RT Protect TI

Руководство по установке и эксплуатации

Версия 1.0.1 от 21 сентября 2024
Разработано компанией АО «РТ-Информационная безопасность»

Оглавление

1. Общие положения

1.1 Назначение и общая информация

RT Protect TI – это программное решение, которое позволяет собирать, обрабатывать, накапливать и распространять данные о киберугрозах (Threat Intelligence), то есть выполняет функции TI-платформы. Решение предоставляет аналитикам информационной безопасности возможность работать с актуальными сведениями об угрозах для эффективных расследований инцидентов и упреждения вредоносной активности.

Модуль управления сервисом сбора TI-данных, находящийся на сервере, предназначен для следующих задач:

– администрирование пользователей, взаимодействующих с сервисом;

– администрирование организаций, взаимодействующих с сервисом;

– подключение различных источников данных с информацией о вредоносных или безопасных артефактах;

– формирование аналитики различных форматов для распространения клиентам TI-портала через API;

– получение вердикта по анализируемым артефактам;

– регистрация действий пользователей.

Программа функционирует под управлением ОС Linux Ubuntu 20.04.5 LTS.

Для распространения сервиса применяется две модели:

– on-premise (покупка дистрибутива и установка на мощностях клиента);

– on-cloud (установка и развертывание осуществляется на мощностях предприятия-разработчика сервиса уполномоченными сотрудниками, доступ к сервису как услуга).

Программа предназначена для обработки информации, не являющейся секретной.

Программа имеет многофункциональный пользовательский интерфейс и подразумевает наличие следующих ролей пользователя:

Пользователь – может осуществлять проверку поддерживаемых платформой артефактов, просматривать отчеты по артефактам, просматривать графики проверки артефактов с распределением по времени.

Администратор – выполняет установку и корректную настройку программы в соответствии с настоящим руководством, регистрирует новых пользователей, подключенных к сервису, регистрирует новые организации, подключенные к сервису, подключает новые источники данных, предоставляющие информацию, и осуществляет другие функции, описанные в данном руководстве;

Аналитик – пользователь, ответственный за анализ поступающих от программы данных. Аналитик принимает решения по дальнейшей реакции на обнаруженные угрозы.

1.2 Идентификация документа

Данный документ кратко можно идентифицировать согласно таблице 1.

Таблица 1 – Идентификация документа

Название документа	«RT Protect TI» Инструкция по установке сервиса
Версия документа	Версия 1.0.1
Идентификация программы	Сервис по предоставлению аналитики «RT Protect TI»
Идентификация разработчика	АО «РТ-Информационная безопасность»

1.3 Аннотация документа

Документ предназначен для организаций, которые заключили договор с компанией поставщиком/производителем (АО «РТ-Информационная безопасность») по модели поставки on-premise (покупка дистрибутива и установка на собственных мощностях).

2. Описание процедуры установки

2.1 Общие положения

Перед началом процедуры установки требуется ознакомиться с требованиями к среде функционирования.

2.2 Требования к среде функционирования

Программа работает на 64-х разрядной платформе семейства Linux (Ubuntu 20.04.5 LTS).

Аппаратная платформа сервера обеспечивает возможность выполнения функциональных требований, предъявляемых к серверу, с учетом технологии его построения, критериев производительности и отказоустойчивости.

Программа пригодна для функционирования на аппаратных платформах, указанных в таблице 2.

Таблица 2 – Программно-аппаратное обеспечение

Характеристики	**Платформа**	
	Минимальные требования	Рекомендуемые требования
Процессор	Не менее 8 ядер частотой минимум 2,4 ГГц	Не менее 10 ядер частотой минимум 2,4 ГГц
Оперативная память	16 ГБ	32ГБ
Жесткий диск (свободное пространство)	1 ТБ	2 ТБ

Программа поддерживает работу в браузерах, представленных в таблице 3.

Таблица 3 – Список поддерживаемых браузеров

№ п/п	**Тип браузера**	**Минимальная рекомендуемая версия**
1	Google Chrome	Версия не ниже 92.0.4515.107
2	Firefox Browser	Версия не ниже 83.0

2.3 Инструкция по развертыванию

Шаг 1. Подготовка окружения

Для развертывания сервера необходимо, чтобы на компьютере было установлено следующее программное обеспечение:

– Python (не ниже версии 3.10.0) [установка].

– Ansible (не ниже версии 5.7.1) [установка].

С компьютера должен быть доступ на сервер (Docker-хост) по SSH, а у удаленного пользователя права sudo (подойдет и root-пользователь).

Для удобства использования Ansible добавьте свой открытый SSH-ключ на сервере, иначе вам может потребоваться установка дополнительной утилиты sshpass.

Далее нужно проверить, что с сервера есть доступ к реестру Docker-образов (http://docker.rt-protect.ru/) и есть доступ в Интернет (для установки deb-пакетов).

Шаг 2. Создание конфигурации сервера

Чтобы создать конфигурацию сервера, необходимо клонировать репозиторий на АРМ, на котором разворачивается сервер TI, перейдя по ссылке https://gitlab.rt-protect.ru/threat-intelligence/ti-deploy, либо осуществите запрос на получение данного репозитория к контактным лицам от поставщика.

Далее необходимо перейти в корень репозитория ti-deploy.

Структура каталогов будет представлена на рисунке 1.

```
. (ti-deploy)
│
├── README.md
├── compose-files
│   ├── docker-compose.yml
│   └── env.j2
├── config
│   └── default
│       ├── config.yml
│       ├── docker-compose.override.yml
│       ├── server.crt
│       └── server.key
├── ti-install.yml
└── ti-update.yml
```

Рисунок 1 – Структура каталогов репозитория ti-deploy

В каталоге config хранятся конфигурации серверов. Сюда нужно будет добавить свою новую конфигурацию. Описание конфигурации:

config_name – название конфигурации, должно совпадать с названием каталога, содержащего данный конфигурационный файл;

host_ip – IP-адрес целевой машины;

home_url – URL, по которому открывается главная страница приложения;

default_culture – локализация сервера по умолчанию, поддерживается 2 варианта «ru-RU» и «en-US»;

docker_registry* – настройки подключения к реестру Docker-образов;

db_user* – имя пользователя и пароль для работы с PostgreSQL, если не планируется подключения в БД извне, можно использовать любую комбинацию имени и пароля;

db_backup_path – путь до каталога, в котором будут храниться бекапы БД;

jwt_token_secret – секрет, на основе которого будут подписываться выпущенные токены, желательно делать сложным и уникальным, минимальная длина - 32 символа;

jwt_token_expiration – время устаревания JWT-токена в минутах, чем меньше, тем безопаснее для сервера;

refresh_token_expiration – время устаревания токена в минутах, если не заходить на сайт в течение указанного времени, потребуется повторный вход в систему;

detect_enabled – признак включения записи «Активность», если false, то страница «Активность» обновляться не будет;

mongo_user* – логин и пароль для инициализации MongoDB, лучше поменять с дефолтных значений;

storage_path – путь, где MongoDB будет хранить свои данные;

mongo_version – версия MongoDB;

redis_password – пароль Redis;

redis_path – путь до хранилища Redis;

ptms_enabled – флаг доступности модуля PT MultiScanner;

ptms_report_lifetime – время актуальности отчета PTMS в днях;

ptms_api_key – ключ доступа до PTMS;

ptms_api_path – путь до API PTMS;

ptms_auto_enrichment – автоматическое обогащение из источника данных;

ptms_sandbox – отправлять файлы на динамический анализ (если позволяет продукт);

ptms_sandbox_image_id – образ динамического анализа (если позволяет продукт);

ptms_sandbox_duration – продолжительность динамического анализа;

kaspersky_enabled – флаг доступности модуля KasperskyTI;

kaspersky_api_key – ключ доступа для KasperskyTI;

kaspersky_auto_enrichment – автоматическое обогащение из источника данных;

vt_enabled – флаг активации модуля VT;

vt_api_key – ключ доступа для VT;

vt_auto_enrichment – автоматическое обогащение из источника данных;

athena_enabled – флаг активации модуля Athena;

athena_url – путь до API Athena;

athena_token – ключ доступа для Athena;

athena_ssl – флаг проверки сертификата при обращении к Athena;

athena_auto_enrichment – автоматическое обогащение из источника данных;

rst_cloud_enabled – флаг активации модуля RST Cloud;

rst_cloud_token – ключ доступа для RST Cloud;

rst_cloud_auto_enrichment – автоматическое обогащение из источника данных;

netlas_enabled – флаг активации модуля Netlas;

netlas_token – ключ доступа для Netlas;

netlas_auto_enrichment – автоматическое обогащение из источника данных;

zk_data_path, zk_logs_path, kafka_data_path – параметры конфигурации Kafka;

minio*– параметры конфигурации MinIO;

smtp* – параметры SMTP;

vulnerabilities_limit – лимит на количество уязвимостей из базы, с которыми будут проверяться компоненты, если установлено -1, значит, без ограничения, рекомендуется установить значение -1;

далее идут версии компонентов и конфигурация сканера уязвимостей;

compose_dir – путь до каталога с файлами docker-compose.yml и env.

Следует обратить внимание, что каталог config не отслеживается гитом (записан в .gitignore) (кроме подкаталога default), поэтому можно добавлять свои собственные конфигурации в любом количестве и не бояться, что чувствительные данные из них попадут в общий репозиторий.

Правильным подходом будет держать все свои конфигурации в одном месте, в каталоге config в соответствующих подкаталогах. Можно одновременно управлять несколькими конфигурациями серверов. Именовать подкаталоги удобно, например, по IP-адресу сервера или домену. Тогда структура каталога config со временем примет вид, представленный на рисунке 2.

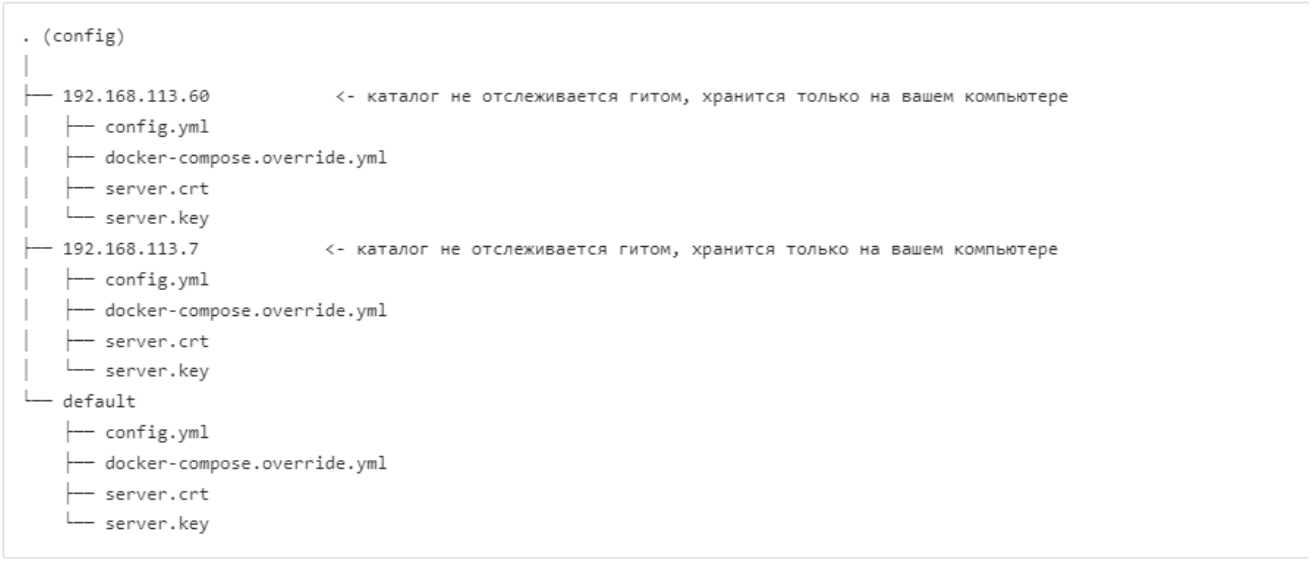

Рисунок 2 – Структура каталога config

В данной структуре имеется две дополнительные конфигурации (помимо дефолтной): для сервера 192.168.113.60 и для сервера 192.168.113.7. Теперь следует создать свой подкаталог в каталоге config и скопировать в него содержимое каталога config/default:

– config.yml – настройки конфигурации;

– docker-compose.override.yml – compose-файл, дает возможность переопределить основной compose-файл на уровне отдельной конфигурации;

– server.crt – открытый сертификат сервера в формате PEM;

– server.key – закрытый ключ сертификата в формате PEM.

Далее следует настроить содержимое каждого файла так, как требуется. Это и будет ваша конфигурация. В файле config.yml содержатся все доступные настройки, включая версии компонентов системы.

Несмотря на то, что docker-compose.yml уже есть и настроен правильно (находится в каталоге compose-files), пользователь, устанавливающий систему, может внести свои коррективы. Это можно сделать на уровне вашей конфигурации, отредактировав файл docker-compose.override.yml. Такие изменения не затронут другие конфигурации, это хороший уровень изоляции.

Файлы server.crt и server.key можно настроить (если есть сертификат, подписанный Центром Сертификации, CA), а можно и не настраивать, если конфигурации тестовая, и доступ извне будет ограничен. В этом случае сертификат будет самоподписанным.

Шаг 3. Запуск скрипта развертывания (Ansible Playbook)

Когда все файлы конфигурации будут настроены, следует перейти в корень репозитория (ti-deploy). Рядом должен находиться файл ti-install.yml. Для запуска скрипта развертывания следует выполнить в консоли команду:

$ ansible-playbook ti-install.yml --extra-vars "@config/192.168.113.60/config.yml" -i 192.168.113.60, -u username --ask-become-pass

Описание аргументов команды:

– @config/192.168.113.60/config.yml – это путь до файла config.yml вашей конфигурации. 192.168.113.60 – это каталог конфигурации. Следует обратить внимание на знак @ в начале пути;

– 192.168.113.60, - это адрес сервера для доступа по SSH (Docker-хост). Следует обратить внимание на знак , в конце.

– username - это имя удаленного пользователя.

Если ваш удаленный пользователь root, то можно сократить команду до:

$ ansible-playbook ti-install.yml --extra-vars "@config/192.168.113.60/config.yml" -i 192.168.113.60, -u root

Начнется процесс развертывания TI-сервера. В начале может потребоваться ввести пароль для доступа по SSH. Введите пароль и нажмите Enter.

Когда Ansible закончит работу, следует перейти в окно веб-браузера, набрать адрес сервера, и проверить подключение, затем выполнить вход в систему с логином и паролем: admin@admin.ru/defaultadminpassword1234567, и поменять пароль по умолчанию.

Сервер готов к эксплуатации.

Шаг 4 (опциональный). Обновление сервера

Если требуется обновить TI-сервер (например, вы изменили версию компонента в config.yml), то выполните команду выше, но замените ti-install.yml на ti-update.yml. Остальную часть команды менять не нужно.

Пример:

$ ansible-playbook ti-update.yml --extra-vars "@config/192.168.113.60/config.yml" -i 192.168.113.60, -u username --ask-become

3. Перечень сокращений

Основные сокращения, указанные в документе, представлены в таблице 4.

Таблица 4 – Перечень сокращений

АО	Акционерное общество
БД	База данных
ОС	Операционная система
VT	Virus Total
JWT	JSON Web Tokens
TI	Threat Intelligence

Цитирование документа допускается только со ссылкой на настоящее руководство. Руководство не может быть полностью или частично воспроизведено, тиражировано или распространено без разрешения АО «РТ-Информационная безопасность».