THREAT PREVENTION API

JAVIER PADILLA

THREAT PREVENTION REGIONAL EXPERT,MEXICO

jpadilla@checkpoint.com

27/Jul/2017

TOPICS

- What is the API and why is so cool!
- API clients can run everywhere.
- API demonstration.
- Share some customer needs.

INTRODUCTION

What, Why, where?

- Malware Analysis as a Service

SandBlast Service Overview

Emulators process file in RAM and are located on dedicated protected Check Point sites

Customer Se

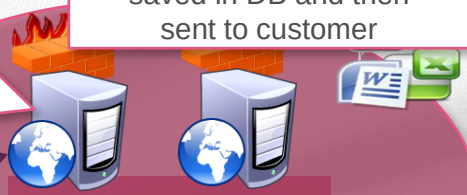
File is reassembled, encrypted and sent to the cloud (Pod)

SandBlast Emulators

Report is sent to Pod, saved in DB and then sent to customer

Emulator asks for work and receives file

Front End Servers and central databases – use Internet Cloud hosting servers. Geo DNS is used to send gateway to local Pod.



Frontend servers

DB

User Center

Enable API on appliance

A. **SK 113599**: SandBlast Agent for Browsers - working with Security Gateway or TE Appliance.
   - Enabling the Threat Prevention API
   - Portal Certificate
B. Latest JHF includes it.
C. Activate scrubd_webservice using GUI DBEdit

Unified Threat Prevention API

Where?

- Local or Cloud

Threat Prevention API capabilities supported

- AV &TE hash reputation. ['av','te']
- Submit a file to be analyzed by TE ['te']
- Get TE early verdict ['te_eb']
- Choose emulation environments.
- TE Forensics report download.
- Covert a file to PDF ['extraction']
- Clean specific content from files.

Exclusive on Cloud

- Include Evaluation Key trough HTTP headers
- Save Cookies for emulation results to be accurate.

How to Use it

Hands On

- We need some little code

Stuff we need to keep in mind when using the API.

- REST Full API
- JSON
- HTTP Methods
- HTTP Headers
- Base64 Encoding/Decoding
- multipart/form-data
- **Cookies !!!**

JSON

- Java Script Object Notation
  - ✓ Syntax for storing and exchanging data between applications.
  - ✓ Easier to use than XML.
  - ✓ Easy to read.
  - ✓ Very native to handle between different languages.
  - ✓ They work like dictionaries.
  - ✓ It will maintain the type of the variables .

JSON

- Query request to AntiVirus using MD5

```
{
    "request": [
        {
            "md5": "8dfa1440953c3d93daafeae4a5daa326",
            "features": ["av"]
        }
    ]
}
```

JSON

- Response from the previous query


```
{
  "response": [
    {
      "status": {
        "code": 1001,
        "label": "FOUND",
        "message": "The request has been fully answered."
      },
      "md5": "8dfa1440953c3d93daafeae4a5daa326",
      "file_type": "",
      "file_name": "",
      "features": [
        "av"
      ],
      "av": {
        "malware_info": {
          "signature_name": "Trojan-Downloader.Win32.Agent.T.kdu",
          "malware_family": 36471,
          "malware_type": 87,
          "severity": 4,
          "confidence": 5
        },
        "status": {
          "code": 1001,
          "label": "FOUND",
          "message": "The request has been fully answered."
        }
      }
    }
  ]
}
```

HTTP METHODS

- HTTP GET- Request information/data.
  - DOWNLOAD *(TE reports / scrubbed files)*
  - QUOTA *(SandBlast **Cloud** Service quota)*
- HTTP POST- Create a resource / Submit data
  - QUERY *(AV & TE hash reputation)*
  - UPLOAD *(Send files to TE & TEX)*

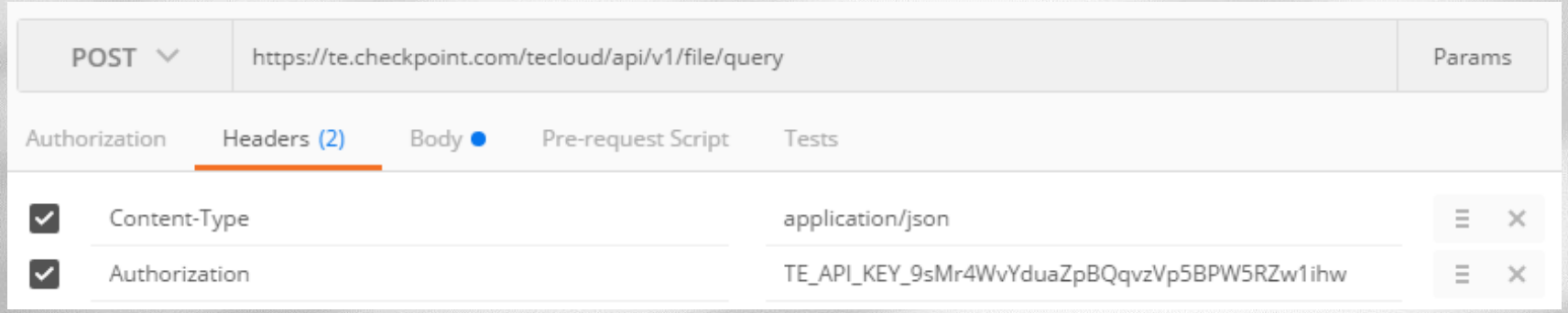
HTTP HEADERS

- Authorization

  This is how we submit our API Key, this header is part of **all requests.**

- Content-Type : application/json

  The type of data we are sending on this request are **JSON** objects



A flexible HTTP Client, some examples

A. Python + requests
B. Curl + Bash
    1. Use jq or JSON.sh to read JSON Data
C. Postman
    1. https://www.getpostman.com/collections/517bd6087b29529b17a7

Malware Analysis proposed flow

Supported
File Format

AV DB /query POST Request
feature: ‘av’, ‘te’
md5 : 23e234242qre
file_name : suspiciousfile.doc

AV DB /query Response
feature: ‘av’, ‘te’
md5 : 23e234242qre
UNKNOWN, FOUND

status: code: 1001
feature: ‘av’
label: FOUND
THIS IS A KNOWN MALWARE

status: code: 1004
feature: ‘av’,’te’
label: NOT_FOUND
IT IS UNKOWN, UPLOAD IT

TE /upload POST Request
feature: ‘te’
md5: 23e234242qre
Images: <image_uuid>,<rev>
report: pdf,xml

status: code: 1003
feature: ‘te’,
label: PENDING

TE Status /query POST
feature: ‘te’
md5 : 23e234242qre
file_name : suspiciousfile.doc

status: code: 1001
feature: ‘te’, label: FOUND
pdf_report: UUID
verdict: malicious / benign

/download GET Request
Params : < report_UUID >
Get forensics report

status: code: 1001
feature: ‘te’
label: FOUND
pdf_report: UUID
Verdict: malicious / benign

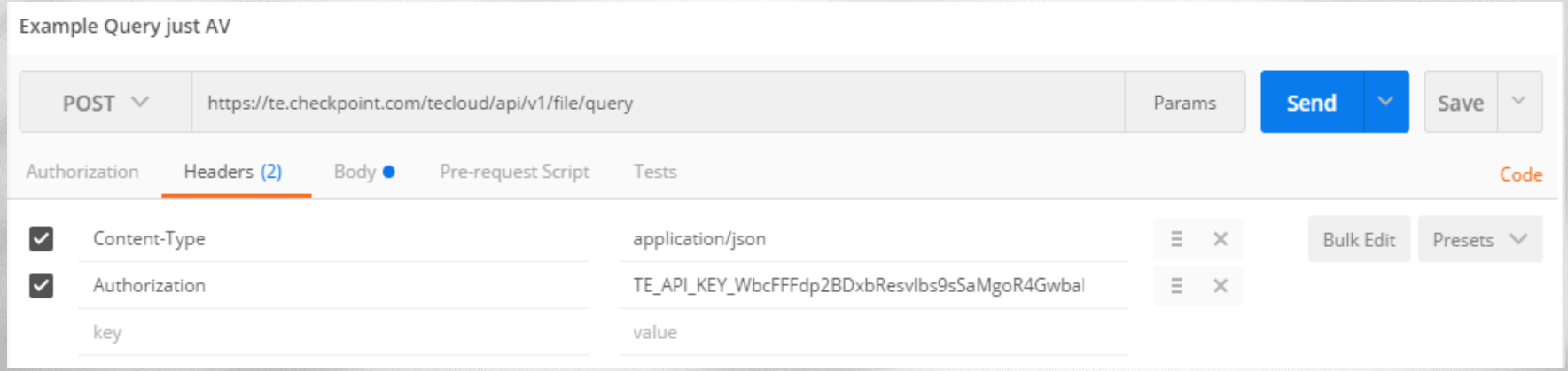
QUERY API

What do you know about this file?

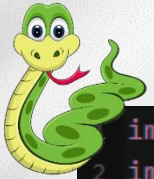
- POST Method

QUERY API

1. API uses POST Method
2. Multiple hashes are supported (SHA1 & MD5*)
3. API Endpoint is https://te.checkpoint.com/tecloud/api/v1/file/query
4. Set Headers:
    1. Content-type : application/json
    2. Authorization : < Evaluation TP API Key>



Is this hash known to be malicious?



```python
import requests
import json

service = "https://172.20.22.220:18194/tecloud/api/v1/file/"
request = { "request":  [
    {   "md5": "764bfd2b86c9eb5ad1432514c8802a10",
    "features": ["av","te"] }]}

response = requests.post(service + 'query', data = json.dumps(request), verify=False)
responseText = json.loads(response.text)
```

QUERY API request for AntiVirus only

Query ar

{ "request":


```
{
  "response": [
    {
      "status": {
        "code": 1001,
        "label": "FOUND",
        "message": "The request has been fully answered."
      },
      "md5": "7234f226dd368a5e2caa7998a758d418",
      "file_type": "",
      "file_name": "",
      "features": [
        "av"
      ],
      "av": {
        "malware_info": {
          "signature_name": "Backdoor.Win32.Androm.lvbh.a",
          "malware_family": 236921,
          "malware_type": 98,
          "severity": 4,
          "confidence": 5
        },
        "status": {
          "code": 1001,
          "label": "FOUND",
          "message": "The request has been fully answered."
        }
      }
    }
  ]
}
```

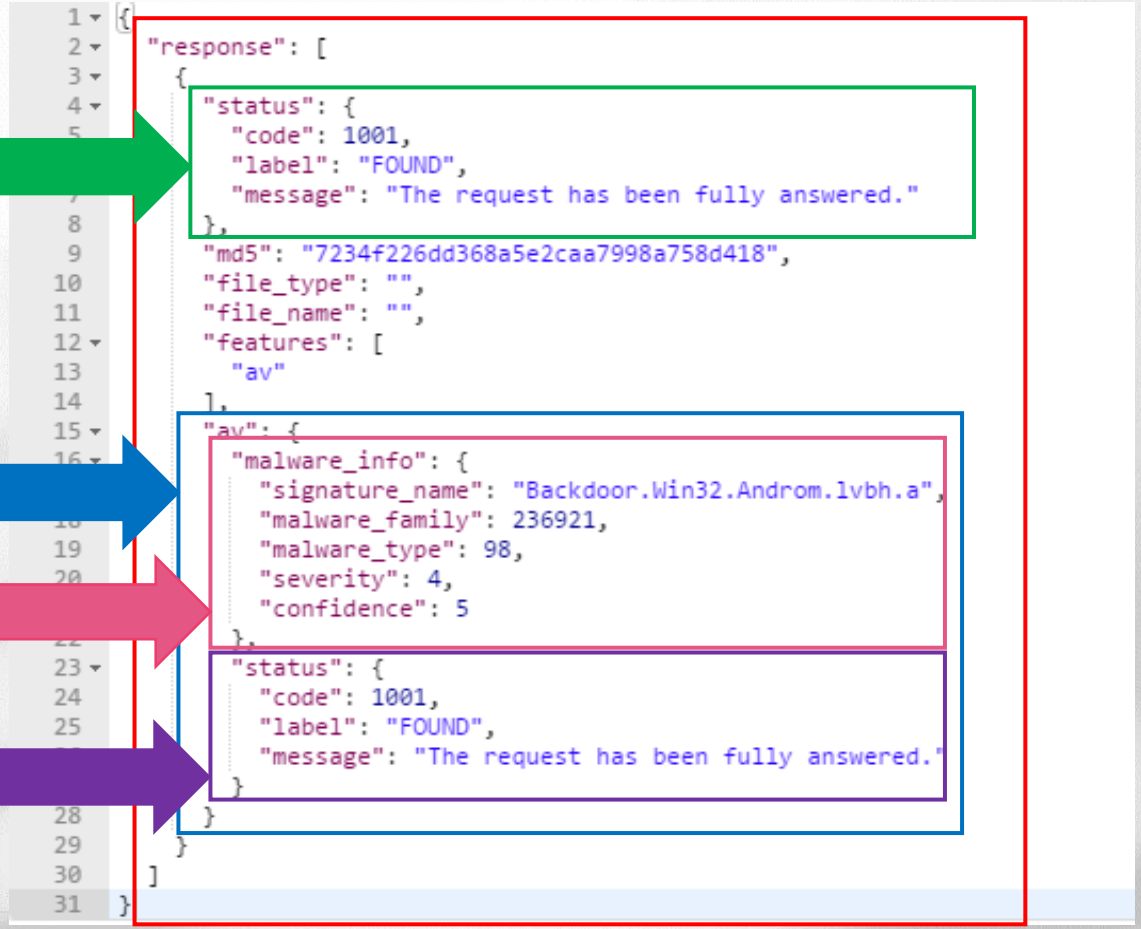
QUERY API request for AntiVirus only

Status of the FULL request.

Antivirus Response details

Malware info for the specified Hash

Status of the AV request



```
{
  "response": [
    {
      "status": {
        "code": 1001,
        "label": "FOUND",
        "message": "The request has been fully answered."
      },
      "md5": "7234f226dd368a5e2caa7998a758d418",
      "file_type": "",
      "file_name": "",
      "features": [
        "av"
      ],
      "av": {
        "malware_info": {
          "signature_name": "Backdoor.Win32.Androm.lvbh.a",
          "malware_family": 236921,
          "malware_type": 98,
          "severity": 4,
          "confidence": 5
        },
        "status": {
          "code": 1001,
          "label": "FOUND",
          "message": "The request has been fully answered."
        }
      }
    }
  ]
}
```

UPLOAD API

What could you know about this file?

- POST Method

UPLOAD API

1. API uses POST Method
2. API Endpoint is https://te.checkpoint.com/tecloud/api/v1/file/upload
3. Set Headers:
    1. Content-type : application/json
    2. Authorization : < Evaluation TP API Key>
4. Files and requests encoded as multipart/form-data
    1. <form action="upload" method="post" enctype="multipart/form-data">

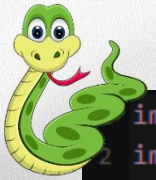
Upload to TE

```python
import requests
import json

service = "https://172.20.22.220:18194/tecloud/api/v1/file/"
print 'Uploading API\n'

with open('mal_files/1fdb378f04b48f77aefde86f38c0473729cdeefa.doc','r') as sample:
    request = {"request": {"md5": 'xxx',"file_name": 'malware-file.doc',
                           #"features": ["te"], .. by default
                           #Selecting no images means emulate with the default set
                    "te": {"reports": ["pdf","xml"]},
                    "images": [{"id": "7e6fe36e-889e-4c25-8704-56378f0830df","revision": 1}, #Win7 32bit
                        {"id": "e50e99f3-5963-4573-af9e-e3f4750b55e2","revision": 1}, #WinXP
                        {"id": "8d188031-1010-4466-828b-0cd13d4303ff","revision": 1},
                        {"id": "5e5de275-a103-4f67-b55b-47532918fa59","revision": 1},
                        {"id": "3ff3ddae-e7fd-4969-818c-d5f1a2be336d","revision": 1},
                        {"id": "6c453c9b-20f7-471a-956c-3198a868dc92","revision": 1}]}}
    files = {'file':(sample),'request': json.dumps(request)}
    print '> Request for uploading a file with content-type multiform/metadata'
    req = requests.post(service + 'upload', files=files , verify=False)
    print req.text
    #print req.cookies['te_cookie']
```

Upload to TEX



```python
import requests
import json

service = "https://te.checkpoint.com/tecloud/api/v1/file/"
headers = {'Authorization': 'TE_API_KEY_6M7ooC3diibjLn5Azfpf5na4AKq6PoaSVQNiiWjp'}
print 'Uploading API\n'

with open('mal_files/1fdb378f04b48f77aefde86f38c0473729cdeefa.doc','r') as sample:
    request = {"request": {"md5": 'xxx',
                           "file_name": 'malware-file.doc',
                           "features": ["extraction"],
                           "extraction": {"method": "convert"}
                }}

    files = {'file':(sample),'request': json.dumps(request)}
    print '> Request for uploading a file with content-type multiform/metadata'
    req = requests.post(service + 'upload', files=files , headers=headers, verify=False)
    print req.text
```

DOWNLOAD API

Give me the reports from the analysis.

- GET Method

DOWNLOAD API

1. API uses GET Method
2. API Endpoint is https://te.checkpoint.com/tecloud/api/v1/file/download
3. Set Headers:
    1. Content-type : No content type defined
    2. Authorization : < Evaluation TP API Key>
4. Request parameters are included in the body of the request
    1. `params = {“id”:<“report_id”> }`

Download TE Report

```python
import requests
import json

service = "https://172.20.22.220:18194/tecloud/api/v1/file/"

print "Fething report "
downloadURL = service + "download"
params = {"id": "64926B0F-C1AC-D84F-9F5D-A68431A636A9"}
req = requests.get( downloadURL, params=params, verify=False)
            #-Iterate trough req.iter_content to get the report
with open ("report.xml","w") as report:
    report.write(req.content)

```

QUOTA API – only for cloud

What are my limits?

- GET Method

QUOTA API

1. API uses GET Method
2. API Endpoint is https://te.checkpoint.com/tecloud/api/v1/file/quota
3. Set Headers:
    1. Content-type : No content type defined
    2. Authorization : < Evaluation TP API Key>

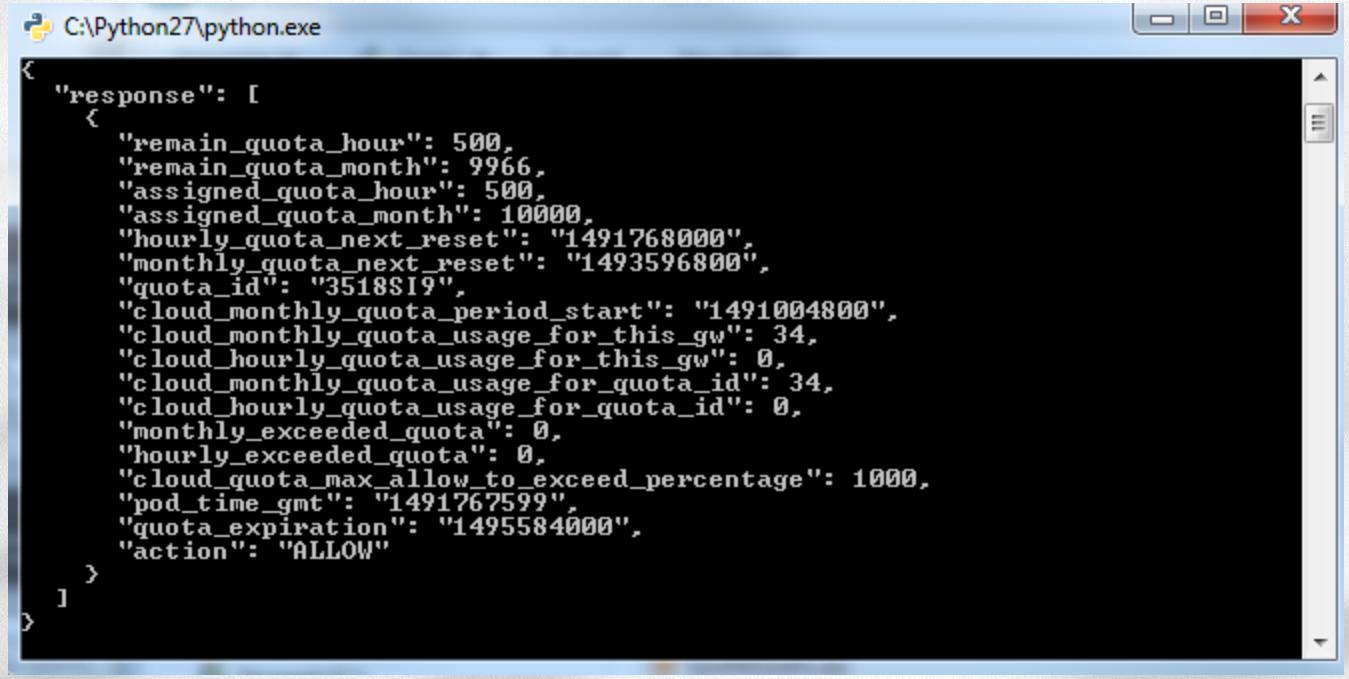
What is the quota status?

```python
import requests
import time

service = "https://te.checkpoint.com/tecloud/api/v1/file/"
token = "TE_API_KEY_6M7ooC3diibjLn5Azfpf5na4AKq6PoaSVQNiiWjp"
headers = {'Authorization': token}
quota = requests.get(service + 'quota', headers = headers, verify=False)

print quota.text
time.sleep(10)
```

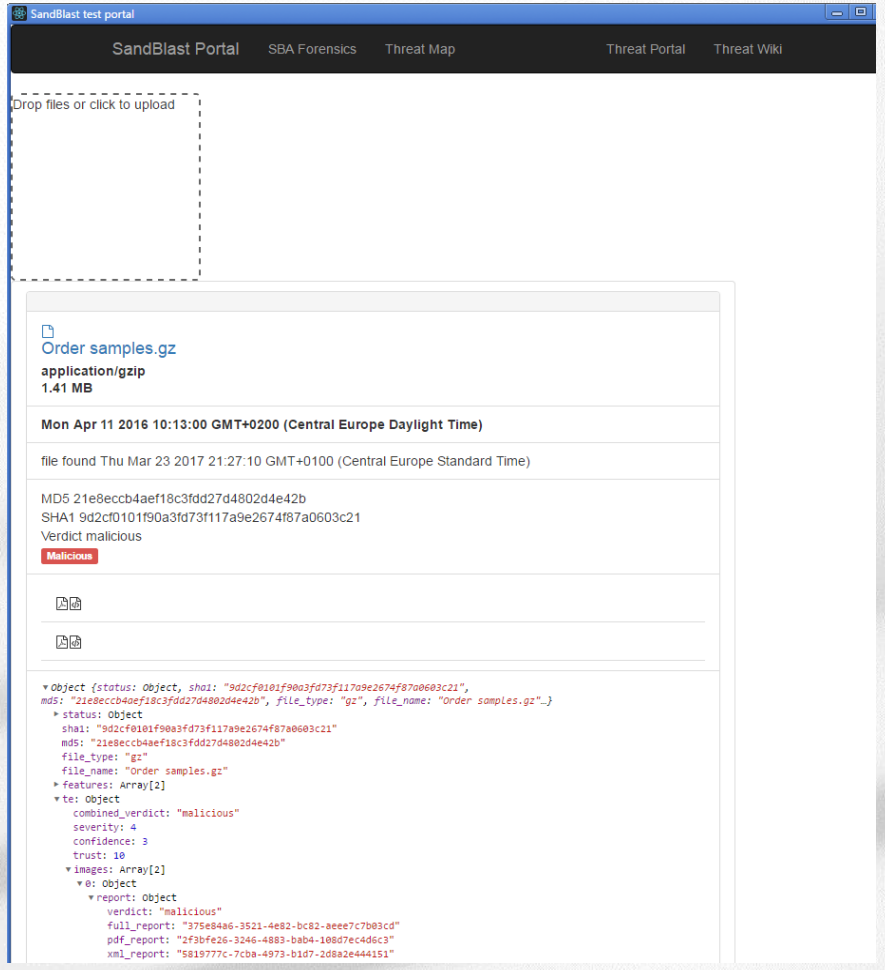
Quota is almost new.



Code Resources

Threat Prevention API reference guide
Threat Prevention API Python Client https://bitbucket.org/chkp/teapi/overview
SandBlast upload portal (NodeJS) by Martin Koldovsky https://github.com/mkol5222/sbport
PooMA (Python) https://bitbucket.org/devilbsd/pooma
Minimal requests (Python) https://bitbucket.org/devilbsd/tpapi/

One File Portal
NodeJs based



In resume

- Check Point's Threat Prevention Unified API is indeed very cool.
- API clients can run everywhere.
- Know about your tools and resources for API projects.

Thank You.