GYTPOL Validator

API guide

Doc: GYT-TEC-014
Release: 2
Date: 22th November 2023

Confidential: GYTPOL and approved recipients

Total pages:	24

Doc. Title:	GYTPOL Validator API Guide

Doc. No.:	GYT-TEC-014	**Classification:**	Confidential
Revision:	2	**Restriction:**	GYTPOL and approved recipients
Date:	27th Nov 2023	**Customer:**	

Owner:	Arie Frusman	**Reviewers/ Approvers:**	Matthew Album Gilad Raz Tal Kollender
Author:	Arie Frusman		

Contents

Introduction

The purpose of this document is to provide instructions to apply API (Application Programming Interface) connection to GYTPOL server from a third-party tools.

Overview

Gytpol API v1.0 covers the following use cases:

- Getting misconfigurations for a given computer
- Getting misconfigurations for all computers
- Getting the list of all computers that had misconfigurations

These use cases are addressed accordingly by the following REST API functions:

- **get_miscon_by_computer**
- **get_misconfigurations_start** to set a filter and get a token for following calls to **get_misconfigurations_next**
- **get_miscon_computers_start** to set a filter and get a token for following calls to **get_miscon_computers_next**
- **add_to_group -** to add computers to your custom group

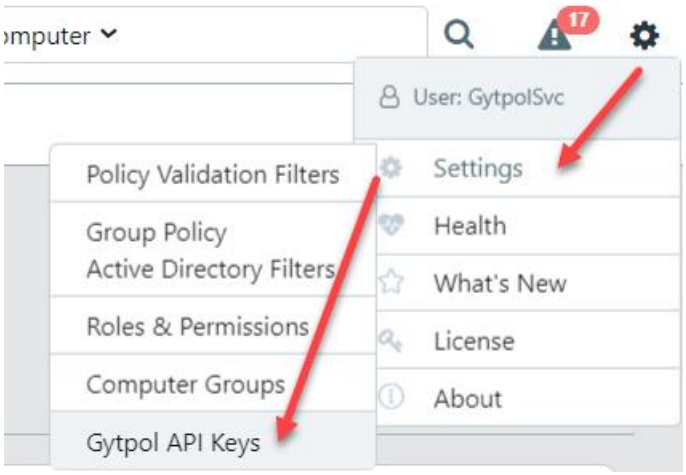
All methods are POST.

API Keys

All HTTPS requests for REST API functions must include the **x-api-key** parameter in the request header, as illustrated in the examples below.

On-Prem Customers:
To generate an API key in the GYTPOL UI, navigate to the gear icon > Settings > GYTPOL API Keys. To create a new key, click the **+** (Plus) icon and assign a name that corresponds to the system. The key will expire after 1 year by default.

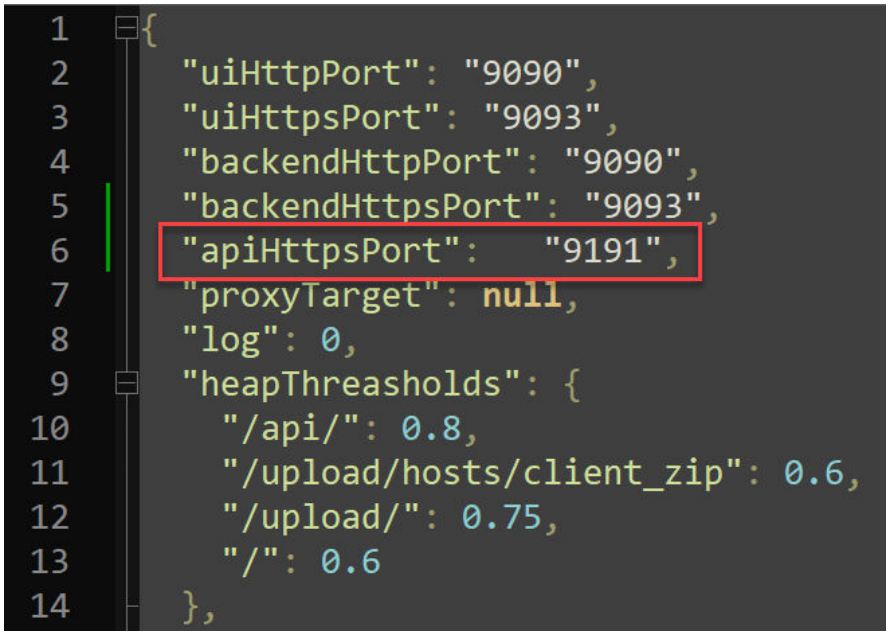
SaaS customers are advised to reach out to their customer success manager to obtain the necessary **x-api-key**.


17
User: GytpolSvc
Settings
Health
What's New
License
About
Policy Validation Filters
Group Policy
Active Directory Filters
Roles & Permissions
Computer Groups
Gytpol API Keys


New API Key
Key Name:
enter a name
Expiration:
26/11/2024
Cancel
Save

API Port

For **On-Prem customers**, the default port for API access is 9191. If you wish to use a different port, you can make the adjustment by modifying the port in the file located at **c:\gytpol\data\webserv_config.json**.
After changing the port, be sure to restart the **gytpol Web UI service** to apply the modifications.



For **SaaS customers**, there is no need to specify a port, as the URL utilizes port **443**.

get_miscon_by_computer

This REST API function returns misconfigurations given a computer name and optionally Windows domain name.

Request Structure

JSON string of the following structure:

computer	string	mandatory	Computer name
domainName	string	optional	For windows computers, Windows domain name

Response Structure

JSON string of the following structure:

computers	object array			
	latestHostReportingDt	datetime		Latest date and time the computer reported to Gytpol
	latestScanDt	datetime		Gytpol client scans computers for misconfigurations and sends the report to Gytpol backend. This is the date and time of the latest misconfiguration scan reported for this computer.
	computer	string		Computer name
	computerOu	string		Name of organizational unit define on this computer
	domainName	string		For windows computers, Windows domain name
	clientVer	string		Gytpol client version currently installed on this computer
	ipAddress	string		Computer's IP address
	os	string		Computer's operating system
	isVdi	bool		Is this computer a VDI
	isServer	bool		Is this computer a server

	isDC	bool		Is this computer a domain controller
	miscon	object array		Array of misconfigurations reported for this computer
		topicCode	string	
		user	string	Username logged into the computer when this misconfiguration had been found
		severity	string	Specifies minimal severity of returned misconfigurations. Supported values are: • Low • Medium High
		addInfo	string	Additional information describing this misconfiguration (this is json string) with \ before " in order to prevent breaking the structure of the response json
		param	string	Parameter providing more details for the misconfiguration
		paramExtra	string	Parameter providing even more details for the misconfiguration

		isRemediable	bool	Is this misconfiguration remediable
		isMuted	bool	Is this misconfiguration muted by one of the mute rules
		mutedByRuleId	number	The Id of the mute rule that muted this misconfiguration
		hostReportingDt	datetime	Datetime when computer reported this misconfiguration to Gytpol backend
		scanTimeDt	datetime	Datetime when Gytpol client installed on the computer found this misconfiguration

Request Example

curl --location --request POST "**{BASE-URL}**/gytpolapi/v2.0/get_miscon_by_computer" --header "**x-api-key: jyUbUQNuVjClzQ5f6sXgmcgGzyoFiaYXA+OvxObvLV8="** --data-raw "{ \"computer\": \"**YOUR-COMPUTER-NAME**\"}"

Powershell:

$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("x-api-key", "**{API KEY}**")

$body = "{`"computer`": `"**{PCName}**`"}"

$response = Invoke-RestMethod 'https://**{BASE-URL}**/gytpolapi/v2.0/get_miscon_by_computer' -Method 'POST' -Headers $headers -Body $body
$response | ConvertTo-Json

Note: Change **BASE-URL** to your base URL. Change **x-api-key** to your API key. For On-prem customers, please incorporate your port (i.e., 9191) into the BASE-URL as follows: ***BASE-URL:PORT***.

Response Example


```
{
  "computers": [
    {
      "latestHostReportingDt": "2023-01-17T18:25:05.5389826+02:00",
      "latestScanTimeDt": "2023-01-17T18:24:47.4503679+02:00",
      "computer": "YOUR-COMPUTER-NAME",
      "computerOu": "COMPUTER-OU",
      "domainName": "YOUR-DOMAIN-NAME",
      "clientVer": "2.26.1.0",
      "ipAddress": "10.67.137.160",
      "os": "Win 10 Enterprise",
      "isVdi": false,
      "isServer": false,
      "isDC": false,
      "miscon": [
        {
          "topicCode": "gytPSVerIssue",
          "user": null,
          "severity": "Medium",
          "addInfo": "{\"PSversions\":\"2; 5.1.19041.2364\",\"occurrences\":1}",
          "param": "2; 5.1.19041.2364",
          "paramExtra": null,
          "isRemediable": true,
          "isMuted": false,
          "mutedByRuleId": 0,
          "hostReportingDt": "2023-01-17T18:25:05.5389826+02:00",
          "scanTimeDt": "2023-01-17T18:24:47.4503679+02:00"
        },
        {
          "topicCode": "gytSmbAnonymous",
          "user": null,
          "severity": "Medium",
          "addInfo": "{\"Current Value\":0,\"Expected Value\":\"1\",\"Registry
Name\":\"RestrictAnonymous\",\"Registry
Path\":\"HKLM:\\\\System\\\\CurrentControlSet\\\\Control\\\\Lsa\",\"Shares with
Everyone\":\"N/A\",\"Shares without Everyone\":\"N/A\",\"occurrences\":1}",
          "param": "No shares; The configured value is not secure",
          "paramExtra": null,
          "isRemediable": true,
          "isMuted": false,
          "mutedByRuleId": 0,
          "hostReportingDt": "2023-01-17T18:25:05.5389826+02:00",
          "scanTimeDt": "2023-01-17T18:24:47.4503679+02:00"
        }
        ]
        }]
}
```

HTTP Return Codes

200 Ok
400 Bad Request
401 Unauthorized
429 Too Many Requests

get_misconfigurations_start

Use this function to initiate a series of calls to get misconfigurations by computer.

Request Structure

Request body should contain json object of the following structure:

computer	string	optional	Computer name
domainName	string	optional	For windows computers, Windows domain name
selTarget	string	optional	Narrows down returned computers to a specified type. One of following possible values can be passed: • All Windows Computers • Windows Endpoints • Windows Servers • Windows Non-VDI • Endpoints Windows VDI • Endpoints Windows DC Servers • Windows Non-DC Servers • Debian Linux Computers • Red Hat Linux Computers • SUSE Linux Computers • Linux Unknown Computers • MAC Computers

severity	string	optional	Specifies minimal severity of returned misconfigurations. Supported values are: • Low • Medium • High
computerOu	string	optional	Return only computers that belong to the provided Organization Unit
topicCodes	string array	optional	Restrict returned misconfigurations only to specify misconfiguration types. The parameter has a form of: ["gytHostsFile", "gytSmbAnonymous", "gytTLSSSLClient"]
fromHostReportingDt	string	optional	Return only computers with latest misconfigurations reported after a given date and time. Example: "2023-01-10T19:43:46+02:00" or just "2023-01-01"
toHostReportingDt	string	optional	Return only computers with latest misconfigurations reported before a given date and time. Example: "2023-01-10T19:43:46+02:00" or just "2023-01-01"
returnMutedTopics	bool	optional	Gytpol product allows to mute misconfigurations meaning that they will not visually appear in the application. Default value is false meaning that muted misconfiguration will not be returned.

Json in the request body looks like in the following example:

```
{
  "computer": "<string> Optional",
  "domainName": "<string> Optional",
  "selTarget": "<string> Optional",
  "severity": "<string> Optional",
  "computerOu": "<string> Optional",
```

```
  "topicCodes": ["<string> topic code", "<string> topic code"],
  "fromHostReportingDt": "<string> that contains datetime Optional",
  "toHostReportingDt": "<string> that contains datetime Optional",
  "returnMutedTopics": "boolean default is false Optional"
}
```

Response Structure

token	string		Encrypted string to use for following calls to the **get_misconfigurations_next** function

Request Example

curl --location --request POST "**{BASE-URL}**/gytpolapi/v2.0/get_misconfigurations_start" --header "**x-api-key: jyUbUQNuVjCIzQ5f6sXgmcgGzyoFiaYXA+OvxObvLV8=**" --data-raw "{ \"computer\": null, \"domainName\": null, \"selTarget\": null, \"severity\": \"Medium\", \"computerOu\": null, \"topicCodes\": null, \"fromHostReportingDt\": \"2023-01-10T19:43:46+02:00\", \"toHostReportingDt\": null, \"returnMutedTopics\": false}"

Powershell:

```
$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("x-api-key", "{API KEY}")
$headers.Add("Content-Type", "application/json")

$body = "{`"severity`":`"High`"}"

$response = Invoke-RestMethod 'https://{BASE-URL}/gytpolapi/v2.0/get_misconfigurations_start' -Method 'POST' -Headers $headers -Body $body
$response | ConvertTo-Json
```

Note: Change **BASE-URL** to your base URL. Change **x-api-key** to your API key.
For On-prem customers, please incorporate your port (i.e., 9191) into the BASE-URL as follows: ***BASE-URL:PORT***.

Response Example

{"token":"7h5vmgiKQgvFiTb3xhrSyum52cbfh77xexcus8kGtOP03mliJxbJL99q8wfC2d8kwpNGXa0QF1VuycY6xnosSJePUkaGGUgCQ61rBmVcJI1J6RkUZMWmmGGD3R/+e9b2SrRlamRNusqUBOCphAeyDpBGb7uliNLpfn7wB2JiDGDJRu73lm6UIt3V7ITZDehfsb+JkWXVLIKNIv9+RvxrBCxVa/7StHvyW10cpGF67P9HfLZFbQOCjFsFOs8Mn6amZJrh1bkpasAblUWI0toXZVrlLHr6lfEYZMRnTadcBNTNlUBBWr6ptLUvdcWqEukmdaBublWlQBpAl++Seqc9rMF2WEex9o2n+5NyQBp8+OnuvsUcUybW/MfjG6J/06d07Tf/ks9mQJgZO2vnuJQAPA=="}

HTTP Return Codes

200 Ok
400 Bad Request
401 Unauthorized
429 Too Many Requests

get_misconfigurations_next

Use this function to initiate a series of calls to get misconfigurations by computer

Request Structure

token	string	Mandatory	Encrypted string to use for following calls to the **get_misconfigurations_next** function

Json in the request body looks like in the following example:

```
{
  "token": "<string> Mandatory
}
```

Response Structure

computers	object array		Same structure as presented in get_miscon_by_computer Response Structure

			Keep calling to get_misconfigurations_next each time with the new token until returned computers array is empty, e.g. []
token	string		Encrypted string to use for following calls to the **get_misconfigurations_next** function

Request Example

curl --location --request POST "**{BASE-URL}**/gytpolapi/v2.0/get_misconfigurations_next" --header "**x-api-key: jyUbUQNuVjClzQ5f6sXgmcgGzyoFiaYXA+OvxObvLV8=**" --data-raw "{
\"token\":
\"7h5vmgiKQgvFiTb3xhrSyum52cbfh77xexcus8kGtOP03mliJxbJL99q8wfC2d8kwpN
GXa0QF1VuycY6xnosSJePUkaGGUgCQ61rBmVcJI1J6RkUZMWmmGGD3R/+e9b2SrRl
amRNusqUBOCphAeyDpBGb7uliNLpfn7wB2JiDGDJRu73Im6Ult3V7ITZDehfsb+JkWX
VLlKNlv9+RvxrBCxVa/7StHvyW10cpGF67P9HfLZFbQOCjFsFOs8Mn6amZJrh1bkpasAbl
UWI0toXZVrlLHr6lfEYZMRnTadcBNTNlUBBWr6ptLUvdcWqEukmdaBublWlQBpAI++Seq
c9rMF2WEex9o2n+5NyQBp8+OnuvsUcUybW/MfjG6J/06d07Tf/ks9mQJgZO2vnuJQ
APA==\"}"

Powershell:

```
$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("x-api-key", "{API KEY}")
$headers.Add("Content-Type", "application/json")

$body = "{`"Token`":{TOKEN}}"

$response = Invoke-RestMethod 'https://{BASE-URL}/gytpolapi/v2.0/get_misconfigurations_next' -Method 'POST' -Headers
$headers -Body $body
$response | ConvertTo-Json
```

Note: Change **BASE-URL** to your base URL. Change **x-api-key** to your API key. For On-prem customers, please incorporate your port (i.e., 9191) into the BASE-URL as follows: ***BASE-URL:PORT***.

Response Example

```
{
  "computers": "same structure as show in Response Example",
  "token" : "new encrypted string for the following call to get_misconfigurations_next"
}
```

HTTP Return Codes

200 Ok
400 Bad Request
401 Unauthorized
429 Too Many Requests

get_miscon_computers_start

Use this function to initiate a series of calls to get the list of computers that have misconfigurations. Computers will be returned in alphabetical order:

Request Structure

Request body should contain json object of the following structure:

computer	string	optional	Computer name
domainName	string	optional	For windows computers, Windows domain name
selTarget	string	optional	Narrows down returned computers to a specified type. One of following possible values can be passed: • All Windows Computers • Windows Endpoints • Windows Servers • Windows Non-VDI • Endpoints Windows VDI

			• Endpoints Windows DC Servers • Windows Non-DC Servers • Debian Linux Computers • Red Hat Linux Computers • SUSE Linux Computers • Linux Unknown Computers • MAC Computers
computerOu	string	optional	Return only computers that belong to the provided Organization Unit

Json in the request body looks like in the following example:

```
{
  "computer": "<string> Optional",
  "domainName": "<string> Optional",
  "selTarget": "<string> Optional",
  "computerOu": "<string> Optional
}
```

Response Structure

token	string		Encrypted string to use for following calls to the **get_misconfigurations_next** function

Request Example

curl --location --request POST "**{BASE-URL}**/gytpolapi/v2.0/get_miscon_computers_start" --header "**x-api-key: jyUbUQNuVjClzQ5f6sXgmcgGzyoFiaYXA+OvxObvLV8=**" --data-raw "{ \"computer\": null, \"domainName\": null, \"selTarget\": \"Windows Servers\"}"

Powershell:

```
$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("x-api-key", "{API KEY}")
$headers.Add("Content-Type", "application/json")

$body = "{domainName: {DomainName}}"
```

```
$response = Invoke-RestMethod 'https://{BASE-URL}/gytpolapi/v2.0/get_miscon_computers_start' -Method 'POST' -Headers $headers -Body $body
$response | ConvertTo-Json
```

Note: Change **BASE-URL** to your base URL. Change **x-api-key** to your API key.
For On-prem customers, please incorporate your port (i.e., 9191) into the BASE-URL as follows: ***BASE-URL:PORT***.

Response Example

{"token":"7h5vmgiKQgvFiTb3xhrSyum52cbfh77xexcus8kGtOP03mliJxbJL99q8wfC2d8kwpNGXa0QF1VuycY6xnosSJePUkaGGUgCQ61rBmVcJI1J6RkUZMWmmGGD3R/+e9b2SrRlamRNusqUBOCphAeyDpBGb7uliNLpfn7wB2JiDGDJRu73lm6UIt3V7ITZDehfsb+JkWXVLlKNIv9+RvxrBCxVa/7StHvyW10cpGF67P9HfLZFbQOCjFsFOs8Mn6amZJrh1bkpasAblUWI0toXZVrlLHr6lfEYZMRnTadcBNTNlUBBWr6ptLUvdcWqEukmdaBublWIQBpAI++Seqc9rMF2WEex9o2n+5NyQBp8+OnuvsUcUybW/MfjG6J/06d07Tf/ks9mQJgZO2vnuJQAPA=="}

HTTP Return Codes

200 Ok
400 Bad Request
401 Unauthorized
429 Too Many Requests

get_miscon_computers_next

Use this function to continue getting results for the list of computers:

Request Structure

token	string	Mandatory	Encrypted string to use for following calls to the **get_misconfigurations_next** function

Json in the request body looks like in the following example:

```
{
  "token": "<string> Mandatory
}
```

Response Structure

computers	object array		Keep calling to get_miscon_computers_next until empty computers array is returned, e.g. computers[]
	latestHostReportingDt	datetime	Latest date and time the computer reported to Gytpol
	latestScanDt	datetime	Gytpol client scans computers for misconfigurations and sends the report to Gytpol backend. This is the date and time of the latest misconfiguration scan reported for this computer.
	computer	string	Computer name
	computerOu	string	Name of organizational unit define on this computer
	domainName	string	For windows computers, Windows domain name
	clientVer	string	Gytpol client version currently installed on this computer
	ipAddress	string	Computer's IP address
	os	string	Computer's operating system
	isVdi	bool	Is this computer a VDI
	isServer	bool	Is this computer a server
	isDC	bool	Is this computer a domain controller
token	string		Encrypted string to use for following calls to the **get_misconfigurations_next** function

Request Example

curl --location --request POST "**{BASE-URL}**/gytpolapi/v2.0/get_miscon_computers_next" --header "**x-api-key: jyUbUQNuVjClzQ5f6sXgmcgGzyoFiaYXA+OvxObvLV8=**" --data-raw "{
\"token\":
\"7h5vmgiKQgvFiTb3xhrSyum52cbfh77xexcus8kGtOP03mliJxbJL99q8wfC2d8kwpN
GXa0QF1VuycY6xnosSJePUkaGGUgCQ61rBmVcJI1J6RkUZMWmmGGD3R/+e9b2SrRl
amRNusqUBOCphAeyDpBGb7uliNLpfn7wB2JiDGDJRu73Im6Ult3V7ITZDehfsb+JkWX
VLlKNlv9+RvxrBCxVa/7StHvyW10cpGF67P9HfLZFbQOCjFsFOs8Mn6amZJrh1bkpasAbl
UWI0toXZVrlLHr6lfEYZMRnTadcBNTNlUBBWr6ptLUvdcWqEukmdaBublWlQBpAI++Seq
c9rMF2WEex9o2n+5NyQBp8+OnuvsUcUybW/MfjG6J/06d07Tf/ks9mQJgZO2vnuJQ
APA==\"}"

Powershell:

```
$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("x-api-key", "{API KEY}")
$headers.Add("Content-Type", "application/json")

$body = "{token: {TOKEN}}"

$response = Invoke-RestMethod 'https://{BASE-URL}/gytpolapi/v2.0/get_miscon_computers_start' -Method 'POST' -Headers $headers -Body $body
$response | ConvertTo-Json
```

Note: Change **BASE-URL** to your base URL. Change **x-api-key** to your API key.
For On-prem customers, please incorporate your port (i.e., 9191) into the BASE-URL as follows: ***BASE-URL:PORT***.

HTTP Return Codes

200 Ok
400 Bad Request
401 Unauthorized
429 Too Many Requests

add_to_group

Use this function to add a computer to a computer group.

Request Structure

groupName	string	Mandatory	Group name
computerName	string	Mandatory	Computer name, case-insensitive

Json in the request body looks like in the following example:

```
{
  "groupName": "Exchange Servers"
  "computerName": "WIN-A951IBBEJC2"
}
```

Response Structure

error	string		Error string, returned only on error

Request Example

curl --location --request POST "**{BASE-URL}**/gytpolapi/v2.0/add_to_group" --header "**x-api-key: jyUbUQNuVjClzQ5f6sXgmcgGzyoFiaYXA+OvxObvLV8=**" --data-raw "{\"groupName\":\"Exchange Servers\",\"computerName\":\"WIN-A951IBBEJC2\"}"

Powershell:

$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("x-api-key", "**{API KEY}**")
$headers.Add("Content-Type", "application/json")

$body = "{'groupName': '**{GROUP}**', computerName: '**{COMPUTER}**'}"

$response = Invoke-RestMethod 'https://**{BASE-URL}**/gytpolapi/v2.0/add_to_group -Method 'POST' -Headers $headers -Body $body
$response | ConvertTo-Json

Note: Change **BASE-URL** to your base URL. Change **x-api-key** to your API key.
For On-prem customers, please incorporate your port (i.e., 9191) into the BASE-URL as follows: ***BASE-URL:PORT***.

HTTP Return Codes

200 Ok
400 Bad Request
401 Unauthorized
429 Too Many Requests