Sandbox версия 5.7

Руководство разработчика

Дата редакции документа: 18.01.2024

Содержание

1. Об этом документе

Руководство разработчика содержит информацию о доступных функциях сервиса публичного API, форматах сообщений syslog, отправляемых во внешние системы, и способах создания дополнительных образов виртуальных машин для поведенческого анализа файлов.

Руководство адресовано специалистам, выполняющим интеграцию PT Sandbox со сторонними системами.

Комплект документации PT Sandbox включает в себя следующие документы:

— Этот документ.

— Руководство администратора — содержит справочную информацию и инструкции по установке, настройке и администрированию продукта.

— Руководство специалиста по безопасности — содержит сценарии использования продукта для управления событиями информационной безопасности.

— Руководство пользователя — содержит инструкции по отправке файлов на проверку через интерфейс продукта или по электронной почте и просмотру результатов проверки.

В этом разделе

1.1. Условные обозначения

В документе приняты условные обозначения.

Таблица 1. Условные обозначения

Пример	Описание
Внимание! При выключении модуля снижается уровень защищенности сети	Предупреждения. Содержат информацию о действиях или событиях, которые могут иметь нежелательные последствия
Примечание. Вы можете создать дополнительные отчеты	Примечания. Содержат советы, описания важных частных случаев, дополнительную или справочную информацию, которая может быть полезна при работе с продуктом
► Чтобы открыть файл:	Начало инструкции выделено специальным значком
Нажмите кнопку **OK**	Названия элементов интерфейса (например, кнопок, полей, пунктов меню) выделены полужирным шрифтом

Пример	Описание
Выполните команду `Stop-Service`	Текст командной строки, примеры кода, прочие данные, которые нужно ввести с клавиатуры, выделены специальным шрифтом. Также выделены специальным шрифтом имена файлов и пути к файлам и папкам
Ctrl+Alt+Delete	Комбинация клавиш. Чтобы использовать комбинацию, клавиши нужно нажимать одновременно
<Название программы>	Переменные заключены в угловые скобки

1.2. Другие источники информации о PT Sandbox

Вы можете найти дополнительную информацию о PT Sandbox на портале технической поддержки.

Портал содержит статьи базы знаний, новости обновлений продуктов Positive Technologies, ответы на часто задаваемые вопросы пользователей. Для доступа к базе знаний и всем новостям нужно создать на портале учетную запись.

Если вы не нашли нужную информацию или решение проблемы самостоятельно, обратитесь в службу технической поддержки (см. раздел 6).

2. Что нового в версии 5.7

Ниже приводится список изменений, которые появились в PT Sandbox.

Поддержка типа содержимого multipart/form-data при загрузке файлов на проверку через API

При проверке файлов через публичный API теперь поддерживается загрузка файлов в PT Sandbox не только в бинарном виде, но и в виде содержимого с типом `multipart/form-data`. Вы можете использовать этот тип содержимого в POST-запросах для загрузки файла при проверке с передаваемыми параметрами `/storage/uploadScanFile` и для запуска проверки файла с выбранными параметрами `/scan/checkFile`.

3. Публичный API

Взаимодействие между вашим приложением и публичным API продукта происходит по протоколу HTTPS. Метод запросов — POST (запрос `maintenance/checkHealth` допускается также отправлять методом GET).

Корневой URL API:

```
https://<Адрес PT Sandbox>/api/v1
```

Например:

```
https://sandbox.example/api/v1
```

Любые параметры должны передаваться в теле запроса в формате JSON.

В ответ на запросы вашего приложения сервис возвращает сообщения в формате JSON.

В этом разделе

3.1. Управление токенами доступа

Любой запрос к API должен содержать токен доступа в поле заголовка X-API-Key. Токен доступа — это уникальная последовательность символов, которая используется для защиты публичного API от несанкционированного доступа.

Сгенерировать токен доступа можно с помощью команды (см. раздел 3.1.1) или через веб-интерфейс PT Sandbox. Вы можете генерировать неограниченное количество токенов доступа, просматривать список сгенерированных токенов, удалять и обновлять их. Управление токенами доступа через веб-интерфейс описано в Руководстве администратора.

Название токена доступа отображается в веб-интерфейсе в списке заданий и в карточке задания рядом с названием источника для проверки.

В этом разделе

Просмотр списка токенов доступа (см. раздел 3.1.3)

Обновление токена доступа (см. раздел 3.1.4)

3.1.1. Генерация токена доступа

► Чтобы сгенерировать токен доступа,

выполните команду:

```
sudo ptmsctl api auth create <Название токена доступа>
```

Например:

```
sudo ptmsctl api auth create apiservice
```

Примечание. Название токена доступа должно быть уникальным и содержать от 5 до 30 символов, среди которых могут быть только буквы латинского алфавита, цифры и дефис. Первым и последним символами могут быть только буквы.

Вы можете также добавить комментарий длиной не более 500 символов и параметр для вывода информации о сгенерированном токене в формате JSON:

```
sudo ptmsctl -o json api auth create -c "<Комментарий>" <Название токена доступа>
```

Например:

```
sudo ptmsctl -o json api auth create -c "Создан для приложения CheckService v1.0"
apiservice
```

Появится информация о сгенерированном токене доступа в виде таблицы или в виде JSON-объекта:

```
{
  "result": {
    "comment": "<Комментарий>",
    "name": "<Название токена доступа>",
    "token": "<Токен доступа>",
    "ts": <Временная метка создания токена>
  }
}
```

Токен доступа сгенерирован.

Внимание! Сохраните токен доступа.

3.1.2. Удаление токена доступа

Вы можете удалить токен доступа, который был создан по ошибке или перестал быть нужным.

► Чтобы удалить токен доступа,

выполните команду:

```
sudo ptmsctl api auth delete <Название токен доступа>
```

Например:

```
sudo ptmsctl api auth delete apiservice
```

Появится таблица с названием токена доступа и строкой `deleted` в столбце `result`.

3.1.3. Просмотр списка токенов доступа

- Чтобы просмотреть список токенов доступа в формате таблицы,

  выполните команду:

  ```
  sudo ptmsctl api auth list
  ```

  Появится таблица со списком токенов.

- Чтобы просмотреть список токенов доступа в формате JSON,

  выполните команду:

  ```
  sudo ptmsctl -o json api auth list
  ```

  Появится список токенов. Каждый токен будет записан в следующем формате:

  ```
  {
    "comment": "<Комментарий>",
    "name": "<Название токена доступа>",
   "token": "***",
    "ts": <Временная метка создания токена>
  }
  ```

3.1.4. Обновление токена доступа

В целях безопасности вы можете обновить используемый вашим приложением токен доступа, например если он был скомпрометирован.

- Чтобы обновить токен доступа,

  выполните команду:

  ```
  sudo ptmsctl api auth update <Название токена доступа>
  ```

  Например:

  ```
  sudo ptmsctl api auth update apiservice
  ```

  Вы можете также обновить комментарий и добавить параметр для вывода информации об обновленном токене в формате JSON:

  ```
  sudo ptmsctl -o json api auth update -c "<Новый комментарий>" <Название токена>
  ```

  Например:

  ```
  sudo ptmsctl -o json api auth update -c "Создан для приложения CheckService v2.0"
  apiservice
  ```

Появится информация об обновленном токене доступа в виде таблицы или в виде JSON-объекта:

```
{
  "result": {
    "comment": "<Комментарий>",
    "name": "<Название токена доступа>",
    "token": "<Обновленный токен доступа>",
    "ts": <Временная метка создания токена>
  }
}
```

Токен доступа обновлен.

Внимание! Сохраните обновленный токен доступа.

3.2. Проверка файлов и ссылок с передаваемыми параметрами

В этом разделе приводится описание запросов для отправки файлов и ссылок на проверку в PT Sandbox с передаваемыми параметрами и получения результатов проверки при помощи публичного API.

В этом разделе

3.2.1. Загрузка файла в PT Sandbox для последующей проверки

Перед запуском проверки файла с помощью публичного API ваше приложение должно загрузить этот файл в PT Sandbox.

Метод и URL запроса:

```
POST <Корневой URL API>/storage/uploadScanFile
```

Параметры запроса отсутствуют.

Файл нужно передавать в бинарном виде (MIME-тип — application/octet-stream) или через multipart/form-data. Каждый запрос может содержать только один файл.

Ответ на запрос

В ответ на успешный запрос сервис публичного API присылает сообщение с кодом 200. Тело ответного сообщения может содержать поля, описанные в таблице ниже.

Таблица 2. Поля в ответе на запрос на загрузку файла

Поле	Тип	Описание
`file_uri`	String	URI файла во внутреннем формате. URI можно использовать до удаления файла из хранилища (см. описание поля `ttl` ниже)
`ttl`	Integer	Время жизни файла в секундах после его загрузки. По истечении этого времени или после запуска проверки файла (см. раздел 3.2.2) PT Sandbox удаляет файл из хранилища
`errors`	Array of JSON objects	Ошибки, возникшие при выполнении запроса. Каждая ошибка описывается по схеме error (см. раздел 3.7.4)

Возможные коды ошибок и их значения:

— 401 — в запросе нет токена доступа (см. раздел 3.1) или он указан неверно;

— 405 — неправильно указан метод запроса.

Пример

Запрос:

```
POST /api/v1/storage/uploadScanFile
<Файл в бинарном формате>
```

Пример запроса в формате cURL:

— при передаче файла в бинарном формате:

```
curl -k https://<Адрес PT Sandbox>/api/v1/storage/uploadScanFile -H "X-API-
Key:<Сгенерированный токен доступа>" -d @test.txt
```

— при передаче файла через multipart/form-data:

```
curl -k https://<Адрес PT Sandbox>/api/v1/storage/uploadScanFile -H "X-API-
Key:<Сгенерированный токен доступа>" -F "file=@test.txt"
```

Ответ на успешный запрос:

```
{
  "data": {
    "file_uri": "sfm-files:///2022-04-26-10/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "ttl": 3600
  },
```

```
  "errors": []
}
```

Ответ на неуспешный запрос:

```
{
  "data": {},
  "errors": [
    {
      "message": "Authorization required",
      "type": "HTTPUnauthorized"
    }
  ]
}
```

3.2.2. Запуск проверки файла

Ваше приложение может запустить проверку файла, ранее загруженного в продукт (см. раздел 3.2.1), и в ответ получить результаты проверки и (или) идентификатор задания на проверку.

Метод и URL запроса:

```
POST <Корневой URL API>/analysis/createScanTask
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 3. Параметры тела запроса на запуск проверки файла

Параметр	Обязательный	Тип	Описание
`file_uri`	Да	String	Временный URI файла, полученный в одноименном параметре в ответе на запрос загрузки файла (см. раздел 3.2.1)
`file_name`	Нет	String	Название проверяемого файла. Если название не указано, в веб-интерфейсе PT Sandbox будет отображаться **<Файл без названия>**
`async_result`	Нет	Boolean	Возвращать только идентификатор задания на проверку. Включение этого параметра может понадобиться для отправки асинхронных запросов на проверку файлов: ваше приложение может не дожидаться результатов проверки, а получать их отдельными запросами. Такие запросы позволяют получать результаты проверки как в общем виде (см. раздел 3.2.4), так и в подробном (см. раздел 3.2.5).

Параметр	Обязательный	Тип	Описание
			Значение по умолчанию — `false`
`short_result`	Нет	Boolean	Возвращать только общий результат проверки. Значение `false` этого параметра игнорируется, если для параметра `async_result` используется значение `true`. Значение по умолчанию — `true`
`options` → `analysis_depth`	Нет	Integer	Глубина проверки. Максимальный уровень декомпозиции объектов с иерархической структурой (например, архивов, электронных писем и ссылок) или уровень декомпрессии сжатых файлов. При значении `0` проверка выполняется без декомпозиции и декомпрессии. Чем больше число, тем дольше может выполняться проверка. Допустимые значения: 0—100. По умолчанию — 0
`options` → `passwords_for_unpack`	Нет	Array of strings	Список паролей для распаковки зашифрованных архивов
`options` → `url_extract_enabled`	Нет	Boolean	Извлекать ссылки из объектов. Значение по умолчанию — `false`
`options` → `mark_suspicious_files_options`	Нет	JSON object	Дополнительные критерии определения потенциально опасных файлов
`options` → `mark_suspicious_files_options` → `encrypted_not_unpacked`	Нет	Boolean	Определять как потенциально опасные зашифрованные и нераспакованные архивы
`options` → `mark_suspicious_files_options` → `max_depth_exceeded`	Нет	Boolean	Определять как потенциально опасные архивы с превышенной глубиной распаковки

Параметр	Обязательный	Тип	Описание
options → mark_suspicious_files_options → office_encrypted	Нет	Boolean	Определять как потенциально опасные зашифрованные файлы офисных форматов
options → mark_suspicious_files_options → office_has_macros	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с макросами
options → mark_suspicious_files_options → office_has_embedded	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с внедренными объектами (OLE)
options → mark_suspicious_files_options → office_has_active_x	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с элементами ActiveX. **Примечание.** Поддерживается только для файлов Microsoft Office версии 2007 или выше
options → mark_suspicious_files_options → office_has_dde	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с функциями DDE. **Примечание.** Поддерживается только для файлов Microsoft Office версии 2007 или выше
options → mark_suspicious_files_options → office_has_remote_data	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с запросами к внешним данным. **Примечание.** Поддерживается только для файлов Microsoft Office версии 2007 или выше
options → mark_suspicious_files_opti	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов, использующие шаблон.

Параметр	Обязательный	Тип	Описание
`ons → office_has_remote_template`			**Примечание.** Поддерживается только для файлов Microsoft Office версии 2007 или выше
`options → mark_suspicious_files_options → office_has_action`	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с настроенными действиями (Actions). **Примечание.** Поддерживается только для файлов Microsoft Office версии 2007 или выше
`options → mark_suspicious_files_options → pdf_encrypted`	Нет	Boolean	Определять как потенциально опасные зашифрованные файлы PDF
`options → mark_suspicious_files_options → pdf_has_embedded`	Нет	Boolean	Определять как потенциально опасные файлы PDF с внедренными объектами (OLE)
`options → mark_suspicious_files_options → pdf_has_open_action`	Нет	Boolean	Определять как потенциально опасные файлы PDF с настроенными действиями при открытии (OpenActions)
`options → mark_suspicious_files_options → pdf_has_action`	Нет	Boolean	Определять как потенциально опасные файлы PDF с настроенными действиями (Actions)
`options → mark_suspicious_files_options → pdf_has_javascript`	Нет	Boolean	Определять как потенциально опасные файлы PDF со сценариями JavaScript

Параметр	Обязательный	Тип	Описание
`options → sandbox → analysis_duration`	Нет	Integer	Продолжительность наблюдения за файлом в ходе поведенческого анализа (в секундах). Допустимые значения: 10—600. По умолчанию — 60
`options → sandbox → analysis_duration_bootkitmon`	Нет	Integer	Продолжительность наблюдения за файлом в ходе поведенческого анализа после перезагрузки ОС (в секундах). Допустимые значения: 10—600. По умолчанию — 60
`options → sandbox → bootkitmon`	Нет	Boolean	Выполнить поведенческий анализ файла с перезагрузкой ОС. Значение по умолчанию — `false`
`options → sandbox → enabled`	Нет	Boolean	Выполнить поведенческий анализ. Значение по умолчанию — `false`
`options → sandbox → file_types`	Нет	Array of strings	Типы и группы типов проверяемых файлов. По умолчанию проверяются все поддерживаемые типы файлов — `["adobe-acrobat/", "executable-files/", "presentations/", "spreadsheets/", "word-processor/"]`
`options → sandbox → image_id`	Да, если значение `options → sandbox → enabled` — `true`	String	Идентификатор образа виртуальной машины (см. раздел 3.5)
`options → sandbox → mitm_enabled`	Нет	Boolean	Включить подмену сертификатов ПО сертификатами PT Sandbox при расшифровке и анализе защищенного трафика. Значение по умолчанию — `false`
`options → sandbox → filter_by_properties`	Нет	JSON object	Фильтрация файлов по группам типов с указанием дополнительных признаков. Архивы и файлы с указанными типами и дополнительными признаками будут отправляться на поведенческий анализ

Параметр	Обязательный	Тип	Описание
options → sandbox → filter_by_properties → pdf	Нет	Array of strings	Дополнительные признаки файлов PDF для их отправки на поведенческий анализ. Возможные значения: — ENCRYPTED — зашифрованные; — HAS_JAVASCRIPT — со сценариями JavaScript; — HAS_OPEN_ACTION — с настроенными действиями при открытии (OpenAction); — HAS_ACTION — с настроенными действиями (Actions); — HAS_EMBEDDED — с внедренными объектами (OLE)
options → sandbox → save_video	Нет	Boolean	Включить запись видео поведенческого анализа. Значение по умолчанию — true

Ответ на запрос

В случае успешного выполнения запроса публичный API возвращает ответ с кодом 200 по схеме ответа с результатами проверки (см. раздел 3.7.1).

Возможные коды ошибок и их значения:

— 401 — в запросе нет токена доступа (см. раздел 3.1) или он указан неверно;

— 404 — файл, URI которого указан в запросе, был удален из временного хранилища или этот URI указан с ошибкой;

— 405 — неправильно указан метод запроса.

Пример

Тело запроса на запуск синхронной проверки файла с выдачей подробных результатов проверки:

```
{
  "file_uri": "sfm-files:///2022-04-26-10/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "short_result": false
}
```

Тело запроса на запуск асинхронной проверки файла (с поведенческим анализом):


```
{
  "file_uri": "sfm-files:///2022-04-26-10/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "file_name": "installer.exe",
  "async_result": true,
  "options": {
    "analysis_depth": 3,
    "passwords_for_unpack": ["P@ssw0rd", "Administr@t0r", "checkpass"],
    "url_extract_enabled": false,
    "mark_suspicious_files_options": {
      "encrypted_not_unpacked": true,
      "max_depth_exceeded": true,
      "office_encrypted": true,
      "office_has_macros": true,
      "office_has_embedded": true,
      "office_has_active_x": true,
      "office_has_dde": true,
      "office_has_remote_data": true,
      "office_has_remote_template": true,
      "office_has_action": true,
      "pdf_encrypted": true,
      "pdf_has_embedded": true,
      "pdf_has_open_action": true,
      "pdf_has_action": true,
      "pdf_has_javascript": true
    },
    "sandbox": {
      "enabled": false,
      "image_id": "string",
      "file_types": [
        "adobe-acrobat/",
        "executable-files/",
        "presentations/",
        "spreadsheets/",
        "word-processor/"
      ],
      "custom_command": "string",
      "procdump_new_processes_on_finish": false,
      "analysis_duration": 60,
      "bootkitmon": false,
      "analysis_duration_bootkitmon": 60,
      "save_video": true,
      "mitm_enabled": false,
      "filter_by_properties": {
        "pdf": [
```

```
            "ENCRYPTED"
          ],
        }
      }
    }
  }
```


Ответ на успешный асинхронный запрос (с включенным параметром `async_result`):


```
{
  "data": {
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
  "errors": []
}
```


Ответ на успешный синхронный запрос с подробными результатами проверки (с выключенными параметрами `async_result` и `short_result`):


```
{
  "data": {
    "artifacts": [
      {
        "artifacts": [],
        "engine_results": [
          {
            "database_time": 1618398601.0,
            "detections": [
              {
                "detect": "Win.Test.EICAR_HDB-1",
                "threat": "VIRUS"
              }
            ],
            "engine_code_name": "clamav",
            "engine_subsystem": "AV",
            "engine_version": "0.102.2",
            "result": {
              "errors": [],
              "scan_state": "FULL",
              "threat": "VIRUS",
              "verdict": "DANGEROUS"
            }
          },
          {
            "database_time": 1617993482.0,
            "database_version": "1.0.0.552",
            "detections": [
              {
                "detect": "tool_win_ZZ_EICAR__Virus",
```

```
                "threat": "VIRUS"
              }
            ],
            "engine_code_name": "ptesc",
            "engine_subsystem": "STATIC",
            "engine_version": "0.0.0.1835+master",
            "result": {
              "errors": [],
              "scan_state": "FULL",
              "threat": "VIRUS",
              "verdict": "DANGEROUS"
            }
          }
        ],
        "file_info": {
          "file_path": "",
          "file_uri":
"sha256:215a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
          "md5": "44d88612fea8a7f36de82e1278abb02f",
          "mime_type": "text/plain; charset=us-ascii",
          "sha1": "3395856ce81f1b7382dee72602f798b642f14140",
          "sha256":
"275a021bbfb5489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
          "size": 69
        },
        "result": {
          "errors": [],
          "scan_state": "FULL",
          "threat": "VIRUS",
          "verdict": "DANGEROUS"
        },
        "type": "FILE"
      }
    ],
    "result": {
      "duration": 0.17754173884168268,
      "errors": [],
      "scan_state": "FULL",
      "threat": "VIRUS",
      "verdict": "DANGEROUS"
    },
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
  "errors": []
}
```

Ответ на неуспешный запрос:

```
{
  "data": {},
  "errors": [
    {
      "message": "File \"sfm-files:///2022-04-26-10/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\" not found",
      "type": "HTTPNotFound"
    }
  ]
}
```

3.2.3. Запуск проверки ссылки

Ваше приложение может запустить проверку ссылки и в ответ получить результаты проверки и (или) идентификатор задания на проверку.

Метод и URL запроса:

```
POST <Корневой URL API>/analysis/createScanURLTask
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 4. Параметры тела запроса на запуск проверки ссылки

Параметр	Обязательный	Тип	Описание
`url`	Да	String	Проверяемая ссылка
`async_result`	Нет	Boolean	Возвращать только идентификатор задания на проверку. Включение этого параметра может понадобиться для отправки асинхронных запросов на проверку ссылок: ваше приложение может не дожидаться результатов проверки, а получать их отдельными запросами. Такие запросы позволяют получать результаты проверки как в общем виде (см. раздел 3.2.4), так и в подробном (см. раздел 3.2.5). Значение по умолчанию — `false`
`short_result`	Нет	Boolean	Возвращать только общий результат проверки. Значение `false` этого параметра игнорируется, если для параметра `async_result` используется значение `true`. Значение по умолчанию — `true`

Параметр	Обязательный	Тип	Описание
`priority`	Нет	Integer	Приоритет выполнения проверки ссылки от 1 до 4, где 4 — самый высокий приоритет. Значение по умолчанию — 3
`options → analysis_depth`	Нет	Integer	Глубина проверки. Максимальный уровень декомпозиции объектов с иерархической структурой (например, архивов, электронных писем и ссылок) или уровень декомпрессии сжатых файлов. При значении `0` проверка выполняется без декомпозиции и декомпрессии. Чем больше число, тем дольше может выполняться проверка. Допустимые значения: 0—100. По умолчанию — 0
`options → passwords_for_unpack`	Нет	Array of strings	Список паролей для распаковки зашифрованных архивов
`options → url_extract_enabled`	Нет	Boolean	Извлекать ссылки из объектов. Значение по умолчанию — `false`
`options → mark_suspicious_files_options`	Нет	JSON object	Дополнительные критерии определения потенциально опасных файлов
`options → mark_suspicious_files_options → encrypted_not_unpacked`	Нет	Boolean	Определять как потенциально опасные зашифрованные и нераспакованные архивы
`options → mark_suspicious_files_options → max_depth_exceeded`	Нет	Boolean	Определять как потенциально опасные архивы с превышенной глубиной распаковки
`options → mark_suspicious_files_opti`	Нет	Boolean	Определять как потенциально опасные зашифрованные файлы офисных форматов

Параметр	Обязатель-ный	Тип	Описание
`ons → office_encrypted`			
`options → mark_suspicious_files_options → office_has_macros`	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с макросами
`options → mark_suspicious_files_options → office_has_embedded`	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с внедренными объектами (OLE)
`options → mark_suspicious_files_options → office_has_active_x`	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с элементами ActiveX. **Примечание.** Поддерживается только для файлов Microsoft Office версии 2007 или выше
`options → mark_suspicious_files_options → office_has_dde`	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с функциями DDE. **Примечание.** Поддерживается только для файлов Microsoft Office версии 2007 или выше
`options → mark_suspicious_files_options → office_has_remote_data`	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с запросами к внешним данным. **Примечание.** Поддерживается только для файлов Microsoft Office версии 2007 или выше
`options → mark_suspicious_files_options → office_has_remote_template`	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов, использующие шаблон. **Примечание.** Поддерживается только для файлов Microsoft Office версии 2007 или выше

Параметр	Обязательный	Тип	Описание
`options` → `mark_suspicious_files_options` → `office_has_action`	Нет	Boolean	Определять как потенциально опасные файлы офисных форматов с настроенными действиями (Actions). **Примечание.** Поддерживается только для файлов Microsoft Office версии 2007 или выше
`options` → `mark_suspicious_files_options` → `pdf_encrypted`	Нет	Boolean	Определять как потенциально опасные зашифрованные файлы PDF
`options` → `mark_suspicious_files_options` → `pdf_has_embedded`	Нет	Boolean	Определять как потенциально опасные файлы PDF с внедренными объектами (OLE)
`options` → `mark_suspicious_files_options` → `pdf_has_open_action`	Нет	Boolean	Определять как потенциально опасные файлы PDF с настроенными действиями при открытии (OpenActions)
`options` → `mark_suspicious_files_options` → `pdf_has_action`	Нет	Boolean	Определять как потенциально опасные файлы PDF с настроенными действиями (Actions)
`options` → `mark_suspicious_files_options` → `pdf_has_javascript`	Нет	Boolean	Определять как потенциально опасные файлы PDF со сценариями JavaScript
`options` → `sandbox` → `enabled`	Нет	Boolean	Выполнить поведенческий анализ. Значение по умолчанию — `false`

Параметр	Обязательный	Тип	Описание
options → sandbox → skip_check_mime_type	Нет	Boolean	Пропустить проверку MIME-типа файла по ссылке перед поведенческим анализом. Значение по умолчанию — false
options → sandbox → image_id	Да, если значение options → sandbox → enabled — true	String	Идентификатор образа виртуальной машины (см. раздел 3.5)
options → sandbox → analysis_duration	Нет	Integer	Продолжительность наблюдения за файлом в ходе поведенческого анализа (в секундах). Допустимые значения: 10—600. По умолчанию — 60
options → sandbox → bootkitmon	Нет	Boolean	Выполнить поведенческий анализ файла по ссылке с перезагрузкой ОС. Значение по умолчанию — false
options → sandbox → analysis_duration_bootkitmon	Нет	Integer	Продолжительность наблюдения за файлом в ходе поведенческого анализа после перезагрузки ОС (в секундах). Допустимые значения: 10—600. По умолчанию — 60
options → sandbox → save_video	Нет	Boolean	Включить запись видео поведенческого анализа. Значение по умолчанию — true
options → sandbox → mitm_enabled	Нет	Boolean	Включить подмену сертификатов ПО сертификатами PT Sandbox при расшифровке и анализе защищенного трафика. Значение по умолчанию — false
options → sandbox → filter_by_properties	Нет	JSON object	Фильтрация файлов по группам типов с указанием дополнительных признаков. Архивы и файлы с указанными типами и дополнительными признаками будут отправляться на поведенческий анализ

Параметр	Обязательный	Тип	Описание
options → sandbox → filter_by_properties → pdf	Нет	Array of strings	Дополнительные признаки файлов PDF для их отправки на поведенческий анализ. Возможные значения: — ENCRYPTED — зашифрованные; — HAS_JAVASCRIPT — со сценариями JavaScript; — HAS_OPEN_ACTION — с настроенными действиями при открытии (OpenAction); — HAS_ACTION — с настроенными действиями (Actions); — HAS_EMBEDDED — с внедренными объектами (OLE)
options → sandbox → file_types	Нет	Array of strings	Типы и группы типов проверяемых файлов. По умолчанию проверяются все поддерживаемые типы файлов — ["adobe-acrobat/", "executable-files/", "presentations/", "spreadsheets/", "word-processor/"]

Ответ на запрос

В случае успешного выполнения запроса публичный API возвращает ответ с кодом 200 по схеме ответа с результатами проверки (см. раздел 3.7.1).

Возможные коды ошибок и их значения:

— 401 — в запросе нет токена доступа или он указан неверно;

— 403 — некорректный номер лицензии или для переданного в запросе токена доступа среди разрешенных действий не выбрана «Проверка с передаваемыми параметрами».

Пример

Тело запроса на запуск синхронной проверки ссылки с выдачей подробных результатов проверки:

```
{
  "url": http://example.ru,
  "short_result": false
}
```

Тело запроса на запуск асинхронной проверки ссылки (с поведенческим анализом):

```
{
  "url": http://example.ru,
  "async_result": true,
  "options": {
    "analysis_depth": 3,
    "passwords_for_unpack": ["P@ssw0rd", "Administr@t0r", "checkpass"],
    "url_extract_enabled": false,
    "mark_suspicious_files_options": {
      "encrypted_not_unpacked": true,
      "max_depth_exceeded": true,
      "office_encrypted": true,
      "office_has_macros": true,
      "office_has_embedded": true,
      "office_has_active_x": true,
      "office_has_dde": true,
      "office_has_remote_data": true,
      "office_has_remote_template": true,
      "office_has_action": true,
      "pdf_encrypted": true,
      "pdf_has_embedded": true,
      "pdf_has_open_action": true,
      "pdf_has_action": true,
      "pdf_has_javascript": true
    },
    "sandbox": {
      "enabled": false,
      "image_id": "string",
      "file_types": [
        "adobe-acrobat/",
        "executable-files/",
        "presentations/",
        "spreadsheets/",
        "word-processor/"
      ],
      "custom_command": "string",
      "procdump_new_processes_on_finish": false,
      "analysis_duration": 60,
      "bootkitmon": false,
      "analysis_duration_bootkitmon": 60,
      "save_video": true,
      "mitm_enabled": false,
      "filter_by_properties": {
        "pdf": [
          "ENCRYPTED"
        ],
```

```
        }
      }
    }
  }
```

Ответ на успешный асинхронный запрос (с включенным параметром `async_result`):

```
{
  "data": {
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
  "errors": []
}
```

Ответ на успешный синхронный запрос с подробными результатами проверки (с выключенными параметрами `async_result` и `short_result`):

```
{
    "errors": [],
    "data": {
      "scan_id": "44c9cb78-74b5-11ee-8184-441ab5b46535",
        "result": {
            "scan_state": "FULL",
            "verdict": "DANGEROUS",
            "errors": [],
            "duration": 16.28870937973261,
            "threat": "VIRUS",
            "duration_full": 16.28870937973261
         },
        "artifacts": [
            {
                "artifacts": [
                    {
                        "artifacts": [],
                        "result": {
                            "scan_state": "FULL",
                            "errors": [],
                            "verdict": "DANGEROUS",
                            "threat": "VIRUS",
                            "duration": 14.614,
                            "duration_full": 14.712
                        },
                        "type": "FILE",
                        "file_info": {
                            "file_uri":
"sha256:275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
                            "file_path": "eicar.com",
                            "mime_type": "text/plain; charset=us-ascii",
                            "md5": "44d88612fea8a8f36de82e1278abb02f",
```

```
                            "sha1": "3395856ce81f2b7382dee72602f798b642f14140",
                            "sha256":
"275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
                            "size": 68
                        },
                        "engine_results": [
                            {
                                "engine_code_name": "clamav",
                                "engine_subsystem": "AV",
                                "engine_version": "0.103.8",
                                "database_time": 1698306473.0,
                                "detections": [
                                    {
                                        "detect": "Eicar-Test-Signature",
                                        "threat": "VIRUS"
                                    }
                                ],
                                "result": {
                                    "scan_state": "FULL",
                                    "verdict": "DANGEROUS",
                                    "errors": [],
                                    "duration": 0.208,
                                    "duration_full": 0.293,
                                    "threat": "VIRUS"
                                }
                            },
                            {
                                "engine_code_name": "ptesc",
                                "engine_subsystem": "STATIC",
                                "engine_version": "5.6.0.61826",
                                "database_time": 1698275624.0,
                                "detections": [
                                    {
                                        "detect": "tool_multi_ZZ_EICAR__Virus",
                                        "threat": "VIRUS"
                                    }
                                ],
                                "database_version": "4.0.0.364",
                                "result": {
                                    "scan_state": "FULL",
                                    "verdict": "DANGEROUS",
                                    "errors": [],
                                    "duration": 0.732,
                                    "duration_full": 0.8,
                                    "threat": "VIRUS"
                                }
```

```
                    },
                    {
                        "engine_code_name": "nano",
                        "engine_subsystem": "AV",
                        "engine_version": "1.0.146.91321",
                        "database_time": 1698392805.0,
                        "detections": [
                            {
                                "detect": "Marker.Dos.EICAR-Test-File.dyb",
                                "threat": "VIRUS"
                            }
                        ],
                        "result": {
                            "scan_state": "FULL",
                            "verdict": "DANGEROUS",
                            "errors": [],
                            "duration": 14.493,
                            "duration_full": 14.531,
                            "threat": "VIRUS"
                        }
                    }
                ],
                "network_objects": []
            }
        ],
        "result": {
            "scan_state": "FULL",
            "errors": [],
            "verdict": "DANGEROUS",
            "threat": "VIRUS",
            "duration": 15.14,
            "duration_full": 15.151
        },
        "type": "URL",
        "file_info": {
            "file_uri": "sha256:",
            "file_path": "https://secure.eicar.org/eicar.com",
            "mime_type": "",
            "md5": "",
            "sha1": "",
            "sha256": "",
            "size": 0
        },
        "engine_results": [],
        "network_objects": []
    }
```

```
        ]
    }
}
```

Ответ на неуспешный запрос:

```
{
    "errors": [
        {
            "type": "ERROR_BAD_REQUEST",
            "message": "1 validation error for SchemaCreateScanURLTask\nurl\n field
required (type=value_error.missing)"
        }
    ],
    "data": {}
}
```

3.2.4. Получение общего результата проверки

Ваше приложение может получить общий результат проверки файла или ссылки при помощи отдельного запроса. Запрос можно использовать, когда проверка была запущена асинхронным методом (при помощи запроса `createScanTask` или `createScanURLTask` с включенным параметром `async_result`), который не подразумевает возвращения результатов проверки.

Метод и URL запроса:

```
POST <Корневой URL API>/analysis/checkTask
```

Тело запроса должно содержать параметр `scan_id` с идентификатором задания на проверку файла или ссылки.

> **Примечание.** Ваше приложение может получить значение для параметра `scan_id` из ответа на запрос проверки файла (см. раздел 3.2.2) или запрос проверки ссылки (см. раздел 3.2.3). В этом случае идентификатор задания можно взять из поля `data` → `scan_id`.

Также тело запроса может содержать включенный параметр `allow_preflight` для получения в ответе предварительного результата проверки.

Ответ на запрос

В случае успешного выполнения запроса публичный API возвращает ответ с кодом 200 по схеме ответа с результатами проверки (см. раздел 3.7.1).

Возможные коды ошибок и их значения:

— 401 — в запросе нет токена доступа (см. раздел 3.1) или он указан неверно;

— 404 — задание на проверку файла не найдено; возможно, оно было создано более трех часов назад и завершилось с ошибкой;

— 405 — неправильно указан метод запроса.

Пример

Тело запроса:

```
{ "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d" }
```

Ответ, когда задание на проверку еще выполняется и нет предварительного результата (с включенным, выключенным или не указанным параметром `allow_preflight`):

```
{
  "data": {
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
  "errors": []
}
```

Ответ, когда задание на проверку еще выполняется и есть предварительный результат (`allow_preflight=true`):

```
{
  "data": {
    "is_preflight": true,
    "result": {
      "duration": 0.17754173884168268,
      "errors": [],
      "scan_state": "FULL",
      "threat": "VIRUS",
      "verdict": "DANGEROUS"
},
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
},
   "errors": []
}
```

Ответ, когда задание на проверку завершилось (с включенным параметром `allow_preflight`):

```
{
  "data": {
    "is_preflight": false,
    "result": {
      "duration": 0.17754173884168268,
      "errors": [],
      "scan_state": "FULL",
```

```
        "threat": "VIRUS",
        "verdict": "DANGEROUS"
      },
      "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
    },
    "errors": []
  }
```

Ответ на неуспешный запрос:

```
  {
    "errors": [
      {
        "type": "ERROR_ITEM_NOT_FOUND",
        "message": "Item \"058f1426-b920-11eb-af49-e757f9105daf\" is not found here."
      }
    ],
    "data": {}
  }
```

См. также

Получение подробных результатов проверки (см. раздел 3.2.5)

3.2.5. Получение подробных результатов проверки

Ваше приложение может получить детальные результаты проверки файла или ссылки при помощи отдельного запроса. Запрос можно использовать, когда проверка была запущена асинхронным методом (при помощи запроса `createScanTask` или `createScanURLTask` с включенным параметром `async_result`), который не подразумевает возвращения результатов проверки.

Метод и URL запроса:

```
POST <Корневой URL API>/analysis/report
```

Тело запроса должно содержать параметр `scan_id` с идентификатором задания на проверку файла или ссылки.

> **Примечание.** Ваше приложение может получить значение для параметра `scan_id` из ответа на запрос проверки файла (см. раздел 3.2.2) или на запрос проверки ссылки (см. раздел 3.2.3). В этом случае идентификатор задания можно взять из поля `data` → `scan_id`.

Ответ на запрос

В случае успешного выполнения запроса публичный API возвращает ответ с кодом 200 по схеме ответа с результатами проверки (см. раздел 3.7.1).

Возможные коды ошибок и их значения:

— 401 — в запросе нет токена доступа (см. раздел 3.1) или он указан неверно;

— 404 — задание на проверку файла не найдено; возможно, оно было создано более трех часов назад и завершилось с ошибкой;

— 405 — неправильно указан метод запроса.

Пример

Тело запроса:

```
{ "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d" }
```

Ответ, когда задание на проверку еще выполняется:

```
{
  "data": {
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
  "errors": []
}
```

Ответ, когда задание на проверку завершилось:

```
{
  "data": {
    "artifacts": [
      {
        "artifacts": [],
        "engine_results": [
          {
            "database_time": 1618398601.0,
            "detections": [
              {
                "detect": "Win.Test.EICAR_HDB-1",
                "threat": "VIRUS"
              }
            ],
            "engine_code_name": "clamav",
            "engine_subsystem": "AV",
            "engine_version": "0.102.2",
            "result": {
              "errors": [],
              "scan_state": "FULL",
              "threat": "VIRUS",
              "verdict": "DANGEROUS"
            }
          },
          {
```

```
            "database_time": 1617993482.0,
            "database_version": "1.0.0.552",
            "detections": [
              {
                "detect": "tool_win_ZZ_EICAR__Virus",
                "threat": "VIRUS"
              }
            ],
            "engine_code_name": "ptesc",
            "engine_subsystem": "STATIC",
            "engine_version": "0.0.0.1835+master",
            "result": {
              "errors": [],
              "scan_state": "FULL",
              "threat": "VIRUS",
              "verdict": "DANGEROUS"
            }
          },
          {
            "engine_code_name": "PT",
            "engine_subsystem": "SANDBOX",
            "details": {
              "sandbox": {
                "image": {
                  "image_id": "win7-sp1-x64",
                  "version": "1.0.0.144",
                  "os": {
                    "architecture": "x64",
                    "locale": "ru-RU",
                    "name": "Microsoft Windows 7 Enterprise",
                    "service_pack": "sp1",
                    "version": "6.1.7601.18741"
                  }
                },
                "artifacts": [
                ],
                "logs": [
                  {
                    "type": "EVENT_CORRELATED",
                    "file_name": "events-correlated.log.gz",
                    "file_uri":
"sha256:d88eb6f52506b8fdeffd37b98438949901055d47ad5298a99d68cbce77296d31"
                  }
                ],
                "analysis_duration": 124.0,
                "network_objects": [
```

```
                {
                  "type": "IP",
                  "value": "93.184.216.34"
                },
                {
                  "type": "DOMAIN",
                  "value": "example.com"
                },
                {
                  "type": "URL",
                  "value": "http://example.com/"
                }
              ]
            }
          }
        }
      ],
      "file_info": {
        "file_path": "",
        "file_uri":
"sha256:215a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
        "md5": "44d88612fea8a7f36de82e1278abb02f",
        "mime_type": "text/plain; charset=us-ascii",
        "sha1": "3395856ce81f1b7382dee72602f798b642f14140",
        "sha256":
"275a021bbfb5489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
        "size": 69
      },
      "network_objects": [
        {
          "type": "URL",
            "value": "https://secure.eicar.org/eicar.com"
        }
      ],
      "result": {
        "errors": [],
        "scan_state": "FULL",
        "threat": "VIRUS",
        "verdict": "DANGEROUS"
      },
      "type": "FILE"
    }
  ],
  "result": {
    "duration": 0.17754173884168268,
    "errors": [],
    "scan_state": "FULL",
```

```
        "threat": "VIRUS",
        "verdict": "DANGEROUS"
      },
      "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
    },
    "errors": []
  }
```

Ответ на неуспешный запрос:

```
  {
    "errors": [
      {
        "type": "ERROR_ITEM_NOT_FOUND",
        "message": "Item \"058f1426-b920-11eb-af49-e757f9105daf\" is not found here."
      }
    ],
    "data": {}
  }
```

См. также

Получение общего результата проверки (см. раздел 3.2.4)

3.3. Проверка файлов и ссылок с параметрами источников

В этом разделе приводится описание запросов для проверки файлов и ссылок с помощью заранее настроенных в PT Sandbox параметров проверки и получения результатов проверки при помощи публичного API.

Для использования запросов вам необходимо в веб-интерфейсе PT Sandbox создать токен доступа с требуемым разрешенным действием, а также создать источник проверки с типом «API с выбранными параметрами проверки». Создание токена доступа и источника проверки приведено в Руководстве администратора.

В этом разделе

3.3.1. Запуск проверки файла с выбранными параметрами

Ваше приложение может запустить проверку файла с заранее выбранными параметрами и в ответ получить результаты проверки и (или) идентификатор задания на проверку.

Метод и URL запроса:

```
POST <Корневой URL API>/scan/checkFile
```

Таблица 5. Параметры запроса на запуск проверки файла с выбранными параметрами

Параметр	Обязательный	Тип	Описание
`file_name`	Нет	String	Название проверяемого файла. Если название не указано, в веб-интерфейсе PT Sandbox будет отображаться **<Файл без названия>**
`short_result`	Нет	Boolean	Возвращать только общий результат проверки. Значение `false` этого параметра игнорируется, если для параметра `async_result` используется значение `true`. Значение по умолчанию — `true`
`async_result`	Нет	Boolean	Возвращать только идентификатор задания на проверку. Включение этого параметра может понадобиться для отправки асинхронных запросов на проверку файлов: ваше приложение может не дожидаться результатов проверки, а получать их отдельными запросами. Такие запросы позволяют получать результаты проверки как в общем виде (см. раздел 3.2.4), так и в подробном (см. раздел 3.2.5). Значение по умолчанию — `false`

Файл нужно передавать в бинарном виде (MIME-тип — application/octet-stream) или через multipart/form-data. Каждый запрос может содержать только один файл.

Ответ на запрос

В случае успешного выполнения запроса публичный API возвращает ответ с кодом 200 по схеме ответа с результатами проверки (см. раздел 3.7.1).

Возможные коды ошибок и их значения:

— 401 — в запросе нет токена доступа, он указан неверно или не привязан к источнику с типом «API с выбранными параметрами проверки»;

— 403 — для переданного в запросе токена доступа среди разрешенных действий не выбрана «Проверка с параметрами источника».

Пример

Тело запроса на запуск синхронной проверки файла с выдачей подробных результатов проверки:

```
POST /api/v1/scan/checkFile?
file_name=some_file.exe&short_result=true&async_restul=false&priority=1
<Файл в бинарном формате>
```

Пример запроса в формате cURL:

— при передаче файла в бинарном формате:

```
curl -k https://<Адрес PT Sandbox>/api/v1/scan/checkFile?file_name=some-name.txt -H
"X-API-Key:<Сгенерированный токен доступа>" -d @test.txt
```

— при передаче файла в multipart/form-data:

```
curl -k https://<Адрес PT Sandbox>/api/v1/scan/checkFile?file_name=some-name.txt -H
"X-API-Key:<Сгенерированный токен доступа>" -F "file=@test.txt"
```

Ответ на успешный асинхронный запрос (с включенным параметром `async_result`):

```
{
  "data": {
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
  "errors": []
}
```

Ответ на успешный синхронный запрос с подробными результатами проверки (с выключенными параметрами `async_result` и `short_result`):

```
{
  "data": {
    "artifacts": [
      {
        "artifacts": [],
        "engine_results": [
          {
            "database_time": 1618398601.0,
            "detections": [
              {
                "detect": "Win.Test.EICAR_HDB-1",
                "threat": "VIRUS"
              }
```

```
            ],
            "engine_code_name": "clamav",
            "engine_subsystem": "AV",
            "engine_version": "0.102.2",
            "result": {
              "errors": [],
              "scan_state": "FULL",
              "threat": "VIRUS",
              "verdict": "DANGEROUS"
            }
          },
          {
            "database_time": 1617993482.0,
            "database_version": "1.0.0.552",
            "detections": [
              {
                "detect": "tool_win_ZZ_EICAR__Virus",
                "threat": "VIRUS"
              }
            ],
            "engine_code_name": "ptesc",
            "engine_subsystem": "STATIC",
            "engine_version": "0.0.0.1835+master",
            "result": {
              "errors": [],
              "scan_state": "FULL",
              "threat": "VIRUS",
              "verdict": "DANGEROUS"
            }
          }
        ],
        "file_info": {
          "file_path": "",
          "file_uri":
"sha256:215a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
          "md5": "44d88612fea8a7f36de82e1278abb02f",
          "mime_type": "text/plain; charset=us-ascii",
          "sha1": "3395856ce81f1b7382dee72602f798b642f14140",
          "sha256":
"275a021bbfb5489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
          "size": 69
        },
        "result": {
          "errors": [],
          "scan_state": "FULL",
          "threat": "VIRUS",
          "verdict": "DANGEROUS"
```

```
          },
          "type": "FILE"
        }
      ],
      "result": {
        "duration": 0.17754173884168268,
        "errors": [],
        "scan_state": "FULL",
        "threat": "VIRUS",
        "verdict": "DANGEROUS"
      },
      "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
    },
    "errors": []
  }
```


Ответ на неуспешный запрос:


```
{
   "errors": [
       {
          "type": "HTTPUnauthorized",
          "message": "PublicAPI entrypoint \"api-entrypoint-name\"disabled"
       }
    ],
    "data": {}
}
```


3.3.2. Запуск проверки ссылки с выбранными параметрами

Ваше приложение может запустить проверку ссылки и в ответ получить результаты проверки и (или) идентификатор задания на проверку.

Метод и URL запроса:

```
POST <Корневой URL API>/scan/checkURL
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 6. Параметры тела запроса на запуск проверки ссылки

Параметр	Обязательный	Тип	Описание
`url`	Да	String	Проверяемая ссылка
`async_result`	Нет	Boolean	Возвращать только идентификатор задания на проверку. Включение этого параметра может понадобиться для отправки асинхронных

Параметр	Обязательный	Тип	Описание
			запросов на проверку ссылок: ваше приложение может не дожидаться результатов проверки, а получать их отдельными запросами. Такие запросы позволяют получать результаты проверки как в общем виде (см. раздел 3.2.4), так и в подробном (см. раздел 3.2.5). Значение по умолчанию — `false`
`short_result`	Нет	Boolean	Возвращать только общий результат проверки. Значение `false` этого параметра игнорируется, если для параметра `async_result` используется значение `true`. Значение по умолчанию — `true`
`priority`	Нет	Integer	Приоритет выполнения проверки ссылки от 1 до 4, где 4 — самый высокий приоритет. Значение по умолчанию — 3

Ответ на запрос

В случае успешного выполнения запроса публичный API возвращает ответ с кодом 200 по схеме ответа с результатами проверки (см. раздел 3.7.1).

Возможные коды ошибок и их значения:

— 401 — в запросе нет токена доступа, он указан неверно или не привязан к источнику с типом «API с выбранными параметрами проверки»;

— 403 — для переданного в запросе токена доступа среди разрешенных действий не выбрана «Проверка с параметрами источника».

Пример

Тело запроса на запуск синхронной проверки ссылки с выдачей подробных результатов проверки:

```
{
  "url": http://example.ru,
  "short_result": false
}
```

Ответ на успешный асинхронный запрос (с включенным параметром `async_result`):


```
{
  "data": {
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
  "errors": []
}
```


Ответ на успешный синхронный запрос с подробными результатами проверки (с выключенными параметрами `async_result` и `short_result`):


```
{
  "data": {
    "artifacts": [
      {
        "artifacts": [],
        "engine_results": [
          {
            "database_time": 1618398601.0,
            "detections": [
              {
                "detect": "Win.Test.EICAR_HDB-1",
                "threat": "VIRUS"
              }
            ],
            "engine_code_name": "clamav",
            "engine_subsystem": "AV",
            "engine_version": "0.102.2",
            "result": {
              "errors": [],
              "scan_state": "FULL",
              "threat": "VIRUS",
              "verdict": "DANGEROUS"
            }
          },
          {
            "database_time": 1617993482.0,
            "database_version": "1.0.0.552",
            "detections": [
              {
                "detect": "tool_win_ZZ_EICAR__Virus",
                "threat": "VIRUS"
              }
            ],
            "engine_code_name": "ptesc",
            "engine_subsystem": "STATIC",
            "engine_version": "0.0.0.1835+master",
```

```
            "result": {
              "errors": [],
              "scan_state": "FULL",
              "threat": "VIRUS",
              "verdict": "DANGEROUS"
            }
          }
        ],
        "file_info": {
          "file_path": "",
          "file_uri":
"sha256:215a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
          "md5": "44d88612fea8a7f36de82e1278abb02f",
          "mime_type": "text/plain; charset=us-ascii",
          "sha1": "3395856ce81f1b7382dee72602f798b642f14140",
          "sha256":
"275a021bbfb5489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
          "size": 69
        },
        "result": {
          "errors": [],
          "scan_state": "FULL",
          "threat": "VIRUS",
          "verdict": "DANGEROUS"
        },
        "type": "FILE"
      }
    ],
    "result": {
      "duration": 0.17754173884168268,
      "errors": [],
      "scan_state": "FULL",
      "threat": "VIRUS",
      "verdict": "DANGEROUS"
    },
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
  "errors": []
}
```


Ответ на неуспешный запрос:


```
{
   "errors": [
       {
          "type": "HTTPUnauthorized",
          "message": "PublicAPI entrypoint \"api-entrypoint-name\"disabled"
       }
```

```
    ],
    "data": {}
}
```

3.3.3. Получение общего результата проверки

Ваше приложение может получить общий результат проверки файла или ссылки при помощи отдельного запроса. Запрос можно использовать, когда проверка была запущена асинхронным методом (при помощи запроса `scan/checkFile` или `scan/checkURL` с включенным параметром `async_result`), который не подразумевает возвращения результатов проверки.

Метод и URL запроса:

```
POST <Корневой URL API>/scan/getStatus
```

Тело запроса должно содержать параметр `scan_id` с идентификатором задания на проверку файла или ссылки.

> **Примечание.** Ваше приложение может получить значение для параметра `scan_id` из ответа на запрос проверки файла (см. раздел 3.3.1) или на запрос проверки ссылки (см. раздел 3.3.2). В этом случае идентификатор задания можно взять из поля `data` → `scan_id`.

Также тело запроса может содержать включенный параметр `allow_preflight` для получения в ответе предварительного результата проверки.

Ответ на запрос

В случае успешного выполнения запроса публичный API возвращает ответ с кодом 200 по схеме ответа с результатами проверки (см. раздел 3.7.1).

Возможные коды ошибок и их значения:

- 401 — в запросе нет токена доступа (см. раздел 3.1) или он указан неверно;
- 404 — задание на проверку файла не найдено; возможно, оно было создано более трех часов назад и завершилось с ошибкой;
- 405 — неправильно указан метод запроса.

Пример

Тело запроса:

```
{ "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d" }
```

Ответ, когда задание на проверку еще выполняется и нет предварительного результата (с включенным, выключенным или не указанным параметром `allow_preflight`):

```
{
  "data": {
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
```

```
    },
    "errors": []
  }
```

Ответ, когда задание на проверку еще выполняется и есть предварительный результат (`allow_preflight=true`):

```
  {
    "data": {
      "is_preflight": true,
      "result": {
        "duration": 0.17754173884168268,
        "errors": [],
        "scan_state": "FULL",
        "threat": "VIRUS",
        "verdict": "DANGEROUS"
  },
      "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
     "errors": []
  }
```

Ответ, когда задание на проверку завершилось (с включенным параметром `allow_preflight`):

```
  {
    "data": {
      "is_preflight": false,
      "result": {
        "duration": 0.17754173884168268,
        "errors": [],
        "scan_state": "FULL",
        "threat": "VIRUS",
        "verdict": "DANGEROUS"
      },
      "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
    },
    "errors": []
  }
```

Ответ на неуспешный запрос:

```
  {
    "errors": [
      {
        "type": "ERROR_ITEM_NOT_FOUND",
        "message": "Item \"058f1426-b920-11eb-af49-e757f9105daf\" is not found here."
      }
    ],
    "data": {}
  }
```

3.3.4. Получение подробных результатов проверки

Ваше приложение может получить детальные результаты проверки файла или ссылки при помощи отдельного запроса. Запрос можно использовать, когда проверка была запущена асинхронным методом (при помощи запроса `scan/checkFile` или `scan/checkURL` с включенным параметром `async_result`), который не подразумевает возвращения результатов проверки.

Метод и URL запроса:

```
POST <Корневой URL API>/scan/getFullReport
```

Тело запроса должно содержать параметр `scan_id` с идентификатором задания на проверку файла или ссылки.

> **Примечание.** Ваше приложение может получить значение для параметра `scan_id` из ответа на запрос проверки файла (см. раздел 3.3.1) или на запрос проверки ссылки (см. раздел 3.3.2). В этом случае идентификатор задания можно взять из поля `data` → `scan_id`.

Ответ на запрос

В случае успешного выполнения запроса публичный API возвращает ответ с кодом 200 по схеме ответа с результатами проверки (см. раздел 3.7.1).

Возможные коды ошибок и их значения:

— 401 — в запросе нет токена доступа (см. раздел 3.1) или он указан неверно;

— 404 — задание на проверку файла не найдено; возможно, оно было создано более трех часов назад и завершилось с ошибкой;

— 405 — неправильно указан метод запроса.

Пример

Тело запроса:

```
{ "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d" }
```

Ответ, когда задание на проверку еще выполняется:

```
{
  "data": {
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
  "errors": []
}
```

Ответ, когда задание на проверку завершилось:

```
{
  "data": {
```

```
"artifacts": [
  {
    "artifacts": [],
    "engine_results": [
      {
        "database_time": 1618398601.0,
        "detections": [
          {
            "detect": "Win.Test.EICAR_HDB-1",
            "threat": "VIRUS"
          }
        ],
        "engine_code_name": "clamav",
        "engine_subsystem": "AV",
        "engine_version": "0.102.2",
        "result": {
          "errors": [],
          "scan_state": "FULL",
          "threat": "VIRUS",
          "verdict": "DANGEROUS"
        }
      },
      {
        "database_time": 1617993482.0,
        "database_version": "1.0.0.552",
        "detections": [
          {
            "detect": "tool_win_ZZ_EICAR__Virus",
            "threat": "VIRUS"
          }
        ],
        "engine_code_name": "ptesc",
        "engine_subsystem": "STATIC",
        "engine_version": "0.0.0.1835+master",
        "result": {
          "errors": [],
          "scan_state": "FULL",
          "threat": "VIRUS",
          "verdict": "DANGEROUS"
        }
      },
      {
        "engine_code_name": "PT",
        "engine_subsystem": "SANDBOX",
        "details": {
          "sandbox": {
```

```
                "image": {
                  "image_id": "win7-sp1-x64",
                  "version": "1.0.0.144",
                  "os": {
                    "architecture": "x64",
                    "locale": "ru-RU",
                    "name": "Microsoft Windows 7 Enterprise",
                    "service_pack": "sp1",
                    "version": "6.1.7601.18741"
                  }
                },
                "artifacts": [
                ],
                "logs": [
                  {
                    "type": "EVENT_CORRELATED",
                    "file_name": "events-correlated.log.gz",
                    "file_uri":
"sha256:d88eb6f52506b8fdeffd37b98438949901055d47ad5298a99d68cbce77296d31"
                  }
                ],
                "analysis_duration": 124.0,
                "network_objects": [
                  {
                    "type": "IP",
                    "value": "93.184.216.34"
                  },
                  {
                    "type": "DOMAIN",
                    "value": "example.com"
                  },
                  {
                    "type": "URL",
                    "value": "http://example.com/"
                  }
                ]
              }
            }
          }
        ],
        "file_info": {
          "file_path": "",
          "file_uri":
"sha256:215a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
          "md5": "44d88612fea8a7f36de82e1278abb02f",
          "mime_type": "text/plain; charset=us-ascii",
          "sha1": "3395856ce81f1b7382dee72602f798b642f14140",
```

```
          "sha256":
"275a021bbfb5489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f",
          "size": 69
        },
        "network_objects": [
          {
            "type": "URL",
              "value": "https://secure.eicar.org/eicar.com"
          }
        ],
        "result": {
          "errors": [],
          "scan_state": "FULL",
          "threat": "VIRUS",
          "verdict": "DANGEROUS"
        },
        "type": "FILE"
      }
    ],
    "result": {
      "duration": 0.17754173884168268,
      "errors": [],
      "scan_state": "FULL",
      "threat": "VIRUS",
      "verdict": "DANGEROUS"
    },
    "scan_id": "16d1f7aa-333a-4fe8-8637-786e2956233d"
  },
  "errors": []
}
```


Ответ на неуспешный запрос:


```
{
  "errors": [
    {
      "type": "ERROR_ITEM_NOT_FOUND",
      "message": "Item \"058f1426-b920-11eb-af49-e757f9105daf\" is not found here."
    }
  ],
  "data": {}
}
```


3.4. Скачивание файла по API

Ваше приложение может скачать файл из PT Sandbox.

Метод и URL запроса:

```
POST <Корневой URL API>/storage/downloadArtifact
```

Тело запроса должно содержать параметр `file_uri`. В качестве значения параметра должны быть указаны название алгоритма, по которому была вычислена хеш-сумма файла (`md5`, `sha1` или `sha256`), двоеточие и сама хеш-сумма.

> **Примечание.** Ваше приложение может получить значение для параметра `file_uri` из ответа на запрос проверки файла (см. раздел 3.2.2). В этом случае URI файла в нужном формате можно взять из поля `data` → `artifacts` → `file_info` → `file_uri`.

Ответ на запрос

В ответ на корректно составленный запрос публичный API присылает сообщение с кодом 200. В теле этого сообщения — запрошенный файл в бинарном виде (MIME-тип — application/octet-stream).

В ответ на неуспешный запрос сервис публичного API присылает сообщение с кодом ошибки. Тело такого сообщения содержит поля, описанные в таблице ниже.

Таблица 7. Поля в ответе на неуспешный запрос на скачивание файла

Поле	Тип	Описание
`data`	JSON object	Не заполняется
`errors`	Array of JSON objects	Ошибки, возникшие при выполнении запроса. Каждая ошибка описывается по схеме error (см. раздел 3.7.4)

Возможные коды ошибок и их значения:

- 400 — запрос составлен неверно (например, нет параметра `file_uri`);
- 401 — в запросе нет токена доступа (см. раздел 3.1) или он указан неверно;
- 404 — файл с указанной хеш-суммой не найден в PT Sandbox;
- 405 — неправильно указан метод запроса.

Пример

Тело запроса:

```
{ "file_uri":
"sha256:32E7802D1E4723F2A529F199EDF62EF74A21640DE379250ECFB7D5C17A2D5A0" }
```

Ответ, когда запрос был составлен неверно:

```
{
  "data": {},
  "errors": [
    {
```

```
      "message": "{\'file_uri\': [\'Missing data for required field.\']}",
      "type": "HTTPBadRequest"
    }
  ]
}
```

3.5. Получение информации об образах виртуальных машин

Ваше приложение может получить список образов виртуальных машин, установленных в PT Sandbox.

Метод и URL запроса:

```
POST <Корневой URL API>/engines/sandbox/getImages
```

Параметры запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис публичного API присылает сообщение с кодом 200. Тело ответного сообщения может содержать поля, описанные в таблице ниже.

Таблица 8. Поля в ответном сообщении на запрос образов виртуальных машин

Поле	Тип	Описание
`data`	Array of JSON objects	Информация об образах виртуальных машин. Каждый образ описывается в виде JSON-объекта по схеме image (см. раздел 3.7.2)
`errors`	Array of JSON objects	Ошибки, возникшие при выполнении запроса. Каждая ошибка описывается по схеме error (см. раздел 3.7.4)

Возможные коды ошибок и их значения:

— 401 — в запросе нет токена доступа (см. раздел 3.1) или он указан неверно;

— 405 — неправильно указан метод запроса.

Пример

Запрос:

```
POST /api/v1/engines/sandbox/getImages
```

Ответ на успешный запрос:

```
{
  "data": [
    {
```

```
        "image_id": "win10-1803-x64",
        "version": "55",
        "os": {
          "name": "Windows 10 Pro",
          "version": "10.0.17134",
          "architecture": "x64",
          "service_pack": "",
          "locale": "en-US"
        }
      },
      {
        "image_id": "win7-sp1-x64",
        "version": "78",
        "os": {
          "name": "Windows 7 ENTERPRISE",
          "version": "6.1.7601"
          "architecture": "x64",
          "service_pack": "Service Pack 1",
          "locale": "en-US"
        }
      }
    ],
    "errors": []
  }
```

Ответ на неуспешный запрос:

```
  {
    "data": {},
    "errors": [
      {
        "message": "405: Method Not Allowed",
        "type": "HTTPMethodNotAllowed"
      }
    ]
  }
```

3.6. Получение состояния API

Ваше приложение может получить текущее состояние публичного API.

Примечание. Запрос может быть отправлен методами POST и GET.

Метод и URL запроса:

```
POST <Корневой URL API>/maintenance/checkHealth
```

Параметры запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис публичного API присылает сообщение с кодом 200. Тело ответного сообщения может содержать поля, описанные в таблице ниже.

Таблица 9. Поля в ответе на запрос состояния публичного API

Поле	Тип	Описание
`data` → `status`	String	Состояние работы сервиса публичного API. Возможное значение — `OK`. При наличии ошибок отсутствует
`errors`	Array of JSON objects	Ошибки, возникшие при выполнении запроса. Каждая ошибка описывается по схеме error (см. раздел 3.7.4)

Пример

Запрос:

```
POST /maintenance/checkHealth
```

Ответ на успешный запрос:

```
{
  "data": {
    "status": "OK"
  },
  "errors": []
}
```

Ответ на неуспешный запрос:

```
{
  "data": {},
  "errors": [
    {
      "message": "405: Method Not Allowed",
      "type": "HTTPMethodNotAllowed"
    }
  ]
}
```

3.7. Описание схем

В этом разделе приводится описание схем, используемых в ответных сообщениях публичного API.

В этом разделе

3.7.1. Схема ответа с результатами проверки

Описывает ответ на запрос на проверку файла или на получение результатов проверки файла.

Таблица 10. Поля с результатами проверки

Поле	Тип	Описание
`data` → `scan_id`	String	Идентификатор задания на проверку
`data` → `result`	JSON object	Общий результат проверки по схеме result (см. раздел 3.7.1.1). Отсутствует в ответах на запросы: — `createScanTask` с включенным параметром `async_result`; — `createScanURLTask` с включенным параметром `async_result`; — `checkTask`, если проверка файла еще не завершена; — `report`, если проверка файла еще не завершена; — `scan/checkFile` с включенным параметром `async_result`; — `scan/checkURL` с включенным параметром `async_result`; — `scan/getStatus`, если проверка файла еще не завершена; — `scan/getFullReport`, если проверка файла еще не завершена
`data` → `artifacts`	Array of JSON objects	Файлы, электронные письма или другие объекты, которые были проверены в ходе проверки файла, URI которого был указан в запросе `createScanTask`, или в ходе проверки ссылок через запросы `scan/checkURL` и

Поле	Тип	Описание
		`createScanURLTask`. Каждый проверенный объект описывается в виде JSON-объекта по схеме artifact (см. раздел 3.7.1.2). Отсутствует в ответах на запросы: — `createScanTask` с включенным параметром `async_result` или `short_result`; — `createScanURLTask` с включенным параметром `async_result` или `short_result`; — `checkTask`; — `scan/checkFile` с включенным параметром `async_result` или `short_result`; — `scan/checkURL` с включенным параметром `async_result` или `short_result`; — `scan/getStatus`
`errors`	Array of JSON objects	Ошибки, возникшие по итогу проверки. Каждая ошибка описывается по схеме error (см. раздел 3.7.4)

В этом разделе

3.7.1.1. Схема result

Описывает результат проверки.

Таблица 11. Поля в схеме `result`

Поле	Тип	Описание
`scan_state`	String	Статус проверки. Возможные значения: — `PARTIAL` — частичная проверка; — `FULL` — полная проверка; — `UNSCANNED` — проверка не проведена
`duration`	Float	Длительность проверки в секундах. Записывается только в общих результатах проверки (в JSON-объекте `data` → `result`). Отсутствует в ответе на запрос `checkTask` с результатами проверки, запущенной асинхронным запросом `createScanTask` (с включенным параметром `async_result`)
`verdict`	String	Результат проверки. Возможные значения: — `CLEAN` — угроз не обнаружено; — `UNWANTED` — потенциально опасный; — `DANGEROUS` — опасный
`threat`	String	Тип обнаруженного вредоносного ПО
`errors`	Array of JSON objects	Ошибки, возникшие в ходе проверки. Каждая ошибка описывается в виде JSON-объекта по схеме error (см. раздел 3.7.4)

3.7.1.2. Схема artifact

Описывает проверяемый файл.

Таблица 12. Поля в схеме `artifact`

Поле	Тип	Описание
`type`	String	Тип проверенного объекта. Возможные значения: — `FILE` — обычный файл; — `ARCHIVE` — архив; — `COMPRESSED` — сжатый файл; — `EMAIL` — электронное письмо

Поле	Тип	Описание
`result`	JSON object	Результат проверки файла по схеме result (см. раздел 3.7.1.1)
`file_info`	JSON object	Информация о проверенном файле по схеме file_info (см. раздел 3.7.1.3)
`engine_results`	Array of JSON objects	Результаты проверки файла конкретными антивирусами или другими компонентами. Каждый результат описывается в JSON-объекте по схеме engine_result (см. раздел 3.7.1.4)
`artifacts`	Array of JSON objects	Файлы, запакованные в архив. Если отправленный на проверку файл не является архивом или превышена допустимая глубина распаковки, массив `artifacts` пустой. Каждый файл в этом массиве описывается в виде JSON-объекта по схеме artifact (см. раздел 3.7.1.2)
`network_objects`	Array of JSON objects	Информация о сетевых объектах выполненного задания. Каждый сетевой объект описывается в виде JSON-объекта по схеме network_object (см. раздел 3.7.1.6)

3.7.1.3. Схема file_info

Описывает свойства файла.

Таблица 13. Поля в схеме `file_info`

Поле	Тип	Описание
`file_uri`	String	Идентификатор файла. Может использоваться для его скачивания (см. раздел 3.4)
`file_path`	String	Путь к файлу (исключая корневой файл структуры), включая его название. Например, для файла `readme.txt` в корне архива `archive.zip` в качестве значения этого поля будет указано `readme.txt`, для самого архива — пустое значение
`mime_type`	String	MIME-тип файла, определенный в процессе проверки
`md5`	String	MD5-хеш-сумма файла
`sha1`	String	SHA-1-хеш-сумма файла
`sha256`	String	SHA-256-хеш-сумма файла

Поле	Тип	Описание
size	Integer	Размер файла в байтах

3.7.1.4. Схема engine_result

Описывает результат проверки конкретным антивирусом или другим проверяющим компонентом.

Таблица 14. Поля в схеме engine_result

Поле	Тип	Описание
engine_subsystem	String	Метод проверки. Возможные значения: — AV — антивирусное сканирование; — SANDBOX — поведенческий анализ; — STATIC — экспертная оценка файлов
engine_code_name	String	Название антивируса или компонента
engine_version	String	Версия антивируса или компонента
database_version	String	Версия антивирусной базы или базы знаний
database_time	Float	Время обновления антивирусной базы или базы знаний
result	JSON object	Результат проверки антивирусом или другим компонентом по схеме result (см. раздел 3.7.1.1)
detections	Array of JSON objects	Описание обнаруженного вредоносного ПО
detections → detect	String	Вредоносное ПО
detections → threat	String	Тип вредоносного ПО
details → sandbox	JSON object	Подробная информация о поведенческом анализе (если проводился)
details → sandbox → image	JSON object	Информация об образе виртуальной машины по схеме image (см. раздел 3.7.2)
details → sandbox → logs	Array of JSON objects	Копия сетевого трафика, видеозапись, журналы событий. Каждый объект описывается по схеме log (см. раздел 3.7.1.5)

Поле	Тип	Описание
`details` → `sandbox` → `artifacts`	Array of JSON objects	Артефакты виртуальной машины — файлы, созданные в ходе поведенческого анализа. Каждый файл описывается в виде JSON-объекта по схеме artifact (см. раздел 3.7.1.2)
`details` → `sandbox` → `network_objects`	Array of JSON objects	Информация о сетевых объектах выполненного задания. Каждый сетевой объект описывается в виде JSON-объекта по схеме network_object (см. раздел 3.7.1.6)

См. также

Схема network_object (см. раздел 3.7.1.6)

3.7.1.5. Схема log

Описывает файлы, созданные для поведенческого анализа: копию сетевого трафика, видеозаписи и журналы событий.

Таблица 15. Поля в схеме `log`

Поле	Тип	Описание
`type`	String	Тип созданного файла. Возможные значения: — `NETWORK` — копия сетевого трафика в формате PCAP; — `SCREENSHOT` — снимок или видеозапись экрана виртуальной машины; — `EVENT_RAW` — необработанный список событий в простом текстовом формате; — `EVENT_CORRELATED` — нормализованные события, проанализированные с помощью правил корреляции для нахождения закономерностей и выявления опасного и потенциально опасного поведения (в формате JSON); — `EVENT_NORMALIZED` — события, приведенные к единому виду, пригодному для дальнейшего анализа (в формате JSON);
`file_uri`	String	Идентификатор файла. Используется для скачивания
`file_name`	String	Название файла

3.7.1.6. Схема network_object

Описывает сетевые объекты выполненного задания.

Таблица 16. Поля в схеме `network_object`

Поле	Тип	Описание
`type`	String	Тип сетевого объекта. Возможные значения: — `DOMAIN` — доменное имя; — `IP` — IP-адрес; — `URL` — URL-адрес
`value`	String	Значение сетевого объекта

3.7.2. Схема image

Описывает образ виртуальной машины.

Таблица 17. Поля в схеме `image`

Поле	Тип	Описание
`image_id`	String	Идентификатор образа виртуальной машины
`version`	String	Версия образа виртуальной машины
`os`	JSON object	Информация об операционной системе образа виртуальной машины по схеме os (см. раздел 3.7.3)

3.7.3. Схема os

Описывает операционную систему образа виртуальной машины.

Таблица 18. Поля в схеме `os`

Поле	Тип	Описание
`name`	String	Название операционной системы
`version`	String	Версия операционной системы
`architecture`	String	Архитектура процессора, которую поддерживает операционная система
`service_pack`	String	Название пакета обновления операционной системы

Поле	Тип	Описание
`locale`	String	Локаль операционной системы

3.7.4. Схема error

Описывает ошибку, возникшую в ходе выполнения запроса или проверки.

Таблица 19. Поля в схеме `error`

Поле	Тип	Описание
`message`	String	Сообщение об ошибке
`type`	String	Тип ошибки

4. Формат сообщений syslog

Вы можете включить и настроить отправку сообщений по протоколу syslog в интерфейсе PT Sandbox для централизованного сбора и анализа событий ИБ в информационной системе вашей организации.

PT Sandbox формирует тело сообщения в формате JSON, а заголовок сообщения в формате, описанном в RFC 5424. Тип сообщения записывается в части MSGID заголовка. Значение приоритета, которое указывается в части PRI заголовка, для всех отправляемых сообщений равно 100.

PT Sandbox может отправлять на сервер системного журнала по протоколу syslog:

- сообщения о ходе сканирования файлов антивирусами;
- результаты поведенческого анализа файлов, обнаруженное опасное поведение файлов, а также информацию об артефактах, полученных в ходе такого анализа;
- информацию о файлах и источниках их получения, распаковке архивов и вредоносном ПО, обнаруженном в файлах в ходе их проверки;
- уведомления об обновлении антивирусов или их баз.

В этом разделе

4.1. Сообщения о проверке файлов

Алгоритм отправки сообщений об обработке задания на проверку файлов представлен на рисунке ниже, где в зеленых блоках — названия типов сообщений.

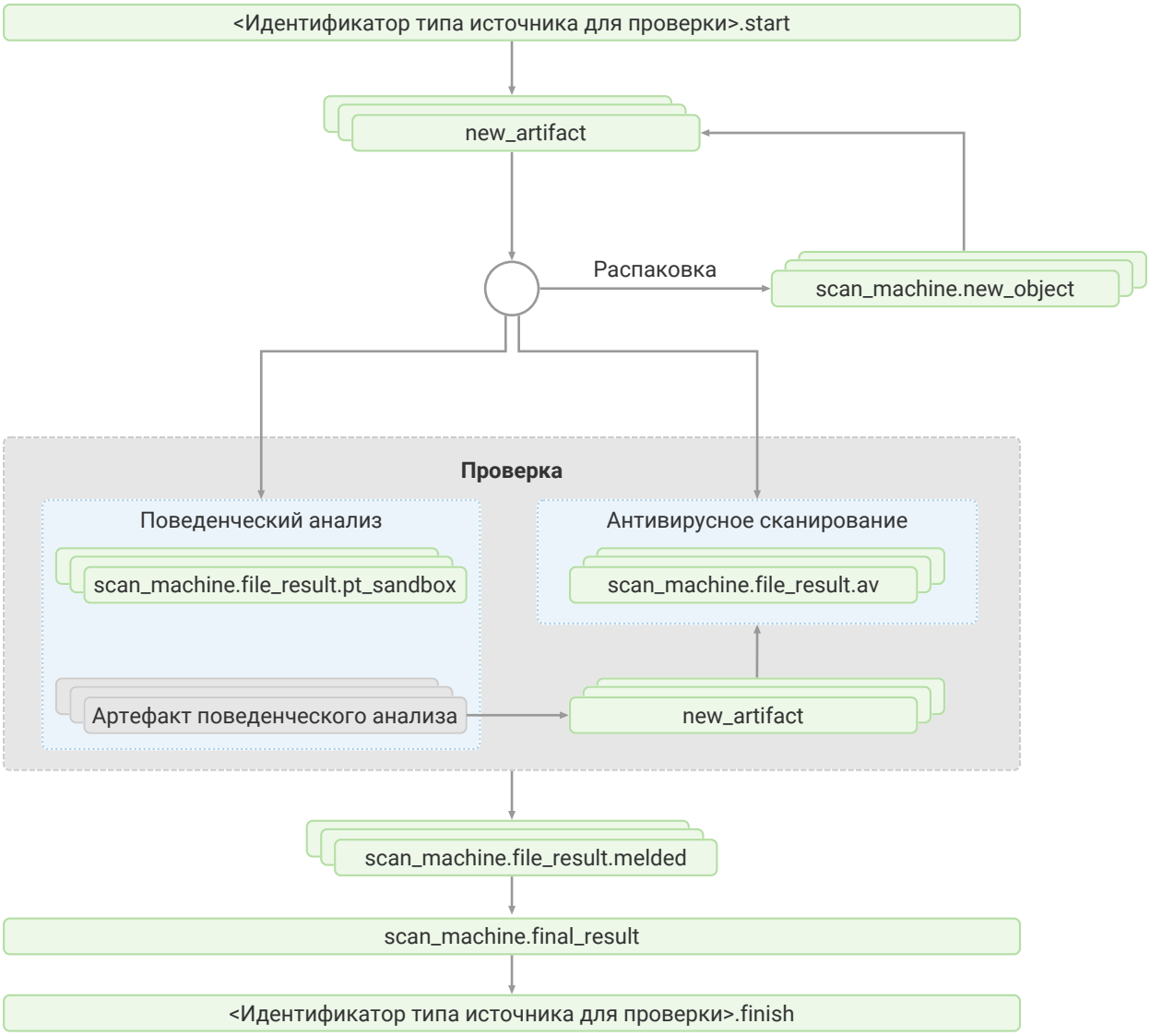

Рисунок 1. Алгоритм отправки сообщений о проверке в syslog

Подробное описание алгоритма:

1. PT Sandbox получает объект для проверки на одну из своих точек входа — на источник для проверки. Объектом может быть файл, электронное письмо или HTTP-сообщение.
2. PT Sandbox создает задание на проверку полученного объекта. Создав задание, PT Sandbox отправляет сообщение типа `<Идентификатор типа источника для проверки>.start` (например, `email.start`). Это сообщение информирует о начале обработки объекта для проверки и содержит уникальный идентификатор задания на проверку, информацию об объекте и о том, как он был получен.

3. PT Sandbox распознает файлы, связанные с объектом для проверки. Например, при получении электронного письма в отдельные файлы выделяются его заголовок, тело и прикрепленные файлы. Для каждого распознанного файла PT Sandbox отправляет сообщение `new_artifact`. Это сообщение содержит информацию о файле, связанном с объектом проверки, а также идентификатор задания, в котором этот файл был получен.

4. PT Sandbox обрабатывает каждый файл, указанный в сообщении `new_artifact`, следующим образом:

   - Если файл является контейнером для других файлов (например, файл является архивом, письмом или ISO-файлом), PT Sandbox пытается извлечь файлы из этого контейнера. При извлечении файла PT Sandbox отправляет в системный журнал сообщение `scan_machine.new_object`. Это сообщение содержит информацию об извлеченных файлах, информацию о контейнере, из которого были извлечены файлы, а также идентификатор задания.

     Примечание. При извлечении каждого файла в свою очередь отправляется отдельное сообщение `new_artifact` (см. предыдущий пункт) с указанием родительского файла.

     Примечание. Глубина извлечения и пароли для архивов определяются для каждого отдельного источника специалистами по безопасности.

   - PT Sandbox отправляет файл на антивирусное сканирование. По завершении сканирования PT Sandbox отправляет сообщение `scan_machine.file_result.av`. Это сообщение содержит информацию о файле, идентификатор задания, в котором этот файл был получен, а также результаты сканирования файла конкретным антивирусом.

   - PT Sandbox проводит поведенческий анализ файла. По его завершении PT Sandbox отправляет сообщение `scan_machine.file_result.pt_sandbox`. Это сообщение содержит информацию о результатах поведенческого анализа и об обнаруженном опасном поведении проверенного файла. Если в ходе поведенческого анализа появляются артефакты (например, дампы памяти, файлы, создаваемые проверяемым приложением), PT Sandbox сообщает о каждом из них сообщением `new_artifact` и отправляет их на антивирусное сканирование.

5. По окончании проверки файла, указанного в сообщении `new_artifact` (получив результаты сканирования от всех антивирусов и результаты поведенческого анализа), PT Sandbox отправляет в системный журнал сообщение `scan_machine.file_result.melded`. Это сообщение содержит информацию о файле, идентификатор задания, в котором этот файл был получен, а также общий результат проверки файла с учетом результатов проверки всех извлеченных из него файлов и артефактов поведенческого анализа.

6. По завершении проверки всех файлов, указанных в сообщениях `new_artifact`, PT Sandbox отправляет в системный журнал сообщение `scan_machine.final_result`. Это сообщение содержит идентификатор задания и итоговый результат проверки объекта, полученного на шаге 1.

7. PT Sandbox отправляет в системный журнал сообщение типа `<Идентификатор типа источника для проверки>.finish` (например, `email.finish`).

Это сообщение информирует об окончании обработки задания и содержит уникальный идентификатор задания и признак успешности обработки задания.

Примечание. Если объект для проверки поступил на источник, работающий в пассивном режиме, PT Sandbox отправляет сообщение об окончании обработки задания еще до получения результатов проверки (сообщение типа `<Идентификатор типа источника для проверки>.finish` приходит до сообщения `scan_machine.final_result`).

В этом разделе

4.1.1. Сообщения <Идентификатор типа источника для проверки>.start

При создании задания на проверку PT Sandbox отправляет в системный журнал сообщение типа `<Идентификатор типа источника для проверки>.start` (например, `email.start`). Это сообщение информирует о начале обработки файлов из задания и содержит уникальный идентификатор задания, информацию о файлах в задании и о том, как они были получены.

В зависимости от типа источника, от которого были получены файлы, сообщения имеют разную структуру и содержание.

В этом разделе

Сообщение icap.start (см. раздел 4.1.1.6)

Информация об HTTP-сообщении в icap.start (см. раздел 4.1.1.7)

Сообщение mail_bcc.start (см. раздел 4.1.1.8)

Сообщение mail_gateway.start (см. раздел 4.1.1.9)

Сообщение public_api.start (см. раздел 4.1.1.10)

Сообщение user_scan.start (см. раздел 4.1.1.11)

См. также

Идентификаторы типов источников для проверки (см. раздел 4.1.11)

4.1.1.1. Сообщение check_me.start

В таблице ниже описываются поля и объекты в сообщении `check_me.start` о начале обработки задания на проверку письма, полученного службой Checkme.

Таблица 20. Поля в сообщении check_me.start

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`scan_id`	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	—	—	UNIX-время генерации сообщения	Float
`entry_point_id`	—	—	Название источника для проверки в интерфейсе PT Sandbox	String
`email` и `envelope`	—	Информация об электронном письме в сообщениях syslog (см. раздел 4.1.9)		
`received`	—	—	UNIX-время формирования задания на проверку	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - check_me.start - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "entry_point_id": "checkme-service",
  "email": {
    "id": "<6fee62f024e0@example.com>",
    "from_address": {
```

```
    "address": "ivanov@example.org",
    "name": "Ivan Ivanov"
  },
  "to": {
    "address": "checkme@example.org",
    "name": ""
  },
  "to_list": [
    {
      "address": "checkme@example.org",
      "name": ""
    }
  ],
  "cc": {
    "address": "username_12@example.org",
    "name": "Ivan Ivanov"
  },
  "cc_list": [
    {
      "address": "username_12@example.org",
      "name": "Ivan Ivanov"
    },
    {
      "address": "username23@example.net",
      "name": ""
```

```
      }
    ],
    "bcc": {
      "address": "admin@example.org",
      "name": ""
    },
    "bcc_list": [
      {
        "address": "admin@example.org",
        "name": ""
      }
    ],
    "subject": "Проверка файла",
    "references": "user@example.org",
    "reply_to": ""
  },
  "envelope": {
    "from_address": "ivanov@example.org",
    "recipients": ["checkme@example.org", "username@example.org", "username23@example.net", "admin@example.org"]
  },
  "received": 1511421762.957363
}
```


4.1.1.2. Сообщение dpi.start

В таблице ниже описываются поля и объекты в сообщении `dpi.start` о начале обработки задания на проверку файла, полученного от PT NAD.

Таблица 21. Поля в сообщении dpi.start

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
scan_id	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	—	—	UNIX-время генерации сообщения	Float
entry_point_id	—	—	Название источника для проверки в интерфейсе PT Sandbox	String
meta	—	src dst filename magic file_id proto state один из двух объектов: http или smtp	Блок данных о полученном файле	JSON object
src	meta	ip port address	Блок данных об отправителе файла	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
dst	meta	ip port address	Блок данных о получателе файла	JSON object
ip	src dst	—	IP-адрес	String
port	src dst	—	Номер сетевого порта	UInt32
address	src dst	—	IP-адрес	String
filename	meta	—	Имя файла	String
magic	meta	—	Текстовое описание автоматически определенного формата файла	String
file_id	meta	—	Идентификатор полученного файла	String
proto	meta	—	Прикладной протокол, который использовался для передачи файла	String

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
state	meta	—	Состояние записи файла. Возможные значения: — UNKNOWN — состояние неизвестно; — COMPLETED — файл записан; — TRUNCATED — файл записан не полностью, но данных больше нет; — ERROR — ошибка записи (при получении ошибки файл удаляется). PT Sandbox отправляет на проверку файлы с состоянием COMPLETED	String
http	meta	referer user_agent host uri	Блок данных с информацией о файле, относящейся к HTTP	JSON object
referer	http	—	Адрес предыдущей веб-страницы, с которой был осуществлен данный HTTP-запрос	String
user_agent	http	—	Название и версия браузера	String
host	http	—	HTTP-адрес узла, с которого был скачан файл	String

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
uri	http	—	URI файла	String
smtp	meta	message_id sender	Блок данных с информацией о файле, относящейся к SMTP	JSON object
message_id	smtp	—	Поле заголовка Message-ID	String
sender	smtp	—	Адрес электронной почты отправителя	String
file	—	Поля с информацией о файле (см. раздел 4.1.10)	Информация о файле	JSON object
received	—	—	UNIX-время формирования задания на проверку	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - dpi.start - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "entry_point_id": "dpi-module",
  "meta": {
    "src": {
      "ip": "203.0.113.43",
      "address": "203.0.113.43",
      "port": 8080
```

```
    },
    "dst": {
      "ip": "203.0.113.11",
      "address": "203.0.113.11",
      "port": 8080
    },
    "filename": "example",
    "magic": "ASCII text, with very long lines",
    "file_id": "10007",
    "proto": "HTTP",
    "state": "COMPLETED",
    "http": {
      "referer": "<unknown>",
      "user_agent": "Wget/1.15 (linux-gnu)",
      "host": "203.0.113.43",
      "uri": "/test"
    }
  },
  "file": {
    "mime_type": "application/x-dosexec; charset=binary",
    "md5": "11aced0fd6535f6e...1495ba1c7be00",
    "sha1": "45e50e2af429e44...6f59e46b18b60",
    "sha256": "23ef04408bb2c...7928e7caf3d7f",
    "size": 64000,
    "name": "software.exe"
  },
  "received": 1511421762.957363
}
```

4.1.1.3. Сообщение email.start

В таблице ниже описываются поля и объекты в сообщении `email.start` о начале обработки задания на проверку письма, полученного почтовым агентом от сервера Microsoft Exchange организации.

Таблица 22. Поля в сообщении email.start

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`scan_id`	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	—	—	UNIX-время генерации сообщения	Float
`entry_point_id`	—	—	Название источника для проверки в интерфейсе PT Sandbox	String
`email` и `envelope`	—	Информация об электронном письме в сообщениях syslog (см. раздел 4.1.9)		
`received`	—	—	UNIX-время формирования задания на проверку	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - email.start - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "entry_point_id": "mail-agent",
  "email": {
    "id": "<6fee62f024e0@example.com>",
    "from_address": {
```

```
    "address": "ivanov@example.org",
    "name": "Ivan Ivanov"
  },
  "to": {
    "address": "username@example.com",
    "name": "Ivan Ivanov"
  },
  "to_list": [
    {
      "address": "username@example.com",
      "name": "Ivan Ivanov"
    }
  ],
  "cc": {
    "address": "username_12@example.org",
    "name": "Ivan Ivanov"
  },
  "cc_list": [
    {
      "address": "username_12@example.org",
      "name": "Ivan Ivanov"
    },
    {
      "address": "username23@example.net",
      "name": ""
```

```
      }
    ],
    "bcc": {
      "address": "admin@example.org",
      "name": ""
    },
    "bcc_list": [
      {
        "address": "admin@example.org",
        "name": ""
      }
    ],
    "subject": "Последняя версия моей программы",
    "references": "",
    "reply_to": ""
  },
  "envelope": {
    "from_address": "ivanov@example.org",
  },
  "received": 1511421762.957363
}
```


4.1.1.4. Сообщение files_inbox.start

В таблице ниже описываются поля и объекты в сообщении `files_inbox.start` о начале обработки задания на проверку файла, обнаруженного в папке-шлюзе.

Таблица 23. Поля в сообщении files_inbox.start

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
scan_id	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	—	—	UNIX-время генерации сообщения	Float
entry_point_id	—	—	Название источника для проверки в интерфейсе PT Sandbox	String
src_file_info	—	url	Блок данных с информацией о файле в папке-шлюзе	JSON object
url	src_file_info	—	Путь до файла в папке-шлюзе	String
file	—	Поля с информацией о файле (см. раздел 4.1.10)	Информация о файле	JSON object
received	—	—	UNIX-время формирования задания на проверку	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - files_inbox.start - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "entry_point_id": "gateway-folder",
  "src_file_info": {
    "url": "smb://example.org/share/inbox/software.exe"
```

```
    },
    "file": {
      "mime_type": "application/x-dosexec; charset=binary",
      "md5": "11aced0fd6535f6e...1495ba1c7be00",
      "sha1": "45e50e2af429e44...6f59e46b18b60",
      "sha256": "23ef04408bb2c...7928e7caf3d7f",
      "size": 64000,
      "name": "software.exe"
    },
    "received": 1511421762.957363
  }
```


4.1.1.5. Сообщение files_monitor.start

В таблице ниже описываются поля и объекты в сообщении `files_monitor.start` о начале обработки задания на проверку файла, обнаруженного в общей папке.

Таблица 24. Поля в сообщении files_monitor.start

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`scan_id`	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	—	—	UNIX-время генерации сообщения	Float
`entry_point_id`	—	—	Название источника для проверки в интерфейсе PT Sandbox	String

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`src_file_info`	—	`url`	Блок данных с информацией о файле в общей папке	Object
`url`	`src_file_info`	—	Путь до файла в общей папке	String
`file`	—	Поля с информацией о файле (см. раздел 4.1.10)	Информация о файле	JSON object
`received`	—	—	UNIX-время формирования задания на проверку	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - files_monitor.start - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "entry_point_id": "shared-folder",
  "src_file_info": {
    "url": "smb://example.org/share/software.exe"
  },
  "file": {
    "mime_type": "application/x-dosexec; charset=binary",
    "md5": "11aced0fd6535f6e...1495ba1c7be00",
    "sha1": "45e50e2af429e44...6f59e46b18b60",
    "sha256": "23ef04408bb2c...7928e7caf3d7f",
    "size": 64000,
    "name": "software.exe"
```

```
    },
    "received": 1511421762.957363
  }
```

4.1.1.6. Сообщение icap.start

В таблице ниже описываются поля и объекты в сообщении `icap.start` о начале обработки задания на проверку контента, полученного по ICAP.

Таблица 25. Поля в сообщении icap.start

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`scan_id`	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	—	—	UNIX-время генерации сообщения	Float
`entry_point_id`	—	—	Название источника для проверки в интерфейсе PT Sandbox	String

`request`	—	`method`
		`url`
		`version`
		`client_ip`
		`client_username`
			Блок данных о запросе	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
		один из двух объектов: `http` (в случае HTTP-запроса) или `file` (в случае файла)		
`method`	`request`	—	Метод запроса. Возможные значения: `REQMOD` (запрос на изменение HTTP REQUEST) или `RESPMOD` (запрос на изменение HTTP RESPONSE)	String
`url`	`request`	—	URL ICAP, по которому идет обращение к PT Sandbox	String
`version`	`request`	—	Версия протокола ICAP-клиента	String
`client_ip`	`request`	—	IP-адрес пользователя, который получил контент или отправил HTTP-запрос (значение извлекается из поля X-Client-IP заголовка HTTP-запроса)	String
`client_username`	`request`	—	Имя пользователя, прошедшего аутентификацию на прокси-сервере (значение извлекается из поля X-Client-Username заголовка HTTP-запроса)	String

`http`	`request`	`direction`
		`http_request`
		`http_response`
			Информация о файле из полей HTTP	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
direction	http	—	Направление сообщения. Возможные значения: REQUEST (запрос к удаленному узлу за прокси-сервером) или RESPONSE (ответ от удаленного узла за прокси-сервером)	String
http_request	http	Поля и JSON-объекты с информацией об HTTP-запросе (см. раздел 4.1.1.7)	Блок данных с информацией об HTTP-запросе. Записывается в случае запроса к удаленному узлу за прокси-сервером, в остальных случаях может отсутствовать или содержать незаполненные поля	JSON object
http_response	http	Поля и JSON-объекты с информацией об HTTP-ответе (см. раздел 4.1.1.7)	Блок данных с информацией об HTTP-ответе. Записывается только в случае ответа от удаленного узла за прокси-сервером	JSON object
file	request	Поля с информацией о файле (см. раздел 4.1.10)	Информация о файле	JSON object
client	—	ip address port	Блок данных об ICAP-клиенте, который обращается к ICAP-серверу PT Sandbox	JSON object
ip	client	—	IP-адрес ICAP-клиента	String
address	client	—	IP-адрес ICAP-клиента	String

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
port	client	—	Номер ICAP-порта	UInt32
received	—	—	UNIX-время формирования задания на проверку	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - icap.start - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "entry_point_id": "icap",
  "request": {
    "method": "RESPMOD",
    "url": "icap://198.51.100.12:1344/bypass",
    "version": "1.0",
    "client_ip": "198.51.100.22",
    "client_username": "",
    "http": {
      "direction": "RESPONSE",
      "http_request": {
        "method": "GET",
        "url": "http://203.0.113.152/MySample/EICAR.COM",
        "version": "1.1",
        "host": "203.0.113.152",
        "referer": "http://203.0.113.152/MySample/",
        "user_agent": "Mozilla/5.0 (Windows NT 6.3; Win64; x64; rv:97.0) Gecko/20100101 Firefox/97.0",
        "content_type": {
```

```
      "mime_type": "",
      "charset": "",
      "name": "",
      "boundary": ""
    },
    "content_length": 0,
    "content_location": "",
    "content_disposition": {
      "type": "",
      "filename": ""
    }
  },
  "http_response": {
    "version": "1.1",
    "code": 200,
    "reason": "OK",
    server": "nginx/1.16.1",
    "content_type": {
      "mime_type": "application/octet-stream",
      "charset": "",
      "name": "",
      "boundary": ""
    },
    "content_length": 9322,
    "content_location": "",
    "content_disposition": {
      "type": "",
      "filename": ""
```

```
          }
        }
      }
    },
    "client": {
      "ip": "198.51.100.13",
      "address": "198.51.100.13",
      "port": 55892
    },
    "received": 1511421762.957363
  }
```

4.1.1.7. Информация об HTTP-сообщении в icap.start

При составлении сообщений syslog `icap.start` PT Sandbox записывает информацию об HTTP-сообщении в поля и JSON-объекты, описанные в таблице ниже.

Таблица 26. JSON-объекты и поля с информацией об HTTP-сообщении

Поле или JSON-объект	Описание	Тип данных	`http_request`	`http_response`
`method`	Метод HTTP-запроса. Возможные значения: `get` или `post`	String	✔	—
`url`	URL HTTP-запроса	String	✔	—
`version`	Версия HTTP	String	✔	✔
`code`	Код ответа	UInt32	—	✔

Поле или JSON-объект	Описание	Тип данных	http_request	http_response
reason	Описание ответа (Reason-Phrase)	String	—	✔
server	Название и версия веб-сервера, от которого был получен HTTP-ответ	String	—	✔
host	Доменное имя сервера, которому был адресован HTTP-запрос	String	✔	—
referer	Адрес предыдущей веб-страницы, с которой был осуществлен данный HTTP-запрос	String	✔	—
user_agent	Название пользовательского приложения (например, браузера), с которого был осуществлен HTTP-запрос	String	✔	—
content_type	Содержимое поля заголовка Content-Type	JSON object	✔	✔
content_type → mime_type	MIME-тип файла согласно полю Content-Type заголовка HTTP-сообщения	String	✔	✔
content_type → charset	Кодировка веб-страницы	String	✔	✔
content_type → name	Значение параметра name в поле Content-Type заголовка HTTP-сообщения	String	✔	✔
content_type → boundary	Значение параметра boundary в поле Content-Type заголовка HTTP-сообщения	String	✔	✔
content_length	Содержимое поля заголовка Content-Length	UInt32	✔	✔

Поле или JSON-объект	Описание	Тип данных	http_request	http_response
content_location	Содержимое поля заголовка Content-Location	String	✔	✔
content_disposition	Содержимое поля заголовка Content-Disposition	JSON object	✔	✔
content_disposition → type	Тип файла согласно полю Content-Disposition заголовка HTTP-сообщения	String	✔	✔
content_disposition → filename	Имя файла согласно полю Content-Disposition заголовка HTTP-сообщения	String	✔	✔

4.1.1.8. Сообщение mail_bcc.start

В таблице ниже описываются поля и объекты в сообщении mail_bcc.start о начале обработки задания на проверку письма, полученного от почтового сервера организации в виде скрытой копии.

Таблица 27. Поля в сообщении mail_bcc.start

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
scan_id	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	—	—	UNIX-время генерации сообщения	Float
entry_point_id	—	—	Название источника для проверки в интерфейсе PT Sandbox	String

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
client	—	ip address port	Блок данных с информацией об SMTP-клиенте	JSON object
ip	client	—	IP-адрес SMTP-клиента	String
address	client	—	IP-адрес SMTP-клиента	String
port	client	—	Порт SMTP-клиента	UInt32
email и envelope	—	Информация об электронном письме в сообщениях syslog (см. раздел 4.1.9)		
received	—	—	UNIX-время формирования задания на проверку	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - mail_bcc.start - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "entry_point_id": "mail-bcc",
  "client": {
    "ip": "203.0.113.33",
    "address": "203.0.113.33",
    "port": 36128
  },
  "email": {
    "id": "<6fee62f024e0@example.com>",
```

```
    "from_address": {
      "address": "ivanov@example.org",
      "name": "Ivan Ivanov"
    },
    "to": {
      "address": "username@example.com",
      "name": "Ivan Ivanov"
    },
    "to_list": [
      {
        "address": "username@example.com",
        "name": "Ivan Ivanov"
      }
    ],
    "subject": "Последняя версия моей программы",
    "references": "",
    "reply_to": ""
  },
  "envelope": {
    "from_address": "ivanov@example.org",
    "recipients": ["bcc@sandbox.local"]
  },
  "received": 1511421762.957363
}
```

4.1.1.9. Сообщение mail_gateway.start

В таблице ниже описываются поля и объекты в сообщении `mail_gateway.start` о начале обработки задания на проверку письма, полученного от почтового сервера Postfix или Exim в режиме фильтрации.

Таблица 28. Поля в сообщении mail_gateway.start

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`scan_id`	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	—	—	UNIX-время генерации сообщения	Float
`entry_point_id`	—	—	Название источника для проверки в интерфейсе PT Sandbox	String

`client`	—	`ip`
		`address`
		`port`
			Блок данных с информацией об SMTP-клиенте	JSON object

`ip`	`client`	—	IP-адрес SMTP-клиента	String
`address`	`client`	—	IP-адрес SMTP-клиента	String
`port`	`client`	—	Порт SMTP-клиента	UInt32
`email` и `envelope`	—	Информация об электронном письме в сообщениях syslog (см. раздел 4.1.9)		
`received`	—	—	UNIX-время формирования задания на проверку	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - mail_gateway.start - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "entry_point_id": "mail-gateway",
  "client": {
    "ip": "203.0.113.33",
    "address": "203.0.113.33",
    "port": 36128
  },
  "email": {
    "id": "<6fee62f024e0@example.com>",
    "from_address": {
      "address": "ivanov@example.org",
      "name": "Ivan Ivanov"
    },
    "to": {
      "address": "username@example.com",
      "name": "Ivan Ivanov"
    },
    "to_list": [
      {
        "address": "username@example.com",
        "name": "Ivan Ivanov"
```

```
      }
    ],
    "subject": "Последняя версия моей программы",
    "references": "",
    "reply_to": ""
  },
  "envelope": {
    "from_address": "ivanov@example.org",
    "recipients": ["username@example.com"]
  },
  "received": 1511421762.957363
}
```


4.1.1.10. Сообщение public_api.start

В таблице ниже описываются поля и объекты в сообщении `public_api.start` о начале обработки задания на проверку файла, полученного при помощи сервиса публичного API.

Таблица 29. Поля в сообщении public_api.start

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`scan_id`	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	—	—	UNIX-время генерации сообщения	Float

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
entry_point_id	—	—	Название источника для проверки в интерфейсе PT Sandbox	String
http_headers	—	user_agent x_forwarded_for referer	Блок данных с информацией об HTTP-заголовках в запросе приложения, отправившего файл на проверку	JSON object
user_agent	http_headers	—	Значение в HTTP-заголовке User-Agent	String
x_forwarded_for	http_headers	—	Значение в HTTP-заголовке X-Forwarded-For	String
referer	http_headers	—	Значение в HTTP-заголовке Referer	String
origin_ip	—	—	Первый IP-адрес из цепочки адресов, записанных в HTTP-заголовок X-Forwarded-For	String
peer_ip	—	—	Фактический IP-адрес приложения, отправившего файл на проверку	String
received	—	—	UNIX-время формирования задания на проверку	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - public_api.start - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
```

```
  "entry_point_id": "api-service",
  "file": {
    "mime_type": "",
    "md5": "11aced0fd6535f6e...1495ba1c7be00",
    "sha1": "45e50e2af429e44...6f59e46b18b60",
    "sha256": "23ef04408bb2c...7928e7caf3d7f",
    "size": 64000,
    "name": "software.exe"
  },
  "http_headers": {
    "user_agent": "curl/7.68.0",
    "x_forwarded_for": "198.51.100.1",
    "referer": ""
  },
  "origin_ip": "198.51.100.1",
  "peer_ip": "198.51.100.1",
  "received": 1511421762.957363
}
```


См. также

Сообщение public_api.finish (см. раздел 4.1.8.9)

4.1.1.11. Сообщение user_scan.start

В таблице ниже описываются поля и объекты в сообщении `user_scan.start` о начале обработки задания на проверку файлов, отправленных пользователем через веб-интерфейс.

Таблица 30. Поля в сообщении user_scan.start

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
scan_id	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	—	—	UNIX-время генерации сообщения	Float
user	—	id name login	Блок данных о пользователе, загрузившем файлы в продукт	JSON object
id	user	—	Идентификатор пользователя в продукте	UInt32
name	user	—	Фамилия, имя и отчество пользователя	String
login	user	—	Логин пользователя	String
http	—	client_ip user_agent	Блок данных об HTTP-запросе браузера при загрузке файла в продукт	JSON object
client_ip	http	—	IP-адрес клиента, с помощью которого был осуществлен запрос	String
user_agent	http	—	Название и версия браузера	String
received	—	—	UNIX-время формирования задания на проверку	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - user_scan.start - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "user": {
    "id": 534333,
    "name": "Ivan Ivanov",
    "login": "username"
  },
  "http": {
    "client_ip": "192.0.2.32",
    "user_agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36"
  },
  "received": 1511421762.957363
}
```

См. также

Сообщение user_scan.finish (см. раздел 4.1.8.10)

4.1.2. Сообщение new_artifact

В таблице ниже описываются поля и объекты в сообщении `new_artifact` об обнаружении файла в задании на проверку, в контейнере при его распаковке или в ходе поведенческого анализа.

Таблица 31. Поля в сообщении new_artifact

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
scan_id	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	—	—	UNIX-время генерации сообщения	Float
new_artifact	—	Один из JSON-объектов: — file; — url; — sandbox_drop; — sandbox_process_memory_dump; — sandbox_memory_dump	Информация о файле или ссылке, поступивших на проверку	JSON object
parent	—	file или url	Информация о файле-контейнере, например письме или архиве, из которого был извлечен файл, или о файле процесса, запуск которого привел к созданию артефакта поведенческого анализа	JSON object
file	new_artifact или parent	Поля с информацией о файле (см. раздел 4.1.10)	Информация о файле	JSON object
url	new_artifact или parent	value	Информация о ссылке	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
value	url	–	Значение	String
sandbox_drop	new_artifact	Поля с информацией о файле (см. раздел 4.1.10)	Информация о файле, который создал, удалил или с которым взаимодействовал проверяемый файл в ходе поведенческого анализа	JSON object
sandbox_process_memory_dump	new_artifact	Поля с информацией о файле (см. раздел 4.1.10)	Информация о дампе памяти, относящемся к процессу, за которым велось наблюдение в ходе поведенческого анализа. Дамп sandbox_process_memory_dump создается после штатного завершения процесса или по истечении времени наблюдения за файлом	JSON object
sandbox_memory_dump	new_artifact	Поля с информацией о файле (см. раздел 4.1.10)	Информация о дампе памяти, относящемся к процессу, за которым велось наблюдение в ходе поведенческого анализа. Дамп sandbox_memory_dump создается при наступлении события, значимого для поведенческого анализа	JSON object

Пример сообщения о файле, распознанном в задании на проверку объекта:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - new_artifact - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "new_artifact": {
    "file": {
      "mime_type": "",
      "md5": "11aced0fd6535f6e...1495ba1c7be00",
      "sha1": "45e50e2af429e44...6f59e46b18b60",
      "sha256": "23ef04408bb2c...7928e7caf3d7f",
      "size": 64000,
      "name": "software.exe"
    }
  }
}
```

Пример сообщения о ссылке, обнаруженной в задании на проверку объекта:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example - new_artifact - {
  "scan_id": "d2157391-6fd6-11ed-8002-cfcd208495d5",
  "created": 1669720395.66726,
  "new_artifact": {
    "url": {
      "value": "http://example.com"
    }
  }
}
```

Пример сообщения о создании дампа памяти `sandbox_process_memory_dump`:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - new_artifact - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "parent": {
    "file": {
      "sha256": "23ef04408bb2c...7928e7caf3d7f",
      "mime_type": "text/x-msdos-batch",
      "md5": "11aced0fd6535f6e...1495ba1c7be00",
      "sha1": "45e50e2af429e44...6f59e46b18b60",
      "name": "sc.exe",
      "size": 64000
    }
  },
  "new_artifact": {
    "sandbox_process_memory_dump": {
      "mime_type": "",
      "md5": "11aced0fd6535f6e...1495ba1c7be00",
      "sha1": "45e50e2af429e44...6f59e46b18b60",
      "sha256": "23ef04408bb2c...7928e7caf3d7f",
      "size": 7148968,
      "name": "",
      "process_name": "windows/system32/sc.exe",
      "process_id": 3060,
      "dump_trigger": "TerminateProcess",
      "dump_create_time": 1601287852.0006383
    }
```

```
    }
  }
```

Пример сообщения о создании дампа памяти `sandbox_memory_dump`:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - new_artifact - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "parent": {
    "file": {
      "sha256": "23ef04408bb2c...7928e7caf3d7f",
      "mime_type": "text/x-msdos-batch",
      "md5": "11aced0fd6535f6e...1495ba1c7be00",
      "sha1": "45e50e2af429e44...6f59e46b18b60",
      "name": "sc.exe",
      "size": 64000
    }
  },
  "new_artifact": {
    "sandbox_memory_dump": {
      "mime_type": "",
      "md5": "11aced0fd6535f6e...1495ba1c7be00",
      "sha1": "45e50e2af429e44...6f59e46b18b60",
      "sha256": "23ef04408bb2c...7928e7caf3d7f",
      "size": 16384,
      "name": "",
      "process_name": "windows/system32/sc.exe",
      "process_id": 1912,
      "dump_trigger": "Possible shellcode detected",
```

```
      "dump_create_time": 20.508
    }
  }
}
```


Пример сообщения об обнаружении другого артефакта, созданного в ходе поведенческого анализа:


```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - new_artifact - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "parent": {
    "file": {
      "sha256": "23ef04408bb2c...7928e7caf3d7f",
      "mime_type": "text/x-msdos-batch",
      "md5": "11aced0fd6535f6e...1495ba1c7be00",
      "sha1": "45e50e2af429e44...6f59e46b18b60",
      "name": "sc.exe",
      "size": 64000
    }
  },
  "new_artifact": {
    "sandbox_drop": {
      "mime_type": "",
      "md5": "11aced0fd6535f6e...1495ba1c7be00",
      "sha1": "45e50e2af429e44...6f59e46b18b60",
      "sha256": "23ef04408bb2c...7928e7caf3d7f",
      "size": 40,
      "process_name": "windows/system32/sppsvc.exe",
      "process_id": 4568,
```

```
        "dump_trigger": "",
        "dump_create_time": 30.801,
        "name": "users/john/appdata/local/google/chrome/user data/crashpad/settings.dat"
      }
    }
  }
```

См. также

Информация о файле в сообщениях syslog (см. раздел 4.1.10)

4.1.3. Сообщение scan_machine.new_object

В таблице ниже описываются поля и объекты в сообщении `scan_machine.new_object` с результатами извлечения файлов из контейнера (например, архива или письма).

Таблица 32. Поля в сообщении scan_machine.new_object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`scan_id`	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	—	—	UNIX-время генерации сообщения	Float
`origin_file`	—	Поля с информацией о файле (см. раздел 4.1.10)	Информация о файле, из которого были извлечены файлы	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
recognized_object	—	Один из объектов: file, email, archive или compressed_file	Информация об извлеченных файлах	JSON object
file	recognized_object	Поля с информацией о файле (см. раздел 4.1.10)	Информация о файле	JSON object
email	—	Информация об электронном письме в сообщениях syslog (см. раздел 4.1.9)		
archive	recognized_object	mime_type	Информация об архиве	JSON object
mime_type	archive	—	MIME-тип архива	String
compressed_file	recognized_object	mime_type file	Информация о сжатом файле	JSON object
mime_type	compressed_file	—	MIME-тип сжатого файла	String
file	compressed_file	Поля с информацией о сжатом файле (см. раздел 4.1.10)	Информация о файле, который был сжат	JSON object

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - scan_machine.new_object - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "origin_file": {
    "mime_type": "",
    "md5": "b53173def106ad...0ea30e504ec860f",
```

```
      "sha1": "c73c71d7858b7...de0272ce1125e01",
      "sha256": "d523708fe3b...e8f9a41b25f3c8d",
      "size": 792
    },
    "recognized_object": {
      "archive": {
        "mime_type": "application/zip; charset=binary"
      }
    }
  }
```

4.1.4. Сообщение scan_machine.file_result.av

В таблице ниже описываются поля и объекты в сообщении `scan_machine.file_result.av` с результатом сканирования файла конкретным антивирусом.

Таблица 33. Поля в сообщении scan_machine.file_result.av

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`scan_id`	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	—	—	UNIX-время генерации сообщения	Float
`file`	—	Поля с информацией о файле (см. раздел 4.1.10)	Информация о файле	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
av_code_name	—	—	Кодовое имя антивируса (см. раздел 4.3), который сканировал файл	String
engine_version	—	—	Версия антивируса, который сканировал файл	String
database_time	—	—	UNIX-время последнего обновления баз антивируса, который сканировал файл	Float
duration	—	—	Продолжительность сканирования файла антивирусом в секундах	Float
result	—	verdict errors state raw_detections	Результат сканирования конкретным антивирусом, ошибки сканирования и состояние сканирования	JSON object
verdict	result	threat_level threat accuracy	Данные о результате проверки	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
threat_level	verdict	—	Результат проверки. Возможные значения: — UNKNOWN — результат проверки неизвестен; — CLEAN — файл без обнаруженных угроз; — UNWANTED — потенциально опасный файл; — DANGEROUS — опасный файл	String
threat	verdict	classification family platform	Информация об обнаруженной угрозе	JSON object
classification	threat	—	Тип обнаруженной угрозы. Если угроза не была обнаружена, в это поле записывается UNKNOWN	String
family	threat	—	Семейство, к которому принадлежит обнаруженное вредоносное ПО	String

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
platform	threat	—	Семейство операционных систем, на которое нацелено вредоносное ПО. Возможные значения: — WINDOWS; — LINUX; — OSX; — ANDROID; — IOS; — NO_PLATFORM — семейство операционных систем не было определено	String
accuracy	verdict	—	Точность результата проверки	Float
errors	result	internal corrupted encrypted limit_exceeded	Ошибки проверки	JSON object
internal	errors	—	Внутренняя ошибка при проверке	Boolean
corrupted	errors	—	Файл поврежден	Boolean

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
encrypted	errors	—	Файл зашифрован паролем, который неизвестен продукту	Boolean
limit_exceeded	errors	—	Проверка файла остановлена по лимиту (например, превышен максимально допустимый уровень вложенности архивов или максимальный размер файла)	Boolean
state	result	—	Состояние проверки. Возможные значения: — UNSCANNED — файл не был проверен из-за ошибки; — PARTIAL — файл был проверен частично; — FULL — файл был проверен полностью	String
raw_detections	result	—	Необработанный результат сканирования (в исходном виде, полученном от антивируса)	Array

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - scan_machine.file_result.av - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "file": {
    "mime_type": "application/x-dosexec; charset=binary",
    "md5": "11aced0fd6535f6e...1495ba1c7be00",
    "name": "software.exe",
    "sha1": "45e50e2af429e44...6f59e46b18b60",
    "sha256": "Хеш-сумма файла, вычисленная по алгоритму SHA-256",
    "size": 64000
  },
  "av_code_name": "clamav",
  "engine_version": "0.99.2",
  "database_time": 1510892498.0,
  "duration": 0.1300000,
  "result": {
    "verdict": {
      "threat_level": "CLEAN",
      "threat": {
        "classification": "UNKNOWN",
        "platform": "NO_PLATFORM",
        "family": ""
      },
      "accuracy": 1.0
```

```
      },
      "errors": {
        "internal": false,
        "corrupted": false,
        "encrypted": false,
      },
      "state": "FULL",
      "raw_detections": []
    }
  }
```

4.1.5. Сообщение scan_machine.file_result.pt_sandbox

В таблице ниже описываются поля и объекты в сообщении `scan_machine.file_result.pt_sandbox` с результатами поведенческого анализа файла.

Таблица 34. Поля в сообщении scan_machine.file_result.pt_sandbox

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`scan_id`	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	—	—	UNIX-время генерации сообщения	Float

`image_info`	—	`name`
		`version`
		`os`
			Информация об образе виртуальной машины, который использовался для развертывания виртуальной машины, на которой проводился поведенческий анализ	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
name	image_info	—	Название образа виртуальной машины, который использовался для развертывания виртуальной машины, на которой проводился поведенческий анализ	String
version	image_info	—	Версия образа виртуальной машины, который использовался для развертывания виртуальной машины, на которой проводился поведенческий анализ	String
os	image_info	architecture locale name service_pack version	Информация об операционной системе образа виртуальной машины	JSON object
architecture	os	—	Архитектура процессора, которую поддерживает операционная система	String
locale	os	—	Локаль операционной системы	String
name	os	—	Название операционной системы	String
service_pack	os	—	Название пакета обновления операционной системы	String
version	os	—	Версия операционной системы	String

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
dpi_rules_version	—	—	Версия правил пакетного анализа трафика, которые использовались в ходе поведенческого анализа	String
correlation_rules_version	—	—	Версия правил корреляции, которые использовались в ходе поведенческого анализа	String
analysis_duration	—	—	Фактическое время поведенческого анализа в секундах	Float
analysis_planned_duration	—	—	Планируемое время поведенческого анализа в секундах	Float
result	—	verdict errors state	Итоговый результат поведенческого анализа с его ошибками и состоянием	JSON object
verdict	result	threat_level threat accuracy	Данные о результате проверки	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
threat_level	verdict	—	Результат проверки. Возможные значения: — UNKNOWN — результат проверки неизвестен; — CLEAN — файл без обнаруженных угроз; — UNWANTED — потенциально опасный файл; — DANGEROUS — опасный файл	String
threat	verdict	classification family platform	Информация об обнаруженной угрозе	JSON object
classification	threat	—	Тип обнаруженной угрозы. Если угроза не была обнаружена, в это поле записывается UNKNOWN	String
family	threat	—	Семейство, к которому принадлежит обнаруженное вредоносное ПО	String

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
platform	threat	—	Семейство операционных систем, на которое нацелено вредоносное ПО. Возможные значения: — WINDOWS; — LINUX; — OSX; — ANDROID; — IOS; — NO_PLATFORM — семейство операционных систем не было определено	String
accuracy	verdict	—	Точность результата проверки	Float
errors	result	internal corrupted encrypted limit_exceeded	Ошибки проверки	JSON object
internal	errors	—	Внутренняя ошибка при проверке	Boolean
corrupted	errors	—	Файл поврежден	Boolean

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
encrypted	errors	—	Файл зашифрован паролем, который неизвестен продукту	Boolean
limit_exceeded	errors	—	Проверка файла остановлена по лимиту (например, превышен максимально допустимый уровень вложенности архивов или максимальный размер файла)	Boolean
state	result	—	Состояние проверки. Возможные значения: — UNSCANNED — файл не был проверен из-за ошибки; — PARTIAL — файл был проверен частично; — FULL — файл был проверен полностью	String
detects	—	Строки с обнаруженным вредоносным ПО	Названия вредоносного ПО, обнаруженного в ходе поведенческого анализа	Array of strings
suspicious_behaviors	—	JSON-объекты, каждый из которых описывает конкретное подозрительное поведение в полях name и version	Перечень обнаруженного подозрительного поведения	Array of JSON objects
name	suspicious_behaviors	—	Внутреннее название обнаруженного подозрительного поведения	String

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
version	suspicious_behaviors	—	Версия правила, при помощи которого было обнаружено подозрительное поведение	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - scan_machine.file_result.pt_sandbox - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "file": {
    "mime_type": "application/x-dosexec; charset=binary",
    "md5": "11aced0fd6535f6e...1495ba1c7be00",
    "name": "software.exe",
    "sha1": "45e50e2af429e44...6f59e46b18b60",
    "sha256": "23ef04408bb2c...7928e7caf3d7f",
    "size": 33
  },
  "image_name": "win7-sp1-x86",
  "image_info": {
    "name": "win7-sp1-x86",
    "version": "1.0.0.50",
    "os": {
      "name": "Windows 7 ENTERPRISE",
      "version": "6.1.7601",
      "architecture": "x86",
      "service_pack": "Service Pack 1",
```

```
    "locale": "en-US"
  }
},
"dpi_rules_version": "1.0.0.138",
"correlation_rules_version": "1.0.0.524",
"analysis_duration": 240.0,
"analysis_planned_duration": 240.0,
"result": {
  "verdict": {
    "threat_level": "CLEAN",
    "threat": {
      "classification": "UNKNOWN",
      "platform": "NO_PLATFORM",
      "family": ""
    },
    "accuracy": 0.0
  },
  "errors": {
    "internal": false,
    "corrupted": false,
    "encrypted": false,
  },
  "state": "FULL"
},
"detects": [],
"suspicious_behaviors": [
  {
    "name": "Read.System.GetCursorPos.CheckVM",
```

```
      "version": ""
    },
    {
      "name": "Read.File.Name.Enumeration",
      "version": ""
    }
  ]
}
```

4.1.6. Сообщение scan_machine.file_result.melded

В таблице ниже описываются поля и объекты в сообщении `scan_machine.file_result.melded` с итоговым результатом проверки файла с учетом результатов проверки всех извлеченных из него файлов и артефактов поведенческого анализа.

Таблица 35. Поля в сообщении scan_machine.file_result.melded

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`scan_id`	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	—	—	UNIX-время генерации сообщения	Float
`file`	—	Поля с информацией о файле (см. раздел 4.1.10)	Информация о проверяемом файле. В сообщении может быть только один проверяемый объект `file` или `url`	JSON object
`url`	—	`value`	Информация о проверяемой ссылке. В сообщении может быть только один проверяемый объект `file` или `url`	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
value	url	—	Ссылка	String
network_objects	—	url domain ip	Обнаруженные сетевые объекты	JSON object
url	network_objects	value	Ссылка	JSON object
domain	network_objects	value	Имя домена	JSON object
ip	network_objects	value	IP-адрес	JSON object
value	url domain ip	—	Значение	String
result	—	verdict errors state	Итоговый результат проверки файла с учетом результатов проверки всех из-влеченных из него файлов и артефактов поведенческого анализа	JSON object
verdict	result	threat_level threat accuracy	Данные о результате проверки	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
threat_level	verdict	—	Результат проверки. Возможные значения: — UNKNOWN — результат проверки неизвестен; — CLEAN — файл без обнаруженных угроз; — UNWANTED — потенциально опасный файл; — DANGEROUS — опасный файл	String
threat	verdict	classification family platform	Информация об обнаруженной угрозе	JSON object
classification	threat	—	Тип обнаруженной угрозы. Если угроза не была обнаружена, в это поле записывается UNKNOWN	String
family	threat	—	Семейство, к которому принадлежит обнаруженное вредоносное ПО	String

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
platform	threat	—	Семейство операционных систем, на которое нацелено вредоносное ПО. Возможные значения: — WINDOWS; — LINUX; — OSX; — ANDROID; — IOS; — NO_PLATFORM — семейство операционных систем не было определено	String
accuracy	verdict	—	Точность результата проверки	Float
errors	result	internal corrupted encrypted limit_exceeded	Ошибки проверки	JSON object
internal	errors	—	Внутренняя ошибка при проверке	Boolean
corrupted	errors	—	Файл поврежден	Boolean

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
encrypted	errors	—	Файл зашифрован паролем, который неизвестен продукту	Boolean
limit_exceeded	errors	—	Проверка файла остановлена по лимиту (например, превышен максимально допустимый уровень вложенности архивов или максимальный размер файла)	Boolean
state	result	—	Состояние проверки. Возможные значения: — UNSCANNED — файл не был проверен из-за ошибки; — PARTIAL — файл был проверен частично; — FULL — файл был проверен полностью	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - scan_machine.file_result.melded - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "file": {
    "mime_type": "application/x-dosexec; charset=binary",
```

```
    "md5": "11aced0fd6535f6e...1495ba1c7be00",
    "name": "software.exe",
    "sha1": "45e50e2af429e44...6f59e46b18b60",
    "sha256": "23ef04408bb2c...7928e7caf3d7f",
    "size": 64000
  },
  "result": {
    "verdict": {
      "threat_level": "CLEAN",
      "threat": {
        "classification": "UNKNOWN",
        "platform": "NO_PLATFORM",
        "family": ""
      },
      "accuracy": 1.0
    },
    "errors": {
      "internal": false,
      "corrupted": false,
      "encrypted": false,
    },
    "state": "FULL"
  }
}
```


4.1.7. Сообщение scan_machine.final_result

В таблице ниже описываются поля и объекты в сообщении `scan_machine.final_result` с итоговым результатом проверки всех файлов задания.

Таблица 36. Поля в сообщении scan_machine.final_result

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
scan_id	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	—	—	UNIX-время генерации сообщения	Float
result	—	verdict errors state	Итоговый результат проверки с ошибками и состоянием проверки	JSON object
verdict	result	threat_level threat accuracy	Данные о результате проверки	JSON object
threat_level	verdict	—	Результат проверки. Возможные значения: — UNKNOWN — результат проверки неизвестен; — CLEAN — файл без обнаруженных угроз; — UNWANTED — потенциально опасный файл; — DANGEROUS — опасный файл	String
threat	verdict	classification	Информация об обнаруженной угрозе	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
		family platform		
classification	threat	—	Тип обнаруженной угрозы. Если угроза не была обнаружена, в это поле записывается UNKNOWN	String
family	threat	—	Семейство, к которому принадлежит обнаруженное вредоносное ПО	String
platform	threat	—	Семейство операционных систем, на которое нацелено вредоносное ПО. Возможные значения: — WINDOWS; — LINUX; — OSX; — ANDROID; — IOS; — NO_PLATFORM — семейство операционных систем не было определено	String
accuracy	verdict	—	Точность результата проверки	Float
errors	result	internal	Ошибки проверки	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
		corrupted encrypted limit_exceeded		
internal	errors	—	Внутренняя ошибка при проверке	Boolean
corrupted	errors	—	Файл поврежден	Boolean
encrypted	errors	—	Файл зашифрован паролем, который неизвестен продукту	Boolean
limit_exceeded	errors	—	Проверка файла остановлена по лимиту (например, превышен максимально допустимый уровень вложенности архивов или максимальный размер файла)	Boolean
state	result	—	Состояние проверки. Возможные значения: — UNSCANNED — файл не был проверен из-за ошибки; — PARTIAL — файл был проверен частично; — FULL — файл был проверен полностью	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - scan_machine.final_result - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "result": {
    "verdict": {
      "threat_level": "CLEAN",
      "threat": {
        "classification": "UNKNOWN",
        "platform": "NO_PLATFORM",
        "family": ""
      },
      "accuracy": 1.0
    },
    "errors": {
      "internal": false,
      "corrupted": false,
      "encrypted": false,
    },
    "state": "FULL"
  }
}
```

4.1.8. Сообщения <Идентификатор типа источника для проверки>.finish

Сообщение типа `<Идентификатор типа источника для проверки>.finish` (например, `email.finish`) информирует об окончании обработки задания на проверку и содержит уникальный идентификатор задания, признак успешности обработки задания.

В этом разделе

См. также

Идентификаторы типов источников для проверки (см. раздел 4.1.11)

4.1.8.1. Сообщение check_me.finish

В таблице ниже описываются поля в сообщении `check_me.finish` о завершении обработки задания на проверку файлов, полученных службой Checkme.

Таблица 37. Поля в сообщении check_me.finish

Поле	Описание	Тип данных
`scan_id`	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	UNIX-время генерации сообщения	Float

Поле	Описание	Тип данных
status	Состояние обработки задания на проверку. Возможные значения: — UNKNOWN — задание еще не обработано; — SUCCESS — задание обработано успешно; — FAIL — обработка задания завершилась с ошибкой	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - check_me.finish - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "status": "SUCCESS"
}
```

4.1.8.2. Сообщение dpi.finish

В таблице ниже описываются поля в сообщении dpi.finish о завершении обработки задания на проверку файла, полученного от PT NAD.

Таблица 38. Поля в сообщении dpi.finish

Поле	Описание	Тип данных
scan_id	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	UNIX-время генерации сообщения	Float
status	Состояние обработки задания на проверку. Возможные значения: — UNKNOWN — задание еще не обработано; — SUCCESS — задание обработано успешно; — FAIL — обработка задания завершилась с ошибкой	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - dpi.finish - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "status": "SUCCESS"
}
```

4.1.8.3. Сообщение email.finish

В таблице ниже описываются поля в сообщении `email.finish` о завершении обработки задания на проверку файлов, полученных почтовым агентом.

Таблица 39. Поля в сообщении email.finish

Поле	Описание	Тип данных
`scan_id`	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	UNIX-время генерации сообщения	Float
`status`	Состояние обработки задания на проверку. Возможные значения: — `UNKNOWN` — задание еще не обработано; — `SUCCESS` — задание обработано успешно; — `FAIL` — обработка задания завершилась с ошибкой	String
`action`	Действие, которое PT Sandbox выполнил с письмом, обработанным в задании. Возможные значения: — `UNKNOWN` — действие неизвестно; — `PASS` — письмо пропущено в информационную систему вместе с вложениями; — `BLOCK` — распространение письма было заблокировано (только если источник работает в блокирующем режиме); — `NOTHING` — во время проверки произошла ошибка и письмо было пропущено в информационную систему вместе с вложениями	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - email.finish - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "status": "SUCCESS",
  "action": "BLOCK"
}
```

4.1.8.4. Сообщение files_inbox.finish

В таблице ниже описываются поля и объекты в сообщении `files_inbox.finish` о завершении обработки задания на проверку файла, обнаруженного в папке-шлюзе.

Таблица 40. Поля в сообщении files_inbox.finish

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
scan_id	—	—	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	—	—	UNIX-время генерации сообщения	Float
status	—	—	Состояние обработки задания на проверку. Возможные значения: — UNKNOWN — задание еще не обработано; — SUCCESS — задание обработано успешно; — FAIL — обработка задания завершилась с ошибкой	String
action	—	—	Действие, которое PT Sandbox выполнил с файлом по результатам его проверки. Возможные значения: — PASS — файл был перемещен в папку для безопасных файлов; — BLOCK — файл был перемещен в папку карантина; — NOTHING — во время проверки произошла ошибка и файл был перемещен в папку для безопасных файлов; — UNKNOWN — действие неизвестно	String
dst_file_info	—	url	Блок данных о перемещенном файле	JSON object

Поле (объект)	Родительский объект	Вложенные поля (объекты)	Описание	Тип данных
`url`	`dst_file_info`	—	Путь к перемещенному файлу в папке для безопасных файлов или в папке карантина	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - files_inbox.finish - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "status": "SUCCESS",
  "action": "NOTHING",
  "dst_file_info": {
    "url": "smb://host/share/safe/software.exe"
  }
}
```

4.1.8.5. Сообщение files_monitor.finish

В таблице ниже описываются поля в сообщении `files_monitor.finish` о завершении обработки задания на проверку файла, обнаруженного в общей папке.

Таблица 41. Поля в сообщении files_monitor.finish

Поле	Описание	Тип данных
`scan_id`	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	UNIX-время генерации сообщения	Float
`status`	Состояние обработки задания на проверку. Возможные значения: — `UNKNOWN` — задание еще не обработано; — `SUCCESS` — задание обработано успешно; — `FAIL` — обработка задания завершилась с ошибкой	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - files_monitor.finish - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "status": "SUCCESS"
}
```

4.1.8.6. Сообщение icap.finish

В таблице ниже описываются поля в сообщении `icap.finish` о завершении обработки задания на проверку контента, полученного от ICAP-сервера PT Sandbox.

Таблица 42. Поля в сообщении icap.finish

Поле	Описание	Тип данных
`scan_id`	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	UNIX-время генерации сообщения	Float
`status`	Состояние обработки задания на проверку. Возможные значения: — `UNKNOWN` — задание еще не обработано; — `SUCCESS` — задание обработано успешно; — `FAIL` — обработка задания завершилась с ошибкой	String
`action`	Действие, которое PT Sandbox выполнил с контентом, обработанным в задании. Возможные значения: — `UNKNOWN` — действие неизвестно; — `PASS` — контент пропущен в информационную систему с указанием типа обнаруженной угрозы в поле X-Virus-ID ответа; — `BLOCK` — распространение контента было заблокировано (только если источник работает в блокирующем режиме); — `NOTHING` — во время проверки произошла ошибка и контент был пропущен в информационную систему	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - icap.finish - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "status": "SUCCESS",
  "action": "BLOCK"
}
```

4.1.8.7. Сообщение mail_bcc.finish

В таблице ниже описываются поля в сообщении `mail_bcc.finish` о завершении обработки задания на проверку письма, полученного от почтового сервера организации в виде скрытой копии.

Таблица 43. Поля в сообщении mail_bcc.finish

Поле	Описание	Тип данных
scan_id	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	UNIX-время генерации сообщения	Float
status	Состояние обработки задания на проверку. Возможные значения: — UNKNOWN — задание еще не обработано; — SUCCESS — задание обработано успешно; — FAIL — обработка задания завершилась с ошибкой	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - mail_bcc.finish - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "status": "SUCCESS"
}
```

См. также

Сообщение mail_bcc.start (см. раздел 4.1.1.8)

4.1.8.8. Сообщение mail_gateway.finish

В таблице ниже описываются поля в сообщении mail_gateway.finish о завершении обработки задания на проверку письма, полученного от почтового сервера Postfix или Exim в режиме фильтрации.

Таблица 44. Поля в сообщении mail_gateway.finish

Поле	Описание	Тип данных
scan_id	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	UNIX-время генерации сообщения	Float
status	Состояние обработки задания на проверку. Возможные значения: — UNKNOWN — задание еще не обработано; — SUCCESS — задание обработано успешно; — FAIL — обработка задания завершилась с ошибкой	String

Поле	Описание	Тип данных
action	Действие, которое PT Sandbox выполнил с письмом, обработанным в задании. Возможные значения: — UNKNOWN — действие неизвестно; — PASS — письмо пропущено в информационную систему вместе с вложениями; — BLOCK — распространение письма было заблокировано (только если источник работает в блокирующем режиме); — NOTHING — во время проверки произошла ошибка и письмо было пропущено в информационную систему вместе с вложениями	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - mail_gateway.finish - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "status": "SUCCESS",
  "action": "BLOCK"
}
```

4.1.8.9. Сообщение public_api.finish

В таблице ниже описываются поля в сообщении public_api.finish о завершении обработки задания на проверку файла, полученного при помощи сервиса публичного API.

Таблица 45. Поля в сообщении public_api.finish

Поле	Описание	Тип данных
scan_id	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
created	UNIX-время генерации сообщения	Float
status	Состояние обработки задания на проверку. Возможные значения: — UNKNOWN — задание еще не обработано; — SUCCESS — задание обработано успешно; — FAIL — обработка задания завершилась с ошибкой	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - public_api.finish - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "status": "SUCCESS"
}
```

См. также

Сообщение public_api.start (см. раздел 4.1.1.10)

4.1.8.10. Сообщение user_scan.finish

В таблице ниже описываются поля в сообщении `user_scan.finish` о завершении обработки задания на проверку файлов, отправленных пользователем через веб-интерфейс.

Таблица 46. Поля в сообщении user_scan.finish

Поле	Описание	Тип данных
`scan_id`	Хеш-сумма задания на проверку, вычисленная по алгоритму продукта	String
`created`	UNIX-время генерации сообщения	Float
`status`	Состояние обработки задания на проверку. Возможные значения: — `UNKNOWN` — задание еще не обработано; — `SUCCESS` — задание обработано успешно; — `FAIL` — обработка задания завершилась с ошибкой	String

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - user_scan.finish - {
  "scan_id": "a36a1f7c-51af-32b1-a144-659b1dfe31c1",
  "created": 1511421763.3728465,
  "status": "SUCCESS"
}
```

См. также

Сообщение user_scan.start (см. раздел 4.1.1.11)

4.1.9. Информация об электронном письме в сообщениях syslog

В сообщениях `check_me.start`, `email.start`, `mail_bcc.start`, `mail_gateway.start` и `scan_machine.new_object` в JSON-объектах `email` и `envelope` содержится информация об электронном письме.

Таблица 47. JSON-объекты с информацией об электронном письме

JSON-объект или поле	Родительский JSON-объект или сообщение	Вложенные поля или объекты	Описание	Тип данных
email	check_me.start (см. раздел 4.1.1.1) email.start (см. раздел 4.1.1.3) mail_bcc.start (см. раздел 4.1.1.8) mail_gateway.start (см. раздел 4.1.1.9) scan_machine.new_object (см. раздел 4.1.3)	id from_address to_list cc_list bcc_list subject references reply_to	Информация о письме	JSON object
id	email	—	Поле заголовка Message-ID	String
from_address	email	address name	Информация об отправителе письма	JSON object

JSON-объект или поле	Родительский JSON-объект или сообщение	Вложенные поля или объекты	Описание	Тип данных
`to`	`email`	`address` `name`	Информация о получателе письма (первом из списка)	JSON object
`to_list`	`email`	Информация о каждом получателе в отдельном JSON-объекте с полями `address` и `name`	Информация о получателях письма	Array of JSON objects
`cc`	`email`	`address` `name`	Информация о получателе копии письма (первом из списка)	JSON object
`cc_list`	`email`	Информация о каждом получателе в отдельном JSON-объекте с полями `address` и `name`	Информация о получателях копии письма	Array of JSON objects
`bcc`	`email`	`address` `name`	Информация о получателе скрытой копии письма (первом из списка)	JSON object
`bcc_list`	`email`	Информация о каждом получателе в отдельном JSON-объекте с полями `address` и `name`	Информация о получателях скрытой копии письма	Array of JSON objects
`address`	`from_address`	—	Адрес электронной почты	String

JSON-объект или поле	Родительский JSON-объект или сообщение	Вложенные поля или объекты	Описание	Тип данных
	to_list cc_list bcc_list			
name	from_address to_list cc_list bcc_list	—	Имя отправителя (получателя) письма	String
subject	email	—	Тема письма	String
references	email	—	Идентификационное поле заголовка references согласно RFC 5322	String
reply_to	email	—	Идентификационное поле заголовка in-reply-to согласно RFC 5322	String
envelope	check_me.start (см. раздел 4.1.1.1) email.start (см. раздел 4.1.1.3) mail_bcc.start (см. раздел 4.1.1.8)	from_address recipients	Данные из поля Envelope электронного письма	JSON object

JSON-объект или поле	Родительский JSON-объект или сообщение	Вложенные поля или объекты	Описание	Тип данных
	mail_gateway.start (см. раздел 4.1.1.9) (см. раздел 4.1.3)			
`from_address`	`envelope`	—	Адрес электронной почты отправителя письма	String
`recipients`	`envelope`	Строки	Данные о получателях письма	Array of strings

4.1.10. Информация о файле в сообщениях syslog

При составлении сообщений syslog PT Sandbox записывает информацию о файлах в поля, описанные в таблице ниже.

Таблица 48. Поля с информацией о файле

Поле	Описание	Тип данных	Примечание
`mime_type`	MIME-тип файла	String	Поле не заполняется (всегда пустая строка) в JSON-объектах сообщений `new_artifact` и `public_api.start`. В остальных сообщениях поле может быть не заполнено
`md5`	Хеш-сумма файла, вычисленная по алгоритму MD5	String	—
`sha1`	Хеш-сумма файла, вычисленная по алгоритму SHA-1	String	—
`sha256`	Хеш-сумма файла, вычисленная по алгоритму SHA-256	String	—
`size`	Размер файла в байтах	UInt64	—
`name`	Имя файла	String	Поле не заполняется (всегда пустая строка) в JSON-объектах `new_artifact` → `sandbox_process_memory_dump` и `new_artifact` → `sandbox_memory_dump` сообщения `new_artifact`

Поля, относящиеся к артефактам поведенческого анализа

Если файл является артефактом поведенческого анализа, в соответствующий JSON-объект (`sandbox_drop`, `sandbox_process_memory_dump`, `sandbox_memory_dump` в объекте `new_artifact` сообщения `new_artifact`) записываются дополнительные поля.

Таблица 49. Поля артефактов поведенческого анализа

Поле	Описание	Тип данных	Примечание
`process_name`	Название процесса, по которому был создан артефакт	String	—
`process_id`	Идентификатор процесса, по которому был создан артефакт	UInt32	—
`dump_trigger`	Название события, которое спровоцировало создание дампа памяти (например, `TerminateProcess` или `Possible shellcode detected`)	String	Поле записывается только в объектах `sandbox_process_memory_dump` и `sandbox_memory_dump`
`dump_create_time`	UNIX-время создания артефакта	Float	—

4.1.11. Идентификаторы типов источников для проверки

Идентификаторами типа источника для проверки могут быть:

- `check_me` — служба Checkme;
- `dpi` — PT NAD;
- `email` — Microsoft Exchange Server с установленным почтовым агентом PT Sandbox;
- `file_inbox` — папка-шлюз;
- `files_monitor` — общая папка;
- `icap` — ICAP-сервер;
- `mail-bcc` — почтовый сервер организации, отправляющий скрытые копии писем в PT Sandbox;
- `mail-gateway` — почтовый сервер Postfix или Exim, отправляющий письма на фильтрацию в PT Sandbox;

— `public_api` — публичный API;

— `user_scan` — веб-интерфейс.

4.2. Сообщение av.update

В таблице ниже описываются поля в сообщении `av.update`, которое формируется при обновлении антивируса и (или) его базы.

Таблица 50. Поля в сообщении av.update

Поле	Описание	Тип данных
`created`	UNIX-время формирования сообщения	Float
`av_code_name`	Кодовое имя антивируса	String
`engine_version`	Версия антивируса	String
`database_time`	UNIX-время выпуска антивирусной базы	Float

Пример сообщения:

```
<100>1 2017-11-23T07:22:44.018041Z host1-example sandbox - av.update - {
  "created": 1520866088.3110101,
  "av_code_name": "bitdefender",
  "engine_version": "11.0.1.18",
  "database_time": 1520862339.0
}
```

4.3. Кодовые имена антивирусов

В таблице ниже приводится список кодовых имен антивирусов. Кодовое имя используется для обозначения антивируса в сообщениях `av.update` и `scan_machine.file_result.av`.

Таблица 51. Кодовые имена антивирусов

Антивирус	Кодовое имя
Avira	`avira`
Bitdefender	`bitdefender`
ClamAV	`clamav`
ESET	`eset`
Kaspersky for Proxy Server	`kaspersky`
Avast Core Security	`avast`
Symantec Protection Engine for Network Attached Storage	`symantec`

Антивирус	Кодовое имя
Dr.Web Server Security Suite	`drweb`

5. Создание образов ВМ

Для создания и настройки Xen-образов для поведенческого анализа файлов в PT Sandbox используется утилита ImageBuilder. С помощью утилиты вы можете:

— создать образ ВМ Windows из ISO-файла с установочным образом ОС;

— создать образ ВМ Windows из готового образа ОС в формате QCOW2;

— скопировать и настроить образ ВМ, уже добавленный в PT Sandbox.

Внимание! Для работы утилиты ImageBuilder требуется доступ через интернет к сайту msdl.microsoft.com.

Утилита представляет собой приложение для работы из командной строки. На узлах с функцией поведенческого анализа исполняемый файл утилиты по умолчанию расположен в каталоге `/opt/ptms/sbin/imagebuilder`. Экранную справку утилиты вы можете вызвать с помощью команды `/opt/ptms/sbin/imagebuilder --help`.

Примечание. При возникновении проблем с утилитой ImageBuilder, отправьте файл журнала `imagebuilder.log`, расположенный в текущем рабочем каталоге, в службу технической поддержки.

В этом разделе

5.1. Предварительная настройка

Утилиту ImageBuilder необходимо запускать на узле с PT Sandbox. Для настройки созданного образа ВМ на узле PT Sandbox необходимо установить и настроить WebVNC-сервер.

Внимание! Для доступа в веб-интерфейс VNC-сервера на узле PT Sandbox необходимо разрешить подключение к порту 8081 (или к другому порту, указанному при запуске сервера).

Установка и настройка VNC-сервера

Внимание! Для установки сервера WebVNC требуется доступ к сайту github.com через интернет.

► Чтобы установить и настроить сервер WebVNC,

выполните команды:

```
wget https://github.com/pgaskin/easy-novnc/releases/download/v1.1.0/easy-novnc_linux-64bit
chmod +x easy-novnc_linux-64bit
mv easy-novnc_linux-64bit /usr/local/bin/easy-novnc
```

Сервер WebVNC установлен и настроен.

5.2. Создание образа ВМ из ISO-файла

Внимание! Создание образа ВМ из ISO-файла поддерживается только для пользовательских Windows версий 7—10 и серверных Windows Server версий 2008—2019. Поддерживаются версии ОС только с 64-разрядной архитектурой.

► Чтобы создать образ ВМ из ISO-файла:

1. На узле, с которого производится настройка, запустите терминальный клиент, поддерживающий протокол SSH.
2. Создайте образ ВМ и установите на него ОС, выполнив команду:

   ```
   imagebuilder build-custom-image --vm-name <Название временной ВМ> --image-dir <Каталог для сохранения образа ВМ> --os-type windows --os-iso <Путь к ISO-файлу> --boot-mode <Режим загрузки ОС> --expose-vnc <Порт подключения к VNC-серверу>
   ```

   В параметре `--image-dir` каталог для сохранения образа должен отличаться от `/opt/ptms/sandbox-images`. Имя каталога должно начинаться с префикса `custom_` и может содержать латинские буквы, цифры и знак подчеркивания.

   В параметре `--boot-mode` необходимо указать режим загрузки Windows с образа ВМ: `uefi` или `bios`.

   Пример:

   ```
   imagebuilder build-custom-image --vm-name custom_win7 --image-dir /tmp/custom_win7 --os-type windows --os-iso /home/win7.iso --boot-mode uefi --expose-vnc 5900
   ```

   Начнется установка и загрузка ОС.
3. На узле, с которого производится настройка, запустите еще один экземпляр терминального клиента.
4. Запустите VNC-сервер, выполнив команду:

   ```
   easy-novnc --addr :<Порт веб-интерфейса VNC-сервера> --basic-ui --no-url-password --host <IP-адрес PT Sandbox> --port <Порт подключения к VNC-серверу>
   ```

   Пример:

   ```
   easy-novnc --addr :8081 --basic-ui --no-url-password --host 10.22.1.4 --port 5900
   ```
5. Откройте браузер и в адресной строке введите:

   ```
   http://<IP-адрес PT Sandbox>:<Порт веб-интерфейса VNC-сервера>
   ```

   Пример:

   ```
   http://10.22.1.4:8081
   ```

6. На открывшейся странице нажмите кнопку **Connect**.

   Запустится мастер установки Windows.

7. Следуйте указаниям мастера до завершения установки.

   После перезагрузки откроется рабочий стол Window.

   Внимание! Если в ОС установлен антивирус или другое ПО для защиты информации отключите или удалите его.

8. Откройте интерфейс командной строки Windows от имени администратора и запустите настройку ОС, выполнив команду:

   ```
   <Имя виртуального CD-привода>:\Invoke-PostInstall.cmd
   ```

   Начнется настройка и перезагрузка ОС.

9. Если требуется, вручную настройте ОС на образе ВМ и установите ПО (см. раздел 5.5).

10. Скопируйте каталог с файлами образа ВМ на каждый узел PT Sandbox с функцией поведенческого анализа в каталог `/opt/ptms/sandbox-images`:

    ```
    scp -r <Каталог с образом ВМ> <Логин на узле>@<IP-адрес PT Sandbox>:/opt/ptms/sandbox-images/<Имя каталога для образа ВМ на узле>
    ```

    Имя каталога с файлами образа ВМ на узле PT Sandbox должно начинаться с префикса `custom_` и может содержать латинские буквы, цифры и знак подчеркивания.

    Пример:

    ```
    scp -r /tmp/custom_win7 Administrator@192.0.3.11:/opt/ptms/sandbox-images/custom_win7
    ```

Образ ВМ из ISO-файла создан и установлен в PT Sandbox.

5.3. Создание образа ВМ из образа диска в формате QCOW2

Внимание! Создание образа ВМ из образа диска в формате QCOW2 поддерживается только для Windows 10.

Примечание. Вы можете конвертировать образ диска в формат QCOW2 с помощью сторонних утилит. Например, из формата RAW с помощью утилиты qemu-img командой `qemu-img convert -c -O qcow2 -o cluster_size=64K image.raw image.qcow2`.

► Чтобы создать образ ВМ Windows из образа диска в формате QCOW2:

1. На узле, с которого производится настройка, запустите терминальный клиент, поддерживающий протокол SSH.
2. Импортируйте образ ВМ из образа диска в формате QCOW2, выполнив команду:

   ```
   imagebuilder import-custom-image --vm-name <Название временной ВМ> --image-dir <Каталог для сохранения образа> --os-type windows --disk-drive <Путь к образу в формате QCOW2> --expose-vnc <Порт подключения к VNC-серверу>
   ```

   В параметре `--image-dir` каталог для сохранения образа должен отличаться от `/opt/ptms/sandbox-images`. Имя каталога должно начинаться с префикса `custom_` и может содержать латинские буквы, цифры и знак подчеркивания.

   Пример:

   ```
   imagebuilder import-custom-image --vm-name custom_win7 --image-dir /tmp/custom_win7 --os-type windows --disk-drive /home/win7.qcow2 --expose-vnc 5900
   ```

   Начнется импорт образа ВМ и загрузка ОС.
3. На узле, с которого производится настройка, запустите еще один экземпляр терминального клиента.
4. Запустите VNC-сервер, выполнив команду:

   ```
   easy-novnc --addr :<Порт веб-интерфейса VNC-сервера> --basic-ui --no-url-password --host <IP-адрес PT Sandbox> --port <Порт подключения к VNC-серверу>
   ```

   Пример:

   ```
   easy-novnc --addr :8081 --basic-ui --no-url-password --host 10.22.1.4 --port 5900
   ```
5. Откройте браузер и в адресной строке введите:

   ```
   http://<IP-адрес PT Sandbox>:<Порт веб-интерфейса VNC-сервера>
   ```

   Пример:

   ```
   http://10.22.1.4:8081
   ```
6. На открывшейся странице нажмите кнопку **Connect**.

   Откроется рабочий стол Windows.

   Внимание! Если в ОС установлен антивирус или другое ПО для защиты информации отключите или удалите его.
7. Откройте интерфейс командной строки Windows от имени администратора и запустите настройку ОС, выполнив команду:

   ```
   <Имя виртуального CD-привода>:\Invoke-PostInstall.cmd
   ```

   Начнется настройка и перезагрузка ОС.

8. Если требуется, вручную настройте ОС на образе ВМ и установите ПО (см. раздел 5.5).
9. Скопируйте каталог с файлами образа ВМ на каждый узел PT Sandbox с функцией поведенческого анализа в каталог `/opt/ptms/sandbox-images`:

   ```
   scp -r <Каталог с образом ВМ> <Логин на узле>@<IP-адрес PT Sandbox>:/opt/ptms/sandbox-images/<Имя каталога для образа ВМ на узле>
   ```

   Имя каталога с файлами образа ВМ на узле PT Sandbox должно начинаться с префикса `custom_` и может содержать латинские буквы, цифры и знак подчеркивания.

   Пример:

   ```
   scp -r /tmp/custom_win7 Administrator@192.0.3.11:/opt/ptms/sandbox-images/custom_win7
   ```

Образ ВМ из образа диска в формате QCOW2 создан и установлен в PT Sandbox.

5.4. Копирование добавленного ранее образа ВМ

Вы можете скопировать и настроить образ ВМ, который уже добавлен в PT Sandbox.

► Чтобы скопировать образ ВМ из образа, добавленного ранее в PT Sandbox:

1. На узле, с которого производится настройка, запустите терминальный клиент, поддерживающий протокол SSH.
2. Скопируйте образ ВМ, выполнив команду:

   ```
   imagebuilder customize-image --vm-name <Название временной ВМ> --image-name <Имя исходного образа ВМ> --customized-image-name <Имя нового образа ВМ> --tmp-dir <Каталог для временных файлов> --expose-vnc <Порт подключения к VNC-серверу>
   ```

   Примечание. Исходный образ ВМ должен находиться в каталоге `/opt/ptms/sandbox-images`.

   Пример:

   ```
   imagebuilder customize-image --vm-name custom_win7 --image-name win7-sp1-x64 --customized-image-name win7-sp1-x64-modified --tmp-dir /tmp/custom_win7 --expose-vnc 5900
   ```

   Образ ВМ будет скопирован в указанный каталог для временных файлов.
3. На узле, с которого производится настройка, запустите еще один экземпляр терминального клиента.
4. Запустите VNC-сервер, выполнив команду:

   ```
   easy-novnc --addr :<Порт веб-интерфейса VNC-сервера> --basic-ui --no-url-password --host <IP-адрес PT Sandbox> --port <Порт подключения к VNC-серверу>
   ```

   Пример:

   ```
   easy-novnc --addr :8081 --basic-ui --no-url-password --host 10.22.1.4 --port 5900
   ```
5. Откройте браузер и в адресной строке введите:

   ```
   http://<IP-адрес PT Sandbox>:<Порт веб-интерфейса VNC-сервера>
   ```

   Пример:

   ```
   http://10.22.1.4:8081
   ```

6. На открывшейся странице нажмите кнопку **Connect**.

   Откроется рабочий стол Windows.

7. Вручную настройте ОС на образе ВМ и установите ПО (см. раздел 5.5).

8. Скопируйте каталог с файлами образа ВМ на каждый узел PT Sandbox с функцией поведенческого анализа в каталог `/opt/ptms/sandbox-images`:

   ```
   scp -r <Каталог с образом ВМ> <Логин на узле>@<IP-адрес PT Sandbox>:/opt/ptms/sandbox-images/<Имя каталога для образа ВМ на узле>
   ```

   Имя каталога с файлами образа ВМ на узле PT Sandbox должно начинаться с префикса `custom_` и может содержать латинские буквы, цифры и знак подчеркивания.

   Пример:

   ```
   scp -r /tmp/custom_win7 Administrator@192.0.3.11:/opt/ptms/sandbox-images/custom_win7
   ```

Образ ВМ скопирован и установлен в PT Sandbox.

5.5. Ручная настройка ОС на образе ВМ

Перед ручной настройкой ОС необходимо создать или скопировать образ ВМ. При ручной настройке ОС вы можете, например, установить ПО, изменить язык интерфейса и активировать Windows.

Внимание! В ОС на образах ВМ не рекомендуется устанавливать антивирусы или другое ПО для защиты информации. Это может привести к неработоспособности образа или большому числу ложных срабатываний при поведенческом анализе файлов. Если такое ПО уже установлено в ОС, рекомендуется отключить или удалить его.

► Чтобы вручную настроить ОС на образе ВМ:

1. Если вы создали образ ВМ из ISO-файла или из образа диска в формате QCOW2, запустите настройку образа ВМ, выполнив команду:

   ```
   imagebuilder run-image --vm-name <Название временной ВМ> --image-dir <Каталог с образом ВМ> --expose-vnc <Порт подключения к VNC-серверу>
   ```

   Пример:

   ```
   imagebuilder run-image --vm-name custom_win7 --image-dir /tmp/custom_win7 --expose-vnc 5900
   ```

   Примечание. На время выполнения команды вы можете разрешить сетевой доступ из операционной системы ВМ к другим узлам, например, к серверу активации Windows в корпоративной сети. Добавьте в команду параметр `--allowed-outbound-connection` и укажите в нем IP-адрес или имя узла (можно добавить несколько таких параметров).

   Начнется загрузка ОС.

2. Через открытый в браузере проводник Windows настройте ОС и установите ПО (см. инструкцию ниже).

3. Завершите настройку образа ВМ:

- Если нужно оптимизировать размер образа, введите `1` и нажмите клавишу Enter.
- Если оптимизация не требуется, введите `2` и нажмите клавишу Enter.

Настройка ОС на образе ВМ завершена.

Установка ПО с помощью менеджера пакетов Chocolatey

► Чтобы установить ПО на образ ВМ с помощью менеджера пакетов Chocolatey:

1. Через открытый в браузере проводник Windows откройте интерфейс командной строки Windows от имени администратора.
2. Если менеджер пакетов Chocolatey не установлен, установите его, выполнив команду:
   ```
   C:\guest-utils\Configure-Sandbox.cmd Install-Chocolatey
   ```
3. Установите ПО, выполнив соответствующие команды, например:
   ```
   choco install visioviewer2016 --version=16.0 -y
   choco install vlc --version=3.0.8 -y
   choco install irfanview --version=4.54 -y
   choco install windjview --version=2.1 -y --ignore-checksum
   choco install paint.net --version=4.2.9 -y
   choco install vcredist-all -y
   ```

ПО установлено.

Установка ПО с локального веб-сервера

► Чтобы установить ПО на образ ВМ с локального веб-сервера:

1. Используя терминальный клиент, перейдите на узле с PT Sandbox в каталог с ПО для установки и выполните команду:
   ```
   ip netns exec <Имя ВМ в Xen> python3 -m http.server <Порт веб-сервера>
   ```
2. Через открытый в браузере проводник Windows откройте интерфейс командной строки Windows от имени администратора.
3. Определите IP-адрес шлюза, выполнив команду:
   ```
   ipconfig
   ```
   IP-адрес шлюза указан в поле Default Gateway.
4. На узле с PT Sandbox запустите браузер и перейдите по ссылке:
   ```
   http://<IP-адрес шлюза>:<Порт веб-сервера>
   ```
5. Скачайте и установите необходимое ПО.

ПО установлено.

Установка ПО с виртуального CD-привода

Если в образе ВМ есть виртуальный CD-привод, вы можете установить ПО из смонтированного на него ISO-файла.

► Чтобы установить ПО с виртуального CD-привода:

1. Используя терминальный клиент, узнайте имя виртуального CD-привода:

   ```
   sudo xl qemu-monitor-command <Название временной ВМ> "info block"
   ```

   Пример:

   ```
   sudo xl qemu-monitor-command custom_win7 "info block"
   ```

2. Смонтируйте ISO-файл с ПО на виртуальный CD-привод:

   ```
   sudo xl qemu-monitor-command <Название временной ВМ> "change <Имя виртуального CD-привода>
   <Абсолютный путь к ISO-файлу>"
   ```

   Пример:

   ```
   sudo xl qemu-monitor-command custom_win7 "change ide1-cd0 /home/soft.iso"
   ```

   Установите необходимое ПО с ISO-файла.

ПО установлено.

6. Обращение в службу технической поддержки

Техническая поддержка продукта включает в себя следующие услуги:

- предоставление рекомендаций по настройке продукта (оптимизации параметров) в процессе его эксплуатации;
- консультации по использованию функциональных возможностей продукта;
- диагностику сбоев, включая поиск причин и информирование клиента о выявленных проблемах;
- предоставление решений или возможностей обойти проблему с сохранением необходимой производительности;
- устранение ошибок в рамках выпуска обновлений;
- рассмотрение предложений по доработке продукта.

Вы можете получать техническую поддержку на специальном портале.

Этот раздел содержит информацию о способах и условиях получения технической поддержки.

В этом разделе

6.1. Техническая поддержка на портале

Портал предоставляет вам возможность создавать запросы на техническую поддержку.

Вы можете создать учетную запись на портале, используя адреса электронной почты, расположенные на официальном домене вашей организации. Вы также можете указывать другие адреса электронной почты для учетной записи в качестве дополнительных. Для оперативной связи укажите в профиле учетной записи название вашей организации и контактный телефон.

Портал содержит статьи базы знаний, новости обновлений продуктов Positive Technologies, ответы на часто задаваемые вопросы пользователей. Для доступа к базе знаний и всем новостям нужно создать на портале учетную запись.

Техническая поддержка на портале предоставляется на русском и английском языках.

6.2. Время работы службы технической поддержки

На портале технической поддержки вы можете круглосуточно создавать и обновлять запросы, читать новости продуктов и пользоваться базой знаний.

6.3. Как служба технической поддержки работает с запросами

При получении вашего запроса специалист службы технической поддержки классифицирует его (присваивает запросу тип и уровень значимости) и выполняет дальнейшие шаги по выполнению запроса.

В этом разделе

6.3.1. Предоставление информации для технической поддержки

Для решения проблем с продуктом вам необходимо предоставить специалисту технической поддержки следующие данные:

- номер лицензии на использование продукта;
- файлы журналов и другие наборы диагностических данных, которые требуются для анализа;
- снимки экрана;
- результаты выполнения рекомендаций специалиста технической поддержки;
- оптимальный канал для удаленного доступа к продукту и его диагностики (выбирается по согласованию).

Если информация не будет предоставлена в течение двух недель с момента запроса, специалист технической поддержки имеет право закрыть заявку, предварительно уведомив вас об этом.

Positive Technologies не несет обязательств по оказанию технической поддержки в случае отказа предоставить указанную выше информацию.

6.3.2. Типы запросов

Специалист технической поддержки относит ваш запрос к одному из следующих типов.

Вопросы по установке, повторной установке и предстартовой настройке продукта

Подразумевается помощь в подготовке продукта к работе, ответы на вопросы на данном этапе эксплуатации продукта. Техническая поддержка по этим вопросам доступна в течение 30 дней с момента активации продукта.

Вопросы по администрированию и настройке продукта

Включают в себя вопросы, возникающие в процессе эксплуатации продукта, рекомендации по оптимизации и настройке параметров продукта.

Восстановление работоспособности продукта

В случае критического сбоя и потери доступа к основной функциональности продукта специалист Positive Technologies оказывает помощь в восстановлении работоспособности продукта. Восстановление заключается либо в помощи по установке продукта заново с потенциальной потерей накопленных до сбоя данных, либо в откате продукта на доступную резервную копию (резервное копирование должно быть настроено заблаговременно). Positive Technologies не несет ответственность за потерю данных в случае неверно настроенного резервного копирования.

Обновление продукта

Positive Technologies поставляет пакеты обновления в течение срока обновления, указанного в лицензии на продукт.

Positive Technologies не несет ответственности за проблемы, возникшие при нарушении регламентированного процесса обновления.

Устранение дефектов продукта

Если по результатам диагностики обнаружен дефект продукта, Positive Technologies обязуется предпринять разумные усилия по предоставлению обходного решения (если возможно), а также включить исправление дефекта в ближайшие возможные обновления продукта.

Доработка продукта

При обращении с предложением по доработке продукта необходимо подробно описать цель такой доработки. Вы также можете поделиться рекомендациями по улучшению продукта и оптимизации его функциональности. Positive Technologies обязуется рассмотреть все предложения, однако не принимает на себя обязательств по реализации каких-либо доработок. Если Positive Technologies принимает решение о доработке продукта, то способы реализации доработки остаются на усмотрение Positive Technologies.

6.3.3. Время реакции и приоритизация запросов

Время реакции на запрос рассчитывается с момента получения запроса до первичного ответа специалиста технической поддержки с уведомлением о взятии запроса в работу.

Время обработки запроса рассчитывается с момента отправки уведомления о взятии вашего запроса в работу до предоставления описания дальнейших шагов по устранению проблемы либо классификации вопроса, указанного в запросе, как дефекта ПО и передачи запроса ответственным лицам для исправления дефекта.

Время реакции и время обработки зависят от указанного вами уровня значимости запроса (см. таблицу 52).

Специалист службы технической поддержки оставляет за собой право переопределять уровень значимости запроса по приведенным ниже критериям. Указанные сроки являются целевыми и подразумевают стремление и разумные усилия исполнителя для их соблюдения, но возможны отклонения от данных сроков по объективным причинам.

Таблица 52. Время реакции на запрос и время его обработки

Уровень значимости запроса	Критерии значимости запроса	Время реакции на запрос	Время обработки запроса
Критический	Аварийные сбои, полностью препятствующие штатной работе продукта (исключая первоначальную установку) либо оказывающие критическое влияние на бизнес	До 4 часов	Не ограничено
Высокий	Сбои, затрагивающие часть функциональности продукта и проявляющиеся в любых условиях эксплуатации либо оказывающие значительное влияние на бизнес	До 24 часов	Не ограничено
Обычный	Сбои, проявляющиеся в специфических условиях эксплуатации продукта либо	До 24 часов	Не ограничено

Уровень значимости запроса	Критерии значимости запроса	Время реакции на запрос	Время обработки запроса
	не оказывающие значительного влияния на бизнес		
Низкий	Вопросы информационного характера либо сбои, не влияющие на эксплуатацию продукта	До 24 часов	Не ограничено

Указанные часы относятся только к рабочему времени специалистов технической поддержки (времени обработки запроса).

6.3.4. Выполнение работ по запросу

По мере выполнения работ по вашему запросу специалист технической поддержки сообщает вам:

— о диагностике проблемы и ее результатах;

— о поиске решения или возможности обойти причины возникновения проблемы;

— о планировании и выпуске обновления продукта (если требуется для устранения проблемы).

Если по итогам обработки запроса необходимо внести изменения в продукт, Positive Technologies включает работы по исправлению в ближайшее возможное плановое обновление продукта (в зависимости от сложности изменений).

Работы по запросу считаются выполненными, если:

— предоставлено решение или возможность обойти проблему, не влияющая на производительность и критически важную функцию продукта;

— диагностирован дефект продукта, собрана техническая информация о дефекте и условиях его воспроизведения; исправление дефекта запланировано к выходу в рамках планового обновления продукта;

— проблема вызвана программными продуктами или оборудованием сторонних производителей, не подпадающих под гарантийные обязательства по продукту;

— проблема классифицирована как неподдерживаемая.

positive technologies

Positive Technologies — лидер рынка результативной кибербезопасности. Компания является ведущим разработчиком продуктов, решений и сервисов, позволяющих выявлять и предотвращать кибератаки до того, как они причинят неприемлемый ущерб бизнесу и целым отраслям экономики. Наши технологии используют более 3300 организаций по всему миру, в том числе 80% компаний из рейтинга «Эксперт-400». Positive Technologies — первая и единственная компания из сферы кибербезопасности на Московской бирже (MOEX: POSI), у нее более 200 тысяч акционеров.