Платформа безопасной разработки «Шерлок»

Руководство по установке

Москва

2025

Содержание

1. Введение

1.1. Область применения

Платформа безопасной разработки «Шерлок» (далее — Система) предназначена для оптимизации процессов безопасной разработки за счет консолидации информации об ИБ-дефектах, получаемой из различных сканеров, и предоставления возможности централизованной одновременной работы с ними различными пользователями.

1.2. Краткое описание возможностей

Платформа безопасной разработки «Шерлок» обладает следующей функциональностью:

– централизованное управление сканерами (создание задач на сканирование, получение и обработка результатов задач на сканирование), позволяющими идентифицировать ИБ-дефекты (потенциальные уязвимости) в анализируемом программном обеспечении. Система поддерживает следующие типы сканеров:

- Static Application Security Testing (SAST);
- Software Composition Analysis (SCA);
- Container Image Analysis (Image Scan).

– консолидация информации, получаемой в результате реализации задач на сканирование, создаваемых различными сканерами программного обеспечения;

– сокращение информационного «шума» за счет дедупликации идентичных (повторяющихся) результатов, получаемых от различных сканеров;

– управление ИБ-дефектами (изменение статусов, уровня критичности, принятие рисков и т.д.), выявленными в анализируемом программном обеспечении;

– выстраивание процесса по управлению ИБ-дефектами за счет интеграции с системами управления задач (Issue Tracker);

– предоставление сводной информации о состоянии информационной безопасности анализируемого программного обеспечения.

2. Назначение и условия применения

2.1. Виды деятельности, функции

Основными функциями системы «Шерлок» являются:

2.1.1. Идентификация уязвимостей. Для этого, как правило, используется набор сканеров, различающихся по типам идентифицируемых уязвимостей и по анализируемым объектам, например:

– Static Application security Testing, SAST. Класс сканеров, анализирующий исходный код ПО;

– Software Composition Analysis, SCA. Класс сканеров, анализирующий зависимости ПО;

– Image Scan. Класс сканеров, анализирующий образы контейнеров.

2.1.2. Анализ ложных срабатываний. Сканеры, используемые на предыдущем этапе, генерируют большое количество данных, включая ложные срабатывания (false positive). Поэтому для дальнейшей работы ИБ-специалисты должны их идентифицировать и исключить из перечня рассматриваемых уязвимостей;

2.1.3. Расстановка приоритетов (triage), постановка задач на устранение. Задача этапа — расставить приоритеты устранения выявленных уязвимостей в зависимости от их типа, уровня критичности, потенциальной опасности для Компании и т.д. После того как приоритеты для всех уязвимостей были определены, ИБ-специалисты создают задачи в Системе управления задачами (Issue Tracker), используемой в Компании;

2.1.4. Контроль устранения уязвимостей. Этап, на котором ИБ-специалисты, получив уведомление о том, что уязвимость была устранена, запускают повторное сканирование, чтобы гарантировать устранение уязвимости и отсутствие новых (которые могут быть привнесены исправлением рассматриваемой);

2.1.5. Подготовка отчетности. Сбор и консолидация информации о проделанной работе по устранению уязвимостей за определенный интервал времени (например, за release, квартал, полугодие, год и т.д.).

2.2. Программные и аппаратные требования к Системе

Возможно несколько способов установки ПБР «Шерлок», описанные в пунктах 2.2.1 — 2.2.4 настоящего Руководства.

2.2.1. Вариант установки Standalone Docker Compose

Для запуска ПБР «Шерлок» в данном варианте установки требуется минимум:

- ОС Linux. Рекомендуемая ОС — Debian 12;
- Docker. Версия: 24.0.2 и выше;
- Docker Compose. Версия: v2.19.1 и выше;
- CPU: 2 vCPU;
- Memory: 8 GB;
- Disk SSD: 50 GB.

2.2.2. Вариант установки Standalone Helm

Для запуска ПБР «Шерлок» в данном варианте установки требуется минимум:

- ОС Linux. Рекомендуемая ОС — Debian 12;
- Kubernetes. Версия: 1.28 и выше. Рекомендуемая 1.31.0;
- Helm. Версия: 3.16.1 и выше;
- Утилиты ctr containerd версии 1.6.20~ds1 и выше, kubectl версии v1.31.4 и выше;
- CPU: 2 vCPU;
- Memory: 8 GB;
- Disk SSD: 50 GB.

2.2.3. Вариант установки High availability Docker Compose

Для запуска ПБР «Шерлок» в данном варианте рекомендуется использовать для БД и хранилищ отдельные выделенные виртуальные машины, рекомендуемые требования представлены в таблице 1.

Для запуска приложения в данном варианте установки требуется установить:

- ОС Linux. Рекомендуемая ОС — Debian 12;
- Docker. Версия: 24.0.2 и выше;
- Docker Compose. Версия: v2.19.1 и выше.

Таблица 1 — Рекомендуемое аппаратное и программное назначение виртуальных машин ПБР «Шерлок»

Тип сервера	Количество ядер в сервере / CPU cores	Объем оперативной памяти, Гб	Объем дискового пространства, Гб	Назначение
Виртуальный	4	8	100	PostgreSQL cluster 1
Виртуальный	4	8	100	PostgreSQL cluster 2
Виртуальный	4	8	100	PostgreSQL cluster 3
Виртуальный	8	8	100	ClickHouse cluster 1
Виртуальный	8	8	100	ClickHouse cluster 2
Виртуальный	8	8	100	ClickHouse cluster 3
Виртуальный	8	8	100	ClickHouse cluster 4
Виртуальный	2	4	10	ClickHouse keeper 1
Виртуальный	2	4	10	ClickHouse keeper 2
Виртуальный	2	4	10	ClickHouse keeper 3
Виртуальный	4	8	100	Minio (S3) instance 1
Виртуальный	4	8	100	Minio (S3) instance 2
Виртуальный	4	8	100	Minio (S3) instance 3
Виртуальный	4	8	100	Minio (S3) instance 4
Виртуальный	4	8	80	MongoDB instance 1
Виртуальный	4	8	80	MongoDB instance 2
Виртуальный	4	8	80	MongoDB instance 3
Виртуальный	4	4	80	Sherlock instance 1
Виртуальный	4	4	80	Sherlock instance 2
Виртуальный	4	4	80	Sherlock instance 3

2.2.4. Вариант установки High availability Helm

Для запуска ПБР «Шерлок» в данном варианте рекомендуется использовать для БД и хранилищ отдельные выделенные виртуальные машины, рекомендуемые требования представлены в таблице 2:

- ОС Linux. Рекомендуемая ОС — Debian 12;
- Kubernetes: версия: 1.28 и выше. Рекомендуемая версия — 1.31.0;
- Helm: версия 3.16.1 и выше.

Таблица 2 — Рекомендуемое аппаратное и программное назначение виртуальных машин ПБР «Шерлок»

Тип сервера	Количество ядер в сервере / CPU cores	Объем оперативной памяти, Гб	Объем дискового пространства, Гб	Назначение
Виртуальный	4	8	100	PostgreSQL
Виртуальный	8	8	100	ClickHouse
Виртуальный	4	8	100	Minio (S3)
Виртуальный	4	8	80	MongoDB
Виртуальный	4	4	80	Sherlock instance 1
Виртуальный	4	4	80	Sherlock instance 2
Виртуальный	4	4	80	Sherlock instance 3

3. Подготовка к работе

3.1. Состав и содержание дистрибутива

Состав и содержание дистрибутива отличается в зависимости от выбранного варианта установки.

3.1.1. Standalone Docker Compose дистрибутив

Предоставляется архив sherlock-dc-v0.1.1.linux-amd64.tar.gz, который содержит следующие файлы и каталоги:

- bin — каталог для бинарных файлов установки, запуска и остановки сервисов;
- storage-clickhouse — каталог данных для установки ClickHouse;
- storage-mongo — каталог данных для установки MongoDB;
- storage-postgres — каталог данных для установки PostgreSQL;
- storage-s3 — каталог данных для установки Minio;
- docker-compose.altair.yml — файл конфигурации запуска Frontend;
- docker-compose.nexus.yml — файл конфигурации запуска Nexus;
- docker-compose.orion.yml — файл конфигурации запуска Backend;
- docker-compose.storage-clickhouse.yml — файл конфигурации запуска ClickHouse;
- docker-compose.storage-mongo.yml — файл конфигурации запуска MongoDB;
- docker-compose.storage-postgres.yml — файл конфигурации запуска PostgreSQL;
- docker-compose.storage-s3.yml — файл конфигурации запуска Minio;
- docker-images.tar — архив с образами контейнеров, необходимых для корректного функционирования ПБР «Шерлок»;
- local_env.env — конфигурационный файл, содержащий параметры запуска ПБР «Шерлок»;
- stand.env — стендовое окружение.

3.1.2. Standalone Helm дистрибутив

Предоставляется архив sherlock-hs-v0.1.1.linux-amd64.tar.gz, который содержит следующие файлы и каталоги:

- rawfile-csi/ — helm-чарт для установки Rawfile CSI;
- sherlock/ — helm-чарт для установки ПБР «Шерлок»;

– ingress-nginx/ — helm-чарт для установки Nginx Ingress;

– nexus/ — helm-чарт для установки Nexus;

– docker-images — директория с архивами образов контейнеров, необходимых для корректного функционирования ПБР «Шерлок»;

– sherlock/values.yaml — конфигурационный файл, содержащий параметры запуска ПБР «Шерлок».

3.1.3. High availability Docker Compose дистрибутив

Предоставляется архив sherlock-hadc-v0.1.1.linux-amd64.tar.gz, содержание которого повторяет архив, описанный в п. 3.1.1.

3.1.4. High availability Helm дистрибутив

Предоставляется архив sherlock-hah-v0.1.1.linux-amd64.tar.gz, содержание которого полностью повторяет архив, описанный в п. 3.1.2.

3.2. Описание параметров конфигурационного файла

Таблица 3 — Описание базовых параметров конфигурационного файла, который используется Системой

Название переменной	Значение по умолчанию	Описание
MACHINE_IP	192.168.0.226	IP-адрес рабочей станции, на которую производится установка Системы
REACT_APP_API_ADDRESS	${MACHINE_IP}:8000	URL API Backend Системы
CLICKHOUSE_HOST	${MACHINE_IP}	IP-адрес хоста с установленным ClickHouse
CLICKHOUSE_PORT	8123	Порт, который слушается установленным ClickHouse
CLICKHOUSE_DATABASE	default	Название БД в ClickHouse
CLICKHOUSE_USERNAME	default	Логин пользователя, под которым Система будет обращаться в ClickHouse

CLICKHOUSE_PASSWORD	default	Пароль пользователя, под которым Система будет обращаться в ClickHouse
S3_HOST	${MACHINE_IP}	IP-адрес хоста с установленным S3
S3_PORT	9000	Порт, который слушается установленным S3
S3_ACCESS_KEY	minio_access_key	Идентификатор ключа доступа к S3
S3_SECRET_KEY	minio_secret_key	Значение секретного ключа для доступа к S3
S3_BUCKET	sherlock	Название корзины в S3 для хранения данных Системы
MONGO_DB_HOST	${MACHINE_IP}	IP-адрес хоста с установленным MongoDB
MONGO_DB_PORT	27017	Порт, который слушается установленным MongoDB
MONGO_DB_DATABASE	orion	Название БД в MongoDB
MONGO_DB_USERNAME	orion	Логин пользователя, под которым Система будет обращаться в MongoDB
MONGO_DB_PASSWORD	orion	Пароль пользователя, под которым Система будет обращаться в MongoDB
POSTGRES_HOST	${MACHINE_IP}	IP-адрес хоста с установленным PostgreSQL
POSTGRES_PORT	5432	Порт, который слушается установленным PostgreSQL
POSTGRES_DATABASE	orion	Название БД в PostgreSQL

POSTGRES_USERNAME	postgres	Логин пользователя, под которым Система будет обращаться в PostgreSQL
POSTGRES_PASSWORD	postgres	Пароль пользователя, под которым Система будет обращаться в PostgreSQL
JWT_SECRET	e57c616fab74489391912415a2da80259c2a790ee1ec239bb4c4c5fd667658863e1fbe5e57a8c381ae54cbb67b571c08da6e3ebeecf3b2b13b1f10a392cf2aa0b7ccdf891d6e1723af8eaff4ad06bd4805d79e2f3919d1ecc4eac27be1ddf070002a814987fb1df7277c12f456bdaf783ac689f6a74431ae478f96015eeb17491e4929414ed5a6f8b85af7b69b2aac226ce94d9106d151ea1f94ac36dfa6da0c31c07fb6adf726b09d28913a0e8f5b8a46d1780c4fba8d345756781d86a84241717314a9c416da626148645695c561bd8ea23119d98d4167a686adc2ff18a4cfa00398eaaf39e63de1e5777c44968e39ad081916b90d0cd7b73e36bf37bbf534	Ключ для генерации JWT при авторизации пользователей. Длина ключа — 256 бит. Для каждого стенда нужно сгенерировать свой ключ при помощи доступного генератора JWT Secret. Например, онлайн генераторы: – https://jwtsecret.com/generate – https://toolcluster.com/tools/jwt-secret-generator

3.3. Установка, запуск и остановка ПО

3.3.1. Установка, запуск и остановка Standalone Docker Compose

Установка standalone-решения подразумевает минимальную настройку приложения перед эксплуатацией. Предполагается, что установка производится на виртуальную машину с установленной ОС Linux (см. рекомендации). Дальнейшие шаги рекомендуется выполнять под пользователем с правами root или с возможностью использовать команду sudo.

1. Установка компонентов:
   - Docker. Инструкцию по установке Docker на выбранную ОС Linux можно получить на сайте (**https://docs.docker.com/engine/install/**);
   - Docker Compose. Инструкцию по установке Docker на выбранную ОС Linux можно получить на сайте (**https://docs.docker.com/compose/install/**).
2. Копирование архива с дистрибутивом в папку <home>(<home> — каталог для Системы. Например, /opt/sherlock).
3. Распаковка архива с дистрибутивом (например, sherlock-dc-v0.1.1._linux-amd64.tar.gz)

```
tar -xf sherlock-dc-v0.1.1.linux-amd64.tar.gz
```

4. В файле с конфигурацией — local_env.env необходимо указать IP-адрес виртуальной машины, на которую идет процесс установки Системы:

```
MACHINE_IP=<ip_adress>
```

5. В файле с конфигурацией — local_env.env необходимо указать JWT Secret:

```
JWT_SECRET=<256_bit_secret>
```

6. Для директории **<home>/bin/** установить права 755;
7. Запуск установки — распаковка образов из архива

```
<home>/bin/install
```

8. Запуск сервисов

```
<home>/bin/up
```

Для остановки сервисов необходимо выполнить команду:

```
<home>/bin/down
```

3.3.2. Установка, запуск и остановка Standalone Helm

Дальнейшие шаги рекомендуется выполнять под пользователем с правами root или с возможностью использовать команду sudo.

1. Установка компонентов:
   - Helm. Инструкцию по установке Helm на выбранную ОС Linux можно получить на сайте (**https://helm.sh/ru/docs/intro/install/**);

- Kubernetes. Инструкцию по установке Kubernetes на выбранную ОС Linux можно получить на сайте (**https://helm.sh/ru/docs/intro/install/**);

2. В кластере Kubernetes необходимо создать namespace, в который будет производиться установка ПБР «Шерлок» (для примера здесь и далее будем использовать имя test)

```
kubectl create ns test
```

3. Копирование архива sherlock-hs-v0.1.1.linux-amd64.tar.gz в папку <home>(<home> — каталог для Системы. Например, /opt/sherlock).
4. Распаковка архива sherlock-hs-v0.1.1.linux-amd64.tar.gz

```
tar -xf sherlock-hs-v0.1.1.linux-amd64.tar.gz
```

5. Сначала необходимо загрузить **все** образы из директории docker-images/ для установки ПБР «Шерлок» в реестр, который будет использоваться для создания контейнеров. После этого необходимо явно указать информацию об образах для всех компонентов, которые планируется использовать, в соответствующих чартах:
   - Sherlock — sherlock/values.yaml (для образов Orion и Altair, строка 7);
   - Clickhouse — sherlock/clickhouse/values.yaml (строка 92 и далее, опционально, зависит от типа установки);
   - Minio — sherlock/minio/values.yaml (строка 75 и далее, опционально, зависит от типа установки);
   - MongoDB — sherlock/mongodb/values.yaml (строка 136 и далее, опционально, зависит от типа установки);
   - PostgreSQL — sherlock/postgresql/values.yaml (строка 115 и далее, опционально, зависит от типа установки);
   - ZooKeeper — sherlock/zookeeper/values.yaml (строка 92 и далее, опционально, зависит от типа установки);
   - Rawfile-CSI — rawfile-csi/charts/rawfile-csi/values.yaml (строка 22 и далее, опционально, зависит от типа установки);
   - Ingress-Nginx — ingress-nginx/values.yaml (строка 8, строка 29, строка 33, опционально, зависит от типа установки);
   - Nexus — nexus/values.yaml (строка 8).

6. В случае отсутствия nginx-ingress-controller в среде Kubernetes необходимо установить его с помощью команд (находясь в корневой директории распакованного архива sherlock-hs-v0.1.1.linux-amd64.tar.gz) — следует обратить внимание, что во второй команде вместо <IP хоста> нужно поставить IP того хоста из worker'ов Kubernetes, с которого планируется иметь доступ до ПБР «Шерлок».

```
kubectl create ns nginx
helm upgrade --install ingress-nginx -n nginx ingress-nginx
kubectl patch svc ingress-nginx-controller -n nginx -p
'{"spec": {"type": "LoadBalancer", "externalIPs":["<IP
хоста>"]}}'
```

7. Если уже имеется rawfile-csi provisioner с уже созданным storageClass, необходимо в rawfile-csi/charts/rawfile-csi/values.yaml на строке 36 указать enabled: false, чтобы предотвратить создание нового storageClass. В таком случае в sherlock/charts/clickhouse/values.yaml в строке 23, sherlock/charts/minio/values.yaml в строке 22, sherlock/charts/mongodb/values.yaml в строке 24, sherlock/charts/porstgresql/values.yaml в строке 21, sherlock/charts/zookeeper/values.yaml в строке 23 необходимо указать соответствующий уже имеющийся storageClass.

В случае отсутствия rawfile-csi provisioner его необходимо установить с помощью команд (находясь в корневой директории распакованного архива sherlock-hs-v0.1.1.linux-amd64.tar.gz)

```
kubectl create ns rawfile-csi
helm upgrade --install rawfile-csi -n rawfile-csi rawfile-
csi
```

8. Далее необходимо установить Nexus с помощью команд (находясь в корневой директории распакованного архива sherlock-hs-v0.1.1.linux-amd64.tar.gz)

```
kubectl create ns nexus
helm upgrade --install nexus -n nexus nexus
```

9. Запуск приложения Sherlock. Для начала нужно произвести конфигурацию в sherlock/values.yaml:

– если нужно использовать уже имеющуюся инфраструктуру (MongoDB, Minio, Postgresql, Clickhouse), доступную с хоста, где установлен кластер Kubernetes, необходимо отключить создание инфраструктуры средствами helm в строках 11, 19, 27, 35, поставив значение переменных deploy: false , а также указать все данные для подключения к инфраструктуре на строках 12–41 (имена пользователей, пароли, порты, хосты, названия баз данных и т.д.);

– если нужно поднимать новую инфраструктуру (MongoDB, Minio, Postgresql, Clickhouse), то это можно сделать, включив создание инфраструктуры средствами helm в строках 11, 19, 27, 35, поставив значение переменных enabled: true (в данном случае остальное наполнение переменных для подключения остается несущественным — их можно оставить либо пустыми, либо шаблонными значениями, которые эти переменные имели изначально сразу после распаковки архива sherlock-hs-v0.1.1.linux-amd64.tar.gz). Если необходимо использовать конкретные образы для деплоя инфраструктурных сервисов, то соответствующие наименования образов следует указать в sherlock/charts/clickhouse/values.yaml в строках 92–95, sherlock/charts/minio/values.yaml в строках 75–78/102–105, sherlock/charts/mongodb/values.yaml в строках 262–265 , sherlock/charts/porstgresql/values.yaml в строках 115–118 , sherlock/charts/zookeeper/values.yaml в строках 92–95;

– указать jwt-secret, необходимый для работы бэкенда ПБР «Шерлок», указанный на строке 96 файла sherlock/values.yaml.

Далее запустить команду (находясь в корневой директории распакованного архива sherlock-hs-v0.1.1.linux-amd64.tar.gz):

```
helm upgrade --install sherlock -n test sherlock
```

Для остановки приложения ПБР «Шерлок» необходимо выполнить команду:

```
helm uninstall sherlock -n test
```

3.3.3. Установка, запуск и остановка High availability Docker Compose

Установка данного варианта полностью повторяет шаги из п. 3.3.3 с разницей в том, что требуется использовать архив с названием sherlock-hadc-v0.1.1.linux-amd64.tar.gz.

3.3.4. Установка, запуск и остановка High availability Helm

Установка данного варианта полностью повторяет шаги из раздела «3.3.2. Установка, запуск и остановка Standalone Helm» с разницей в том, что требуется в файле sherlock/values.yaml на строке 5 поставить необходимое число реплик приложения.

3.4. Проверка работоспособности установленного ПО

Когда конфигурирование системы будет закончено, администратор может выполнить первоначальную проверку установленного им программного обеспечения. Для этого необходимо из окна браузера выполнить HTTP GET запрос по 80-му порту на доменное имя или IP-адрес машины с установленным на нее ПБР «Шерлок».

Открытие окна авторизации в окне браузера приложения обозначает его полное функционирование в данной установке.

Появление каких-либо ошибок при первом запуске говорит о неправильной установке или настройке программного обеспечения.

4. Резервное копирование

Резервное копирование требуется только для сервисов хранения данных и осуществляется резервным копированием виртуальных машин с учетом корпоративных правил и процедур для создания резервных копий. Сервисы хранения данных:

- Clickhouse — сервис хранения статистических данных;
- MongoDB — сервис хранения ИБ-дефектов;
- PostgreSQL — сервис хранения пользовательских данных;
- S3 — сервис хранения файловых данных.

Если конфигурация запуска сервисов подразумевает использование сервисов хранения, установленных не с использованием инсталлятора, то их DRP обеспечит надежность хранения данных приложения.

5. Перечень терминов и сокращений

Термин, сокращение	Описание
ИБ-дефект	Дефект информационной безопасности
ОС	Операционная система
Система	Платформа безопасной разработки «Шерлок»
ПБР	Платформа безопасной разработки
ПО	Программное обеспечение
CPU	Центральный процессор
DRP	План по восстановлению работоспособности Системы
IP	Интернет-протокол
REST	Протокол взаимодействия информационных систем
SSD	Твердотельный накопитель данных
Image Scan	Container Image Analysis
Issue Tracker	Система управления задач
SAST	Static Application Security Testing
SCA	Software Composition Analysis