Disarmer API

User Guide and Reference

Version 8.3

February, 2019

www.votiro.com

Contents

1 Using this Document

This API document describes version 3.1 of the programming interface to Votiro's Disarmer engine using a hosted Web Service in the cloud or on premises.

For more information about Disarmer, Votiro's core sanitization engine, refer to the Disarmer User Guide.

2 Introduction

Votiro's Disarmer exposes a REST API for adding Disarmer's protection to any application. The Votiro solution disarms threats from documents stored, accessed, shared, and collaborated on across multiple devices and data sources. In addition to email, file servers, and desktop applications, Votiro Disarmer offers various capabilities to help secure structured data, in many known formats, across the entire organization.

Maintaining full functionality of safe files, Votiro Disarmer protects against all known and unknown malicious content threats.

Additional information regarding Disarmer technology, sanitization policies, and supported file types can be found in the Votiro Disarmer User Guide and on the Votiro website: https://www.votiro.com.

2.1 Named Policies

It is recommended to use Votiro Management for defining policies that meet the requirements that are specific to your organization. The result is a named policy.

For more information, see the sections on Management and Policies in the Disarmer User Guide.

2.2 Simple Sanitization Policies

Simple policies provide an alternative to named policies.

Simple policies are predefined, action-value pairs that are attached to every sanitization request. The value is always true or false. Using simple policies enables you to set up your system quickly. By default, all attributes in a simple policy are true.

When simple private policies reach the Disarmer server, they are translated to a predefined, hard-coded set of policy rules. The translation cannot be changed.

A description of simple sanitization policies is provided here.

> Note
> Simple policies are more limited than named policies and are not recommended.

Table 1: Simple Sanitization Policies

Parameter Name	Description
CleanPdf	Specifies that all PDF files are to be inspected and sanitized.
CleanImages	Specifies that all image files are to be inspected and sanitized.

Parameter Name	Description
CleanOffice	Specifies that all Microsoft Office, and Hancom Office files are to be inspected and sanitized.
CleanCad	Specifies that AutoCAD SFC, JWW, DWS, DWT, DWG, and DXF files are to be inspected and sanitized.
ExtractEmls	Extracts and sanitizes email files and their attachments recursively.
BlockPasswordProtectedArchives	Password-protected archive files are blocked by the Disarmer system, meaning that the files cannot be inspected. This attribute does not block Microsoft Office files or PDF files. You might want to set the value of this attribute to false, if you have recipients in your organization who must receive password-protected Office files.
BlockPasswordProtectedOffice	Password-protected Microsoft files are blocked by the Disarmer system, meaning that the files cannot be inspected. You might want to set the value of this attribute to false, if you have recipients in your organization who must receive password-protected Office files.
BlockPasswordProtectedPdfs	Password-protected PDF files are blocked by the Disarmer system, meaning that the files cannot be inspected. You might want to set the value of this attribute to false if you have recipients in your organization who must receive password-protected PDF files.
BlockAllPasswordProtected	All password-protected files are blocked, regardless of file type. This attribute is used by the Disarmer system, meaning that the files cannot be inspected. This password-protection attribute overrides any other password-protection attribute that has been given a false value.
BlockUnsupported	File types that are unsupported by Disarmer sanitization engine are blocked automatically.

Parameter Name	Description
BlockFakeFiles	When a file extension does not match the characteristics specified for files of that type (by the organization who created the specific file type) the files are determined to be fake and are blocked.
BlockUnknownFiles	Files that are not recognized by the TTD scanner are blocked.
ExtractArchiveFiles	Extracts and sanitizes archive files recursively.
BlockScriptFiles	Files that are recognized by the TTD as script files are blocked.
BlockBinaryFiles	Files that are recognized by the TTD as binary files are blocked.
BlockEquationOleObject	Files that are recognized by the TTD as Equation OLE objects are blocked.
ScanVirus	Files are scanned by integrated antivirus engines. If a virus is detected, the file is blocked.

2.3 File Process Statuses

Table 2: File Process Statuses

Status Code	Status Name	Status Description
0	Passed	File was not processed because none of the specified sanitization policies matched the file.
1	Sanitized	File was sanitized.
2	Blocked	File was blocked.

2.4 Responses and Error Codes

Votiro uses conventional HTTP response codes to indicate the success or failure of an API request. In general, codes in the 2xx range indicate success, codes in the 4xx range indicate an error that failed because of the information provided (for example, a required parameter was omitted, a bad parameter formation, and so on), and codes in the 5xx range indicate an error with Votiro's servers.

Table 3: Error Codes

Status Code	Code Value	Description
200	OK	The request was successful.
400	Bad Request	The request included a non-existent resource, for example, an incorrect RequestID.
409	Conflict	There was an issue with the request parameters. The response includes detailed error information.
429	Too many requests	The request queue is full.
500	Internal Server Error	The server failed to process the request.

Error Response Parameters

Table 4: Error Response Parameters

Parameter	Description	Type
Errors	List of error description objects.	list
Errors [n] Description	The ASCII encoded file name. If the file was sanitized successfully, this is the file name. If the file was blocked, then the suffix _blocked.pdf is appended to the file name.	string

Error Response Example

```
{
    "Errors": [
        {
            "Description": "File name is empty."
        }
    ]
}
```

3 Reference

This reference provides descriptions of the calls in the Votiro API.

3.1 Upload a File to Sanitize

3.1.1 Functional Overview

You upload potentially malicious files to the Votiro server, where it is queued for sanitization, and then sanitized. You can specify sanitization policies that dictate how the file is scanned and sanitized.

3.1.2 Description

Path	http[s]://<*base address*>/v3/upload/file?filename=<*filename*>
Method	POST

3.1.3 Request Data

Votiro recommends that you encode the URL using URL Encoding. This means that the file name can contain Unicode characters.

Request URL Parameters

Request parameters are case sensitive.

Parameter	Description	Type	Required/optional
base address	The base address is configured when you install the Votiro API, or configured directly by Votiro.	string	Required
filename	Name of the file that you are uploading for sanitization. To achieve the most accurate sanitization, you must include the file extension. The file name and file extension should be identical to the file that the user received.	string	Required
PolicyRule	A collection of policy rules. See **Simple Sanitization Policies** on page 5.	boolean	Optional

Parameter	Description	Type	Required/optional
PolicyName	File name of a predefined policy rules collection (also called *named policy*). For PolicyName, specify the name of the policy, without any extension such as .xml.	string	Optional
Password	Allows sanitization of password-protected archive files with a file type of .zip and .7z. See **Simple Sanitization Policies** on page 5. Password will be used only with PolicyName.	string	Optional

Notes

- Do not include PolicyName and PolicyRule in the same request.
- Uploading 0-byte files is not supported.

Request Body

In the request body, supply the content of the file. The content should contain the file's binary data, that is, in the same way it is saved in the user's storage.

Request Example

Posting a password-protected file named test.zip, providing the password 123456 and the context identifier joe.

```
POST https://api.votiro.com/v3/upload/file?filename
=test.zip&policyname=xxxxxx&password=123456
HTTP/1.1
Content-Length: 712692
Content-Type: application/octet-stream
Accept: */*
Ocp-Apim-Subscription-Key: <YOUR SUBSCRIPTION KEY>
ContextIdentifier: joe
```

Notes

When using the Password parameter, you must set the password-protected policy to allow. For more information, see the "Simple Policies" section in the Disarmer User Guide.

The password must be provided via the URL using the format password=<password>
and encoded....

In case of a password-protected 7-Zip file that is being delivered with a wrong password, file will be blocked with an "error in sanitization process".

Request Header Parameters

Request parameters are case sensitive.

Parameter	Description	Type	Required/optional
Ocp-Apim-Subscription-Key	The subscription key you received from Votiro.	string	Required
Content-Type	Value must be "Application/octet-stream"	string	Required
ContextIdentifier	Identifier that enables easy identification.	string	Optional

Request Examples

Upload a PDF file

```
POST https://api.votiro.com/v3/upload/file?filename
=test.pdf HTTP/1.1
Content-Length: 712692

Content-Type: application/octet-stream
Accept*/*:
Ocp-Apim-Subscription-Key: <YOUR SUBSCRIPTION KEY

[Actual File Binary Content]
```

Upload a PDF file with a Unicode name

```
POST https://api.votiro.com/
v3/upload/file?filename=%2F%E3%83%86%E3%82%B9%E3%83%88.pdf
HTTP/1.1
Content-Length: 712692
Content-Type: application/octet-stream
Accept*/*: Ocp-Apim-Subscription-Key: <YOUR SUBSCRIPTION KEY>
[Actual File Binary Content]
```

Upload a ZIP file using policy rules

```
POST https://api.votiro.com/v3/upload/file?filename
=test.zip&BlockPasswordProtectedArchives=false&BlockUnsupported
```

```
=false HTTP/1.1
Content-Length: 712692
Content-Type: application/octet-stream
Accept*/*:
Ocp-Apim-Subscription-Key: <YOUR SUBSCRIPTION KEY>

[Actual File Binary Content]
```

Upload File Using Named Policy

```
POST https://ap-
i.votiro.com/v3/upload/file?filename=test.zip&PolicyName
=xxxxxx HTTP/1.1
Content-Length: 712692
Content-Type: application/octet-stream
Accept*/*:
Ocp-Apim-Subscription-Key: <YOUR SUBSCRIPTION KEY>

[Actual File Binary Content]
```

3.1.4 Response Data

Response Parameters

Parameter	Description	Type
RequestID	Upload request ID.	string
UsedRules	Policy rules used in the request.	boolean
PolicyName	Contains the name of predefined policy rules collection (named policy), instead of policy rules.	string

Response Examples

Response to PDF File Upload

```
{
  "RequestID": "4d6888d1-5ab5-4cf5-9d19-d43f16fd01d8"
      "UsedRules": {
        "CleanOffice": true,
        "CleanPdf": true,
        "CleanImages": true,
        "CleanCad": true,
        "ExtractEmls": true,
        "BlockPasswordProtectedArchives": true,
        "BlockPasswordProtectedOffice": true,
        "BlockPasswordProtectedPdfs": true,
        "BlockAllPasswordProtected": true,
        "Blockunsupported": true,
        "ScanVirus": true,
        "BlockUnknownFiles": true,
        "ExtractArchiveFiles": true,
        "BlockEquationOleObject": true,
        "BlockBinaryFiles": true,
        "BlockScriptFiles": true,
        "BlockFakeFiles": true
```

```
    }
}
```

Response to File Upload Using Named Policy

```
{
    "RequestID": "4d6888d1-5ab5-4cf5-9d19-d43f16fd01d8",
    "UsedNamedPolicy": "Sales and Marketing"
}
```

3.2 Check the Status of a Sanitization Request

3.2.1 Functional Overview

After you upload a file for sanitization, you check the file-processing status using the RequestID, that you received in the `POST` method (see **Upload a File to Sanitize** on page 9).

3.2.2 Description

Path	http[s]://<*base address*>/v3/file/<*RequestID*>/status
Method	GET

3.2.3 Request Data

Request URL Parameters

Request parameters are case sensitive.

Parameter	Description	Type	Required/optional
RequestID	Upload request ID	string	Required

Request Header Parameters

Request parameters are case sensitive.

Parameter	Description	Type	Required/optional
Ocp-Apim-Subscription-Key	The subscription key you received from Votiro.	string	Required

Request Example

```
GET https://api.votiro.com/v3/file/c94f00a4-7a36-4152-977d-dda71c-ccfb95/status HTTP/1.1
```

```
Ocp-Apim-Subscription-Key: <YOUR SUBSCRIPTION KEY>
```

3.2.4 Response Data

Response Parameters

Parameter	Description	Type
Status	The status of the sanitization request	string

Sanitization Request Statuses

Status Name	Description
Queued	The file you uploaded is in the sanitization queue.
Processing	The file you uploaded is being processed by the Votiro system.
Done	The file you uploaded is sanitized, and you can download the file.
Error	The file you uploaded was not sanitized because of an internal Votiro error.
Blocked	The file you uploaded was blocked in the sanitization process. Download the file for more information on why the file was blocked.
LimitExceeded	Exceeded the number of uploads.

Response Example

```
{
  "Status": Done
}
```

3.3 Download a Sanitized File

3.3.1 Functional Overview

You can download a sanitized file with the status of Done or Blocked.

The download request for a blocked file returns a PDF with the reason the file was blocked in the sanitization process.

The file name is returned in all response headers.

3.3.2 Description

Path	http[s]://<*base address*>/v3/file/<*RequestID*>
Method	GET

3.3.3 Request Data

Request URL Parameters

Request parameters are case sensitive.

Parameter	Description	Type	Required/optional
RequestID	Upload request ID	string	Required

Request Header Parameters

Request parameters are case sensitive.

Parameter	Description	Type	Required/optional
Ocp-Apim-Subscription-Key	The subscription key you received from Votiro	string	Required

Request Example

```
GET https://api.votiro.com/v3/file/c94f00a4-7a36-4152-977d-dda71c-ccfb95 HTTP/1.1
Ocp-Apim-Subscription-Key: <YOUR SUBSCRIPTION KEY>
```

3.3.4 Response Data

Response Body (Content Disposition)

Parameter	Description	Type
attachment		string
filename	The ASCII encoded file name. If the file was sanitized successfully, this is the file name. If the file was blocked, then the suffix _ blocked.pdf is appended to the file name.	string

Parameter	Description	Type
filename*	The UTF-8 URL encoded file name. If the file was sanitized successfully, this is the file name. If the file was blocked, then the suffix _ blocked.pdf is appended to the file name.	string

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/octet-stream
Server: Microsoft-HTTPAPI/2.0
Content-Disposition: attachment; filename=test.pdf; filename*=UTF-
8''test.pdf
Date: Fri. 13 Nov 2016 13:21:59

[Actual File Binary Content]
```

3.4 Download a Sanitization Report

3.4.1 Functional Overview

The report is a summary detailing which engines were executed on each item and sub-item during the sanitization process.

3.4.2 Description

Path	http[s]://<*base address*>/v3/-file/<*RequestID*>/report
Method	GET

3.4.3 Request Data

Request URL Parameters

Request parameters are case sensitive.

Parameter	Description	Type	Required/optional
RequestID	Upload request ID	string	Required

Request Header Parameters

Request parameters are case sensitive.

Parameter	Description	Type	Required/optional
Ocp-Apim-Subscription-Key	The subscription key you received from Votiro	string	Required

Request Example

```
GET https://api.votiro.com/v3/file/c94f00a4-7a36-4152-977d-dda71c-ccfb95/report HTTP/1.1
Ocp-Apim-Subscription-Key: <YOUR SUBSCRIPTION KEY>
```

3.4.4 Response Data

Response Body

Parameter	Description	Type
FileName	Relative path of artifact file.	string
FileType	File type information for the current artifact as determined by Votiro File Type Discoverer.	object
FileType.Code	File type code	number
FileType.Type	Artifact file type description	string
FileType.Family	Artifact file type family description	string
Events	Events for the current artifact.	array of Event objects
Event.Id	Event code. See the following table.	number
Event.Details	Event textual description. Subject to changes.	string
Event.Severity	Urgency of the event: The valid integer values are 0-6, where 6 is the most severe.	number
Event.Category	Category. Can be one of: • Trace • System • Indicator	object
Event.SubCategory	Subcategory. See the following table. The subcategory name is unique within the category that is belongs to.	object

Parameter	Description	Type
Event.Value	Event type. It is unique within the subcategory that is belongs to.	number
Event.Name	SubCategory name. See the following table.	string
Children	Sub-objects that were processed separately from the sanitized file, for example, a ZIP archive file or nested Office document.	array

Report Events

Category	Event Code	Event Name	Sub-Category	Details
Trace	10000010	True File Type	File Type Discoverer	File {FileName} was recognized as {FileType}.
Trace	10010010	Antivirus Scan	AV Scan	File {FileName} was successfully scanned by AV {AVEngine}.
Trace	10020110	Sanitization Done	File Process	File {FileName} sanitization process successfully ended.
Trace	10020200	File Block	File Process	File {FileName} was blocked as a result of the sanitization process.
Trace	10050100	Block - Policy	Blocker	File {FileName} was blocked due to your organization policy violation [{Policy}] in the sanitization process.
Trace	10050200	Block - Antivirus	Blocker	Virus found by {AVEngine} in file {FileName}.
Trace	10050500	Block - Error	Blocker	File {FileName} was blocked due to an error in Sanitization process.
Indicator	50010000	Suspicious Macro	Macro Analyzer	Suspicious Office macro detected.
Indicator	50010010	Suspicious Auto Execution Macro	Macro Analyzer	Suspicious Office macro detected [Auto Execution].
Indicator	50010020	Suspicious File System Activity Macro	Macro Analyzer	Suspicious Office macro detected [File System Activity].

Category	Event Code	Event Name	Sub-Category	Details
Indicator	50010030	Suspicious Out Of Document Interaction Macro	Macro Analyzer	Suspicious Office macro detected [Out-Of-Document Interaction].
Indicator	50020010	Suspicious Fake File	File Type Discoverer	Suspicious fake file [Extension does not match file structure] was detected in the artifact.
Indicator	50020020	Suspicious Unknown File	File Type Discoverer	Unknown file [Data file or unidentified file type] was detected in the artifact.
Indicator	50020110	Suspicious Executable File	File Type Discoverer	Executable file was detected in the artifact.
Indicator	50020120	Suspicious Script File	File Type Discoverer	Script file was detected in the artifact.
Indicator	50030100	Suspicious Threat File	AV	AV {AVEngine} detects a threat ({ThreatType}) in file {FileName}.
Indicator	50040010	External Program Run Action	Active Element	External Program Run Action detected in file {Filename}.
Indicator	50050010	Dynamic code exception	JavaScript Analyzer	Dynamic code exception detected in file {Filename}.

Event codes respect the following scheme:

LLRCCTTR

where L, R, C, T are digits [0-9].

- LL specifies the event main category.
- CC specifies the sub-category.
- TT specifies the specific event type.
- R is reserved for future use and must be ignored.

Examples

- 50020110 represents an Indicator event (LL=50) of category Suspicious Executable File (C=20), specifying that an executable artifact (TT=11) was found.
- 10000010 represents a Trace event (LL=10) of category FTD (C=00), specifying that a discovered file type (TT=01) was found.

Response Example

```
HTTP/1.1 200 OK
{
  "FileName": "sample_file.ppt",
  "FileType": {
    "Code": 16,
    "Type": "Power Point",
    "Family": "Microsoft Office"
  },
  "Events": [
    {
      "Id": 10000010,
      "Details": "File sample_file.ppt was recognized as [16]
Power Point (Microsoft Office)",
      "Severity": 2,
      "Category": {
        "Value": 10000000,
        "Name": "Trace"
      },
      "SubCategory": {
        "MainCategory": {
          "Value": 10000000,
          "Name": "Trace"
        },
        "Value": 0,
        "Name": "File Type Discoverer"
      },
      "Value": 10,
      "Name": "File Type Discoverer"
    },
    {
      "Id": 10010010,
      "Details": "File sample_file.ppt was successfully scanned by
AV AviraAntiVirus",
      "Severity": 2,
      "Category": {
        "Value": 10000000,
        "Name": "Trace"
      },
      "SubCategory": {
        "MainCategory": {
```

```
            "Value": 10000000,
            "Name": "Trace"
          },
          "Value": 10000,
          "Name": "AV Scan"
        },
        "Value": 10,
        "Name": "AV Scan"
      },
      {
        "Id": 10020110,
        "Details": "File sample_file.ppt sanitization process suc-
cessfully ended",
        "Severity": 2,
        "Category": {
          "Value": 10000000,
          "Name": "Trace"
        },
        "SubCategory": {
          "MainCategory": {
            "Value": 10000000,
            "Name": "Trace"
          },
          "Value": 20000,
          "Name": "File Process"
        },
        "Value": 110,
        "Name": "File Process"
      }
    ],
    "Children": []
}
```


3.5 Get Service Information

3.5.1 Functional Overview

You can query the API version and Disarmer version of the system. The response is in the form of a JSON structure.

3.5.2 Description

Path	http[s]://<*base address*>/v3/info
Method	GET

3.5.3 Request Data

Request Header Parameters

Request parameters are case sensitive.

Parameter	Description	Type	Required/optional
Ocp-Apim-Sub-scription-Key	The subscription key you received from Votiro	string	Required

Request Example

```
GET https://api.votiro.com/v3/info HTTP/1.1
Ocp-Apim-Subscription-Key: <YOUR SUBSCRIPTION KEY>
```

3.5.4 Response Data

Response Parameters

Parameter	Description	Type
ApiVersion	The Disarmer API version	string
SdsVersion	The Disarmer version	string

Response Example

```
{
  "ApiVersion": "3",
  "SdsVersion": "2.2.2.333"
}
```