

ПРОТОКОЛ № 16791/2024

проведения совместных испытаний программного обеспечения «Система управления доступом Identity Management» версии 23.10 и операционной системы специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (очередное обновление 1.7)

г. Москва

09.02.2024

1 Предмет испытаний

1.1 В настоящем протоколе зафиксирован факт проведения в период с 30.01.2024 по 09.02.2024 совместных испытаний программного обеспечения «Система управления доступом Identity Management» версии 23.10 (далее – ПО), разработанного ООО «Руск Солюшенс», и операционной системы специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (очередное обновление 1.7) (далее – Astra Linux SE 1.7.0), включая Astra Linux SE 1.7.0 с установленным оперативным обновлением безопасности БЮЛЛЕТЕНЬ № 2023-1023SE17 (оперативное обновление 1.7.5) (далее – Astra Linux SE 1.7.5), разработанной ООО «РусБИТех-Астра».

2 Объект испытаний

2.1 Перечень компонентов, эксплуатировавшихся в ходе проведения данных испытаний, относящихся к ПО, представлен в Таблице 1.

Таблица 1 – Перечень компонентов, относящихся к ПО

Описание	Наименование	MD5	Источник
Файлы программных пакетов дистрибутивов ПО	env-config-uidm-fstec-astra-sso_1.0.0-406_all.deb	48ff22c2612e2cf109d566e926254a12	Репозиторий разработчика ПО RooX Solutions.
	model-dbsync_8.65.12-606_all.deb	ae717d816d734bf0694c7ac8dba705ce	
	audit-dbsync_8.65.12-606_all.deb	e65e4d4abf8db2ce83041837bb761ab1	
	federation-dbsync_8.65.12-606_all.deb	b8c98e79a8c2533849f55b6c9f83297f	
	token-storage-10_1.0.16-105_all.deb	13f562c5a9fc0d07942dd76e90f97f9d	
	roox-tomcat-addons-10_1.0-10_all.deb	ae6969d8525b414bc2b460cab49339dc	



roox-tomcat-oracle-jdbc-112_11.2.0-4_all.deb	12667f3d82d84ba119219bd12321d82d
roox-tomcat-postgresql-jdbc-93_9.3-4_all.deb	2ff7383b1d5bcfe77778d002c3fb309d
sso-scripts_12.1.0-43424_all.deb	930c9d5bd32c9e22b2a69e983ed57d4b
sso-tools_12.1.0-35_all.deb	7a812913535ca2cca4b90e7da8fe9c07
ua-detection-dictionaries-51degrees-freeware_2022.12-18_all.deb	077e082b331f4576bfce172c003e7e1
sso-server-ocb-tomcat7_0.1.231005-4876_all.deb	1523352fe9d0c62d6cd87c9f4ac6b244
api-gateway-arm-10_1.0.0-76_all.deb	cad49cfe514e9eef0ff214b7a9db98a8
audit-search-api-10_1.0.21-120_all.deb	d57fc29d76d6c9befc8fce9d8a1f8fbf
audit-writer-10_1.0.18~66-66_all.deb	2e33d399f69521de3dd33b199f61b2d8
certificate-service-10_1.0.6-32_all.deb	85340de1390a8e1f31370e565a220a1c
customer-webapi-10_1.0.38-179_all.deb	422980264e207c13a91d10ebd827e1b3
federation-webapi-21_2.1.32-10_all.deb	f03a2733d72d2a6f016090ab85d94b5a
oauth2-consumer-server-10_1.0.18-87_all.deb	cb4a52cc8ef9aa062501337656e62821
sync-service-wtl-10_1.0.4-38_all.deb	d4b01b3d8e9236476fc332cf6ac9a4ea
uidm-admin-panel_0.1.0-294_all.deb	ea8deb703f64eb2bb0a9360325a2e6aa
webapi-notifier-10_1.0.58-130_all.deb	612f5e5579d48313de676ab5e581e7e5



	webapi-server-otp-settings-10_1.0.25-100_all.deb	4c88ce0b76f3b289a9b9f81eb7f11bff	
	widgets-uidm-login_1.0-124_all.deb	7d6c43f2502029cc4d4fe5c58c3cc8d2	
	widgets-uidm-selfservices_1.0-139_all.deb	94a2b637f8d3fe7cbc64406d ddcbd7c8	
Архив с автотестами проверки МФ	rooxteam-autotests-astra-linux-0-0-2.tar	587cc053cb1827cf9e8f7641de1b7128	Репозиторий разработчика ПО RooX Solutions.
Инструкция по установке RooX UIDM	Инструкция по установке RooX UIDM.pdf	95c4ab4bfb5a5ab5d1a910c078d7a7b4	Сторона разработчика ПО

3 Ход испытаний

3.1 В ходе проведения настоящих испытаний были выполнены проверки корректности функционирования ПО в средах: Astra Linux SE 1.7.0, Astra Linux SE 1.7.5 в объеме, указанном в Приложении 1.

3.2 Перечень используемых репозиториях приведен в Приложении 2.

3.3 Информация об эксплуатируемых неофициальных репозиториях ПО для указанных сред отражена в Приложении 2.

3.4 По информации от разработчика ПО не поддерживает работу с активным режимом ЗПС. Испытания проводились при отключенном режиме ЗПС.

3.5 Проверка корректности функционирования ПО в условиях ненулевого уровня конфиденциальности механизма мандатного разграничения доступа (далее – МРД) указанных сред не проводилась по причине отсутствия поддержки ПО соответствующей функциональности ОС. Информация об отсутствии упомянутой поддержки была заявлена стороной разработчика ПО.

3.6 Тестирование на ядрах hardened не проводилось из-за неработоспособности docker-контейнеров на ядрах hardened, потому что в качестве пререквизита используется (см. Приложение 2) IMDB Tarantool в docker-исполнении.

3.7 Перед установкой ПО необходимо убедиться, что у сервера есть доменное имя, и что оно разрешает обращение к 127.0.0.1 согласно системным требованиям ПО (см. Приложение 2).

3.8 Для функционирования ПО необходимо не менее 16 Гб ОЗУ, что установлено в результате тестирования.



3.9 ПО может быть установлено на несколько серверов, но тестировалась полная установка на одном сервере.

3.10 В ряде команд установки ПО требуются логин и пароль для доступа к репозиториям RooX Solutions. Эти данные разработчиком передаются отдельно.

3.11 После установки пререквизита docker.io и добавления пользователя в группу docker необходимо перелогиниться, во избежание ошибок работы X11-сервера (см. Приложение 2).

3.12 Тестирование минимального базового функционала ПО проводились посредством автотестов, предоставленных разработчиком ПО.

3.13 В среде Astra Linux SE 1.7.0 выявлено не критичное нарушение целостности файлов операционной системы, реализующих и (или) поддерживающих работу функций безопасности.

3.14 Удаление ПО происходит не полностью, с предупреждениями о неполном удалении, и требуется ручное доудаление согласно Приложение 2.

4 Результаты испытаний

4.1 ПО корректно функционирует в средах: Astra Linux SE 1.7.0, Astra Linux SE 1.7.5.

5 Вывод

5.1 ПО и операционная система специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (очередное обновление 1.7) совместимы, принимая во внимание информацию, содержащуюся в разделах 3, 4 и Приложении 2.

6 Состав рабочей группы и подписи сторон

6.1 Данный протокол составлен участниками рабочей группы:

Проканюк Д. С. – начальник сектора отдела технологической совместимости департамента развития технологического сотрудничества ДВиС ООО «РусБИТех-Астра»;

Брянцева Л. С. – инженер отдела инфраструктуры рабочих мест Департамента внедрения и сопровождения ООО «АйСиЭл Астра Сервис».



Перечень проверок совместимости ПО и Astra Linux SE 1.7.0, Astra Linux SE 1.7.5

№ п/п	Наименование проверки	Результат проверки ПО и Astra Linux SE									
		1.7.0 с ядром ОС		1.7.5 с ядром ОС							
		5.4.0-54-generic	5.4.0-54-hardened	5.4.0-162-generic	5.4.0-162-hardened	5.10.190-1-generic	5.10.190-1-hardened	5.15.0-83-generic	5.15.0-83-hardened	5.15.0-83-lowlatency	6.1.50-1-generic
1.	Установка ПО	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Успешно
2.	Запуск, остановка выполнения ПО	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Успешно
3.	Эксплуатация минимальной базовой функциональности ПО	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Успешно
4.	Функционирование ПО в условиях низкого уровня целостности механизма МКЦ ОС	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Успешно
5.	Функционирование ПО в условиях ненулевого уровня конфиденциальности механизма МРД ОС	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась
6.	Отсутствие нарушений требований подраздела 17.3 «Руководство по КСЗ Ч. 1»	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Успешно
7.	Соответствие объектов ФС ОС дистрибутиву ОС при эксплуатации ПО	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Успешно
8.	Удаление ПО	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Успешно
9.	Функционирование ПО в условиях включённого механизма ЗПС ОС	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась	Не проводилась
10.	Отсутствие нарушений требований подраздела 17.2 «Руководство по КСЗ Ч. 1»	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Не проводилась	Успешно	Успешно



Инструкция по установке и удалению ПО в средах: Astra Linux SE 1.7.0, Astra Linux SE 1.7.5

1 Используемые репозитории:

в Astra Linux SE 1.7.0:

- deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.0/repository-base/ 1.7_x86-64 main contrib non-free
- deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-extended/ 1.7_x86-64 main contrib non-free
- сторонние репозитории:
deb [arch=all signed-by=/etc/apt/trusted.gpg.d/roox.asc]
<https://repo2.rooxcloud.com/customers/repository/customers-astra-deb/> astra main

в Astra Linux SE 1.7.5:

- deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-base/ 1.7_x86-64 main contrib non-free
- deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-extended/ 1.7_x86-64 main contrib non-free
- сторонние репозитории:
deb [arch=all signed-by=/etc/apt/trusted.gpg.d/roox.asc]
<https://repo2.rooxcloud.com/customers/repository/customers-astra-deb/> astra main

2 Установка ПО:

2.1 выполнить действия:

1.1.1 До начала тестирования:

Смените имя сервера на `fstec-astra.demo.rooxteam.com`:

```
hostnamectl set-hostname fstec-astra.demo.rooxteam.com
```

Добавьте к записи `127.0.0.1` в `/etc/hosts`:

```
sed -i 's/127.0.0.1(.*)/127.0.0.1 localhost localhost.localdomain localhost4  
localhost4.localdomain4 fstec-astra.demo.rooxteam.com/g' /etc/hosts
```

перезагрузитесь.

1.1.2 Установите необходимые зависимости для среды AstraLinux SE 1.7.0

```
sudo apt install -y curl
```

Необходимо заменить {LOGIN} и {PASSWORD} на переданные логин и пароль, либо



объявить соответствующие переменные перед выполнением команд:

```
export LOGIN=username
```

```
export PASSWORD=password
```

1.1.3. Установка Axiom Standard JDK 8

Загрузка deb-пакета с официального сайта вендора и последующая установка:

```
curl -O https://download.axiomjdk.ru/axiomjdk-pro/8u382+6/axiomjdk-jdk-pro8u382+6-  
linux-amd64.deb && apt install -y ./axiomjdk-jdk-pro8u382+6-linux-amd64.deb
```

1.1.4. Установка haproxy

Сгенерируйте сертификат для haproxy:

```
sudo mkdir -p /etc/ssl/{private,csr}
```

```
sudo openssl genrsa -out /etc/ssl/private/server.key 2048
```

```
sudo openssl req -new -key /etc/ssl/private/server.key -out /etc/ssl/csr/server.csr -nodes -  
subj "/C=US/ST=Denial/L=Springfield/O=Dis/CN=fstec-astra.demo.rooxteam.com"
```

```
sudo openssl x509 -req -days 365 -in /etc/ssl/csr/server.csr -signkey
```

```
/etc/ssl/private/server.key -out /etc/ssl/certs/server.crt
```

```
sudo bash -c 'cat /etc/ssl/certs/server.crt /etc/ssl/private/server.key >
```

```
/etc/ssl/certs/server.bundle.pem'
```

Добавьте сгенерированный самоподписанный сертификат в хранилище доверенных сертификатов Java:

```
echo 'changeit' | sudo keytool -importcert -v -keystore /usr/lib/jvm/axiomjdk-java8-pro-  
amd64/jre/lib/security/cacerts -file /etc/ssl/certs/server.crt -srcstorepass changeit -
```

```
noprompt -alias "fstec-astra"
```

Установите haproxy:

для среды AstraLinux SE 1.7.0 подключите репозиторий ALSE 1.7.5:

```
echo ""deb http://download.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-base
```

```
1.7_x86-64 main contrib non-free
```

```
deb http://download.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-update 1.7_x86-  
64 main contrib non-free
```

```
"" | sudo tee /etc/apt/sources.list.d/astra175.list
```

```
sudo apt update
```

```
sudo apt install -y haproxy
```

Проверка:смотрим active, enabled, отсутствие критичных ошибок

```
sudo systemctl status haproxy.service
```



для среды AstraLinux SE 1.7.0 удалите подключенный репозиторий ALSE 1.7.5:

```
sudo rm -rf /etc/apt/sources.list.d/astra175.list && sudo apt update
```

1.1.5. Установка nginx

Подключите расширенный репозиторий AstraLinux (extended):

для среды AstraLinux SE 1.7.0:

```
echo ""deb http://download.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-base  
1.7_x86-64 main contrib non-free
```

```
deb http://download.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-extended  
1.7_x86-64 main contrib non-free
```

```
"" | sudo tee /etc/apt/sources.list.d/extended.list
```

для среды AstraLinux SE 1.7.5 :

```
echo ""deb http://download.astralinux.ru/astra/frozen/1.7_x86-64/1.7.5/repository-  
extended 1.7_x86-64 main contrib non-free"" | sudo tee  
/etc/apt/sources.list.d/extended.list
```

Обновите список пакетов:

```
sudo apt update
```

Установите nginx из расширенного репозитория AstraLinux (extended):

```
sudo apt install -y nginx
```

Проверка: смотрим active, enabled, отсутствие критичных ошибок

```
sudo systemctl status nginx.service
```

Удалите расширенный репозиторий AstraLinux (extended):

```
sudo rm -Rf /etc/apt/sources.list.d/extended.list
```

Обновите список пакетов:

```
sudo apt update
```

1.1.6. Установка СУБД Postgres 11 из base репозитория

```
sudo apt install -y postgresql
```

Проверка: смотрим active, enabled, отсутствие критичных ошибок

```
sudo systemctl status postgresql.service
```

Создайте скрипт для создания схем и пользователей в БД:

```
echo ""CREATE DATABASE \"RX_SSO\";  
CREATE USER \"RX_SSO\" WITH PASSWORD 'RX_SSO';  
GRANT ALL PRIVILEGES ON DATABASE \"RX_SSO\" TO \"RX_SSO\";  
CREATE DATABASE \"RX_AUDIT\";  
CREATE USER \"RX_AUDIT\" WITH PASSWORD 'RX_AUDIT';
```




```
GRANT ALL PRIVILEGES ON DATABASE \"RX_AUDIT\" TO \"RX_AUDIT\";
CREATE DATABASE \"RX_FEDERATION\";
CREATE USER \"RX_FEDERATION\" WITH PASSWORD 'RX_FEDERATION';
GRANT ALL PRIVILEGES ON DATABASE \"RX_FEDERATION\"
TO \"RX_FEDERATION\";\"\" > /tmp/create_db_users.sql
```

Сделайте пользователя postgres владельцем этого скрипта и выполните скрипт от имени пользователя postgres:

```
sudo chown postgres:postgres /tmp/create_db_users.sql && sudo -u postgres psql -f
/tmp/create_db_users.sql
```

Проверка: предупреждением "could not change directory to "/home/XXX": Отказано в доступе", пренебречь. Оно из за того, что запуск команд идет не из директории Postgresql.

Если сделать cd /usr/lib/postgresql/11/bin и выполнить команды, то этого предупреждения не будет

проверка БД:

```
sudo -u postgres psql -l
```

выведет все БД в кластере, убедиться, что БД RX_SSO,RX_AUDIT,RX_FEDERATION создались.

проверка пользователей:

```
sudo -u postgres psql -c "SELECT * FROM pg_catalog.pg_user;"
```

выведет список пользователей, убедиться что пользователи RX_SSO,RX_AUDIT,RX_FEDERATION создались.

Настройка параметров Postgresql выполняются в соответствии с потребностями.

Пример настроек для проведения тестирования:

1. соблюдение указаний п.17.2.4.4 "Руководство по КСЗ Ч. 1" ОС Astra Linux SE 1.7:

При использовании защищенного сервера СУБД из состава ОС в режиме мандатного управления доступом необходимо в конфигурационном файле кластера

postgresql.conf для параметра enable_bitmapscan установить значение off и для параметра ac_ignore_socket_maclabel установить значение false

```
sudo sed -i 's/#enable_bitmapscan = on/enable_bitmapscan = off/'
```

```
/etc/postgresql/11/main/postgresql.conf
```

```
sed1=$(cat /etc/postgresql/11/main/postgresql.conf | grep "ac_ignore_socket_maclabel")
```

```
&& sed2=$(echo $sed1 | sed 's/true/false/') && sudo sed -i "s/$sed1/$sed2/"
```

```
/etc/postgresql/11/main/postgresql.conf
```



2. общие рекомендации от вендора PostgreSQL:

```
sudo sed -i "s/^#listen_addresses.*listen_addresses = */g"
/etc/postgresql/11/main/postgresql.conf
```

```
sudo sed -i "s/^max_connections.*max_connections = 200 /g"
/etc/postgresql/11/main/postgresql.conf
```

```
sudo sed -i "s/^shared_buffers.*shared_buffers = 256MB/g"
/etc/postgresql/11/main/postgresql.conf
```

3. параметры в pg_hba.conf с учетом подключений пользователей

RX_SSO,RX_AUDIT,RX_FEDERATION с паролями созданных только в postgresql, и не существующих среди пользователей ОС:

заменить в строке 90: "local all all peer" метод аутентификации peer (получает данные пользователя из ОС) на password (по паролю).

```
sudo sed -i '90s/peer/password/' /etc/postgresql/11/main/pg_hba.conf
```

проверка (копировать команду со всеми пробелами)

```
sudo cat /etc/postgresql/11/main/pg_hba.conf | grep -n 'local all all'
```

сделать то же самое для host подключений, в строках 93 и 95:

```
sudo sed -i '93,95s/md5/password/' /etc/postgresql/11/main/pg_hba.conf
```

проверка:

```
sudo cat /etc/postgresql/11/main/pg_hba.conf | grep -n 'host all all'
```

Перезапустите postgresql и проверьте его статус:смотрим active, enabled, отсутствие критичных ошибок

```
sudo systemctl restart postgresql.service && sudo systemctl status postgresql.service
```

1.1.7. Установка IMDB Tarantool

В данной инструкции рассматривается установка IMDB Tarantool на сервер в виде Docker-образа.

Установите docker из базового репозитория AstraLinux (base):

```
apt install -y docker.io
```

Проверка установки - сервис установлен, запущен и включен автозапуск, отсутствуют критичные ошибки:

```
sudo systemctl status docker.service
```

Внести текущего пользователя (который выполняет установку) в группу docker

```
sudo usermod -aG docker $USER && exec su - $USER
```

и перелогиниться для предупреждения ошибок X11-сервер



Создайте директорию для работы tarantool:

```
sudo mkdir -p /data/{tarantool,etc/tarantool/instances.available}
```

Создайте инициализационный скрипт tarantool:

```
echo ""box.cfg {
listen = '0.0.0.0:3301';
io_collect_interval = nil;
readahead = 16320;
memtx_memory = 629145600;
memtx_max_tuple_size = 16 * 1024;
wal_mode = "none";
wal_max_size = 629145600;
checkpoint_interval = 60 * 60;
checkpoint_count = 6;
force_recovery = true;
log_level = 5;
too_long_threshold = 0.5;
}

local function bootstrap()
local space = box.schema.create_space('token')
space:create_index('primary', {unique = true, parts = {{field = 1, type =
'string'}}})
local metrics = require('metrics')
metrics.enable_default_metrics()
box.schema.user.create('test', {password = 'test_password'})
box.schema.user.grant('test', 'read,write,execute,create,alter,drop', 'universe')
end
box.once('users', bootstrap)
"" | sudo tee /data/etc/tarantool/instances.available/roox.lua
```

Запустите tarantool в docker-контейнере:

```
docker run -d --name tarantool -p 3301:3301 -m 768m -e RX_ENV=dev -e
TARANTOOL_USER_NAME=admin_user -e
TARANTOOL_USER_PASSWORD=admin_password -e
TARANTOOL_MEMTX_MEMORY=629145600 --restart=always -v
/data/tarantool:/var/lib/tarantool:rw -v
```



```
/data/etc/tarantool/instances.available/roox.lua:/etc/tarantool/instances.available/
roox.lua:rw tarantool/tarantool:latest tarantool /etc/tarantool/instances.available/roox.lua
```

1.1.8. Установка RabbitMQ

Установите RabbitMQ из базового репозитория AstraLinux (base):

```
apt install -y rabbitmq-server
```

Проверка установки - сервис установлен, запущен и включен автозапуск, отсутствуют критичные ошибки:

```
sudo systemctl status rabbitmq-server.service
```

Активируйте плагин для управления:

```
sudo /usr/sbin/rabbitmq-plugins enable rabbitmq_management
```

Создайте пользователя и очереди:

```
sudo /usr/sbin/rabbitmqctl add_user testuser testpassword && \
```

```
sudo /usr/sbin/rabbitmqctl set_user_tags testuser administrator && \
```

```
sudo /usr/sbin/rabbitmqctl set_permissions -p / testuser ".*" ".*" ".*" && \
```

```
sudo /usr/bin/rabbitmqadmin declare exchange --vhost=/ name=audit.raw type=fanout && \
```

```
sudo /usr/bin/rabbitmqadmin declare queue --vhost=/ name=audit.events durable=true
```

```
&& \
```

```
sudo /usr/bin/rabbitmqadmin declare queue --vhost=/ name=roox.autotests.audit
```

```
durable=true && \
```

```
sudo /usr/bin/rabbitmqadmin --vhost="/" declare binding source="audit.raw"
```

```
destination_type="queue" destination="audit.events" && \
```

```
sudo /usr/bin/rabbitmqadmin --vhost="/" declare binding source="audit.raw"
```

```
destination_type="queue" destination="roox.autotests.audit"
```

1.1.9. Установка tomcat

Создайте пользователя tomcat с домашней директорией /usr/share/tomcat:

```
sudo groupadd tomcat && sudo useradd --gid tomcat --home-dir /usr/share/tomcat --
```

```
shell /sbin/nologin tomcat && sudo mkdir /usr/share/tomcat/ && sudo chown tomcat:tomcat
```

```
/usr/share/tomcat/
```

Установите пакет tomcat 7 на сервер:

```
TOMCAT_VERSION=7.0.108 && CATALINA_HOME=/etc/tomcat/ && \
```

```
cd /tmp/ && curl -O
```

```
https://archive.apache.org/dist/tomcat/tomcat-7/v$TOMCAT_VERSION/bin/apache-tomcat-
```

```
$TOMCAT_VERSION.tar.gz && \
```

```
tar -xvf apache-tomcat-$TOMCAT_VERSION.tar.gz && rm apache-tomcat-
```



```

$TOMCAT_VERSION.tar.gz && \
sudo mv ./apache-tomcat-$TOMCAT_VERSION $CATALINA_HOME && \
sudo rm -r $CATALINA_HOME/webapps/* && \
sudo chown -Rf tomcat.tomcat $CATALINA_HOME && \
sudo ln -s /etc/tomcat/bin/ /usr/share/tomcat/ && \
sudo ln -s /etc/tomcat/conf/ /usr/share/tomcat/ && \
sudo ln -s /etc/tomcat/lib/ /usr/share/tomcat/

```

1.2. Установка компонентов UIDM на сервер

1.2.1. Подключение репозитория RooX Solutions

Добавьте GPG-ключ для аутентификации репозитория RooX Solutions:

будьте внимательны: между 2 и 3 ей строкой стоит пустая строка, она важна

```
echo ""-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
Version: GnuPG v2.0.22 (GNU/Linux)
```

```

mQENBGNto4EBCADNvLghrrUcfnrThHTFr/G9KfSQgKV+LeqCfS0dxdD9mFIPbIJQI
Z/wzMOafNI27pGvwchFIRpGDII4oamaKXhTQFIF105MPPWshWt0xDw6MIESPLSAd
R8NY9oVNnTWC5J8J2KXBKO/ZfMYzhkK1TgLUtVNFuObbN+3hFTnO9mmXUqWO1wj
fYwoax6aonL9G/S2VWoNhXqX5MXZggh/BPNbbzIm47NYIR9QIaUk8RU3zSHGRQpb
ddJOqi1pQVStvYJ8XOqfg2B4ZFu/7iGZQ0vAJ5atjE9TxfSb1sW8h/IYGFpZ1YM
vix7HkZxCdGR0sDG7DXxFvBmZXFdIUmEPHnABEBAAG0QIjvb1ggU29sdXRpb25z
IChodHRwczovL3VpZG0ucnUgaHR0cHM6Ly9yb294LnJ1KSA8c3VwcG9ydEByb294
LnJ1PokBPwQTAQIAKQUCY22jgQIbAwUJCWYBgAcLCQgHAwIBBhUIAgkKCwQWAgMB
Ah4BAheAAAoJEOTTC8VtWEPsF4sH/RicUwwWckVfbXqMGh0jmLvEAX9w8Fcaunq/
SblOpfWf5sDC5Uim3z37QwjADVnbBnoSttGWLeh1SKRB6F3iTuu5ldOOnxuP//fL
oC8eyKGH2mBOMABqECfxswgb6vZ/bbTryc2loR2UCmhTXyOauDOX295/x4EAoDwo
BEfDhgWtFzSECbcqc03eJQhP/fb4TaVZSITzUOW2pAFVrYE3Uj6Rcm9PHWdd8ZKF
Vfn3PiXC3rchIjx71iQRgMMPpEm0m9/ocBmpS7UGh8Semuf0EvV9HmVTKU8bAJY8
z7LTF4mCiTEAfWOTciYWQIxpWlpj7UL81DOtAq2FNxIUDE5aaC5AQ0EY22jgQEI
ANnasalAfQ/Q4MAyogeUix5ntCoBCjkm3BAXon+beAjWlf/qG0qiaYB975wKgU7/
KAvshY9olvx/FKWLnYLRtzcL6Oz64xqB0QHyLT7fUFv3c/B8vmAhSgTzhJVftXfA
Z8WITRUAR/5Jo0hkcmFK4/nQK88SxMvjTPbajURuGZpYXAm2Dr6k0Vh+ktfaPiSL
YLTPXKFaQZE3FWVuk8toXBqlydpHRWWhbEtpH970f5H1zr/NGFrMrscoYwQrZpVvQ
QQvaVxbDBsDIGjGnoc39GebSFeWg8V0mHLcHLJmAGSpFf6blgU+becpYrxuPYzPI
shBp2wYT9obN+A8CiGeM8fsAEQEAAAYkBJQQYAQIADwUCY22jgQIbDAUJCWYBgAAK

```



```
CRDk0wvFbVhD7DJ6B/4zd2gv3A/u13BT66fW9O4wSFPREy+hVR5i4LE0IXtXa36
pQdQdE6oNLAC9QNr0XrooW/jeSrSzpgHVbq8ezBvRZWWfMrHWazSkIIalogZR0/V
3U9COVqflbhxf5YiqVY9e/cN+OhK0GX2EI3DCn6C+w9lvVxGUKJ9qMQ9diObYrB
Hu3IGpNmO6fXo7ucjn2cuY3ii+3O5yaofhd3MRLNq5Yg7IRy7aDUZ4akBTIc7edM
GRpNUMEWi4MoYYM1agyT4f9kXZtR1m1PO9LJHke4yIHcgK2i943J14W8CYq5j0Sq
XtGInuT86nbHZlxpJ3/K70Y2pH4MFWEV5e9A1UZn=ShCh
```

```
-----END PGP PUBLIC KEY BLOCK-----"" | sudo tee /etc/apt/trusted.gpg.d/roox.asc
```

Добавьте файл с аутентификационной информацией для доступа к репозиторию RooX Solutions (логин и пароль передаются отдельно от инструкции):

```
echo "machine repo2.rooxcloud.com login ${LOGIN} password ${PASSWORD}" | sudo tee
/etc/apt/auth.conf.d/roox.conf
```

Подключите репозиторий RooX Solutions для загрузки пакетов:

```
echo "deb [arch=all signed-by=/etc/apt/trusted.gpg.d/roox.asc]
https://repo2.rooxcloud.com/customers/repository/customers-astra-deb/ astra main"| sudo
tee /etc/apt/sources.list.d/roox.list
```

Обновите список пакетов:

```
sudo apt update
```

1.2.2. Установка конфигурационного пакета на сервер

Установите пакет с файлами конфигурации из репозитория RooX Solutions: `sudo apt install -y env-config-uidm-fstec-astra-ss0=1.0.0-406`

После установки перезапустите сервисы haproxy, nginx: `sudo systemctl restart haproxy nginx`

проверка запуска:смотрим active, enabled, отсутствие критичных ошибок

```
sudo systemctl status haproxy.service
```

```
sudo systemctl status nginx.service
```

1.2.3. Установка model-dbsync из репозитория RooX Solutions: `sudo apt install -y model-dbsync=8.65.12-606`

Изменить параметр `zero_if_notfound` с "no" на "yes" в конфигурационном файле `/etc/parsec/mswitch.conf`.

По умолчанию, при подключении пользователя к защищенной СУБД PostgreSQL из состава ОС Astra Linux, выполняется проверка его мандатных атрибутов. Если пользователя с таким именем нет в ОС, попытка проверки его мандатных атрибутов завершается ошибкой.

Изменение параметра `zero_if_notfound` на `yes` означает присвоение нулевых



значений мандатных атрибутов, если они не назначены, и подключение пользователя к защищенной СУБД Postgresql из состава ОС Astra Linux проходит без ошибок.

Пользователей БД RX_SSO, RX_AUDIT, RX_FEDERATION нет в ОС, и у них нет мандатных атрибутов, поэтому надо внести это изменение:

```
sudo sed -i 's/zero_if_notfound: no/zero_if_notfound: yes/' /etc/parsec/mswitch.conf
```

Запустите процедуру обновления БД: `sudo`

```
/opt/roox-model-dbsync/bin/dbsync_as_tomcat.sh
```

Проверьте, что необходимая структура базы данных создана в схеме RX_SSO и появились записи в таблице `databasechangelog`.

```
sudo -u postgres psql -d RX_SSO -c "select  
pg_size_pretty(pg_relation_size('databasechangelog'));"
```

если все правильно (таблица существует, и не пуста) выведет ненулевой размер таблицы, при тестировании получили 88kB

1.2.4. Установка `audit-dbsync` из репозитория RooX Solutions: `sudo apt install -y audit-dbsync=8.65.12-606`

Запуск процедуры обновления БД: `sudo`

```
/opt/roox-audit-dbsync/bin/dbsync_as_tomcat.sh
```

Проверить, что необходимая структура базы данных создана в схеме RX_AUDIT и есть записи в таблице `databasechangelog`

```
sudo -u postgres psql -d RX_AUDIT -c "select  
pg_size_pretty(pg_relation_size('databasechangelog'));"
```

если все правильно (таблица существует, и не пуста) выведет ненулевой размер таблицы, при тестировании получили 8192 bytes

1.2.5. Установка `federation-dbsync` из репозитория RooX Solutions: `sudo apt install -y federation-dbsync=8.65.12-606`

Запуск процедуры обновления БД: `sudo`

```
/opt/roox-federation-dbsync/bin/dbsync_as_tomcat.sh
```

Проверить, что необходимая структура базы данных создана в схеме RX_SSO и есть записи в таблице `databasechangelog`

```
sudo -u postgres psql -d RX_FEDERATION -c "select  
pg_size_pretty(pg_relation_size('databasechangelog'));"
```

если все правильно (таблица существует, и не пуста) выведет ненулевой размер таблицы, при тестировании получили 16 kB



1.2.6. Установка roox-token-storage-10 из репозитория RooX Solutions: `apt install -y token-storage-10=1.0.16-105`

Убедитесь, что сервис запущен: `curl http://127.0.0.1:43108/token-storage-1.0/isAlive.jsp`
В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.7. Установка sso-tools из репозитория RooX Solutions: `sudo apt install -y sso-tools=12.1.0-35`

1.2.8. Установка sso-scripts из репозитория RooX Solutions: `sudo apt install -y sso-scripts=12.1.0-43424`

1.2.9. Установка roox-tomcat-addons из репозитория RooX Solutions: `sudo apt install -y roox-tomcat-addons-10=1.0-10`

1.2.10. Установка пакетов с драйверами для подключения к базе данных из репозитория RooX Solutions:

`sudo apt install -y roox-tomcat-postgresql-jdbc-93=9.3-4 roox-tomcat-oracle-jdbc-112=11.2.0-4`

1.2.11. Установка пакета словарей User-Agent с зависимостями: `sudo apt install -y libatomic1 ua-detection-dictionaries-51degrees-freeware=2022.12-18`

1.2.12. Установка roox-sso-01 из репозитория RooX Solutions: `sudo apt install -y sso-server-ocb-tomcat7=0.1.231005-4876`

Убедитесь, что сервис запущен: `curl http://127.0.0.1:15018/sso/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.13. Конфигурирование sso-server

Очистите существующую конфигурацию sso-server (выполняется от имени пользователя tomcat): `sudo -u tomcat /opt/ssoTools/scripts/sso-clean.sh`

Перезапустите sso-server после завершения очистки: `sudo systemctl restart roox-sso-01`

Проверка:смотрим active, enabled, отсутствие критичных ошибок

`sudo systemctl status roox-sso-01.service`

Убедитесь, что сервис запущен: `curl http://127.0.0.1:15018/sso/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии



компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

Запустите процесс конфигурирования сервиса sso-server (выполняется от имени пользователя tomcat): `sudo -u tomcat /opt/ssoTools/scripts/sso-configuration.sh`

Во время процесса конфигурации не должно быть ошибок. Проверить состояние сервиса: `curl http://127.0.0.1:15018/sso/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

Перезапустите sso-server после завершения конфигурации: `sudo systemctl restart roox-sso-01`

Проверка:смотрим active, enabled, отсутствие критичных ошибок

`sudo systemctl status roox-sso-01.service`

Убедитесь, что сервис запущен: `curl http://127.0.0.1:15018/sso/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.14. Установка roox-api-gateway-arm-10 из репозитория RooX Solutions: `sudo apt install -y api-gateway-arm-10=1.0.0-76`

Убедитесь, что сервис запущен: `curl http://127.0.0.1:14108/api-gateway-arm/isAlive`

В ответ на запрос должно вывестись "STATUS: ОК". Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.15. Установка roox-audit-search-api-10 из репозитория RooX Solutions: `sudo apt install -y audit-search-api-10=1.0.21-120`

Убедитесь, что сервис запущен: `curl`

`http://127.0.0.1:8388/audit-search-api-1.0/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.16. Установка roox-audit-writer-10 из репозитория RooX Solutions: `sudo apt install -y audit-writer-10=1.0.18~66-66`

Убедитесь, что сервис запущен: `curl http://127.0.0.1:39108/audit-writer-10/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам



тестирования)

1.2.17. Установка roox-certificate-service-10 из репозитория RooX Solutions: `sudo apt install -y certificate-service-10=1.0.6-32`

Убедитесь, что сервис запущен: `curl`

`http://127.0.0.1:42108/certificate-service-1.0/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.18. Установка roox-customer-webapi-10 из репозитория RooX Solutions: `sudo apt install -y customer-webapi-10=1.0.38-179`

Убедитесь, что сервис запущен: `curl`

`http://127.0.0.1:33108/customer-webapi-1.0/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.19. Установка roox-federation-webapi-21 из репозитория RooX Solutions: `sudo apt install -y federation-webapi-21=2.1.32-10`

Убедитесь, что сервис запущен: `curl`

`http://127.0.0.1:13218/federation-webapi-2.1/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.20. Установка roox-oauth2-consumer-server-10 из репозитория RooX Solutions: `sudo apt install -y oauth2-consumer-server-10=1.0.18-87`

Убедитесь, что сервис запущен: `curl http://127.0.0.1:27108/oauth2-consumer/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.21. Установка roox-sync-service-10 из репозитория RooX Solutions: `sudo apt install -y sync-service-wtl-10=1.0.4-38`

Убедитесь, что сервис запущен: `curl http://127.0.0.1:8488/sync-service-wtl-1.0/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)



1.2.22. Установка roox-webapi-notifier-10 из репозитория RooX Solutions: `sudo apt install -y webapi-notifier-10=1.0.58-130`

Убедитесь, что сервис запущен: `curl http://127.0.0.1:34108/notifier-1.0/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.23. Установка roox-webapi-server-otp-settings-10 из репозитория RooX Solutions: `sudo apt install -y webapi-server-otp-settings-10=1.0.25-100`

Убедитесь, что сервис запущен: `curl http://127.0.0.1:29108/otp-settings-1.0/isAlive.jsp`

В ответ на запрос должно вывестись "STATUS: ОК" и информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.24. Установка widgets-uidm-login из репозитория RooX Solutions: `sudo apt install -y widgets-uidm-login=1.0-124`

Убедитесь, что ресурсы приложения раздаются: `curl`

`http://127.0.0.1:13108/widgets/widgets-uidm-login/version.json`

В ответ на запрос должна вывестись информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.25. Установка widgets-uidm-selfservices из репозитория RooX Solutions: `sudo apt install -y widgets-uidm-selfservices=1.0-139`

Убедитесь, что ресурсы приложения раздаются: `curl`

`http://127.0.0.1:13108/widgets/widgets-uidm-selfservices/version.json`

В ответ на запрос должна вывестись информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.2.26. Установка uidm-admin-panel из репозитория RooX Solutions: `sudo apt install -y uidm-admin-panel=0.1.0-294`

Убедитесь, что ресурсы приложения раздаются: `curl http://127.0.0.1:13108/uidm-admin-panel/version.json`

В ответ на запрос должна вывестись информация о версии компонента. Возможна задержка в получении ответа до 3х минут (по результатам тестирования)

1.3. Тестирование

Загрузите multipart-архив, содержащий docker-образ с автотестами (логин и пароль передаются отдельно от инструкции):

`curl --user ${LOGIN}:${PASSWORD} https://repo.rooxcloud.com/share-astra/rooxteam-`



```
autotests-astra-linux-0-0-2.tar.0 -o /tmp/rooxteam-autotests-astra-linux-0-0-2.tar.0 && \
curl --user ${LOGIN}:${PASSWORD} https://repo.rooxcloud.com/share-astra/rooxteam-
autotests-astra-linux-0-0-2.tar.1 -o /tmp/rooxteam-autotests-astra-linux-0-0-2.tar.1 && \
curl --user ${LOGIN}:${PASSWORD} https://repo.rooxcloud.com/share-astra/rooxteam-
autotests-astra-linux-0-0-2.tar.2 -o /tmp/rooxteam-autotests-astra-linux-0-0-2.tar.2
```

Объедините multipart-архив:

```
cat /tmp/rooxteam-autotests-astra-linux-0-0-2.tar.* > /tmp/rooxteam-autotests-astra-linux-0-0-2.tar
```

Загрузите docker-образ из архива:

```
docker load -i /tmp/rooxteam-autotests-astra-linux-0-0-2.tar
```

Создайте директорию, в которую будут сохранены отчеты о результатах тестирования (пользователь с UID=1000 должны быть владельцем директории):

```
mkdir -p /tmp/test-reports && chown 1000:1000 /tmp/test-reports
```

Получаем IP и записываем в переменную serv_ip:

```
serv_ip=$(hostname -I | cut -d " " -f 1) && echo $serv_ip
```

Или, если несколько сетевых интерфейсов, то вместо переменной serv_ip

подставляем нужный ip адрес в команде ниже

Запускаем автотесты:

```
docker run --rm --env Host=fstec-astra.demo.rooxteam.com --add-host fstec-
astra.demo.rooxteam.com:$serv_ip --volume /dev/shm:/dev/shm --volume /tmp/test-
reports:/app/test-reports repo2.rooxcloud.com:8084/rooxteam-autotests-astra-linux:0.0.2
python3 tests.py astra_linux.test_ml_k_api
```

После завершения работы тестов в директории /tmp/test-reports будет создан отчет о результатах тестирования в формате html.

3 Удаление ПО:

3.1 выполнить системные команды, действия:

1. отключить автозапуск и остановить службы установленного ПО:

1.1. отключить автозапуск: `sudo systemctl disable roox-api-gateway-arm-10.service roox-audit-search-api-10.service roox-audit-writer-10.service roox-certificate-service-10.service roox-federation-21.service roox-oauth2-consumer-10.service roox-ss0-01.service roox-sync-service.service roox-token-storage.service roox-webapi-customer-10.service roox-webapi-notifier-10.service roox-webapi-otp-settings-10.service`

1.2. остановить службы: `sudo systemctl stop roox-api-gateway-arm-10.service roox-audit-search-api-10.service roox-audit-writer-10.service roox-certificate-service-10.service`



roox-federation-21.service roox-oauth2-consumer-10.service roox-sso-01.service roox-sync-service.service roox-token-storage.service roox-webapi-customer-10.service roox-webapi-notifier-10.service roox-webapi-otp-settings-10.service

2. удалить установленное ПО:

```
sudo apt purge -y env-config-uidm-fstec-astra-sso=1.0.0-406 model-dbsync=8.65.12-606
audit-dbsync=8.65.12-606 federation-dbsync=8.65.12-606 token-storage-10=1.0.16-105
sso-tools=12.1.0-35 sso-scripts=12.1.0-43424 roox-tomcat-addons-10=1.0-10 roox-
tomcat-postgresql-jdbc-93=9.3-4 roox-tomcat-oracle-jdbc-112=11.2.0-4 ua-detection-
dictionaries-51degrees-freeware=2022.12-18 sso-server-ocb-tomcat7=0.1.231005-4876
api-gateway-arm-10=1.0.0-76 audit-search-api-10=1.0.21-120 audit-writer-10=1.0.18~66-
66 certificate-service-10=1.0.6-32 customer-webapi-10=1.0.38-179 federation-webapi-
21=2.1.32-10 oauth2-consumer-server-10=1.0.18-87 sync-service-wtl-10=1.0.4-38
webapi-notifier-10=1.0.58-130 webapi-server-otp-settings-10=1.0.25-100 widgets-uidm-
login=1.0-124 uidm-admin-panel=0.1.0-294 widgets-uidm-selfservices=1.0-139
```

3. удалить docker-контейнеры, docker-образы:

3.1. определить контейнеры, которые требуется удалить: `docker ps -a`

3.2. остановить контейнеры: `docker stop ...`

3.3. удалить контейнеры: `docker rm -v ...`

3.4. определить образы, которые требуется удалить: `docker images`

3.5. удалить образы: `docker rmi`

4. удалить созданные базы данных и пользователей в postgres:

4.1. Создайте скрипт для удаления БД и пользователей:

```
echo """"DROP DATABASE \"RX_SSO\";
DROP USER \"RX_SSO\";
DROP DATABASE \"RX_AUDIT\";
DROP USER \"RX_AUDIT\";
DROP DATABASE \"RX_FEDERATION\";
DROP USER \"RX_FEDERATION\" ;
"""" > /tmp/drop_db_users.sql
```

4.2 Сделайте пользователя postgres владельцем этого скрипта и выполните скрипт от имени пользователя postgres:

```
sudo chown postgres:postgres /tmp/create_db_users.sql && sudo -u postgres psql -f
/tmp/drop_db_users.sql
```

проверка БД:



```
sudo -u postgres psql -l
```

выведет все БД в кластере, убедиться, что БД RX_SSO,RX_AUDIT,RX_FEDERATION отсутствуют.

проверка пользователей:

```
sudo -u postgres psql -c "SELECT * FROM pg_catalog.pg_user;"
```

выведет список пользователей, убедиться что пользователи RX_SSO,RX_AUDIT,RX_FEDERATION отсутствуют.

5. удалить директории

```
sudo rm -Rf /opt/roox-* /etc/tomcat /usr/share/tomcat/ /data /var/log/roox-*
```



Перечень используемых сокращений

«Руководство по КСЗ Ч. 1» – документ «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1» РУСБ.10015-01 97 01-1;

Astra Linux SE 1.7.0 – операционная система специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (очередное обновление 1.7);

Astra Linux SE 1.7.5 – операционная система специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (очередное обновление 1.7) с установленным оперативным обновлением безопасности БЮЛЛЕТЕНЬ № 2023-1023SE17 (оперативное обновление 1.7.5);

ДВиС – дирекция внедрения и сопровождения;

ЗПС – замкнутая программная среда;

КСЗ – комплекс средств защиты;

МКЦ – мандатный контроль целостности;

МРД – мандатное управление доступом;

ОС – операционная система;

ПО – программное обеспечение «Система управления доступом Identity Management» версии 23.10.

Идентификатор документа d0c8f01f-0f31-4602-a835-7f463f3fd8c8

Документ подписан и передан через оператора ЭДО АО «ПФ «СКБ Контур»

Организация, сотрудник

Доверенность: рег. номер, период действия и статус

Сертификат: серийный номер, период действия

Дата и время подписания

Подписи отправителя:



ООО "РУСБИТЕХ-АСТРА"
Проканюк Дмитрий Сергеевич



Не приложена при подписании

043C5A7100B6B007A24D9A5E4F
91BFE299
с 10.11.2023 09:42 по 10.11.2024
09:42 GMT+03:00

21.02.2024 10:54 GMT+03:00
Подпись соответствует файлу
документа

