FMGTool

Release notes

p2.2

Table of contents

Changelog

Current version of the tool can be found by running it with parameter “-version”.

p2.2

(Released on May 10th, 2023)

- Added command “workaroundSDWAN70”.

p2.1

(Released on October 7th, 2022)

- Added command “listDevices”.

p2.0

(Released on August 4th, 2022)

- Public version created.

Distribution

FMGTool is distributed as binary application for all 3 main operating systems (Window, Linux, MacOS).

It has no dependencies and is supposed to be run without any installation.

This application is available to Fortinet customers and partners.

Supported commands

FMGTool is application that connects to FortiManager via JSON API and performs actions as specified on command line. Following actions are available in this version:

getConfigs: Download config files from managed FortiGates

Config files are then stored locally on the computer where FMGTool was executed. Additional file with the same name as each config file but ".xml" suffix is created and contains additional information about the device retrieved directly from FortiManager.

listDevices: List devices managed by FortiManager

By default, managed devices including platform, serial number, management IP, firmware version, etc. are shown on standard output. With additional parameter it is possible to generate Excel file.

workaroundSDWAN70: Convert Central Management SD-WAN profiles between v6 and v7

This is designed to help with migrating SDWAN central managed profiles from FortiManager v6 to v7.

Security

Application can be run on any computer from which FortiManager HTTPs management interface is reachable. No special privileges are needed on the computer itself.

Computer doesn't need any other network access – FMGTool doesn't need - and it doesn't attempt to - connect anywhere else then FortiManager HTTPs API.

Downloaded config files are the same files that can be downloaded manually by logging in to FortiGate web interface and taking full config backup. That means that the config file contains encrypted passwords, encrypted SSL key files, etc.

This tool uses FortiManager's proxy API to request the backup to be taken directly on target FortiGate. That is why the tunnel between FortiManager and FortiGate must be up.

Prerequisites

- Administrator with JSON API read (or read-write) privileges must exist on FortiManager.
- Tunnels between FortiGates and FortiManager are up.
- User running this application has write access to local filesystem.

Usage

This tool is executed from command line.

Common use cases

```
> fmgtool.exe -version
```

Print current version and exit.

```
> fmgtool.exe -host 10.0.0.21 -user api -read-password getConfigs
```

Get config files from all FortiGates reachable from FortiManager running at 10.0.0.21. User with API access is “api” and password is requested on terminal.

```
> fmgtool.exe -host 10.0.0.21 -user api -password api getConfigs
```

Same as above, but this time the password “api” for user “api” is directly specified on command line.

```
> fmgtool.exe -host 10.0.0.21 -user api -password api -adom root getConfigs
```

Collect config files only for devices in ADOM “root”.

```
> fmgtool.exe -host 10.0.0.21 -user api -password api getConfigs
```

Same as above, but this time the password “api” for user “api” is directly specified on command line.

```
> fmgtool.exe -host 10.0.0.21 -user api -password api getConfigs -sn ^FG140E
```

Consider only FortiGates with serial number starting “FG140E”.

```
> fmgtool.exe -host 10.0.0.21 -user api -password api getConfigs -directory out
```

Save all config and XML files into sub-directory “out”.

```
> fmgtool.exe -host 10.0.0.21 -user api -password api listDevices -excel
out.xlsx
```

Create Excel file “out.xlsx” containing list of all devices managed by FortiManager.

Global parameters

As can be noticed from the last example above, parameters are split into several groups.

First group of parameters specifies how to reach FortiManager and possible which ADOM to operate on after successfully connected. The main recognized parameters are:

`-host`	Specifies the IP or hostname of FortiManager (mandatory)
`-user`	Name of administrator with JSON API enabled (default “admin”)
`-password`	Used to specify password directly on command line (this can be dangerous on shared machines)
`-read-password`	Alternative to `-password`. Application will ask for password on terminal without printing it
`-port`	HTTPs API interface port. Must be specified is FortiManager is listening on custom port (or some kind of DNAT is used)
`-adom`	If specified, further commands are limited to that ADOM only

Then the command name and command specific parameters follow.

Commands details

getConfigs

This command downloads config files of all FortiGates currently reachable from FortiManager.

Warning:
Since FortiManager does not store the full config file of FortiGate in generally known text format, this command needs to request FortiManager to download it from running FortiGate. This is the reason why FortiGates need to be reachable by FortiManager at the time when this command is executing.

Devices known to FortiManager as unreachable are skipped with no delay. However, if there are some devices that have reachability problems FortiManager is not (yet) aware of, about 1 minute timeout is added for each round (see `-size` parameter).

Suggestion:
For huge deployments it is suggested to use global parameter `-loglevel warning` that hides terminal output that does not report any problem.

By default, config is retrieved from all (reachable) FortiGates, but regular expressions can be used on serial numbers and/or device names to limit the scope.

By default, all ADOMs are used when looking for FortiGates, but global parameter `-adom` can be used to limit the scope to single ADOM.

Downloaded configs

All config files are stored in the current directory unless this is changed with command line parameter `-directory` .

Together with config files with suffix ".conf" there are also files with suffix ".xml" and the same name created in the same directory. The XML file contains meta-information for the device known by FortiManager.

Parallelism

By default, FortiManager is instructed to download configs from 10 devices simultaneously. This operation is quite fast, so it is not a problem even with many FortiGates managed by FortiManager.

However, for FortiManagers with hundreds or even thousands of managed FortiGates, it might make sense to increase the parameter `-size` to make things faster. In that case, please monitor the FortiManager resources to make sure it is not using too much CPU/RAM.

Command specific parameters

-device	Regular expression applied on device names as known to FortiManager - only matching devices will be considered
-sn	Regular expression applied on device serial number - only matching devices will be considered
-directory	Directory where to save downloaded config files; Directory must already exist (current directory by default)

listDevices

This command retrieves information about all FortiGates managed by FortiManager.

By default, it will only print them on console, but it can also create nicely formatted Excel file.

If no global `-adom` parameter is specified, this command will collect FortiGates across all ADOMs present on FortiManager.

Following details are shown for each device:

- Device name
- Platform
- SN
- IP (management IP)
- ADOM
- Version (human readable, like 6.4.5)
- Build
- HA Name
- HA Role (standalone/master/slave)
- Connection (up/down)
- VDOMs (disabled/enabled + count)

In Excel output, following additional columns are shown:

- Authorized (yes/no, unauthorized device can be missing some information)
- VDOM names (only those assigned to “current ADOM”)

Unauthorized devices

Unauthorized device usually (but not always) has FortiOS version reduced to major.minor (i.e. missing the patch version).

FortiGate with VDOMs in different ADOMs

On FortiManager it is possible to assign every FortiGate VDOM to different FortiManager ADOM.

To match this logic, command listDevices will show the device as many times as number of ADOMs it is (partially) member of. Columns showing VDOM counts and names are only relevant for that ADOM.

That means that if the device has two VDOMs, each in different ADOM, it will be shown twice, and each time with only one VDOM.

Command specific parameters

`-excel`	Name of the Excel file to create

`-only-master`	For every cluster, including only master device in the list

workaroundSDWAN70

When upgrading FortiManager from 6.4 to 7.0 with "SD-WAN Central Management" enabled, the auto-updated SD-WAN Template configuration is not correct in many cases. This is a general problem when upgrading from v6 to v7.

This is because in FortiManager v7, now there can be only physical interface names used in SD-WAN template, while in 6.4, Normalized interface names were used and then there were several levels of mapping applied before installing the configuration to physical device. For more details about this change, see our official documentation.

Unfortunately, due to these changes in logic, FortiManager is unable to correctly update the configuration during the upgrade process. That results in "SD-WAN Provisioning Templates" (successor of former "SD-WAN Central Management Templates") that do not reflect the desired state after the upgrade and must be manually fixed (sometimes completely re-created) by customers.

Workaround

Before upgrading the device (it must be still running the latest 6.4) admin needs to use this tool to export old SDWAN central management profiles to local file.

After upgrading, admin uses this tool to update SDWAN profiles based on data in the file created previously. This will overwrite the relevant profiles on FortiManager!

Meta fields

To simulate central SDWAN management configuration on 7.0, this tool creates meta fields for interface, source IP and destination IP in SD-WAN Provisioning Templates. This is necessary to allow reusing the same profile for multiple devices.

However, on bigger configuration this can easily fail on maximal number of meta fields that FortiManager supports (255). To deal with this, there is `-no-default-meta` option that avoids creating meta fields for source and destination addresses if those would have no benefit – for example when only single device is assigned to profile, or all assigned devices use the same value.

Warning

Be aware that it is not technically possible to fully convert v6 configuration to v7. This is because some configurations have no exact equivalent.

Command specific parameters

`-file`	Name of the Excel file to create
`-collect`	For every cluster, including only master device in the list
`-restore`	Update 7.0 FortiManager with configurations from file
`-no-default-meta`	Avoids creating meta-fields with no benefit (see above)

-no-ipv6	Avoids creating meta-fields for IPv6 addresses