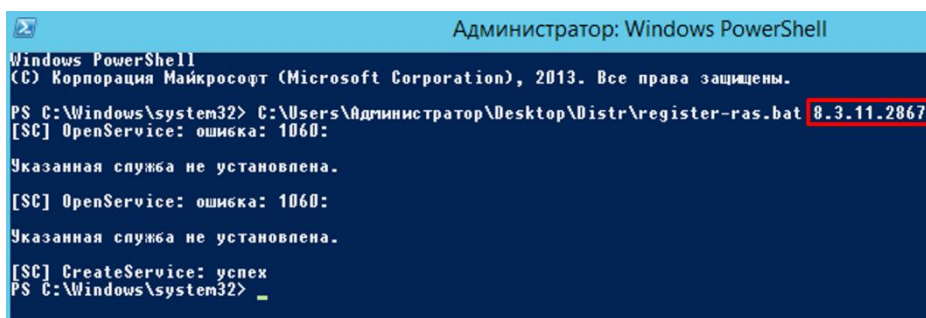
В bat-файле необходимо указать следующие данные:

- Имя пользователя и пароль (переменные SrvUserName и SrvUserPwd), под которыми будет запускаться служба. Это необходимо для корректной работы службы с правами, соответствующими заданному пользовательскому аккаунту.
- Адрес и порт агента сервера (переменные AgentName и CtrlPort), которые будут использоваться для подключения к серверу, который подлежит администрированию.
- Имя службы и сетевой порт (переменные RASPort и SrvcName), на которых будет работать сервер администрирования. Эти параметры являются изменяемыми. Если требуется запустить несколько экземпляров серверов администрирования, например, для обслуживания различных серверов 1С, чтобы избежать конфликта портов.

Эти значения необходимо настроить в соответствии с заданными требованиями предприятия, чтобы соответствовать специфике сети и инфраструктуры.

После того как все параметры настроены, необходимо запустить bat-файл для регистрации службы.

В качестве единственного параметра bat-файла выступает текущая версия платформы 1С. Это необходимо для того, чтобы служба могла корректно взаимодействовать с нужной версией платформы при запуске (рис. 14).



```
Администратор: Windows PowerShell
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.
PS C:\Windows\system32> C:\Users\Администратор\Desktop\Distr\register-ras.bat 8.3.11.2867
[SC] OpenService: ошибка: 1060:
Указанная служба не установлена.
[SC] OpenService: ошибка: 1060:
Указанная служба не установлена.
[SC] CreateService: успех
PS C:\Windows\system32>
```

Рисунок 14 – Запуск скрипта

После регистрации службы в системе необходимо открыть окно «Службы» (рис. 15), найти службу с указанным именем и запустите её, выбрав соответствующий пункт в контекстном меню. После этого сервер администрирования начнет работать в фоновом режиме.

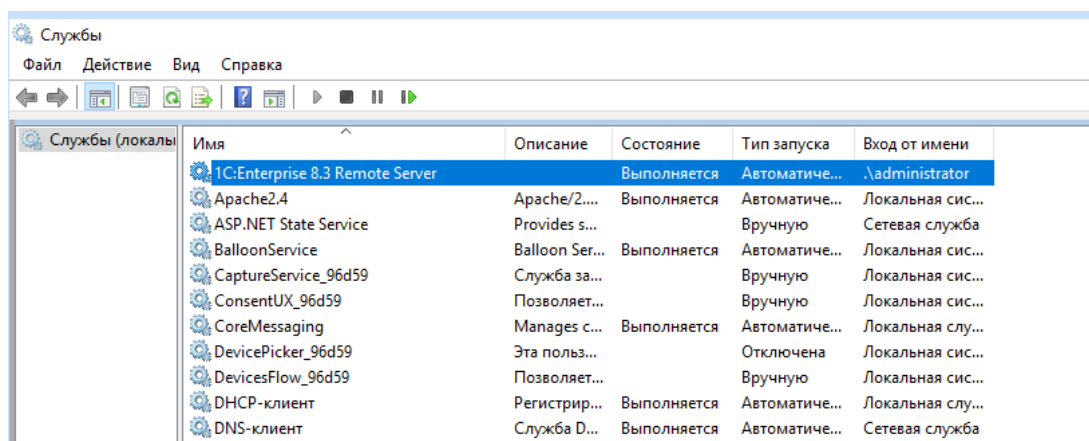


Рисунок 15 – Запуск сервера администрирования в качестве службы

8 Необходимые настройки для подключения источников типа «Файловая система» для раздела «Статический анализатор»

При установке контейнера `target-manager` автоматически создаётся папка `se_em_sources_folder`, в которой будут храниться файлы для анализа источников типа файловая система для статического анализа кода.

Путь к директории, в которой будет расположена эта папка, настраивается через переменную окружения `HOST_HOMEDIR` в файле `.env`. Подробное описание `.env` представлено в разделе 4.

Примечания:

1. В корневой папке могут храниться директории, архивы `zip` и отдельные файлы с программным кодом. Все эти объекты можно добавить как источник в цели сканирования.
2. Субдиректории или отдельные файлы для сканирования можно выбрать в поле «Объекты» при добавлении таких источников в проект. Отдельные файлы из `zip` архива выбрать нельзя.

9 Необходимые настройки для подключения почтового сервера

Для корректной работы почтового сервера необходимо задать переменные в файле .env . Подробное описание .env представлено в разделе 4 .

Пример заполнения переменных представлен ниже:

```
MAIL_HOST=mailserver.corp.lan
MAIL_USERNAME=admin@corp.lan
MAIL_PASSWORD=password
MAIL_PORT=25
MAIL_DEBUG_ENABLED=false
```

Примечания:

1. В полях необходимо указать следующую информацию:
 - MAIL_HOST – хост почтового сервера, который использует сервис почтовой рассылки;
 - MAIL_USERNAME – данные для подключения ПК «SAFEERP EM» к почтовому сервису;
 - MAIL_PASSWORD – пароль от учетной записи пользователя;
 - MAIL_PORT – порт для подключения к почтовому серверу;
 - MAIL_DEBUG_ENABLED – добавление отладочных сообщений к логу работы.
2. В файле .env доступны следующие переменные:
 - SENDER_EMAIL – адрес электронной почты;
 - SENDER_NAME – имя отправителя. применяется только в паре с SENDER_EMAIL.Если эти переменные не указаны, отправитель письма по умолчанию you-sender-name <you-sender@you-domain.net>
3. Для применения изменений из файла .env в контейнере mailing (отвечающий за почтовую рассылку) необходимо выполнить перезагрузку контейнера. Для этого необходимо ввести команду:
`docker compose up -d --force-recreate mailing`

10 Настройки с системой отслеживания задач «Jira»

Чтобы изменения в системе отслеживания задач «Jira» поступали в ПК «SafeERP EM», необходимо выполнить соответствующую настройку в этой системе.

Необходимо выполнить следующие действия:

- 1) Перейти по следующему пути «Администрирование» → «Система» → «Дополнительно» → «Веб-хуки».
- 2) Нажать «+Создать веб-хук» в разделе «Веб – хуки (Webhooks)».
- 3) Выбрать статус «Включен» и заполнить необходимые поля в окне создания веб-хук (рис. 16).

Имя* Наименование нового webhook

Статус* ☒ Включен ☐ Выключен

URL*

Можно использовать следующие дополнительные переменные в URL: \${board.id}, \${comment.id}, \${issue.id}, \${issue.key}, \${mergedVersion.id}, \${modifiedUser.key}, \${modifiedUser.name}, \${project.id}, \${project.key}, \${sprint.id}, \${version.id}

[Подробнее](#)

Описание

События **События, относящиеся к проблеме**

События для проблем и журналов работ. Вы можете указать запрос JQL только для отправки событий, инициируемых соответствующими проблемами.

☒ project = SafeERP

[Описание синтаксиса](#)

Комментарий	Журнал работ	Проблема	Ссылка на проблему
☒ создано	☐ создано	☐ создано	☐ создано
☒ обновлено	☐ обновлено	☒ обновлено	☐ удалено
☐ удалено	☐ удалено	☒ удалено	
		☐ журнал работ изменен	

События, относящиеся к проекту

События для проектов и версий проектов.

Проект	Версия
☐ создано	☐ выпущено
☐ обновлено	☐ не выпущено
☐ удалено	☐ создано
☐ в архиве	☐ перемещено
☐ восстановленные	☐ обновлено
	☐ объединено
	☐ удалено

Рисунок 16 – Настройка области «События»

- 4) Заполнить область «События» согласно рис. 16.
- 5) Нажать кнопку «Создать».

Примечания:

1. В поле «Имя» необходимо внести название нового веб-хука.
2. В поле «URL» необходимо ввести адрес, на котором расположен task-orchestrator в следующем формате:

[https://example:8459/task-orchestrator/api/jira/webhook?project_id=\\${project.id}&issue_key=\\${issue.key}](https://example:8459/task-orchestrator/api/jira/webhook?project_id=${project.id}&issue_key=${issue.key}) ,

где example - имя хоста, где развернут сервис.

11 Обновление SSL-сертификатов

Обновлять сертификаты необходимо в том случае, если у сертификатов истекает срок действия.

В frontend HTTPS обслуживается через nginx внутри контейнера frontend-react.

Для работы контейнера необходимы следующие порты:

- 8080 – внутри контейнера при использовании http-протокола;
- 8443 – внутри контейнера при использовании https-протокола.

На хосте эти порты прописаны следующим образом:

ports:

- "\${FRONTEND_HTTP_PORT:-80}:8080"
- "\${FRONTEND_HTTPS_PORT:-443}:8443"

Снаружи frontend открывается по `https://<host>/`, а внутри контейнера nginx работает на 8443.

Расположение сертификатов для nginx

По умолчанию TLS-файлы находятся по следующему адресу:

`/certs/tls.crt`

`/certs/tls.key`

Где:

- `tls.crt` — сертификат frontend. Обычно это fullchain: сам сертификат + промежуточные сертификаты УЦ;
- `tls.key` — приватный ключ от этого сертификата.

В проекте также есть SSL-настройки nginx:

`nginx/conf.d/01-frontend-ssl-params.conf`

Обычно этот файл менять не нужно. Он отвечает за параметры TLS:

- cipher suite;
- HSTS и прочие настройки nginx.

Перед заменой сертификатов необходимо выполнить следующие действия:

- 1) Проверить сертификата (см. подраздел 11.1).
- 2) Обновить сертификат (см. подраздел 11.2).

11.1 Проверка сертификатов

Перед заменой сертификата необходимо проверить, что сертификат читается. Для проверки необходимо выполнить следующие действия:

- 1) Ввести следующее:

```
openssl x509 -in tls.crt -noout -subject -issuer -dates
```

Примечание. Пример ожидаемого вывода:

```
subject=...
```

```
issuer=...
```

```
notBefore=...
```

```
notAfter=...
```

- 2) Проверить приватный ключ:
`openssl pkey -in tls.key -noout`
- 3) Проверить, что сертификат и ключ совпадают:
`openssl x509 -in tls.crt -pubkey -noout | openssl pkey -pubin -outform DER | sha256sum`
`openssl pkey -in tls.key -pubout -outform DER | sha256sum`

Примечания:

1. Хэши должны совпасть.
2. Если хэши разные, то сертификат и ключ не пара. nginx с ними не заработает.

11.2 Обновление сертификатов

Обновление сертификатов осуществляется следующими способами:

- загрузка в контейнер;
- загрузка в compose-файл;
- внесение информации в .env-файл;
- ключ с паролем;
- выгрузка сертификатов с другим расширением.

11.2.1 Загрузка сертификата в контейнер

Для загрузки контейнера необходимо выполнить следующие действия:

- 1) Создать каталог, где будут расположены сертификаты:
`sudo mkdir -p /home/SafeERP-EM/services/frontend/certs`
- 2) Загрузить файлы:
`/home/SafeERP-EM/services/frontend/certs/tls.crt`
`/home/SafeERP-EM/services/frontend/certs/tls.key`
- 3) Присвоить права:
`sudo chmod 644 /home/SafeERP-EM/services/frontend/certs/tls.crt`
`sudo chmod 640 /home/SafeERP-EM/services/frontend/certs/tls.key`
- 4) Добавить поле «Volumes»:
frontend-react:
container_name: frontend-react
hostname: frontend_react.safeerp-em
image: frontend-react
ports:
- "\${FRONTEND_HTTP_PORT:-80}:8080"
- "\${FRONTEND_HTTPS_PORT:-443}:8443"
volumes:
- /opt/safeerp/frontend/certs:/certs:ro # <---
environment:
<<: *env-common
- 5) Пересоздать контейнер:
`docker compose up -d --force-recreate frontend-react`

Примечание. Если контейнер был остановлен, то необходимо ввести следующее:
`docker compose up -d frontend-react`

11.2.2 Загрузка в compose файл

Если сертификаты необходимо хранить рядом с compose-файлом, то необходимо выполнить следующие действия:

- 1) Ввести каталог, где расположены сертификаты
`./nginx/runtime-certs/tls.crt`
`./nginx/runtime-certs/tls.key`
- 2) Добавить место расположение сертификатов в область «volumes»:
`volumes:`
`- ./nginx/runtime-certs:/certs:ro`

Структура будет выглядеть следующим образом:

```
nginx/  
  runtime-certs/  
    tls.crt  
    tls.key
```

11.2.3 Внесение информации в .env-файл

В .env-файле указать свои пути явно, задав переменные `NGINX_SSL_CERTIFICATE` и `NGINX_SSL_CERTIFICATE_KEY`.

Например:

```
frontend-react:  
  environment:  
    <<: *env-common  
    NGINX_SSL_CERTIFICATE: /run/certs/company.fullchain.pem  
    NGINX_SSL_CERTIFICATE_KEY: /run/certs/company.key  
  volumes:  
    - /opt/certs/company.fullchain.pem:/run/certs/company.fullchain.pem:ro  
    - /opt/certs/company.key:/run/certs/company.key:ro
```

В большинстве случаев рекомендуется использовать стандартные пути:

```
/certs/tls.crt  
/certs/tls.key
```

11.2.4 Ключ с паролем

Для `nginx` рекомендуется использовать приватный ключ без пароля. В противном случае контейнер не сможет нормально стартовать без дополнительной настройки, так как `nginx` должен получить пароль при запуске.

Для проверки ключа необходимо ввести следующее:

```
openssl pkey -in tls.key -noout
```

Примечание. Если команда просит пароль, следовательно, ключ зашифрован. Для того чтобы снять пароль с ключа необходимо выполнить следующие действия:

1. Ввести:

```
cp tls.key tls.key.encrypted  
openssl pkey -in tls.key.encrypted -out tls.key
```
2. Ввести следующее для проверки ключа:

```
openssl pkey -in tls.key -noout
```

Примечания:

1. Если команда завершилась без ошибок и не спросила пароль, следовательно, ключ готов.
2. Если по требованиям пароль с ключа снимать нельзя, нужно использовать NGINX_SSL_PASSWORD_FILE.

Например:

```
frontend-react:
```

```
environment:
```

```
<<: *env-common
```

```
NGINX_SSL_PASSWORD_FILE: /run/secrets/nginx_ssl_password
```

volumes:

```
- /home/SafeERP-EM/services/frontend/certs:/certs:ro
```

```
-
```

```
/home/SafeERP-
```

```
EM/services/frontend/secrets/nginx_ssl_password:/run/secrets/nginx_ssl_password:ro
```

3. Файл, в котором находится пароль от приватного ключа, расположен по следующему адресу:

```
/home/SafeERP-EM/services/frontend/secrets/nginx_ssl_password
```

11.2.5 Выгрузка сертификатов с другим расширением

Если сертификат имеет расширение отличное от tls.crt и tls.key, то доступна выгрузка сертификатов в пути контейнера.

Для выгрузки сертификатов с другим расширением необходимо ввести следующее:

volumes:

```
- /opt/certs/company.fullchain.pem:/certs/tls.crt:ro
```

```
- /opt/certs/company.key:/certs/tls.key:ro
```

Для nginx внутри контейнера файлы будут выглядеть следующим образом:

```
/certs/tls.crt
```

```
/certs/tls.key
```

11.3 Возможные ошибки при обновлении сертификата

При обновлении сертификатов могут возникнуть ошибки, которые представлены ниже.

11.3.1 nginx не стартует и выдает ошибку: «cannot load certificate»

Одна из причин:

- файл /certs/tls.crt нет внутри контейнера;
- файл называется иначе;

- volume смонтирован не туда;
- сертификат некорректен или не в расширении .PEM.

Для проверки необходимо ввести:

```
docker compose exec frontend-react sh -lc 'ls -l /certs && file /certs/tls.crt'
```

11.3.2 nginx не стартует и выдает ошибку: «cannot load certificate key»

Необходимо проверить наличие ключа:

```
docker compose exec frontend-react sh -lc 'ls -l /certs/tls.key'
```

Проверить ключ:

```
docker compose exec frontend-react sh -lc 'openssl pkey -in /certs/tls.key -noout'
```

Если команда спрашивает пароль, следовательно, ключ зашифрован.

11.3.3 nginx выдает ошибку: «key values mismatch»

Сертификат и ключ не подходят друг к другу.

Необходимо проверить пару:

```
openssl x509 -in tls.crt -pubkey -noout | openssl pkey -pubin -outform DER | sha256sum
```

```
openssl pkey -in tls.key -pubout -outform DER | sha256sum
```

Хэши должны быть одинаковыми.

11.3.4 Браузер показывает старый сертификат

Одна из причин:

- контейнер не был перезапущен после замены;
- сертификаты заменены не в том каталоге;
- используется другой compose-файл;
- запрос идет не на тот сервер или не на тот порт;
- перед frontend стоит балансировщик или reverse proxy, и TLS завершается на нем, а не в nginx.

Проверить сертификат напрямую:

```
openssl s_client -connect localhost:443 -servername localhost </dev/null 2>/dev/null | openssl x509 -noout -subject -issuer -dates
```

11.3.5 При mount /certs пропали сертификаты из образа

Так и должно быть. Docker volume перекрывает каталог внутри контейнера.

Если указать:

volumes:

- /opt/safeerp/frontend/certs:/certs:ro

то nginx будет видеть только файлы из /opt/safeerp/frontend/certs на хосте.

Поэтому в этом каталоге обязательно должны расположены по следующим адресам:

tls.crt

tls.key

12 Возможные ошибки и их устранения

12.1 Ошибки в ПК «SafeERP PS EM»

12.1.1 Не отвечает Агент 1С

При сканировании большого объема данных в разделе «Анализ платформы» агент 1С может быть недоступен.

Для корректной работы раздела «Анализ платформы» необходимо выполнить следующие действия:

- 1) Выключить контейнеры:
`docker-compose down`
- 2) Добавить следующую информацию в .env файл:
`AGENT_1C_READ_TIMEOUT=600`
- 3) Добавить строку в раздел "platform security" файла `docker-compose.yml` :
`AGENT_1C_READ_TIMEOUT: ${AGENT_1C_READ_TIMEOUT}`
- 4) Запустить контейнеры
`docker-compose up -d`
- 5) Запустить сканирование в разделе «Анализ платформы».

12.1.2 Не запускается контейнер «authentication»

Если в .env-файл не указан пользователь с ролью «ADMIN», то контейнер «authentication» не запустится.

Для того чтобы запустить контейнер «authentication» необходимо внести изменения в .env-файл. Подробную информацию см в разделе 4

Приложение 1. docker-compose.yml файл для установки ПК «SAFEERP EM» и подключению к внешней СУБД

```
x-services-common: &services-common
  restart: unless-stopped
  networks:
    - SAFEERP

services:
  mongodb:
    image: redos-mongo-6.0
    container_name: mongodb
    hostname: mongodb.SAFEERP-em
    volumes:
      - mongodb_data:/data/db
    healthcheck:
      test: ["CMD-SHELL", "mongo --quiet --eval 'db.adminCommand(\"ping\").ok'"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 10s
    <<: *services-common

  opensearch:
    container_name: opensearch
    hostname: opensearch.SAFEERP-em
    image: redos-opensearch-2.9.0
    ulimits:
      memlock:
        hard: -1
        soft: -1
      nofile:
        soft: 65536
        hard: 65536
    volumes:
      - opensearch_data:/var/lib/opensearch/data
    healthcheck:
      test: ["CMD-SHELL", "curl -s http://localhost:9200/_cluster/health"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 30s
    <<: *services-common

  graylog:
    container_name: graylog
    hostname: graylog.SAFEERP-em
```

```
image: redos-graylog-6.0.11
depends_on:
  opensearch:
    condition: "service_healthy"
  mongodb:
    condition: "service_healthy"
user: root
environment:
  GRAYLOG_HOME: "/usr/share/graylog-server"
  PATH:
"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/share/gr
aylog-server/bin"
  GRAYLOG_DISABLE_HOSTNAME_VERIFICATION: "true"
  GRAYLOG_HTTP_ENABLE_TLS: "true"
  GRAYLOG_HTTP_TLS_CERT_FILE: "/etc/graylog/certs/graylog-cert.pem"
  GRAYLOG_HTTP_TLS_KEY_FILE: "/etc/graylog/certs/graylog-key.pem"
  GRAYLOG_NODE_ID_FILE: "/usr/share/graylog-server/data/config/node-id"
  GRAYLOG_PASSWORD_SECRET: "pepperandsaltsmellsgoodright"
  GRAYLOG_ROOT_PASSWORD_SHA2:
8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918
  GRAYLOG_HTTP_BIND_ADDRESS: "0.0.0.0:9000"
  GRAYLOG_HTTP_EXTERNAL_URI: "https://localhost:9000/"
  GRAYLOG_HTTP_PUBLISH_URI: "https://graylog.SAFEERP-em:9000/"
  GRAYLOG_ELASTICSEARCH_HOSTS: "http://opensearch.SAFEERP-em:9200"
  GRAYLOG_MONGODB_URI: "mongodb://mongodb.SAFEERP-em:27017/graylog"
ports:
  - "9000:9000/tcp" # Server API
  - "12201:12201/tcp" # GELF TCP
volumes:
  - graylog_data:/var/lib/graylog/data/data
  - graylog_journal:/var/lib/graylog/data/journal
healthcheck:
  test: ["CMD-SHELL", "curl -sk https://graylog.SAFEERP-
em:9000/api/system"]
  interval: 30s
  timeout: 10s
  retries: 5
  start_period: 60s
<<: *services-common

kafka:
  container_name: kafka
  image: kafka
  environment:
    KAFKA_CFG_NODE_ID: 0
    KAFKA_CFG_PROCESS_ROLES: controller,broker
    KAFKA_CFG_LISTENERS:
PLAINTEXT://:9092,CONTROLLER://:9093,EXTERNAL://0.0.0.0:9094
    KAFKA_CFG_ADVERTISED_LISTENERS:
PLAINTEXT://kafka:9092,EXTERNAL://localhost:9094
```

```
KAFKA_CFG_LISTENER_SECURITY_PROTOCOL_MAP:
PLAINTEXT:PLAINTEXT,CONTROLLER:PLAINTEXT,EXTERNAL:PLAINTEXT
KAFKA_CFG_CONTROLLER_QUORUM_VOTERS: '0@kafka:9093'
KAFKA_CFG_CONTROLLER_LISTENER_NAMES: CONTROLLER
healthcheck:
  test: [ "CMD", "kafka-topics.sh", "--bootstrap-server",
"localhost:9092", "--list" ]
  start_period: 15s
  interval: 5s
  timeout: 10s
  retries: 10
depends_on:
  graylog:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "kafka"
<<: *services-common

target-manager:
  container_name: target-manager
  hostname: target-manager.SAFEERP-em
  image: target-manager
  environment:
    SERVICE: target-manager
    SERVER_PORT: 8451
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
  ports:
    - "8451:8451"
    - "5011:5011"
  volumes:
    - /opt/1cv8:/opt/1cv8
    - ./target_certs:/app/certs
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "target-manager"
  <<: *services-common

code-security:
  container_name: code-security
  hostname: code-security.SAFEERP-em
```

```
image: code-security
volumes:
  - /opt/1cv8:/opt/1cv8
  - ./target_certs:/app/certs
ports:
  - "8082:8082"
  - "8445:8445"
  - "5006:5006"
environment:
  JVM_OPTS: "-Xms4g -Xmx16g"
  SERVICE: code-security
  SERVER_PORT: 8445
  DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
  DATABASE_USERNAME: Safeerp_sa
  DATABASE_PASSWORD: password
depends_on:
  graylog:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "code-security"
<<: *services-common

platform-security:
  container_name: platform-security
  hostname: platform-security.SAFEERP-em
  image: platform-security
  environment:
    SERVICE: platform-security
    SERVER_PORT: 8447
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
  ports:
    - "8447:8447"
    - "5008:5008"
  volumes:
    - h2data:/app/h2data
    - ./target_certs:/app/certs
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "platform-security"
  <<: *services-common
```

```
authentication:
  container_name: authentication
  hostname: authentication.SAFEERP-em
  image: authentication
  environment:
    SERVICE: authentication
    SERVER_PORT: 8449
    JWT_LIFE_TIME: 36000
    COOKIE-MAX_AGE: 36000
    SESSION_TIMEOUT: 36000
    PASSWORD_UPDATE: "true"
    KERBEROS_AUTHENTICATION_ENABLE: "false"
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
```

Устанавливаются пароли для каждой из базовых ролей

```
ADMIN_USERNAME: system
ADMIN_PASSWORD: System__12
OPERATOR_USERNAME: operator
OPERATOR_PASSWORD: Operator__12
SEC_ADMIN_USERNAME: securityAdmin
SEC_ADMIN_PASSWORD: SecurityAdmin__12
ports:
  - "8449:8449"
  - "5009:5009"
volumes:
  - h2data:/app/h2data
depends_on:
  graylog:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "authentication"
<<: *services-common
```

```
gateway:
  container_name: gateway
  hostname: gateway.SAFEERP-em
  image: gateway
  ports:
    - "8443:8443"
    - "5007:5007"
  environment:
    SERVER_PORT: 8443
    SPRING_PROFILE: docker,swagger
  depends_on:
    graylog:
```

```
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "gateway"
<<: *services-common

frontend-react:
  container_name: frontend-react
  hostname: frontend_react.SAFEERP-em
  image: frontend-react
  ports:
    - "80:80"
    - "443:443"
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "frontend-react"
  <<: *services-common

pentest-agent:
  image: "pentest-agent:latest"
  container_name: pentest-agent
  ports:
    - "$PENTEST_AGENT_PORT:8080"
    - '$PENTEST_AGENT_PROXY_PORT:10000'
  volumes:
    - type: bind
      source: $PENTEST_AGENT_SHARED_FOLDER/logs
      target: /home/SAFEERP_pentest/logs
    - type: bind
      source: $PENTEST_AGENT_SHARED_FOLDER/license
      target: /home/SAFEERP_pentest/license
  environment:
    # Хосты
    SERP_HOST_NAME: ${SERP_HOST_NAME}

    # Pentest Agent
    PENTEST_AGENT_PORT: ${PENTEST_AGENT_PORT}
    PENTEST_AGENT_PROXY_PORT: ${PENTEST_AGENT_PROXY_PORT}
    PENTEST_AGENT_DNS_TIMEOUT: ${PENTEST_AGENT_DNS_TIMEOUT}

    # Pentest Backend
    PENTEST_BACKEND_PROTO: ${PENTEST_BACKEND_PROTO}
    PENTEST_BACKEND_PORT: ${PENTEST_BACKEND_PORT}
```

```
# Аутентификация в SafeERP Authentication
USER_INIT_LIST: ${USER_INIT_LIST}

# Логирование
PENTEST_LOG_LEVEL: ${PENTEST_LOG_LEVEL}
PENTEST_AGENT_LOGS_LIFETIME: ${PENTEST_AGENT_LOGS_LIFETIME}
restart: unless-stopped
networks:
  - SAFEERP
depends_on:
  graylog:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "pentest-agent"

pentest-backend:
  image: "pentest-backend:latest"
  container_name: pentest-backend
  environment:
    # Хосты
    SERP_HOST_NAME: ${SERP_HOST_NAME}

    # Pentest Agent
    PENTEST_AGENT_PORT: ${PENTEST_AGENT_PORT}

    # Pentest Backend
    PENTEST_BACKEND_PROTO: ${PENTEST_BACKEND_PROTO}
    PENTEST_BACKEND_PORT: ${PENTEST_BACKEND_PORT}

    # БДУ
    INFO_DB_PASSWORD: ${INFO_DB_PASSWORD}

    # Аутентификация в SafeERP Authentication
    PENTEST_IN_SAFE_ERP_CONTOUR: ${PENTEST_IN_SAFE_ERP_CONTOUR}
    USER_INIT_LIST: ${USER_INIT_LIST}

    # Логирование
    DJANGO_LOG_LEVEL: ${DJANGO_LOG_LEVEL}
    PENTEST_LOG_LEVEL: ${PENTEST_LOG_LEVEL}
    PENTEST_INFO_LOG_FILE_SIZE_IN_MB: ${PENTEST_INFO_LOG_FILE_SIZE_IN_MB}
    PENTEST_ERROR_LOG_FILE_SIZE_IN_MB: ${PENTEST_ERROR_LOG_FILE_SIZE_IN_MB}
  networks:
    - SAFEERP
  restart: unless-stopped
  depends_on:
    pentest-webgui-db:
```

```

    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "pentest-backend"
pentest-info-db:
  image: "pentest-info-db:latest"
  container_name: pentest-info-db
  networks:
    - SAFEERP
  restart: unless-stopped
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "pentest-info-db"
pentest-webgui-db:
  image: "pentest-webgui-db:latest"
  container_name: pentest-webgui-db
  volumes:
    - type: volume
      source: pentest-webgui-db-data
      target: /var/lib/postgresql/data
  environment:
    - POSTGRES_PASSWORD=init_pass
  healthcheck:
    test: ["CMD-SHELL", "sh -c 'pg_isready -U postgres -d $${POSTGRES_PASSWORD}'"]
    interval: 10s
    timeout: 3s
    retries: 3
  shm_size: '256mb'
  command: -c 'max_connections=250'
  restart: unless-stopped
  networks:
    - SAFEERP
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "pentest-webgui-db"

```

```
infodb-c2server:
  image: infodb-c2server:latest
  container_name: infodb-c2server
  hostname: infodb-c2server.safeerp-em
  environment:
    <<: *env-common
    C2SERVER_INFODB__USER: ${C2SERVER_INFODB__USER}
    C2SERVER_INFODB__PASSWORD: ${C2SERVER_INFODB__PASSWORD}
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: 'tcp://localhost:12201'
      tag: 'infodb-c2server'
  <<: *services-common

web-security-service:
  container_name: web-security-service
  hostname: web-security-service.SAFEERP-em
  image: web-security-service
  environment:
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
    - WSC_CONNECTOR__COMMAND_RECEIVER__REST__API_HOST="0.0.0.0"
    - WSC_CONNECTOR__COMMAND_RECEIVER__REST__API_PORT=8080
    -
WSC_CONNECTOR__MESSAGE_SENDER__KAFKA__BOOTSTRAP_SERVERS="kafka:9092"
  - WSC_CONNECTOR__MESSAGE_SENDER__KAFKA__TOPIC="wsc-data"
  - WSC_PROXY__SERVER="http://proxy-auth:8899"
  - WSC_VERBOSE=1
  - WSC_INFODB__HOST="pentest-info-db"
  - WSC_INFODB__PORT=5432
  - WSC_CVES_FROM_CPES=true
  ports:
    - "8280:8080"
  volumes:
    - wsc_data:/app/.storage
  deploy:
    resources:
      limits:
        memory: "5000M"
        cpus: "1.0"
  depends_on:
    kafka:
      condition: service_healthy
  graylog:
```

```
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "web-security-service"
<<: *services-common

dast:
  container_name: dast
  hostname: dast.SAFEERP-em
  image: dast
  ports:
    - "8453:8453"
    - "5010:5010"
    - "7090:7090"
  environment:
    SERVICE: dast
    SERVER_PORT: 8453
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
  depends_on:
    kafka:
      condition: service_healthy
    graylog:
      condition: service_healthy
    postgres:
      condition: service_started
    web-security-service:
      condition: service_started
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "dast"
  <<: *services-common

proxy-auth:
  container_name: proxy-auth
  hostname: proxy-auth.SAFEERP-em
  image: proxy-auth
  ports:
    - "8899:8899"
  environment:
    - PYTHONPATH="."
  volumes:
    - ./ca:/app/ca
    - ./target_certs:/app/certs
  depends_on:
```

```
kafka:
  condition: service_healthy
graylog:
  condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "proxy-auth"
<<: *services-common
```

```
volumes:
  mongodb_data:
    driver: local
  opensearch_data:
    driver: local
  graylog_data:
    driver: local
  graylog_journal:
    driver: local
  pgdata:
    driver: local
  h2data:
    driver: local
  nginx:
    driver: local
  certs:
    driver: local
  private:
    driver: local
  postgres_data:
    driver: local
  pentest-webgui-db-data:
    driver: local
  wsc_data:
    driver: local
```

```
networks:
  SAFEERP:
    name: SAFEERP
    driver: bridge
  ipam:
    config:
      - subnet: 172.20.0.0/16
        gateway: 172.20.0.1
```

Приложение 2. docker-compose.yml файл для установки ПК SAFEERP CS EM и подключению к внешней СУБД

```
x-services-common: &services-common
  restart: unless-stopped
  networks:
    - SAFEERP

services:
  mongodb:
    image: redos-mongo-6.0
    container_name: mongodb
    hostname: mongodb.SAFEERP-em
    volumes:
      - mongodb_data:/data/db
    healthcheck:
      test: ["CMD-SHELL", "mongo --quiet --eval 'db.adminCommand(\"ping\").ok'"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 10s
    <<: *services-common

  opensearch:
    container_name: opensearch
    hostname: opensearch.SAFEERP-em
    image: redos-opensearch-2.9.0
    ulimits:
      memlock:
        hard: -1
        soft: -1
      nofile:
        soft: 65536
        hard: 65536
    volumes:
      - opensearch_data:/var/lib/opensearch/data
    healthcheck:
      test: ["CMD-SHELL", "curl -s http://localhost:9200/_cluster/health"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 30s
    <<: *services-common

  graylog:
    container_name: graylog
    hostname: graylog.SAFEERP-em
    image: redos-graylog-6.0.11
```

```
depends_on:
  opensearch:
    condition: "service_healthy"
  mongodb:
    condition: "service_healthy"
user: root
environment:
  GRAYLOG_HOME: "/usr/share/graylog-server"
  PATH:
"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/share/gr
aylog-server/bin"
  GRAYLOG_DISABLE_HOSTNAME_VERIFICATION: "true"
  GRAYLOG_HTTP_ENABLE_TLS: "true"
  GRAYLOG_HTTP_TLS_CERT_FILE: "/etc/graylog/certs/graylog-cert.pem"
  GRAYLOG_HTTP_TLS_KEY_FILE: "/etc/graylog/certs/graylog-key.pem"
  GRAYLOG_NODE_ID_FILE: "/usr/share/graylog-server/data/config/node-id"
  GRAYLOG_PASSWORD_SECRET: "pepperandsaltsmellsgoodright"
  GRAYLOG_ROOT_PASSWORD_SHA2:
8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918
  GRAYLOG_HTTP_BIND_ADDRESS: "0.0.0.0:9000"
  GRAYLOG_HTTP_EXTERNAL_URI: "https://localhost:9000/"
  GRAYLOG_HTTP_PUBLISH_URI: "https://graylog.SAFEERP-em:9000/"
  GRAYLOG_ELASTICSEARCH_HOSTS: "http://opensearch.SAFEERP-em:9200"
  GRAYLOG_MONGODB_URI: "mongodb://mongodb.SAFEERP-em:27017/graylog"
ports:
  - "9000:9000/tcp" # Server API
  - "12201:12201/tcp" # GELF TCP
volumes:
  - graylog_data:/var/lib/graylog/data/data
  - graylog_journal:/var/lib/graylog/data/journal
healthcheck:
  test: ["CMD-SHELL", "curl -sk https://graylog.SAFEERP-
em:9000/api/system"]
  interval: 30s
  timeout: 10s
  retries: 5
  start_period: 60s
<<: *services-common

kafka:
  container_name: kafka
  image: kafka
  environment:
    KAFKA_CFG_NODE_ID: 0
    KAFKA_CFG_PROCESS_ROLES: controller,broker
    KAFKA_CFG_LISTENERS:
PLAINTEXT://:9092,CONTROLLER://:9093,EXTERNAL://0.0.0.0:9094
    KAFKA_CFG_ADVERTISED_LISTENERS:
PLAINTEXT://kafka:9092,EXTERNAL://localhost:9094
    KAFKA_CFG_LISTENER_SECURITY_PROTOCOL_MAP:
```

```
PLAINTEXT:PLAINTEXT,CONTROLLER:PLAINTEXT,EXTERNAL:PLAINTEXT
  KAFKA_CFG_CONTROLLER_QUORUM_VOTERS: '0@kafka:9093'
  KAFKA_CFG_CONTROLLER_LISTENER_NAMES: CONTROLLER
  healthcheck:
    test: [ "CMD", "kafka-topics.sh", "--bootstrap-server",
"localhost:9092", "--list" ]
    start_period: 15s
    interval: 5s
    timeout: 10s
    retries: 10
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "kafka"
<<: *services-common

target-manager:
  container_name: target-manager
  hostname: target-manager.SAFEERP-em
  image: target-manager
  environment:
    SERVICE: target-manager
    SERVER_PORT: 8451
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
  ports:
    - "8451:8451"
    - "5011:5011"
  volumes:
    - /opt/1cv8:/opt/1cv8
    - ./target_certs:/app/certs
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "target-manager"
<<: *services-common

code-security:
  container_name: code-security
  hostname: code-security.SAFEERP-em
  image: code-security
```

```
volumes:
  - /opt/1cv8:/opt/1cv8
  - ./target_certs:/app/certs
ports:
  - "8082:8082"
  - "8445:8445"
  - "5006:5006"
environment:
  JVM_OPTS: "-Xms4g -Xmx16g"
  SERVICE: code-security
  SERVER_PORT: 8445
  DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
  DATABASE_USERNAME: Safeerp_sa
  DATABASE_PASSWORD: password
depends_on:
  graylog:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "code-security"
<<: *services-common
```

```
authentication:
  container_name: authentication
  hostname: authentication.SAFEERP-em
  image: authentication
  environment:
    SERVICE: authentication
    SERVER_PORT: 8449
    JWT_LIFE_TIME: 36000
    COOKIE-MAX_AGE: 36000
    SESSION_TIMEOUT: 36000
    PASSWORD_UPDATE: "true"
    KERBEROS_AUTHENTICATION_ENABLE: "false"
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
```

```
# Устанавливаются пароли для каждой из базовых ролей
ADMIN_USERNAME: system
ADMIN_PASSWORD: System__12
OPERATOR_USERNAME: operator
OPERATOR_PASSWORD: Operator__12
SEC_ADMIN_USERNAME: securityAdmin
SEC_ADMIN_PASSWORD: SecurityAdmin__12
ports:
  - "8449:8449"
  - "5009:5009"
volumes:
```

```
- h2data:/app/h2data
depends_on:
  graylog:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "authentication"
<<: *services-common
```

```
gateway:
  container_name: gateway
  hostname: gateway.SAFEERP-em
  image: gateway
  ports:
    - "8443:8443"
    - "5007:5007"
  environment:
    SERVER_PORT: 8443
    SPRING_PROFILE: docker,swagger
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "gateway"
  <<: *services-common
```

```
frontend-react:
  container_name: frontend-react
  hostname: frontend_react.SAFEERP-em
  image: frontend-react
  ports:
    - "80:80"
    - "443:443"
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "frontend-react"
  <<: *services-common
```

```
volumes:
  mongodb_data:
```

```
    driver: local
opensearch_data:
  driver: local
graylog_data:
  driver: local
graylog_journal:
  driver: local
pgdata:
  driver: local
h2data:
  driver: local
nginx:
  driver: local
certs:
  driver: local
private:
  driver: local
postgres_data:
  driver: local
pentest-webgui-db-data:
  driver: local
wsc_data:
  driver: local

networks:
  SAFEERP:
    name: SAFEERP
    driver: bridge
    ipam:
      config:
        - subnet: 172.20.0.0/16
          gateway: 172.20.0.1
```

Приложение 3. docker-compose.yml файл для установки ПК SAFEERP PS EM и подключению к внешней СУБД

```
x-services-common: &services-common
  restart: unless-stopped
  networks:
    - SAFEERP

services:
  mongodb:
    image: redos-mongo-6.0
    container_name: mongodb
    hostname: mongodb.SAFEERP-em
    volumes:
      - mongodb_data:/data/db
    healthcheck:
      test: ["CMD-SHELL", "mongo --quiet --eval 'db.adminCommand(\"ping\").ok'"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 10s
    <<: *services-common

  opensearch:
    container_name: opensearch
    hostname: opensearch.SAFEERP-em
    image: redos-opensearch-2.9.0
    ulimits:
      memlock:
        hard: -1
        soft: -1
      nofile:
        soft: 65536
        hard: 65536
    volumes:
      - opensearch_data:/var/lib/opensearch/data
    healthcheck:
      test: ["CMD-SHELL", "curl -s http://localhost:9200/_cluster/health"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 30s
    <<: *services-common

  graylog:
    container_name: graylog
    hostname: graylog.SAFEERP-em
    image: redos-graylog-6.0.11
    depends_on:
```

```
opensearch:
  condition: "service_healthy"
mongodb:
  condition: "service_healthy"
user: root
environment:
  GRAYLOG_HOME: "/usr/share/graylog-server"
  PATH:
"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/share/gr
aylog-server/bin"
  GRAYLOG_DISABLE_HOSTNAME_VERIFICATION: "true"
  GRAYLOG_HTTP_ENABLE_TLS: "true"
  GRAYLOG_HTTP_TLS_CERT_FILE: "/etc/graylog/certs/graylog-cert.pem"
  GRAYLOG_HTTP_TLS_KEY_FILE: "/etc/graylog/certs/graylog-key.pem"
  GRAYLOG_NODE_ID_FILE: "/usr/share/graylog-server/data/config/node-id"
  GRAYLOG_PASSWORD_SECRET: "pepperandsaltsmellsgoodright"
  GRAYLOG_ROOT_PASSWORD_SHA2:
8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918
  GRAYLOG_HTTP_BIND_ADDRESS: "0.0.0.0:9000"
  GRAYLOG_HTTP_EXTERNAL_URI: "https://localhost:9000/"
  GRAYLOG_HTTP_PUBLISH_URI: "https://graylog.SAFEERP-em:9000/"
  GRAYLOG_ELASTICSEARCH_HOSTS: "http://opensearch.SAFEERP-em:9200"
  GRAYLOG_MONGODB_URI: "mongodb://mongodb.SAFEERP-em:27017/graylog"
ports:
  - "9000:9000/tcp" # Server API
  - "12201:12201/tcp" # GELF TCP
volumes:
  - graylog_data:/var/lib/graylog/data/data
  - graylog_journal:/var/lib/graylog/data/journal
healthcheck:
  test: ["CMD-SHELL", "curl -sk https://graylog.SAFEERP-
em:9000/api/system"]
  interval: 30s
  timeout: 10s
  retries: 5
  start_period: 60s
<<: *services-common

kafka:
  container_name: kafka
  image: kafka
  environment:
    KAFKA_CFG_NODE_ID: 0
    KAFKA_CFG_PROCESS_ROLES: controller,broker
    KAFKA_CFG_LISTENERS:
PLAINTEXT://:9092,CONTROLLER://:9093,EXTERNAL://0.0.0.0:9094
    KAFKA_CFG_ADVERTISED_LISTENERS:
PLAINTEXT://kafka:9092,EXTERNAL://localhost:9094
    KAFKA_CFG_LISTENER_SECURITY_PROTOCOL_MAP:
PLAINTEXT:PLAINTEXT,CONTROLLER:PLAINTEXT,EXTERNAL:PLAINTEXT
```

```
KAFKA_CFG_CONTROLLER_QUORUM_VOTERS: '0@kafka:9093'
KAFKA_CFG_CONTROLLER_LISTENER_NAMES: CONTROLLER
healthcheck:
  test: [ "CMD", "kafka-topics.sh", "--bootstrap-server",
"localhost:9092", "--list" ]
  start_period: 15s
  interval: 5s
  timeout: 10s
  retries: 10
depends_on:
  graylog:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "kafka"
<<: *services-common

target-manager:
  container_name: target-manager
  hostname: target-manager.SAFEERP-em
  image: target-manager
  environment:
    SERVICE: target-manager
    SERVER_PORT: 8451
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
  ports:
    - "8451:8451"
    - "5011:5011"
  volumes:
    - /opt/1cv8:/opt/1cv8
    - ./target_certs:/app/certs
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "target-manager"
  <<: *services-common

platform-security:
  container_name: platform-security
  hostname: platform-security.SAFEERP-em
  image: platform-security
```

```
environment:
  SERVICE: platform-security
  SERVER_PORT: 8447
  DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
  DATABASE_USERNAME: Safeerp_sa
  DATABASE_PASSWORD: password
ports:
  - "8447:8447"
  - "5008:5008"
volumes:
  - h2data:/app/h2data
  - ./target_certs:/app/certs
depends_on:
  graylog:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "platform-security"
<<: *services-common
```

```
authentication:
  container_name: authentication
  hostname: authentication.SAFEERP-em
  image: authentication
  environment:
    SERVICE: authentication
    SERVER_PORT: 8449
    JWT_LIFE_TIME: 36000
    COOKIE-MAX_AGE: 36000
    SESSION_TIMEOUT: 36000
    PASSWORD_UPDATE: "true"
    KERBEROS_AUTHENTICATION_ENABLE: "false"
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
```

```
# Устанавливаются пароли для каждой из базовых ролей
  ADMIN_USERNAME: system
  ADMIN_PASSWORD: System__12
  OPERATOR_USERNAME: operator
  OPERATOR_PASSWORD: Operator__12
  SEC_ADMIN_USERNAME: securityAdmin
  SEC_ADMIN_PASSWORD: SecurityAdmin__12
ports:
  - "8449:8449"
  - "5009:5009"
volumes:
  - h2data:/app/h2data
depends_on:
```

```
graylog:
  condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "authentication"
<<: *services-common
```

```
gateway:
  container_name: gateway
  hostname: gateway.SAFEERP-em
  image: gateway
  ports:
    - "8443:8443"
    - "5007:5007"
  environment:
    SERVER_PORT: 8443
    SPRING_PROFILE: docker,swagger
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "gateway"
  <<: *services-common
```

```
frontend-react:
  container_name: frontend-react
  hostname: frontend_react.SAFEERP-em
  image: frontend-react
  ports:
    - "80:80"
    - "443:443"
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "frontend-react"
  <<: *services-common
```

```
pentest-agent:
  image: "pentest-agent:latest"
  container_name: pentest-agent
```

```
ports:
  - "$PENTEST_AGENT_PORT:8080"
  - '$PENTEST_AGENT_PROXY_PORT:10000'
volumes:
  - type: bind
    source: $PENTEST_AGENT_SHARED_FOLDER/logs
    target: /home/SAFEERP_pentest/logs
  - type: bind
    source: $PENTEST_AGENT_SHARED_FOLDER/license
    target: /home/SAFEERP_pentest/license
environment:
  # Хосты
  SERP_HOST_NAME: ${SERP_HOST_NAME}

  # Pentest Agent
  PENTEST_AGENT_PORT: ${PENTEST_AGENT_PORT}
  PENTEST_AGENT_PROXY_PORT: ${PENTEST_AGENT_PROXY_PORT}
  PENTEST_AGENT_DNS_TIMEOUT: ${PENTEST_AGENT_DNS_TIMEOUT}

  # Pentest Backend
  PENTEST_BACKEND_PROTO: ${PENTEST_BACKEND_PROTO}
  PENTEST_BACKEND_PORT: ${PENTEST_BACKEND_PORT}

  # Аутентификация в SafeERP Authentication
  USER_INIT_LIST: ${USER_INIT_LIST}

  # Логирование
  PENTEST_LOG_LEVEL: ${PENTEST_LOG_LEVEL}
  PENTEST_AGENT_LOGS_LIFETIME: ${PENTEST_AGENT_LOGS_LIFETIME}
restart: unless-stopped
networks:
  - SAFEERP
depends_on:
  graylog:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "pentest-agent"

pentest-backend:
  image: "pentest-backend:latest"
  container_name: pentest-backend
  environment:
    # Хосты
    SERP_HOST_NAME: ${SERP_HOST_NAME}

    # Pentest Agent
    PENTEST_AGENT_PORT: ${PENTEST_AGENT_PORT}
```

```
# Pentest Backend
PENTEST_BACKEND_PROTO: ${PENTEST_BACKEND_PROTO}
PENTEST_BACKEND_PORT: ${PENTEST_BACKEND_PORT}

# БДУ
INFO_DB_PASSWORD: ${INFO_DB_PASSWORD}

# Аутентификация в SafeERP Authentication
PENTEST_IN_SAFE_ERP_CONTOUR: ${PENTEST_IN_SAFE_ERP_CONTOUR}
USER_INIT_LIST: ${USER_INIT_LIST}

# Логирование
DJANGO_LOG_LEVEL: ${DJANGO_LOG_LEVEL}
PENTEST_LOG_LEVEL: ${PENTEST_LOG_LEVEL}
PENTEST_INFO_LOG_FILE_SIZE_IN_MB: ${PENTEST_INFO_LOG_FILE_SIZE_IN_MB}
PENTEST_ERROR_LOG_FILE_SIZE_IN_MB: ${PENTEST_ERROR_LOG_FILE_SIZE_IN_MB}
networks:
  - SAFEERP
restart: unless-stopped
depends_on:
  pentest-webgui-db:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "pentest-backend"

pentest-info-db:
  image: "pentest-info-db:latest"
  container_name: pentest-info-db
  networks:
    - SAFEERP
  restart: unless-stopped
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "pentest-info-db"

pentest-webgui-db:
  image: "pentest-webgui-db:latest"
  container_name: pentest-webgui-db
  volumes:
    - type: volume
      source: pentest-webgui-db-data
```

```
    target: /var/lib/postgresql/data
environment:
  - POSTGRES_PASSWORD=init_pass
healthcheck:
  test: ["CMD-SHELL", "sh -c 'pg_isready -U postgres -d
$$POSTGRES_PASSWORD'"]
  interval: 10s
  timeout: 3s
  retries: 3
shm_size: '256mb'
command: -c 'max_connections=250'
restart: unless-stopped
networks:
  - SAFEERP
depends_on:
  graylog:
    condition: service_healthy
logging:
  driver: gelf
  options:
    gelf-address: "tcp://localhost:12201"
    tag: "pentest-webgui-db"

infodb-c2server:
  image: infodb-c2server:latest
  container_name: infodb-c2server
  hostname: infodb-c2server.safeerp-em
  environment:
    <<: *env-common
    C2SERVER_INFODB__USER: ${C2SERVER_INFODB_USER}
    C2SERVER_INFODB__PASSWORD: ${C2SERVER_INFODB_PASSWORD}
  depends_on:
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: 'tcp://localhost:12201'
      tag: 'infodb-c2server'
  <<: *services-common

web-security-service:
  container_name: web-security-service
  hostname: web-security-service.SAFEERP-em
  image: web-security-service
  environment:
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
    - WSC_CONNECTOR__COMMAND_RECEIVER__REST__API_HOST="0.0.0.0"
```

```
- WSC_CONNECTOR__COMMAND_RECEIVER__REST__API_PORT=8080
-
WSC_CONNECTOR__MESSAGE_SENDER__KAFKA__BOOTSTRAP_SERVERS="kafka:9092"
- WSC_CONNECTOR__MESSAGE_SENDER__KAFKA__TOPIC="wsc-data"
- WSC_PROXY__SERVER="http://proxy-auth:8899"
- WSC_VERBOSE=1
- WSC_INFODB__HOST="pentest-info-db"
- WSC_INFODB__PORT=5432
- WSC_CVES_FROM_CPES=true
ports:
- "8280:8080"
volumes:
- wsc_data:/app/.storage
deploy:
  resources:
    limits:
      memory: "5000M"
      cpus: "1.0"
  depends_on:
    kafka:
      condition: service_healthy
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "web-security-service"
<<: *services-common

dast:
  container_name: dast
  hostname: dast.SAFEERP-em
  image: dast
  ports:
    - "8453:8453"
    - "5010:5010"
    - "7090:7090"
  environment:
    SERVICE: dast
    SERVER_PORT: 8453
    DATABASE_URL: jdbc:postgresql://<IP-адрес>:<Порт>/SAFEERP_db
    DATABASE_USERNAME: Safeerp_sa
    DATABASE_PASSWORD: password
  depends_on:
    kafka:
      condition: service_healthy
    graylog:
      condition: service_healthy
    postgres:
```

```
    condition: service_started
  web-security-service:
    condition: service_started
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "dast"
  <<: *services-common
```

```
proxy-auth:
  container_name: proxy-auth
  hostname: proxy-auth.SAFEERP-em
  image: proxy-auth
  ports:
    - "8899:8899"
  environment:
    - PYTHONPATH="."
  volumes:
    - ./ca:/app/ca
    - ./target_certs:/app/certs
  depends_on:
    kafka:
      condition: service_healthy
    graylog:
      condition: service_healthy
  logging:
    driver: gelf
    options:
      gelf-address: "tcp://localhost:12201"
      tag: "proxy-auth"
  <<: *services-common
```

```
volumes:
  mongodb_data:
    driver: local
  opensearch_data:
    driver: local
  graylog_data:
    driver: local
  graylog_journal:
    driver: local
  pgdata:
    driver: local
  h2data:
    driver: local
  nginx:
    driver: local
  certs:
    driver: local
```

```
private:
  driver: local
postgres_data:
  driver: local
pentest-webgui-db-data:
  driver: local
wsc_data:
  driver: local

networks:
  SAFEERP:
    name: SAFEERP
    driver: bridge
    ipam:
      config:
        - subnet: 172.20.0.0/16
          gateway: 172.20.0.1
```

Приложение 4. Генерация токена для источников типа GIT

В ПК SafeERP CS EM доступно подключение источника с помощью токена. Подробную информацию см. в документе «Руководство администратора ПК SafeERP CS EM».

Для подключения репозитория необходимо сгенерировать токен.

Примечания:

1. При генерации токена необходимо выбрать доступ на чтение репозитория.
2. Большинство сервисов показывают токен только один раз. Если закрыть или обновить страницу, то восстановить его будет невозможно, и придется генерировать новый токен.

Доступно создание токена в следующих git-хостингах:

- github;
- gitlab.

Генерация токена в git-репозитории Github

Для создания токена доступа необходимо выполнить следующие действия:

- 1) Выбрать «Fine-grained personal access token» → «Repository permissions: Contents (доступ: Read-only)».

Примечание. При настройке токена необходимо убедиться, что пользователю доступно чтение репозитория: «All repositories» или «Only select repositories».

- 2) Выбрать «Personal Access Token (classic)» → полномочие «repo (full control of private repositories)».

Генерация токена в git-репозитории Gitlab

Для создания токена доступа необходимо выполнить следующие действия:

- 1) Выбрать «Personal Access Token» → «read_repository».

Перечень сокращений

ИБ	–	Информационная безопасность
ПК	–	Программный комплекс
СУБД	–	Система управления базами данных