

RT Protect TI

Руководство по установке и эксплуатации

Версия 1.0.11 от 07 мая 2026

Разработано компанией АО «РТ-Информационная безопасность»



1. Общие положения	3
1.1 Назначение и общая информация	3
1.2 Идентификация документа	4
1.3 Аннотация документа	4
2. Описание процедуры установки	5
2.1 Общие положения	5
2.2 Требования к среде функционирования	5
2.3 Инструкция по развертыванию	6
2.3.1. Развертывание сервера через docker-hub (онлайн вариант установки)	6
Шаг 1. Подготовка окружения	6
Шаг 2. Создание конфигурации сервера	7
Шаг 3. Запуск скрипта развертывания (Ansible Playbook)	18
Шаг 4 (опциональный). Обновление сервера	20
2.3.2. Развертывание сервера с помощью предоставленного архива (офлайн вариант установки)	20
2.4 Активация/деактивация используемых анализаторов или провайдеров	21
3. Мониторинг состояния компонентов сервера	23
4. Настройка конфигурации сервера с помощью конфигуратора	25
5. Перечень сокращений	26
6. Заключение	27

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.

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 1.0.11
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

В рамках пилотного проекта рекомендуется использовать Ubuntu 20.04 или Ubuntu 22.04. Для серверной части программы допустимо использование и других ОС семейства Linux в регулярном (не тестовом) формате применения RT Protect TI.

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	32 ГБ
Жесткий диск (свободное пространство)	1 ТБ	2 ТБ

Процессор сервера, на котором будет установлен RT Protect TI, должен поддерживать набор инструкций AVX.

Проверку на наличие инструкций можно выполнить с помощью команды: `cat /proc/cpuinfo | grep avx`

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

Развертывание сервера RT Protect TI пользователям по модели поставки On-Premise можно произвести двумя способами:

- 1) Онлайн через docker-hub;
- 2) Оффлайн из архива.

2.3.1. Развертывание сервера через docker-hub (онлайн вариант установки).

Шаг 1. Подготовка окружения

Для развертывания сервера необходимо, чтобы на компьютере с ОС Linux 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 может потребоваться установка дополнительной утилиты sshpass.

Для того чтобы добавить открытый ssh-ключ на сервер требуется выполнить команды:

- ssh-keygen -t rsa -b 4096
- ssh-copy-id <USERNAME>@<HOST>

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

Проверку доступа до реестра докер-образов можно выполнить, набрав одну из следующих команд:

- curl -v https://docker.rt-protect.ru
- nc -zv docker.rt-protect.ru

- `ncat -zv docker.rt-protect.ru`
- `wget -v --spider https://docker.rt-protect.ru`
- `telnet docker.rt-protect.ru`

Шаг 2. Создание конфигурации сервера

Чтобы создать конфигурацию сервера, необходимо копировать комплект установочных файлов на АРМ, из которого будет запускаться установка.

Далее необходимо перейти в корень репозитория `ti-deploy`.

Структура каталогов представлена на рисунке 1.

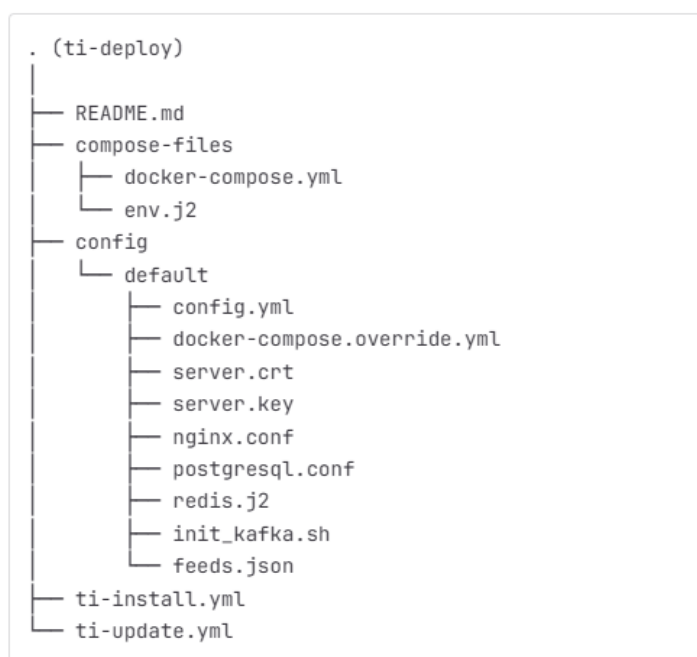


Рисунок 1 – Структура каталогов репозитория `ti-deploy`

В каталоге `config` хранятся конфигурации серверов.

Свою конфигурацию требуется добавить именно в каталог `config`.

Правильным подходом будет держать все свои конфигурации в одном месте, в каталоге `config` в соответствующих подкаталогах. Можно одновременно управлять несколькими конфигурациями серверов. Именовывать подкаталоги удобно, например, по IP-адресу сервера или домену.

Пример структуры каталога `config` с дефолтной конфигурацией и добавленными конфигурациями, представлена на рисунке 2.

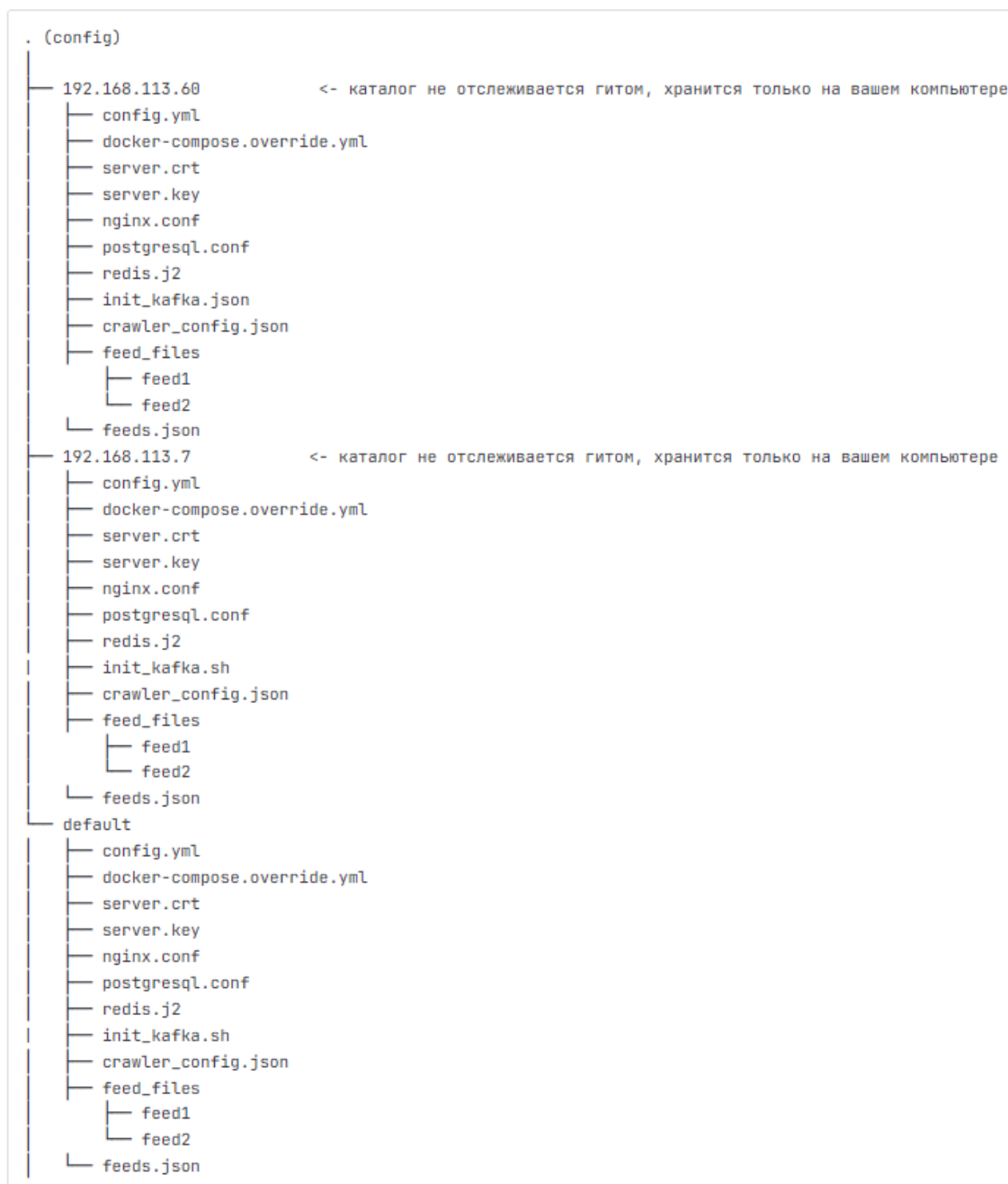


Рисунок 2 – Структура каталога config

Описанные в конфигурационных файлах параметры являются изменяемыми и даны в данном руководстве для примера.

Далее следует создать свой подкаталог в каталоге config, и скопировать в него содержимое каталога config/default.

Описание содержимого каталога config представлено в списке:

- 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;
- feeds.json — JSON-файл с источниками данных по умолчанию для указанного сервера;
- redis.j2 — конфигурационный файл Redis (при отсутствии необходимости лучше оставить по умолчанию);
- init_kafka.sh - скрипт для инициализации Kafka (при отсутствии необходимости лучше оставить по умолчанию);
- crawler_config.json - JSON-файл с источниками для получения статей по умолчанию для указанного сервера;
- feed_files - вложенный каталог, содержащий набор файлов источников данных. Используется совместно с файлом feeds.json;
- nginx.conf — конфигурационный файл NGINX (при отсутствии необходимости лучше оставить по умолчанию);
- postgresql.conf — конфигурационный файл PostgreSQL (при отсутствии необходимости лучше оставить по умолчанию).

Файл конфигурации config.yml не требуется настраивать вручную, для удобства настройки юыл разработан инструмент Конфигуратор (см. раздел 4 данного руководства). С помощью Конфигуратора настройка осуществляется быстро, но если требуется более подробное описание параметров то можно воспользоваться описанием порядка настройки приведенным в таблице 4.

Таблица 4 – Описание порядка настройки параметров в файле config.yml

№ п.п	Параметр	Описание	Пример значения	Рекомендации
1	config_name:	Название конфигурации	"default"	Должно совпадать с названием каталога, содержащего данный конфигурационный файл.
2	host_ip:	IP-адрес целевой машины	31.13.33.68	Требуется указать IP машины, на которую ставится RT Protect TI.
3	home_url:	URL, по которому будет	"https://ti.rt-protect.ru/"	Вместо доменного имени в URL может быть

		открываться главная страница приложения		указан IP-адрес машины на которую ставится RT Protect TI.
4	insecure:	Признак того, что сервер будет работать с самоподписным сертификатом	"false"	Установить в true, если доступ к платформе будет осуществляться через самоподписной сертификат.
5	internal_token:	Токен для внутренних API- вызовов	"KaDDUNn4jjqd5G8JvFvt Dg0VeSfyMN"	Рекомендуется заменить строку на аналогичную случайную.
6	default_culture:	Локализация сервера по умолчанию.	"ru-RU"	Поддерживается 2 варианта: "ru-RU" и "en-US"
7	detect_worker_count:	Количество воркеров для фиксации обнаружений	10	Рекомендуется оставить значение по умолчанию.
Настройки подключения к реестру Docker-образов				
8	docker_registry:		https://docker.rt- protect.ru	Образы хранятся в реестре по адресу https://docker.rt- protect.ru/containers Данные настройки актуальны только при установке через репозиторий. В значении параметра требуется указать переданные данные от учетных записей в сообщении для docker- репозитория.
9	docker_registry_user:		"user"	
10	docker_registry_password:		"pass"	
Параметры конфигурации PostgreSQL				
11	db_user_name:	Имя пользователя для работы с БД PostgreSQL.	"ti_server_user"	Если не планируется подключения в БД извне, можно использовать любую комбинацию имени и пароля (лучше не по умолчанию).
12	db_user_password:	Пароль для работы с БД PostgreSQL.	"pass"	

13	db_files_path:	Путь до каталога, в котором БД будет хранить данные	"/media/ti/database"	Требуется, чтобы на разделе, на который указывает этот путь, было достаточно места для хранения данных (согласно системным требованиям)
14	db_backup_path:	Путь до каталога, куда будут складываться бекапы БД	"/media/ti/db_backup"	Требуется, чтобы на разделе, на который указывает этот путь, было достаточно места для хранения данных (согласно системным требованиям)
Параметры конфигурации JWT				
15	jwt_token_secret:	Секрет, на основе которого будут подписываться выпущенные токены.	"хувfmwehrpqwrkcmihuujytiaiwatbjouk"	Лучше делать сложным и уникальным. Минимальная длина - 32 символа
16	jwt_token_issuer:		"ti.rt-protect.ru"	Эту настройку и следующие настройки можно оставить по умолчанию.
17	jwt_token_audience:		"ti.rt-protect.ru"	
18	jwt_token_expiration:	Время устаревания JWT токена в минутах.	"120"	Чем меньше, тем безопаснее для сервера.
19	refresh_token_expiration:	Время устаревания refresh токена в минутах.	"10080"	Если не заходить на сайт в течении этого времени, потребуется повторный вход в систему.
Параметры конфигурации сервера				
20	detect_enabled:	Признак включения записи "Активность".	"true"	Если false, то страница "Активность" обновляться не будет.
Параметры конфигурации MongoDB				

21	mongo_user_name:	Логин для инициализации MongoDB.	"root"	Нужно поменять значения по умолчанию
22	mongo_user_password:	Пароль для инициализации MongoDB.	"pass"	
23	storage_path:	Путь, где MongoDB будет хранить свои данные	"/media/ti/storage"	Требуется, чтобы на разделе, на который указывает этот путь, было достаточно места для хранения данных (согласно системным требованиям)
24	mongo_version:	Версия MongoDB	"7.0.5"	Оставить по умолчанию
Параметры конфигурации Redis				
25	redis_password:	Пароль Redis	"zWQaR1uNF8fcbI3XsQho6pIV"	Поменять со значения по умолчанию.
26	redis_path:	Путь до хранилища Redis	"/media/ti/redis"	Требуется, чтобы на разделе, на который указывает этот путь, было достаточно места для хранения данных (согласно системным требованиям).
Параметры конфигурации Kafka				
27	zk_data_path:		"/media/ti/zk-data"	Требуется, чтобы на разделе, на который указывает этот путь, было достаточно места для хранения данных (согласно системным требованиям).
28	zk_logs_path:		"/media/ti/zk-txn-logs"	
29	kafka_data_path:		"/media/ti/kafka-data"	
Параметры конфигурации MinIO				
30	minio_path:		"/media/ti/minio"	Требуется, чтобы на разделе, на который указывает этот путь, было достаточно места для хранения данных (согласно системным требованиям).

31	minio_access_key:		"7TjqfXUAsG2eTtK2L45U"	Поменять со значения по умолчанию на случайную строку аналогичной или большей длины.
32	minio_secret_key:		"rWf8E62JXi61yHDubhP5"	Поменять со значения по умолчанию на случайную строку аналогичной или большей длины.
Параметры SMTP				
33	smtp_host:	Хост сервера SMTP	"smtp.yandex.ru"	Это и далее заполнять если smtp_enabled: true
34	smtp_port:	Порт сервера SMTP	"465"	
35	smtp_username:	Имя пользователя для авторизации на сервере SMTP	"username"	
36	smtp_password:	Пароль для авторизации на сервере SMTP	"password"	
37	smtp_use_ssl:	Флаг использования ssl для доступа к SMTP серверу	true	
38	smtp_enabled:	Признак включения интеграции с сервером SMTP	false	
39	smtp_from_name:	Имя в поле from для отправляемых сообщений	"RT Protect TI"	Это и далее заполнять если smtp_enabled: true
40	smtp_from_address:	Адрес отправителя сообщений	"ti@rt-ib.ru"	

Параметры работы с уязвимостями				
41	vulnerabilities_limit:	Лимит на количество уязвимостей из базы, с которыми будут проверяться компоненты	-1	По умолчанию установлено: -1 (-1, значит без ограничения). Рекомендуется оставить и не изменять.
42	vuln_sync_enabled:	Включение синхронизации уязвимостей с NVD и БДУ	true	
43	nvd_client_api_key:	Токен доступа к NVD	"key"	Требуется если vuln_sync_enabled: true
Параметры работы с модулем TI Reports				
44	report_enabled:	Флаг активации модуля ленты угроз	true	
45	report_worker_url:	URL до сервиса TI Reports	"https://ml-rt-ti-reports.rt-ib.ru"	
46	report_api_key:	Ключ для доступа к сервису TI Reports	"key"	
Настройки версий компонентов				
47	back_version:			Набор версий компонентов требуется взять из текущего файла config.yml в папке default, там находится самый актуальный набор версий на момент релиза. При обновлении сервера нужно заменить СРАЗУ ВСЕ номера версий на вновь выданные в дистрибутиве обновления.

	front_version:			
48	detect_worker_version:			
49	migrator_version:			
50	feed_worker_version:			
51	feed_downloader_worker_version:			
52	vuln_worker_version:			
53	vuln_downloader_worker_version:			
54	vuln_engine_worker_version:			
55	yara_update_worker_version:			
56	analyzer_worker_version:			
57	crawler_worker_version:			
58	report_worker_version:			
Настройки подключенных анализаторов				
59	kaspersky_api_key:		"key"	В данных полях заполняется значение ключей для анализаторов полученные от соответствующих вендоров
60	vt_api_key:		"key"	
61	netlas_api_key:		"key"	
62	rst_cloud_api_key:		"key"	
63	pt_sandbox_api_key:		"key"	
64	pt_sandbox_server:	Адрес сервера PT Sandbox	"https://localhost"	
65	pt_sandbox_enabled:	Флаг включения поведенческого анализа	true	

66	pt_sandbox_duration:	Длительность выполнения поведенческого анализа (если pt_sandbox_enabled: true)	180	
67	pt_sandbox_image_id:	Идентификатор образа виртуальной машины, на которой будет выполняться поведенческий анализ (если pt_sandbox_enabled: true)	"win7-sp1-x64"	
68	meta_defender_api_key:		"key"	Ключи, логины и пароли должны быть получены у соответствующих вендоров
69	malshare_api_key:		""	
70	geo_ip_login:		""	
71	geo_ip_password:		""	
72	abuse_ipdb_api_key:		""	
73	alien_vault_api_key:		""	
74	kaspersky_analyzer_version:		"1.0.7"	
75	vt_analyzer_version:		"3.0.19"	
76	netlas_analyzer_version:		"1.0.9"	
77	rst_cloud_analyzer_version:		"1.0.12"	
78	static_analyzer_version:		"3.0.5"	
79	yara_analyzer_version:		"3.0.2"	
80	fuzzy_family_analyzer_version:		"1.0.9"	
81	pt_sandbox_version:		"1.0.3"	

82	meta_defender_analyzer_version:		"1.0.4"	
83	geo_ip_analyzer_version:		"1.0.2"	
84	abuse_ipdb_analyzer_version:		"1.0.1"	
85	alien_vault_analyzer_version:		"1.0.6"	
86	malshare_analyzer_version:		"1.0.1"	
Настройки подключенных провайдеров данных				
87	fincert_login:		"login"	
88	fincert_password:		"password"	
89	fincert_base_url:		"http://127.0.0.1:3456"	
90	pt_feeds_api_key:		"api_key"	
Названия коллекций PT Feeds				
91	pt_feeds_collections:		["current-act"]	
92	fincert_version:		"1.0.5"	
93	pt_feeds_version:		"1.0.1"	
94	meta_defender_provider_version:		"1.0.4"	
Пути на сервере (на Docker-хосте)				
95	compose_dir:	Путь до каталога с файлами docker-compose.yml и env	./rt-protect/ti	Определяют местоположение файлов, которые передаются на сервер. Без острой необходимости менять их не рекомендуется.

Примечание к таблице:

В таблице имеется цветовая идентификация параметров, описанная следующего списка:

– **config_name** (параметры, обозначенные красным), обязательные для установки вручную (или валидации значения по умолчанию).

– **kaspersky_api_key** (параметры, обозначенные оранжевым) – переменные желательные для настройки, но значения по умолчанию будет достаточно для работы сервера.

– **mongo_user_name** (параметры, обозначенные зеленым) – переменные, которые можно оставить по умолчанию или поменять при желании/необходимости.

– **vt_analyzer_version** (параметры, обозначенные черным) – номера версий и те параметры, которые крайне не рекомендуется менять при нормальной работе сервера.

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 собственной конфигурации, отредактировав файл `docker-compose.override.yml`. Такие изменения не затронут другие конфигурации.

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.

Опционально в каталог `config` можно положить файл `feeds.json` из стандартной конфигурации, либо полученный с другого TI сервера. Если такой файл присутствует, при выполнении `ti-install` Ansible-скрипт выполнит инициализацию таблицы источников данных тем наборов источников, который содержится в файле `feeds.json`.

Вместе с файлом `feeds.json` в каталоге `config` может находиться подкаталог `feed_files`, в котором должны находиться файлы, используемые по умолчанию при инициализации списка источников данных. Имя файла должно строго совпадать с именем источника.

Также опционально в каталоге располагается файл `crawler_config.json`. Он используется для первичной инициализации страницы "Источники" внутри Ленты угроз.

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.

Дополнительно можно ввести проверку, что Ansible может достучаться до хоста, выполнив команду:

```
ansible all -i <HOST>, -m ping -u <USERNAME> -k
```

В случае, если используется нестандартный SSH порт при установке, требуется выполнить 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 -e "ansible_port=2222"
```

В случае, если нужно использовать sshpass, в команду по установке ещё нужно вписать флаги –kK:

```
ansible-playbook ti-install.yml --extra-vars "@config/192.168.113.60/config.yml" -i 192.168.113.60, -u username -kK
```

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
```

2.3.2. Развертывание сервера с помощью предоставленного архива (офлайн вариант установки)

Для развертывания сервера по офлайн варианту установки на АРМ с отсутствием подключения к интернету, пользователь должен выполнить следующие шаги:

1) Подготовить окружение.

Необходимо, чтобы на компьютере с ОС Linux было установлено следующее программное обеспечение:

- Python (не ниже версии 3.10.0) [\[установка\]](#).

- Ansible (не ниже версии 5.7.1) [\[установка\]](#).

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

2) Скопировать на АРМ с носителя, файлы установочного архива переданного в рамках поставки, в составе:

- compose-files;

- config/default;

- files;

- .gitignore;

- Redme.md;

- ti-install-offline.yml

3) В каталоге config добавить свою конфигурацию.

4) Скопировать в папку с собственной конфигурацией содержимое каталога config/default.

5) Выполните в консоли команду:

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

Примечание: Для успешной offline-установки требуется наличие актуальных образов всех используемых контейнеров в папке `./files/images/`.

Версии компонентов в файле `config.yml` должны совпадать с фактическими версиями имеющихся контейнеров.

Обновление платформы в режиме Offline.

Для обновления TI платформы в offline режиме, требуется запросить у поставщика платформы новый файл с установочным архивом, в котором будут присутствовать новые образы.

Заменить файлы docker образов и в файле `config.yml` прописать актуальные версии компонентов.

Далее выполнить установку запустив скрипт установки, описанный выше.

2.4 Активация/деактивация используемых анализаторов или провайдеров

Для активации/деактивации анализатора/провайдера необходимо при обновлении RT Protect TI в папке `config/default` в файле `config.yml` в соответствующих разделах установить параметр `true/false`.

По умолчанию для анализаторов/провайдеров, которые требуют дополнительной настройки установлен параметр `false`. Требуется установить параметр `true` и настроить требуемые поля, например токены доступа.

Для анализаторов которые не требуют дополнительных настроек параметр `true` предустановлен.

Текущий список имен контейнеров анализаторов:

- `static_analyzer`;
- `vt_analyzer`;
- `kaspersky_analyzer`;
- `yara_analyzer`;
- `fuzzy_family_analyzer`;
- `rst_cloud_analyzer`;
- `netlas_analyzer`;
- `pt_sandbox_analyzer`;
- `meta_defender_analyzer`;

- geo_ip_analyzer;
- abuse_ipdb_analyzer;
- alien_vault_analyzer;
- malshare_analyzer;
- malware_bazaar_analyzer.

Текущий список имен контейнеров провайдеров:

- fincert_provider
- pt_feeds_provider
- meta_defender_provider
- malware_bazaar_provider
- x_ti_provider

Списки провайдеров и анализаторов расширяются, полный список можно увидеть в актуальной документации, а имена контейнеров можно посмотреть в файле `compose-files/docker-compose.yml`.

3. Мониторинг состояния компонентов сервера

Администратор может просматривать состояние компонентов сервера (контейнеров) в реальном времени.

Для этого после развертывания сервера необходимо настроить сервис мониторинга Gatus и Dozzle.

Для настройки компонентов Gatus и Dozzle следует выполнить следующие действия, описанные ниже.

1) Скопировать (либо запросить у производителя) репозиторий ti-monitoring на целевую машину, где установлен сервер. Перейти в каталог репозитория.

2) Заполнить конфигурацию файла gatus.env для настройки системы мониторинга Gatus.

Параметры для заполнения:

- SCHEME=https - схема обращения к серверу TI;
- HOST=ti-dev.rt-protect.ru - Адрес сервера TI;
- INSECURE=false - Если true, обращаться без проверки сертификата;
- PRODUCT_TYPE=TI DEV - Тип продукта (используется в веб интерфейсе Gatus и в алертах);

Для того чтобы сообщения (алерты) о неисправности компонентов приходили в Telegram настроить следующие параметры:

- TELEGRAM_ENABLED=false;
- TELEGRAM_BOT_TOKEN= - Токен бота выпускается через бота @BotFather;
- TELEGRAM_USER_ID= - Идентификатор чата или канала, в который будут отправляться алерты.

Необходимо создать канал, затем добавить в него ранее созданного бота в качестве админа. После этого при помощи запроса <https://api.telegram.org/bot<токен вашего бота>/getUpdates> в браузере можно получить идентификатор группы в объекте вида {"chat":{"id":-4880480591,"title"...`

Для того чтобы сообщения (алерты) о неисправности компонентов приходили на email, настроить следующие параметры:

- EMAIL_ENABLED=false
- EMAIL_HOST=
- EMAIL_FROM=
- EMAIL_USERNAME=
- EMAIL_PASSWORD=

– EMAIL_TO=

3) Сделать файлы run-dozzle.sh и run-gatus.sh исполняемыми выполнив команды:

```
chmod +x run-gatus.sh
```

```
chmod +x run-dozzle.sh,
```

4) Выполнить запуск с помощью команд:

```
./run-dozzle.sh и run-gatus.sh.
```

5) После этого на порту 8888 машины где установлен сервер будет доступен интерфейс Dozzle, позволяющий просматривать логи всех запущенных на машине контейнеров в реальном времени. На порту 9090 будет запущен сервис Gatus, выполняющий healthcheck всех компонентов TI. Данный сервис также будет отправлять алерты в Telegram и на почту, если такой функционал настроен.

4. Настройка конфигурации сервера с помощью конфигуратора

Для удобства настройки файла конфигурации сервера `config.yml` можно воспользоваться приложением «Конфигуратор», открыв в браузере файл `configurator.html`.

Общий вид страницы по настройке файла конфигурации представлен на рисунке 3.

Конфигуратор RT Protect TI

+ Создать файл | Открыть файл | Сохранить | Сохранить как

Редактор конфигурации

Предпросмотр YAML

Основные настройки

Имя конфигурации

Должно совпадать с названием каталога, содержащего данный конфигурационный файл!

IP-адрес целевой машины

URL, по которому будет открываться главная страница приложения

<https://ti.rt-protect.ru/>

☒ Доступ через самоподписной сертификат

Локализация сервера по умолчанию

Токен для внутренних API-вызовов

Рекомендуется оставить текущую строку

Настройки подключения к реестру Docker-образов

Путь до реестра Docker-образов

`docker.rt-protect.ru:5000`

Имя пользователя для доступа к реестру

`user`

Пароль для доступа к реестру

`pass`

Конфигурация PostgreSQL

Имя пользователя для работы с БД PostgreSQL

`ti_server_user`

Пароль для работы с БД PostgreSQL

Не заполнены обязательные поля

Рисунок 3 – Общий вид страницы Конфигуратор RT Protect TI

5. Перечень сокращений

Основные сокращения, указанные в документе, представлены в таблице 5.

Таблица 5 – 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».