Руководство пользователя ПО «CTT Downloader»

ОГЛАВЛЕНИЕ

1 Введение 3
1.1 Описание функциональных характеристик ПО 3
1.2 О программном обеспечении 3
1.3 Системные требования 3
2 Установка ПО «CTT Downloader» 5
2 Начало работы 6
2.1 Единоразовый запуск 6
2.1 Запуск по расписанию 7
Приложение 1. Описание параметров запуска и конфигурационных файлов 8
Параметры запуска 8
Описание конфигурационных файлов 9
config.yml 9
filters.yml 10
logging.yml 12
Приложение 2. Спецификация открытого протокола взаимодействия с внешними сторонними сервисами. 13
Приложение 3. Спецификация формата JSON с индикаторами 14

1 ВВЕДЕНИЕ

1.1 Описание функциональных характеристик ПО

Программное обеспечение «CTT Downloader» разработано для работы с облачным сервисами реализующими, сбор, очистку и обогащение индикаторов компрометации из множества открытых источников.

«CTT Downloader» берет на себя все проблемы, связанные с доступом к облачным сервисам, поддерживающим определенную спецификацию протокола REST API, а также представлению полученной информации в формате, удобном для дальнейшей работы с ней на стороне пользователя.

«CTT Downloader» решает следующие задачи:

1. Загрузка индикаторов компрометации по REST API из источников данных.
2. Конвертирование индикаторов, загруженных из источника, в формат JSON, CSV.
3. Сохранение данных файлов на локальном на диске (один файл для каждого типа индикаторов: IP, Domain, URL, Hash).
4. Фильтрация загруженных данных по: требуемому набору полей, индикаторам с заданными тегами, индикаторам с уровнем опасности, превышающим заданный.

1.2 О программном обеспечении

ПО «CTT Downloader» предназначено для получения индикаторов из источников данных, в частном случае таким источником данных может являться облако ООО «Технологии киберугроз». Также ПО «CTT Downloader» может взаимодействовать с иными источниками, реализующими протокол, описанный в Приложении 2 и Приложении 3.

1.3 Системные требования

ПО «CTT Downloader» распространяется в виде Standalone-приложения - заказчику предоставляются инструкция и дистрибутив для локальной установки ПО на вычислительном устройстве заказчика, причем заказчику доступна загрузка дистрибутива ПО с сайта правообладателя ПО по ссылке, предоставляемой после приобретения ПО.

Минимальные системные требования для установки:

- Свободное место на диске: 400 МБ.
- ОС: Windows 10 (Home, Pro), Linux (Debian 9, 10).
- Прямой доступ в интернет, или доступ через Proxy.

2 УСТАНОВКА ПО «CTT DOWNLOADER»

ПО «CTT Downloader» поставляется в виде архива с бинарными исполняемыми файлами. ПО не требует доустановки каких-либо сторонних пакетов в ОС.

Для установки ПО необходимо:

1. Распаковать архив с ПО в требуемую директорию.
2. Отредактировать **./conf/config.yml** (описание параметров в Приложении 1).
    a. Указать **basedir**. Значение параметра предоставляется правообладателем, после покупки ПО.
    b. Указать **apikey**. Значение параметра предоставляется правообладателем, после покупки ПО.
    c. Указать пути в секции **dirs.**
    d. Указать **dirs.target** – целевая директория для сохранения файла с индикаторами
    e. Указать **dirs.tmp** – временная директория для распаковки архива с индикаторами
3. Отредактировать **./conf/logging.yml** (описание параметров в Приложении 1).
    a. Минимальная настройка: Полный путь до лог-файла
4. Отредактировать **./conf/filters.yml** (описание параметров в Приложении 1).
    a. Минимальная настройка: убрать комментарий (#) с тех полей, которые необходимо получить в итоговом файле с индикаторами.
5. Дать права на исполнение cttdownloader (для nix-систем)
    a. chmod +x cttdownloader

2 НАЧАЛО РАБОТЫ

2.1 Единоразовый запуск

После установки ПО «CTT Downloader» готово к запуску.

Одним из сценариев работы является разовый запуск для получения выгрузки индикаторов.

Рассмотрим сценарий, когда нам необходимо единоразово получить список индикаторов с IP-адресами в формате CSV:

1. Запустите интерпретатор командной строки в ОС (bash, sh в nix-системах, cmd.exe, в Windows)
2. Перейдите в директория с ПО.
3. Выполните команду:
   `./cttdownloader -f csv -t ip`
4. Дождитесь завершения выполнения команды.
5. Зайдите в директорию **./target**
6. В данной директории будет находится CSV-файл с индикаторами.
7. В файле будут находится индикаторы с теми полями, которые были заданы в конфигурационном файле ./conf/filters.yml

Рассмотрим сценарий, когда нам необходимо единоразово получить список индикаторов с Доменами, с уровнем опасности более 50-ти, в формате CSV:

1. Запустите интерпретатор командной строки в ОС (bash, sh в nix-системах, cmd.exe, в Windows)
2. Перейдите в директория с ПО.
3. Измените в **./conf/filters.yml** параметр **score.threshold** на значение **50**.
4. Выполните команду:
   `./cttdownloader -f csv -t domain`
5. Дождитесь завершения выполнения команды.
6. Зайдите в директорию **./target**
7. В данной директории будет находится CSV-файл с индикаторами.
8. В файле будут находится индикаторы с теми полями, которые были заданы в конфигурационном файле **./conf/filters.yml** и **score >= 50**.

Рассмотрим сценарий, когда нам необходимо единоразово получить список только индикаторов с IP-адресами, которые являются управляющими адресами вредоносного программного обеспечения:

1. Запустите интерпретатор командной строки в ОС (bash, sh в nix-системах, cmd.exe, в Windows)
2. Перейдите в директория с ПО.
3. Измените в **./conf/filters.yml** параметр **tags** на значение **['c2 ', 'malware']**, либо снимите комментируй с существующей строки в конфигурационном файле.
4. Выполните команду:
   `./cttdownloader -f csv -t domain`
5. Дождитесь завершения выполнения команды.
6. Зайдите в директорию **./target**
7. В данной директории будет находится CSV-файл с индикаторами.
8. В файле будут находится индикаторы с теми полями, которые были заданы в конфигурационном файле **./conf/filters.yml** и **score >= 50**.

2.1 Запуск по расписанию

Список индикаторов может обновляться на регулярной основе. В таком случае удобно запланировать регулярный запуск ПО «CTT Downloader», для чего необходимо:

1. Добавить запуск ПО в cron (для nix-систем, или планировщик заданий в Windows). К примеру, индикаторы компрометации на серверах ООО «Технологии киберугроз» обновляются 1 раз в сутки в интервале с 04:00 - 04:30 по Мск. Т.о. загрузку рекомендуется планировать на 05:00 по Мск. На серверах других поставщиков время обновления может отличаться.
2. При добавлении ПО загрузки в cron необходимо задать каталог ПО в качестве рабочей директории, либо явно определить параметры -c, -e, -l
3. После запуска, в директории **./target** (см. config.yml) появится файл, подготовленный для дальнейшего использования на стороне заказчика.

ПРИЛОЖЕНИЕ 1. ОПИСАНИЕ ПАРАМЕТРОВ ЗАПУСКА И КОНФИГУРАЦИОННЫХ ФАЙЛОВ

Параметры запуска

```
-c, --config                              Путь до config.yml
                                          По умолчанию: ./conf/config.yml

-e, --filters                             Путь до filters.yml.
                                          По умолчанию: ./conf/filters.yml

-l, --log                                 Путь до logging.yml.
                                          По умолчанию: ./conf/logging.yml

-f, --convformat    [обязательный]        Формат для конвертирования фида.
                                          По умолчанию: none
                          none – не конвертировать, только скачать.
                          csv – Конвертировать в CSV-формат
                          mpsiem – Конвертировать в формат MP SIEM
                          qradar – Конвертировать в формат IBM QRadar
                          arcsight - Конвертировать в формат Micro Focus ArcSight
                          redis - Конвертировать в формат Redis
                          memcached - Конвертировать в формат Memcached

-t, --feedtype      [обязательный]        Тип фида.
                                          ip - IP-адреса
                                          domain – Домены
                                          url – URL
                                          hash - Хэши

--upload / --not-upload   Флаг. Попытаться установит в SIEM
                          (только для mpsiem и qradar).
                          По умолчанию: --upload

--cleartmp / --not-cleartmp      Флаг. Очищать временную директорию после
                                 успешной конвертации.
                                 По умолчанию: --cleartmp

--cday <число>            Принудительно выставить дату, за которую
                          надо искать файл фида.
                          Unix timestamp (int 32)

--help                                    Показать справку
```

Описание конфигурационных файлов

config.yml

```
connection:
  proxy:          # Удалить, если proxy не используется
    type: 'https' # http/https
    url: 'socks5://123.123.123.123:8888' # format
<http/socks5>://<user:pass>@<ip/fqdn>:<port>

  cttcloud:
    baseurl: '' # Ваша ссылка на поток данных
    apikey: '' # Ваш ключ для подключения
    contimeout: 10   # Можно повысить, если большой пинг до сервера
    readtimeout: 20 # Можно повысить, если медленное соединение
    retry: 2        # Кол-во попыток переподключения
    delete_gz: true
    feeds:
      filetype: 'json'

dirs:
  target: './target' # Целевая директория для сконвертированного файла
  tmp: './tmp '      # Временная директория для скаченного файла. Очищается после
конвертации
  alert: './'        # Директория для cttdownloader.alert
  state: './ '       # Директория для cttdownloader.state
```

filters.yml

```
export:
  fields:                   # Список экспортируемых из фида полей.
    ip:
      #0: '_last_changed' # для MP SIEM ( < R24)
      1: 'ip_v4'
      #2: 'ip_num'
      3: 'fseen'
      4: 'lseen'
      5: 'collect'
      8: 'tags_str'
      #9: 'tags_codes'
      10: 'asn_num'
      #11: 'asn_firstip_netv4'
      #12: 'asn_firstip_num'
      #13: 'asn_lastip_netv4'
      #14: 'asn_lastip_num'
      15: 'asn_cloud'
      16: 'asn_domains'
      17: 'asn_org'
      18: 'asn_isp'
      #19: 'geo_city'
      20: 'geo_country'
      #21: 'geo_region'
      22: 'related_domains'
      24: 'score_src'
      25: 'score_tags'
      26: 'score_frequency'
      27: 'score_total'
      28: 'fp_alarm'
      29: 'fp_descr'
      30: 'threat'
      31: 'cve'
      32: 'industry'
      33: 'src_report'
      34: 'id'
      35: 'title'
      36: 'description'
      37: 'ports'
    domain:
      #0: '_last_changed' # для MP SIEM ( < R24)
      1: 'domain'
      3: 'fseen'
      4: 'lseen'
      5: 'collect'
      8: 'tags_str'
      #9: 'tags_codes'
      10: 'resolved_ip_a'
      11: 'resolved_ip_alias'
      12: 'resolved_ip_cname'
      13: 'resolved_whois_created'
      14: 'resolved_whois_updated'
      #15: 'resolved_whois_expires'
      #16: 'resolved_whois_age'
      17: 'resolved_whois_registrar'
      18: 'resolved_whois_registrant'
      #19: 'resolved_whois_havedata'
      20: 'score_src'
      21: 'score_tags'
```

```
      22: 'score_frequency'
      23: 'score_total'
      24: 'fp_alarm'
      25: 'fp_descr'
      26: 'threat'
      27: 'cve'
      28: 'industry'
      29: 'src_report'
      30: 'id'
      31: 'title'
      32: 'description'
      33: 'ports'
    url:
      #0: '_last_changed' # для MP SIEM ( < R24)
      1: 'url'
      2: 'fseen'
      3: 'lseen'
      4: 'collect'
      5: 'tags_str'
      #6: 'tags_codes'
      7: 'score_src'
      8: 'score_tags'
      9: 'score_frequency'
      10: 'parsed_schema'
      11: 'parsed_domain'
      12: 'parsed_port'
      13: 'parsed_path'
      14: 'parsed_params'
      15: 'parsed_anchor'
      16: 'resolved_status'
      17: 'score_total'
      18: 'fp_alarm'
      19: 'fp_descr'
      20: 'threat'
      21: 'cve'
      22: 'industry'
      23: 'src_report'
      24: 'id'
      25: 'title'
      26: 'description'
    hash:
      #0: '_last_changed' # для MP SIEM ( < R24)
      1: 'fseen'
      2: 'lseen'
      3: 'collect'
      4: 'md5'
      5: 'sha1'
      6: 'sha256'
      7: 'filename'
      8: 'tags_str'
      #9: 'tags_codes'
      10: 'score_src'
      11: 'score_tags'
      12: 'score_frequency'
      13: 'score_total'
      14: 'fp_alarm'
      15: 'fp_descr'
      16: 'threat'
      17: 'cve'
      18: 'industry'
```

```
      19: 'src_report'
      20: 'id'
      21: 'title'
      22: 'description'
  falsealarm:# Экспорт индикаторов с определенным уровнем ложных срабатываний.
    0: 'true'
    1: 'false'
    2: 'possible'
  tags:            # Оставлять только индикаторы с любым из указанных тегов
    - c2
    - malware
  score:
    threshold: 40   # Отбросить индикаторы со score_total меньше указанного
    highrisk: 40    # Применимо для QRadar. Создавать отдельный Set с
индикаторами,
                     чей score_total выше указанного.
  ignorefilters:
    keeptags:      # Игнорировать ограничения на score и оставлять индикаторы с
любым,
                       указанным, тегом
      - tor_node
      - tor_exit
```

logging.yml

```
version: 1
disable_existing_loggers: True
formatters:
  simple:
    format: "%(asctime)s - %(name)s - %(levelname)s - %(message)s"
handlers:
  console:
    class: logging.StreamHandler
    level: DEBUG
    formatter: simple
    stream: ext://sys.stdout
  file_handler:
    class: logging.handlers.RotatingFileHandler
    level: DEBUG
    formatter: simple
    filename: ./var/log/cttdownloader.log #Установить путь до файла лога.
Убедиться, что есть права на запись в директорию
    maxBytes: 10485760 # 10MB
    backupCount: 10
    encoding: utf8
loggers:
  cttdownloader:
    level: INFO      # Изменить уровень журналирования, если необходимо
    handlers: [console, file_handler]
    propagate: no
root:
  level: INFO
  handlers: [console, file_handler]
```

ПРИЛОЖЕНИЕ 2. СПЕЦИФИКАЦИЯ ОТКРЫТОГО ПРОТОКОЛА ВЗАИМОДЕЙСТВИЯ С ВНЕШНИМИ СТОРОННИМИ СЕРВИСАМИ.

1. Для работы с ПО «CTT Downloader» сторонние сервисы обязаны реализовать следующую спецификацию.
2. Протокол: HTTP/HTTPS
3. Аутентификация: Каждый запрос ПО «CTT Downloader» содержит HTTP-заголовок "x-api-key". Значение заголовка содержит API-key, заданный в конфигурационном файле.
4. Основной URL обязан удовлетворять шаблону: <http|https>://<FQDN>/static/v2/full/
5. Реализация запросов метода HEAD:
    a. Точка входа: <Основной URL>/<ip|domain|url|hash>
    b. В заголовке ответа должен присутствовать "Last-Modified", удовлетворяющий форматной строке "%a, %d %b %Y %H:%M:%S %Z"
6. Реализация запросов метода GET:
    a. Точка входа: <Основной URL>/<ip|domain|url|hash>/?type=json&date=<date>
    b. Параметр date должен удовлетворять форматной строке "%Y%m%d" и формироваться на основе поля "Last-Modified" из HEAD-запроса
    c. Ответ: JSON, либо GZ-архив с JSON-файлом, удовлетворяющим формату из Приложения 2.

ПРИЛОЖЕНИЕ 3. СПЕЦИФИКАЦИЯ ФОРМАТА JSON С ИНДИКАТОРАМИ

Тип индикатора: IP

```
{
  {
  "ip": {
    "v4": <string>,
    "num": <string>      - IPv4 в формате UInt32
  },
  "fseen": <int>,      - timestamp первого появления
  "lseen": <int>,      - timestamp последнего появления
  "collect": <int>,    - timestamp сбора
  "src": {
    "name": <string>      - название источника IoC
    "report": <string>        - URL источника IoC
  }
  "tags": {               - категория индикатора
    "str": [<string>],
  },
  "cve": <string>,                - связь с CVE
  "industry": <string>,     - в каких секторах экономики наблюдался IoC
  "threat": [<string>],  - атрибуция с группировкой/ВПО/хакерской утилитой
  "score": {                - уровень опасности
    "total": <int>,             - итоговый уровень
    "src": <double>,            - доверие к источникам
    "tags": <double>,           - уровень опасности по контексту
    "frequency": <double>         - частота появления IoC
  },
  "fp": {                   - вероятного ложного срабатывания
    "alarm": <string>,          - флаг "false"/"true"
    "descr": <string>             - описание причины выставления флага
  },
  "asn": {
    "num": <int>,           - номер ASN
    "firstip": {
      "netv4": <string>, - начало диапазона IP ASN
      "num": <string>   - IPv4 в формате UInt32
    },
    "lastip": {
      "netv4": <string>, - окончание диапазона IP ASN
      "num":  <string>       - IPv4 в формате UInt32
    },
    "cloud": <string>,              - название облачного провайдера, или CDN (AWS, GCP и т.д.)
    "domains": <int>,         - кол-во доменов, зарегистрированных в ASN
    "org": <string>,    - владелец ASN
    "isp": <string>         - владелец ASN
  },
  "geo": {                  - Геоданные
    "city": <string>,
    "country": <string>,
    "region": <string>
  },
  "related": {
    "domains": [<string>]  - связанные вредоносные домены
  }
}
```

Тип индикатора: Domain

```
{
  {
  "domain": <string>,
  "fseen": <int>,       - timestamp первого появления
  "lseen": <int>,       - timestamp последнего появления
  "collect": <int>,     - timestamp сбора
  "src": {
    "name": <string>      - название источника IoC
    "report": <string>        - URL источника IoC
  }
  "tags": {               - категория индикатора
    "str": [<string>],
  },
  "cve": <string>,                - связь с CVE
  "industry": <string>,     - в каких секторах экономики наблюдался IoC
  "threat": [<string>],  - атрибуция с группировкой/ВПО/хакерской утилитой
  "score": {                - уровень опасности
    "total": <int>,             - итоговый уровень
    "src": <double>,            - доверие к источникам
    "tags": <double>,           - уровень опасности по контексту
    "frequency": <double>         - частота появления IoC
  },
  "fp": {                   - вероятного ложного срабатывания
    "alarm": <string>,          - флаг "false"/"true"
    "descr": <string>             - описание причины выставления флага
  },
  "resolved": {           - дополнительная информация (whois, dns)
    "ip": {
      "a": [            - DNS A-записи
        <string>
      ],
      "alias": <string>,        - основной домен, если IoC является CNAME-записью
      "cname": [<string>]         - DNS CNAME-записи
    },
    "whois": {            - WHOIS
      "created": <string>,
      "updated": <string>,
      "expires": <string>,
      "age": <int>,
      "registrar": <string>,
      "registrant": <string>,
      "havedata": "none"  - "false" - whois сервис не нашел записи
                  "true"  - whois нашел запись
                  "none"  - whois не ответил
    }
  }
}
```

Тип индикатора: URL

```
{
  {
  "url": <string>,
  "fseen": <int>,       - timestamp первого появления
  "lseen": <int>,       - timestamp последнего появления
  "collect": <int>,     - timestamp сбора
  "src": {
    "name": <string>      - название источника IoC
    "report": <string>        - URL источника IoC
  }
  "tags": {               - категория индикатора
```

```
    "str": [<string>],
  },
  "cve": <string>,                  - связь с CVE
  "industry": <string>,     - в каких секторах экономики наблюдался IoC
  "threat": [<string>],  - атрибуция с группировкой/ВПО/хакерской утилитой
  "score": {                  - уровень опасности
    "total": <int>,               - итоговый уровень
    "src": <double>,              - доверие к источникам
    "tags": <double>,             - уровень опасности по контексту
    "frequency": <double>          - частота появления IoC
  },
  "fp": {                    - вероятного ложного срабатывания
    "alarm": <string>,          - флаг "false"/"true"
    "descr": <string>              - описание причины выставления флага
  },
"resolved": {            - HTTP-код ответа от Web-сервера на момент сбора IoC
    "status": <int>
  },
  "parsed": {              - декомпозиция URL
       "schema": <string>,
       "domain": <string>,
       "port": <int>,
       "path": <string>,
       "params": <string>,
       "anchor": <string>
  }
}
```

Тип индикатора: Hash

```
{
  {
  "md5": <string>,
  "sha1": <string>,
  "sha256": <string>,
  "filename": [<string>],
  "fseen": <int>,      - timestamp первого появления
  "lseen": <int>,      - timestamp последнего появления
  "collect": <int>,    - timestamp сбора
  "src": {
    "name": <string>      - название источника IoC
    "report": <string>         - URL источника IoC
  }
  "tags": {                - категория индикатора
    "str": [<string>],
  },
  "cve": <string>,                  - связь с CVE
  "industry": <string>,     - в каких секторах экономики наблюдался IoC
  "threat": [<string>],  - атрибуция с группировкой/ВПО/хакерской утилитой
  "score": {                  - уровень опасности
    "total": <int>,               - итоговый уровень
    "src": <double>,              - доверие к источникам
    "tags": <double>,             - уровень опасности по контексту
    "frequency": <double>          - частота появления IoC
  },
  "fp": {                    - вероятного ложного срабатывания
    "alarm": <string>,          - флаг "false"/"true"
    "descr": <string>              - описание причины выставления флага
  }
}
```