24 July 2017

Threat Prevention API

1.0

Reference Guide

Check Point
SOFTWARE TECHNOLOGIES LTD.

Important Information

Latest Software

We recommend that you install the most recent software release to stay up-to-date with the latest functional improvements, stability fixes, security enhancements and protection against new and evolving attacks.

Latest Version of this Document

Download the latest version of this document http://supportcontent.checkpoint.com/documentation_download?ID=43199.

To learn more, visit the Check Point Support Center http://supportcenter.checkpoint.com.

Feedback

Check Point is engaged in a continuous effort to improve its documentation.

Please help us by sending your comments mailto:cp_techpub_feedback@checkpoint.com?subject=Feedback on Threat Prevention API 1.0 Reference Guide.

Revision History

Date	Description
24 July 2017	Updated access information for Threat Extraction blade ("Introduction" on page 5).
3 April 2017	Added extraction Request (on page 12) and response ("extraction Response Format" on page 19)
16 February 2017	Updated te Request (on page 12)
22 December 2016	Updated Global Request Headers (on page 6) and HTTP Download Request (on page 29)
15 June 2016	Added av Response Format (on page 19) Updated Upload Request Format (on page 25) Updated Images Object Format (on page 16) CR02054303 CR02052248
20 April 2016	General updates
6 July 2015	CR01783343: Corrected URL in HTTP Download Request (on page 29)
22 June 2015	First release of this document

Contents

Important Information 3
Introduction 5
Accessing the API 6
Global Request Headers 6
JSON Request and Response Structure 7
Request Object Structure 7
Response Object Structure 7
Response Status Codes 8
Best Practices 9
Query API 10
HTTP Query Request 10
Query Request HTTP Headers 10
Query Request Format 11
te Request 12
av Request 12
extraction Request 12
Query Request Example 14
Query Response HTTP Headers 14
Query Response Format 15
te Response 15
Images Object Format 16
XML Report Structure 18
av Response Format 19
extraction Response Format 19
Query Response Example 22
Upload API 24
HTTP Upload Request 24
Upload Request HTTP Headers 24
Upload Request Format 25
Upload Request Example 26
Upload Response Format 27
Upload Response Example 28
Download API 29
HTTP Download Request 29
Download Request HTTP Headers 29
Download Request Format 29
Quota API 30
HTTP Quota Request 30
Quota Request Headers 30
Quota Response Format 30
Quota Response Example 31

Introduction

In This Section:

Accessing the API 6
Global Request Headers 6
JSON Request and Response Structure 7
Request Object Structure 7
Response Object Structure 7
Response Status Codes 8
Best Practices 9

The Check Point Threat Prevention API lets you control Threat Prevention products through cloud web services. The request/response API has functionality similar to Next Generation Threat Extraction (NGTX) and Threat Extraction appliances.

Notes:

- On local gateways, the Check Point Threat Extraction API is only supported with Threat Extraction engine 4.1.
- Use the Threat Extraction API only after you follow the steps in the Check Point Threat Extraction API, sk113599 http://supportcontent.checkpoint.com/solutions?id=sk113599.

For further inquiry or information, contact Check Point Support at CheckMates - SandBlast API https://community.checkpoint.com/community/threat-prevention/sandblast-api.

General Details:

All services use the HTTP POST method, unless stated otherwise.

The body of requests and responses is in JSON format, unless stated otherwise.

All services have mandatory headers and request body fields. If you do not enter values for mandatory variables, there will be errors. Optional variables have predefined defaults.

The response to some requests includes detailed status messages in the response body (see "Response Status Codes" on page 8).

License:

NGTX and TX packages include the API usage license.

To evaluate the cloud service, contact Check Point for an Evaluation API Key.

On local gateways, see sk113599 http://supportcontent.checkpoint.com/solutions?id=sk113599 to generate an API key for Threat Extraction use.

Accessing the API

A request format depends on the required API Web service.

Access the API with the URL:

```
https://<service_address>/tecloud/api/<version>/file/<API_name>
```

Variable	Description
service address	`te.checkpoint.com` for the Check Point Threat Prevention cloud services, or the IP address of a TE appliance **Note** - To run Check Point Threat Prevention API on a local gateway, you must specify the port - 18194. That is: `https://<service_address>:18194/tecloud/api/<version>/file/query`
version	Current version is 1.0. Valid value: `v1`
API name	Valid values: query ("Query API" on page 10) upload ("HTTP Upload Request" on page 24) download ("Download API" on page 29) quota ("Quota API" on page 30) (for cloud services only)

Global Request Headers

Every request for cloud services must have an authorization parameter in the header, with a valid API Key as the value. For requests on local gateways, it is not required.

Mandatory in request headers:

Header	Value	Example
Authorization	A valid API key	`Authorization: YWJjZDEyMzQ`
`te_cookie:`*<string>*	Cookie name, to make sure all query requests go to the same server. Get the cookie name from the first header response.	`te_cookie:remember`

JSON Request and Response Structure

The request body can be a single JSON object, or it can contain an array of objects, called **request object**. A JSON request has this structure:

```
{
   "request": <request body>
}
```

The response body has one object, or an array of response objects. A response has this structure:

```
{
   "response": <response body>
}
```

Request Object Structure

A request object has JSON fields that are relevant to the request, with metadata about the request. The body of the request can also have a detailed section, with data for each feature that can be used. If not given, the Threat Prevention service uses the best default values.

Example:

```
{
   "request": {
       "sha1": "af1aab67180197681016fc32654ec3cac2850109",
       "features": [
           "te"
       ],
       "te": {
           "reports": [
               "xml"
           ]
       }
   }
}
```

In this example, **sha1** and **features** are general fields, to ask the service for a Threat Emulation knowledge base to report known data on the SHA-1 digest with the "af1..." ID. The **te** section is the feature: get the report in XML format.

Response Object Structure

The response object has general and feature-specific fields and properties.

Note - If the request gave an array of request objects, the response body will have a response object for each request object, in the same order.

Response Status Codes

HTTP Code	Label	Description
200	OK	Request served successfully.
301	Moved permanently	Service is not available anymore.
400	Bad request	Incorrect request format.
401	Unauthorized	Authentication failed.
403	Forbidden	Unauthorized access to the service.
404	Not found	Service does not exist.
500	Internal server error	There was an error in the service.
503	Service unavailable	Currently this request cannot be served.

API code	Label	Example
1001	FOUND	Request fully answered.
1002	UPLOAD_SUCCESS	File uploaded successfully. Query with the same hash and file type every 30 seconds.
1003	PENDING	File is pending. Query with the same hash and file type in about 3 seconds.
1004	NOT_FOUND	Request not found. Upload the file.
1005	NO_QUOTA	There is no quota for this API key. Contact Check Point.
1006	PARTIALLY_FOUND	Part of the request found. If the missing data is required, upload the file.
1007	FILE_TYPE_NOT_SUPPORTED	File type is illegal.
1008	BAD_REQUEST	Request format is not valid. Make sure the request follows this documentation.
1009	INTERNAL_ERROR	There is a temporary error with the service. Try again in a few minutes.
1010	FORBIDDEN	You do not have permissions to use the requested feature. Contact Check Point.
1011	NOT_ENOUGH_RESOURCES	There is a temporary error with the service. Try again in few seconds.

Best Practices

We recommend this flow of requests:

1. Query the API for cached entries for the features of the request.
2. If the features are not found, upload files for processing.
3. If the response is a 503 error, wait a few minutes before you try the request again. Web services can be temporarily overloaded or down for maintenance.
4. Send a number of requests in one array, rather than each request separately.

Query API

In This Section:

HTTP Query Request 10
Query Request HTTP Headers 10
Query Request Format 11
Query Request Example 14
Query Response HTTP Headers 14
Query Response Format 15

Use the Query API to have a client application look for:

- The analysis report of a specific file on the Check Point Threat Prevention service databases.
- The status of a file, uploaded for analysis.

When the Query API is called with a message digest, the response includes all known data about that digest. The `features` field narrows the results (project) for relevant features.

Best Practice: Make sure the request is for the minimum required data, to save lookup and processing time.

HTTP Query Request

```
HTTP POST: https://<service_address>/tecloud/api/<version>/file/query
```

The value of the service address depends on your environment ("Accessing the API" on page 6).

Query Request HTTP Headers

Header	Value	Example
Authorization	A valid API Key	`Authorization: YWJjZDEyMzQ`
Content-Type	`application/json`	`Content-type: application/json`
Content-Length	Payload size	`Content-length: 1234`

Query Request Format

Mandatory fields:

Field	Type	Value	Notes
`md5` | `sha1` | `sha256`	String	File digest	Only one digest is mandatory. **Note** - On local gateways, only sha1 digest format is supported.

Optional fields:

Field	Type	Value	Default	Notes
`file_type`	String	File extension		Service identifies the type. **Note** - This file is required in requests to local gateways.
`file_name`	String	Name of file to extract		Mandatory for `extraction`, it is the same file as given with the file digest
`features`	String array	Available features	`te, av`	
<feature>	JSON object	JSON fields that describe expected result data and format (see next sections)	*<feature default>*	There is one JSON object for each feature. Features that were not in the `features` array are ignored.
`quota`	Boolean	`true` | `false`	`false`	If true, response delivers the quota data ("Quota API" on page 30) (for cloud services only).

te Request

Feature Field Name `te`

Check Point Name Threat Emulation

Field	Type	Value	Default	Notes
`images`	Array	ID and revision of available OS images	All	An image is an operating system configuration.
> `id`	String	Image ID		
> `revision`	Integer	Image revision		
`reports`	String array	`pdf` | `xml` | `tar`		Supported report formats

av Request

Feature Field Name `av`

Check Point Name Anti-Virus

No fields.

extraction Request

Feature Field Name `extraction`

Check Point Name Threat Extraction

Field	Type	Value
`method`	String	`clean` | `pdf` Default = `pdf`
`extracted_parts_codes`	Integer array	extracted_parts_codes Values (on page 13) Only relevant if `method = clean`

Example of extraction only request:

```
 {
      "request": {
      "sha1": "4307473c14351751a3563e07bc6c4a96bb2b2f5d",
      "features": ["extraction"],
      "file_name": "example.xls"
      "extraction": {
            "extracted_parts_codes": [1034, 1026, 1019, 1018, 1139, 1142, 1143,
1141, 1150, 1151, 1137, 1021],
            "method": "clean"
      },

      }
}
```

extracted_parts_codes Values

Threat Extraction cleans files. If the components of files are not given in the **extracted_parts_codes** field, the default parts are cleaned.

Default value of **extracted_parts_codes** is `1025,1026,1034,1137,1139,1141,1142,1143,1150,1151,1018,1019,1021`

Code	Description
1025	Linked Objects
1026	Macros and Code
1034	Sensitive Hyperlinks
1137	PDF GoToR Actions
1139	PDF Launch Actions
1141	PDF URI Actions
1142	PDF Sound Actions
1143	PDF Movie Actions
1150	PDF JavaScript Actions
1151	PDF Submit Form Actions
1018	Database Queries
1019	Embedded Objects
1021	Fast Save Data
1017	Custom Properties
1036	Statistic Properties

Code	Description
1037	Summary Properties
	Images in side files. *For future use*

Query Request Example

```
{
    "request": [
        {
            "md5": "8dfa1440953c3d93daafeae4a5daa326",
            "features": [
                "te",
                "av"
                "extraction"
            ],
            "file_name": "example.xls"
            "te": {
                "reports": [
                    "xml",
                    "pdf"
                ]
            }
            "extraction": {
                "method": "pdf"
            }
        }
    ]
}
```

This example sends a web service query to the databases for Threat Emulation, Anti-Virus, and Threat Extraction results for the file with the given MD5, named `example.xls`

Results will be given in XML and PDF formats, on all supported images.

Anti-Virus does not have configuration fields. It is enough to list it as a value of the `features` object.

Query Response HTTP Headers

Header	Value	Example
Content-Type	`application/json`	`Content-type: application/json`
Content-Length	Payload size	`Content-length: 1234`

Query Response Format

Field	Type	Value	Notes
`status`	JSON Object	`code`, `label`, `message`	Status of requested features in machine code and human readable label and message.
> `code`	Integer	Status Code	
> `label`	String	Readable code label	
> `message`	String	Readable status message	
`md5 | sha1 | sha256`	String	Message digest	If found, all three, otherwise only the requested digest.
`file_name`	String	Name of file saved on Check Point databases	
`features`	String array	Available features	See next sections
<feature>	JSON object	JSON fields that describe expected result data and format	There is one JSON object for each feature. Features that were not in the `features` array are ignored.
`file_type`	String	Identifier file type	File type can be different from the one sent in the query request (according to the type to identify)

te Response

The te feature for Threat Emulation can return these fields in the response.

Field	Type	Value	Notes
`status`	JSON Object	`code`, `label`, `message`	Status of Threat Emulation on the requested file.
> `code`	Integer	Status Code	
> `label`	String	Readable code label	
> `message`	String	Readable status message	

Field	Type	Value	Notes
`combined_verdict`	String	`benign` | `malicious`	Combined verdict of all the images. **Note** - Benign reports are not supported for local gateways.
`severity`	Integer	{1 - 4}	Combined severity of threats found. If none found, this field is not given.
`confidence`	Integer	{1 - 3}	Rating of the threat data and its relevance to this instance.
`images`	JSON object	Data for each image	See next section.

Images Object Format

Field	Type	Value	Notes
`status`	String	`found` | `not_found`	
`id`	String	Image identification string	
`revision`	Integer	Image revision number	
`report`	JSON object	Image `verdict` and all requested reports	Each report is an ID string for the Download API (on page 29).
> `verdict`	String	`benign` | `malicious`	
> `xml_report`	String	XML report ID	In response if requested. If not requested, omitted.
> `tar_report`	String	TAR report ID	In response if requested. If not requested, omitted. The `*tar.gz` file has XML report, VM snapshots of the emulation images.

Note - On local gateways, only requests for `tar.gz` reports are supported.

Available OS Image ID	Revision	Image OS and Application
e50e99f3-5963-4573-af9e-e3f4750b55e2	1	**Microsoft Windows**: XP - 32bit SP3 **Office**: 2003, 2007 **Adobe Acrobat Reader**: 9.0 **Flash Player** 9r115 and **ActiveX** 10.0 **Java Runtime:** 1.6.0u22
7e6fe36e-889e-4c25-8704-56378f0830df	1	**Microsoft Windows**: 7 - 32bit **Office**: 2003, 2007 **Adobe Acrobat Reader**: 9.0 **Flash Player:** 10.2r152 (**Plugin** & **ActiveX**) **Java Runtime:** 1.6.0u0
8d188031-1010-4466-828b-0cd13d4303ff	1	**Microsoft Windows**: 7 - 32bit **Office**: 2010 **Adobe Acrobat Reader**: 9.4 **Flash Player:** 11.0.1.152 (**Plugin** & **ActiveX**) **Java Runtime:** 1.7.0u0
5e5de275-a103-4f67-b55b-47532918fa59	1	**Microsoft Windows**: 7 - 32bit **Office**: 2013 **Adobe Acrobat Reader**: 11.0 **Flash Player:** 15 (**Plugin** & **ActiveX**) **Java Runtime:** 1.7.0u9
3ff3ddae-e7fd-4969-818c-d5f1a2be336d	1	**Microsoft Windows**: 7 - 64bit **Office**: 2013 (32bit) **Adobe Acrobat Reader**: 11.0.01 **Flash Player:** 13 (**Plugin** & **ActiveX**) **Java Runtime:** 1.7.0u9

XML Report Structure

```
 <?xml version="1.0" encoding="UTF-8"?>
<report>
  <reporttype>Summary</reporttype>
  <operating_system_reports>
    <operating_system_report>
      <osid> image id event profile id </osid>
       <Document>
         <FileName> file name </FileName>
         <FileType> file type </FileType>
         <Md5> md5 </Md5>
         <Sha1> sha1 </Sha1>
         <FileSize> file size </FileSize>
         <FileLink> name of tar.gz with malicious file </FileLink>
         <Verdict> verdict </Verdict>
         <Score> score </Score>
      </Document>
      <System>
         <Osname> image name, file name </Osname>*
         <OsInfo> image description </OsInfo>
      </System>
      <Activities>
       <Command>
         <CommandName>FileSystemEvent</CommandName>
         <ID>6</ID>
         <Time>00:00:17</Time>
         <Src>C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE</Src>
         <Dst>C:param.txt</Dst>
         <Action>Create</Action>
       </Command>

      ....
      </Activities>
      <Residues>
        <ResiduesEvent>
          <ResidueType>FileSystemEvent</ResidueType>
          <Residue>
            <ResiduePath>C: param.txt</ResiduePath>
            <ResidueAction>Create</ResidueAction>
          </Residue>

      ....
        </ResiduesEvent>

      ....
      </Residues>
      <More>
        <More> Advisories result blob </More>
      </More>
    </operating_system_report>

      ....
   </operating_system_reports>
   <reportDate> date and time </reportDate>
</report>
```

av Response Format

Field	Type	Value	Notes
`status`	Json Object	Status of Anti-Virus on the requested file: `code`, `label`, `message`	
> `code`	Integer	Status code	
> `label`	String	Readable code label	
> `message`	String	Readable status message	
`malware_info`	Json Object	Analysis of Anti-Virus for on the requested file: `signature_name`, `malware_family`, `malware_type`, `confidence`, `severity`	
`signature_name`	String	Signature name	If the file is not detected by Anti-Virus, the signature name is empty
`malware_family`	Integer	ID for malware family, if available: `{0-}`	
`malware_type`	Integer	ID for malware type, if available: `{0-}`	
`confidence`	Integer	`{0-5}`	0 for benign files
`severity`	Integer	`{0-4}`	0 for benign files

extraction Response Format

Field	Type	Value
`tex_product`	Boolean	`true` | `false` True if the queried file is already a Sandblast-safe copy.
`status`		Status of Threat Extraction on the requested file.
> `code`	Integer	Status code

Field	Type	Value
> label	String	Readable code label If value is FOUND, the other fields are given.
> message	String	Readable status message
extract_result	String	CP_EXTRACT_RESULT_UNKNOWN (Default - returned if the POD did not receive an answer from the Threat Extraction engine in 60 seconds) CP_EXTRACT_RESULT_SUCCESS CP_EXTRACT_RESULT_FAILURE CP_EXTRACT_RESULT_TIMEOUT CP_EXTRACT_RESULT_UNSUPPORTED_FILE CP_EXTRACT_RESULT_NOT_SCRUBBED CP_EXTRACT_RESULT_INTERNAL_ERROR CP_EXTRACT_RESULT_DISK_LIMIT_REACHED CP_EXTRACT_RESULT_ENCRYPTED_FILE CP_EXTRACT_RESULT_DOCSEC_FILE CP_EXTRACT_RESULT_OUT_OF_MEMORY
extracted_file_download_id	String	The download id of the extracted file, for download request ("Download API" on page 29). Only sent when extract_result = CP_EXTRACT_RESULT_SUCCESS
extraction_data	JSON object	Data of the extracted file.

extraction_data:

Field	Type	Value
input_extension	String	Uploaded filename-extension as sent by the client
input_real_extension	String	Extension as resolved by Threat Extraction
orig_file_url	String	Url to original filename - empty in cloud response
output_file_name	String	The name of the output file. The file extension may be modified after extraction, for example: docm > docx
risk	Float	Represents the risk of the part that was extracted from the document
scrub_method	String	Convert to PDF | Clean Document
protection_type	String	Protection done for scrub_method: Conversion to PDF | Content Removal
protection_name	String	"Potential malicious content extracted"
scrub_result	Float	Code result from Threat Extraction
message	String	Status message for scrub_result
scrub_activity	String	Readable result from Threat Extraction
scrub_time	Float	Threat Extraction process time
scrubbed_content	String	Content that was removed

Query Response Example


```
{
   "response": {
       "status": {
           "code": 1001,
           "label": "FOUND",
           "message": "The requested data has been found."
       },
       "md5": "da855ff838250f45d528a5a05692f14e",
       "features": [
           "av",
           "te"
           "extraction"
       ],
       "te": {
           "combined_verdict": "malicious",
           "severity": 4,
           "confidence": 3,
           "images": [
               {
                   "report": {
                       "verdict": "malicious",
                       "xml_report": "ef5f38d8-c35e-42fa-b3f1-388e681e18b9"
                   },
                   "status": "found",
                   "id": "5e5de275-a103-4f67-b55b-47532918fa59",
                   "revision": 1
               },
               {
                   "report": {
                       "verdict": "malicious",
                       "xml_report": "c7486ce7-9cde-484d-9ba4-bfc51fd88f99"
                   },
                   "status": "found",
                   "id": "7e6fe36e-889e-4c25-8704-56378f0830df",
                   "revision": 1
               }
           ],
           "status": {
               "code": 1001,
               "label": "FOUND",
               "message": "The requested data has been found"
           }
       },
       "av": {
           "malware_info": {
               "signature_name": "Exploit.JS.Pdfka.fma.W.ygrku",
               "malware_family": 22,
               "malware_type": 104,
               "sevirity": 4,
               "confidence": 5
           },
           "status": {
               "code": 1001,
               "label": "FOUND",
               "message": "The requested data has been found"
           }
       }
       "extraction":{
           "method": "pdf",
           "extract_result": "CP_EXTRACT_RESULT_SUCCESS",
           "extracted_file_download_id":
"82f67772-2116-4d29-a5be-245a434af2ae",
```

```
            "output_file_name": "MyFile.docx.pdf",
            "time": "1.374",
            "extract_content": "Database Queries",
            "extraction_data": {
                "input_extension": "docx",
                "input_real_extension": "xls",
                "message": "OK",
                "orig_file_url": "",
                "output_file_name": "MyFile.cleaned.xls.pdf",
                "protection_name": "Potential malicious content extracted",
                "protection_type": "Conversion to PDF",
                "risk": 2.0,
                "scrub_activity": "Active content was found - XLS file was
converted to PDF",
                "scrub_method": "Convert to PDF",
                "scrub_result": 0.0,
                "scrub_time": "1.374",
                "scrubbed_content": "Database Queries"
                 },
            "tex_product": false,
            "status": {
                "code": 1001,
                "label": "FOUND",
                "message": "The requested data has been found."
            }
        }
    }
}
```

Upload API

In This Section:

HTTP Upload Request 24
Upload Request HTTP Headers 24
Upload Request Format 25
Upload Request Example 26
Upload Response Format 27
Upload Response Example 28

Use the Upload API to have a client application request that Check Point Threat Prevention modules scan and analyze a file. When you upload a file to the service, the file is encrypted. It is un-encrypted during analysis, and then deleted.

Upload Requests are submitted with multiple parts: body (`request`) and file buffer (`file`).

Best Practice: Use this API after the client application used the Query API. Only if the service does not have the required data about that file, upload the file.

Note: The Upload Response headers ("Query Response HTTP Headers" on page 14) and format ("Query Response Format" on page 15) are the same as in the Query API.

HTTP Upload Request

```
HTTP POST: https://<service_address>/tecloud/api/<version>/file/upload
```

The value of the service address depends on your environment ("Accessing the API" on page 6).

Upload Request HTTP Headers

Header	Value	Example
Authorization	A valid API Key	`Authorization: YWJjZDEyMzQ`
Content-Type	Multipart boundary	`Content-Type: multipart/form-data; boundary=123456789123456789123456678`
Content-Length	Payload size	`Content-length: 1234`

Upload Request Format

Mandatory fields:

Field	Type	Value	Notes
String	File Name	File name	Service calculates the file name from the part name.

Optional fields:

Field	Type	Value	Default	Notes
`md5 | sha1 | sha256`	String	Message digest	Service calculates digests	If not given, successful upload is not validated.
`file_type`	String	File extension		Service identifies the type. **Note** - This filed is required in requests to local gateways.
features	String array	Available features	te	
<feature>	JSON object	te Request (on page 12)		There is one JSON object for each feature. Features that were not in the `features` array are ignored.

Upload Request Example

```
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="request";
Content-Type: application/json
{
    "request": {
        "md5": "da855ff838250f45d528a5a05692f14e",
        "file_name": "MyFile.docx",
        "file_type": "docx",
        "features": ["te"],
        "te": {
            "reports": ["pdf", "xml"],
            "images": [
                {
                    "id": "7e6fe36e-889e-4c25-8704-56378f0830df",
                    "revision": 1
                },
                {
                    "id": "e50e99f3-5963-4573-af9e-e3f4750b55e2",
                    "revision": 1
                }
            ]
        }
    }
}

----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="file"; filename="MyFile.docx"
Content-Type:
application/vnd.openxmlformats-officedocument.wordprocessingml.document

[Binary content of MyFile.docx]

----WebKitFormBoundary7MA4YWxkTrZu0gW
```

Upload Response Format

Field	Type	Value	Notes
status	JSON Object	`code`, `label`, `message`	Status of requested features in machine code and human readable label and message.
> code	Integer	Status Code	
> label	String	Readable code label	
> message	String	Readable status message	
`md5 | sha1 | sha256`	String	Message digest	If found, all three, otherwise only the requested digest.
`file_name`	String	Name of file saved on Check Point databases	
features	String array	Available features	See next sections
<feature>	JSON object	JSON fields that describe expected result data and format	There is one JSON object for each feature. Features that were not in the `features` array are ignored.
`file_type`	String	Identifier file type	File type can be different from the one sent in the upload request (according to the type to identify)

Upload Response Example


```
{
 "response": {
   "status": {
     "code": 1002,
     "label": "UPLOAD_SUCCESS",
     "message": "The file was uploaded successfully."
   },
   "md5": "da855ff838250f45d528a5a05692f14e",
   "file_name": "MyFile.docx",
   "file_type": "docx",
   "features": [ "te" ],
   "te": {
     "images": [
       {
         "report": {
           "verdict": "unknown"
         },
       "status": "not_found",
       "id": "5e5de275-a103-4f67-b55b-47532918fa59",
       "revision": 1
       },
       {
       "report": {
         "verdict": "unknown"
       },
       "status": "not_found",
       "id": "7e6fe36e-889e-4c25-8704-56378f0830df",
       "revision": 1
       }
     ],
   "status": {
       "code": 1001,
       "label": "FOUND",
       "message": "The requested data has been found."
       }
   }
 }
}
```

Download API

In This Section:

HTTP Download Request 29
Download Request HTTP Headers 29
Download Request Format 29

Use the Download API to have a client application download files generated by the Check Point Threat Prevention service: analysis reports, Threat Emulation sandbox outputs, and more. The request must have the ID of the file to download. Use the Query API or Upload API to get the ID.

Note: The Upload Response headers ("Query Response HTTP Headers" on page 14) are the same as in the Query API. The Upload Response format is the binary buffer with the requested file.

HTTP Download Request

```
HTTP POST:
https://<service_address>/tecloud/api/<version>/file/download?id=<id>
```

The value of the service address depends on your environment ("Accessing the API" on page 6).

The Download Request includes the ID of the file to download.

Example:

```
HTTP GET:
https://te.checkpoint.com/tecloud/api/v1/file/download?id=ef5f38d8-c35e
-42fa-b3f1-388e681e18b9
```

Download Request HTTP Headers

Header	Value	Example
Authorization	A valid API Key	`Authorization: YWJjZDEyMzQ`

Download Request Format

This request does not have a body. The ID is in the HTTP Request ("HTTP Download Request" on page 29).

Quota API

In This Section:

HTTP Quota Request 30
Quota Request Headers 30
Quota Response Format 30
Quota Response Example 31

Use the Quote API to have a client application get the current license and quota status of the API Key that you use in the `authorization` of the other APIs.

The Quota Request does not have a body.

Note - The Quota Request is not supported for services on local gateways.

HTTP Quota Request

```
HTTP GET: https://<service_address>/tecloud/api/<version>/file/quota
```

The value of the service address depends on your environment ("Accessing the API" on page 6).

Quota Request Headers

Header	Value	Example
Authorization	A valid API Key	`Authorization: YWJjZDEyMzQ`

Quota Response Format

The Quota Response HTTP Headers are the same as Query Response HTTP Headers (on page 14).

Response format:

Field	Type
remainQuotaHour	Integer
remainQuotaMonth	Integer
assignedQuotaHour	Integer
assignedQuotaMonth	Integer
hourlyQuotaNextReset	String

Field	Type
monthlyQuotaNextReset	String
quotaId	String
cloudMonthlyQuotaPeriodStart	String
cloudMonthlyQuotaUsageForThisGw	Integer
cloudHourlyQuotaUsageForThisGw	Integer
cloudMonthlyQuotaUsageForQuotaId	Integer
cloudHourlyQuotaUsageForQuotaId	Integer
monthlyExceededQuota	Integer
hourlyExceededQuota	Integer
cloudQuotaMaxAllowToExceedPercentage	Integer
podTimeGmt	String
quotaExpiration	String

Quota Response Example

```
{
"response": {
      "remain_quota_hour": 0,
      "remain_quota_month": 0,
      "assigned_quota_hour": 0,
      "assigned_quota_month": 0,
      "hourly_quota_next_reset": "0",
      "monthly_quota_next_reset": "0",
      "cloud_monthly_quota_period_start": "0",
      "cloud_monthly_quota_usage_for_this_gw": 0,
      "cloud_hourly_quota_usage_for_this_gw": 0,
      "cloud_monthly_quota_usage_for_quota_id": 0,
      "cloud_hourly_quota_usage_for_quota_id": 0,
      "monthly_exceeded_quota": 0,
      "hourly_exceeded_quota": 0,
      "cloud_quota_max_allow_to_exceed_percentage": 0,
      "pod_time_gmt": "0",
      "quota_expiration": "0"
    }
}
```