FortiGate / FortiManager - Communications Protocol Guide

Version 6.2

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO GUIDE
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/support-and-training/training.html

NSE INSTITUTE
https://training.fortinet.com

FORTIGUARD CENTER
https://fortiguard.com/

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

June 07, 2019
FortiGate / FortiManager 6.2 Communications Protocol Guide
01-620-282493-20190607

TABLE OF CONTENTS

Change Log 4
Overview 5
Exceptions 5
Protocol components 6
FGFM header 6
Keep-Alive messages 6
Customizing the Keep-Alive settings 6
FortiManager passive mode 7
FortiGate to FortiManager authentication 8
Tunneling 9
Tunnel setup details 9
Communication hardening 10
Protocol operation on FortiGate 11
Security concerns 11
Protocol operation on FortiManager 12
Topology scenarios 13
Scenario 1: FortiGate has public IP address, FortiManager is behind NAT 13
Scenario 2: FortiManager on a routable public IP address / FortiGate behind NAT 14
Scenario 3: Both FortiManager and FortiGate have public IP addresses 14
Scenario 4: Mixed topology 15
Scenario 5: Both devices behind NAT 15
FGFM built-in recovery 16
Making changes to the FortiGate management IP address 16
FGFM recovery logic 16
Example 17

Change Log

Date	Change Description
2019-06-07	Initial release.

Overview

The fgfm protocol implements a secure communication protocol with the following functions:

1. FortiGate reachability status (from FortiManager)
2. FortiManager reachability status (from FortiGate)
3. Configuration installation and retrieval
4. Script push
5. JSON monitoring via RTM

Exceptions

The following communications between FortiGate and FortiManager units are handled outside of the fgfm protocol and are managed by the FortiGuard protocol:

1. FortiGuard package downloads (AV, IPS, Virus Scan, etc.)
2. FortiGuard query (WF, AS)
3. Firmware Downloads

Protocol components

The fgfm protocol runs over SSL (Secure Sockets Layer) using TCP port 541 under IPv4. FortiManager 6.2 supports the use of IPv6.

Both FortiGate and FortiManager units have a fgfm daemon running exclusively for FortiGate to FortiManager communication. The FortiManager unit listens on TCP port 541 for an incoming session request. The FortiGate unit establishes an SSL session with the FortiManager. Both units use TCP port 541 for sending and receiving messages.

The fgfm daemon handles all FortiGate to FortiManager (and vice versa) authentication, keep-alive messages and actions resulting from them (such as instructing another daemon on a FortiGate device to update its configuration or various database files).

FGFM header

The fgfm header is a simple header that indicates the purpose of the message, such as keep-alive, authentication and packet forwarding.

Keep-Alive messages

The FortiGate unit sends keep-alive messages to the FortiManager every 120 seconds or 2 minutes. If the FortiManager unit does not receive 3 consecutive messages (360 seconds or 6 minutes), it considers that specific FortiGate unit to be unreachable, disabled or otherwise offline.

When that unit comes back online, it must re-establish an SSL connection with the FortiManager before management functions can continue. It will attempt to do so using the last-known IP address and serial number of the FortiManager device. The FortiManager will do the same.

The keep-alive message contains information that assists the FortiManager in managing the FortiGate unit, such as current OS version, platform, configuration checksum and versions of the unit's AV and IPS databases.

Customizing the Keep-Alive settings

You can customize how quickly the FortiManager can detect an issue or failure of a managed FortiGate device.

Keep in mind that shortening the interval between the units will let the FortiManager find a device failure more quickly, it can also generate a substantial amount of processing overhead to the FortiManager system when the unit is managing many devices.

To change the keep-alive interval (default values are listed):

```
get system dm
      .......
      fgfm-sock-timeout: 360
```

```
    fgfm_keepalive_itvl: 120
    .......
end
```

Please note the difference between the two above commands (dash versus underscore).

FortiManager passive mode

After a FortiGate unit receives a keep-alive message from a FortiManager unit containing the unit's OS version, AV and IPS database versions and configuration information, it compares that information with its local versions. If the FortiManager unit has a newer version of any of the above and the FortiGate is configured to receive automatic updates, the fgfm daemon running on the FortiGate will then notify the FortiGuard daemon running. The FortiGuard daemon will then issue an update request to the FortiManager unit. The information sent by the FortiManager is not sent via the SSL connection on TCP port 541; FortiManager uses UDP port 9443 to send this information.

The keep-alive message is the de facto 'push' action for delivering update notifications. The FortiManager unit will never send updates to the FortiGate unit; the FortiGate unit instead downloads updates from the FortiManager.

FortiGate to FortiManager authentication

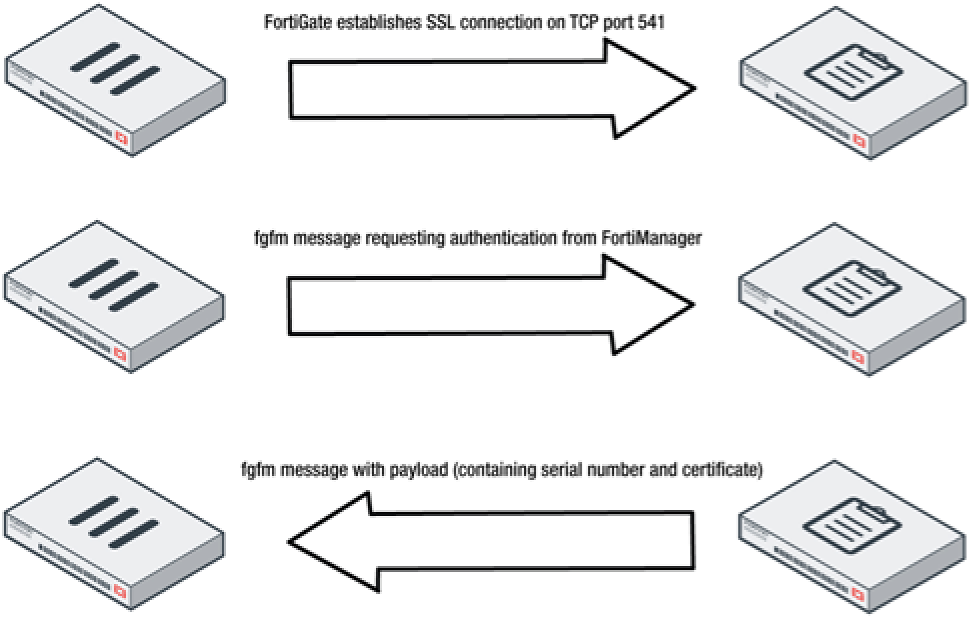
Once a FortiManager has accepted an incoming connection from a FortiGate unit, the FortiManager will send its certificate to the FortiGate unit via SSL.

The FortiGate unit then authenticates the certificate as follows:

1. It compares the serial number provided in the authentication message against the one in the certificate;
2. It verifies that the serial number is in its local allowed serial number table;
3. It then verifies the validity of the certificate.

If any of the above steps fail, authentication fails and the FortiGate unit will not accept management from the FortiManager. This authentication process occurs at the start of any new SSL session.

Communication sequence for authentication


FortiGate establishes SSL connection on TCP port 541
fgfm message requesting authentication from FortiManager
fgfm message with payload (containing serial number and certificate)

Each FortiGate unit maintains a local trusted list containing up to ten unique entries. This contains a list of FortiManager serial numbers that the FortiGate is willing to yield management rights to. If a FortiManager unit with a serial number not on that FortiGate's trust list attempts to connect to the FortiGate, the unit will immediately terminate the connection and refuse to be managed. The local trusted list on the FortiGate is not configurable and the serial number of the managing FortiManager is added to the device's trusted list when added to the FortiManager's list of managed devices.

Tunneling

All other management traffic, which at this point will only be RTM traffic, is tunneled through the SSL connection with an fgfm header identifying the packet data as an IP packet to be extracted and passed to the device over a tunnel interface (see next section for more details).

Tunnel setup details

The following settings are sent from FortiManager to the FortiGate unit during the setup of the fgfm tunnel:

To enable the following viewing, you must log in to the FortiGate CLI with the administrative account and enter the following debug commands:

```
# diagnose debug enable
# diagnose debug application fgfmd 255
```

After entering the above commands, you will see the following log printed out on the FortiGate CLI during fgfm tunnel setup:

```
......
FGFMs: Set managment id 247331677 OK.
FGFMs: [__chg_by_fgfm_msg] set keepalive_interval: 300
FGFMs: [__chg_by_fgfm_msg] set channel buffer/window size to 32768 bytes
FGFMs: [__chg_by_fgfm_msg] set sock timeout: 900
FGFMs: [fgfm_msg_put_tuninfo] vdom=’root’, physical_intf=, intf=’wan1’
FGFMs: client:send:
get ip
first_fmgid=
probe_mode=yes
vdom=root
intf=wan1
FGFMs: client:
reply 200
overwrite_fmgid=1
request=ip
ip=169.254.0.2
mgmtid=247331677
register_status=1
fmg_ip=192.168.48.46
keepalive_interval=300
chan_window_sz=32768
sock_timeout=900
FGFMs: [__chg_by_fgfm_msg] set keepalive_interval: 300
FGFMs: [__chg_by_fgfm_msg] set channel buffer/window size to 32768 bytes
FGFMs: [__chg_by_fgfm_msg] set sock timeout: 900
......
```

Communication hardening

FortiManager allows you to customize the level of security and the encryption algorithms used to securely communicate with managed FortiGate devices.

FortiManager allows you to limit the cipher suites used by the device to prevent the possibility of a crypto downgrade attack such as that found in the Logjam vulnerability or other protocol downgrade attacks.

In the FortiManager CLI, you can change the supported cipher suites with the following command:

```
config system global
   set enc-algorithm {high | medium | low}
end
```

The default value is `high`.

The following cipher suites are used for each level:

- **LOW**: EDH-RSA-DES-CDBC-SHA, DES-CBC-SHA, DES-CBC-MD5
- **MEDIUM**: RC4-SHA, RC4-MD5, RC4-MD
- **HIGH**: ECDHE-RSA-AES256-GCM-SHA384 , DHE-RSA-AES256-GCM-SHA384 , ECDHE-RSA-AES128-GCM-SHA256

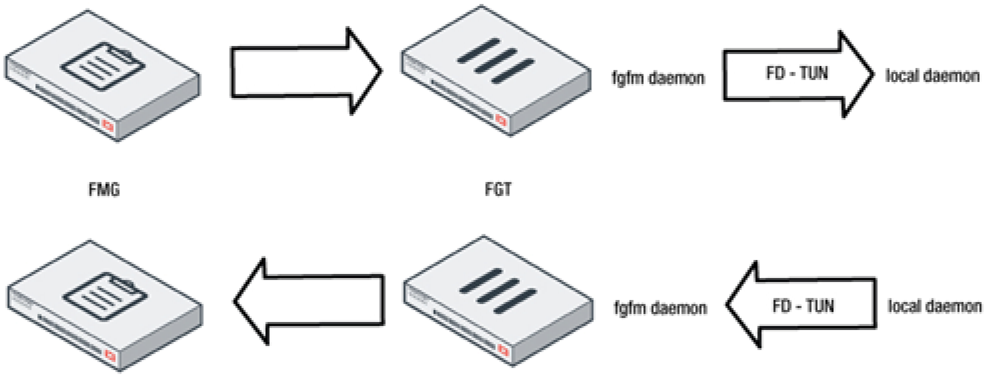
For the certificate used to encrypt communications, the FortiManager uses the BIOS certificate burned into the unit at time of manufacture in order to prevent tampering of the certificate.

Protocol operation on FortiGate

In order to isolate management traffic on the FortiGate unit and to prevent decrypted traffic from being routed to an external device, all management traffic is routed through an isolated virtual domain on the FortiGate unit. This contains only a single tunnel network device, TUN. The fgfm daemon owns a file descriptor, FD, which is linked directly to TUN. These steps are transparent to the user, and occur to segregate management traffic from all other traffic that may be passing through the FortiGate unit.

On the FortiGate unit, the fgfm daemon routes fgfm-encapsulated traffic from FortiManager through FD. These packets are received by their respective local daemons via the TUN device and respond via the TUN device. Therefore, their responses are always received by the fgfm daemon over FD and sent over the SSL connection to the FortiManager unit.

Protocol operation on FortiGate



Security concerns

Using the old IPSec tunneling method, packets can be routed anywhere by a FortiGate unit after decryption, which is a potential security issue. The fgfm protocol performs tunneling/detunneling exclusively in the fgfm daemon, sending packets to the FortiGate's TUN device and no other network devices. Packets are unable to leak out of the FortiGate and incoming data is dropped if it cannot be delivered to daemons via the local stack.

Protocol operation on FortiManager

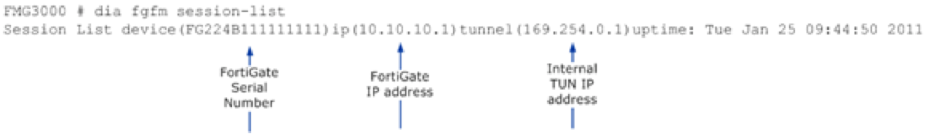
Much like how a TUN device is created on the FortiGate side of the connection, the FortiManager unit also creates a similar TUN device. This device is configured to have the same IP address as the physical interface through which the FortiManager communicates with the FortiGate unit.

The FortiManager will have a daemon receiving fgfm messages and upon reception of a message compares the FortiGate's declared IP address to the actual remote IP address to determine if the FortiGate unit is behind a NAT device.

A routing table is maintained for TUN such that traffic destined for any FortiGate behind NAT will be routed through the TUN based on each FortiGate's unique serial number and IP address. The the fgfm daemon running on the FortiManager assigns unique internal-use IP addresses to each FortiGate behind NAT so that it can distinguish between each unit and route traffic to the appropriate device via SSL.

Initially, the FortiGate's virtual TUN device has no IP address. If the FortiManager detects that the FortiGate is behind NAT, it allocates a unique internal IP address and notifies the FortiGate of this address.

Command line output of FortiManager tunneling



```
FMG3000 # dia fgfm session-list
Session List device(FG224B111111111)ip(10.10.10.1)tunnel(169.254.0.1)uptime: Tue Jan 25 09:44:50 2011
```

Regardless whether or not the FortiGate unit is behind NAT, the FortiManager always sends management traffic via the secure tunnel. The FortiGate unit assigns the address provided by the FortiManager to its TUN device so that traffic sent to the FortiManager appears to have come from that address.

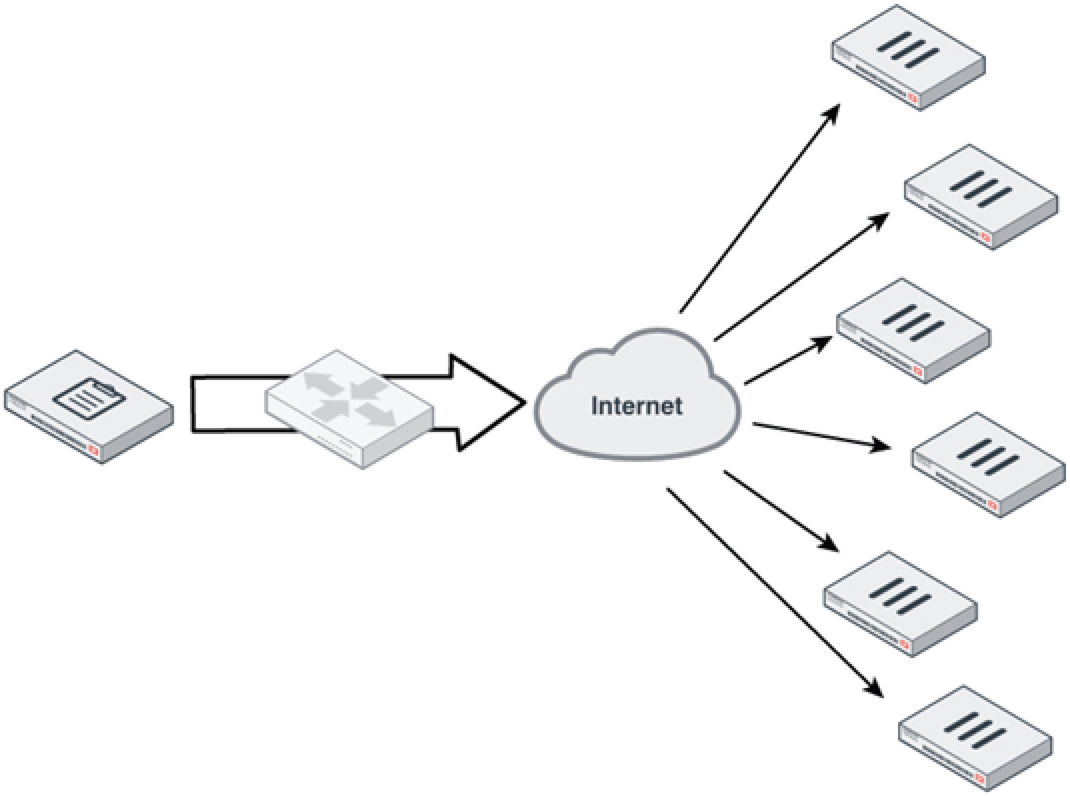
Conversely, the FortiManager sends any traffic destined for the FortiGate to the same address, routing it over the FortiManager's TUN device via SSL back to the FortiGate.

Topology scenarios

The fgfm protocol supports four basic scenarios:

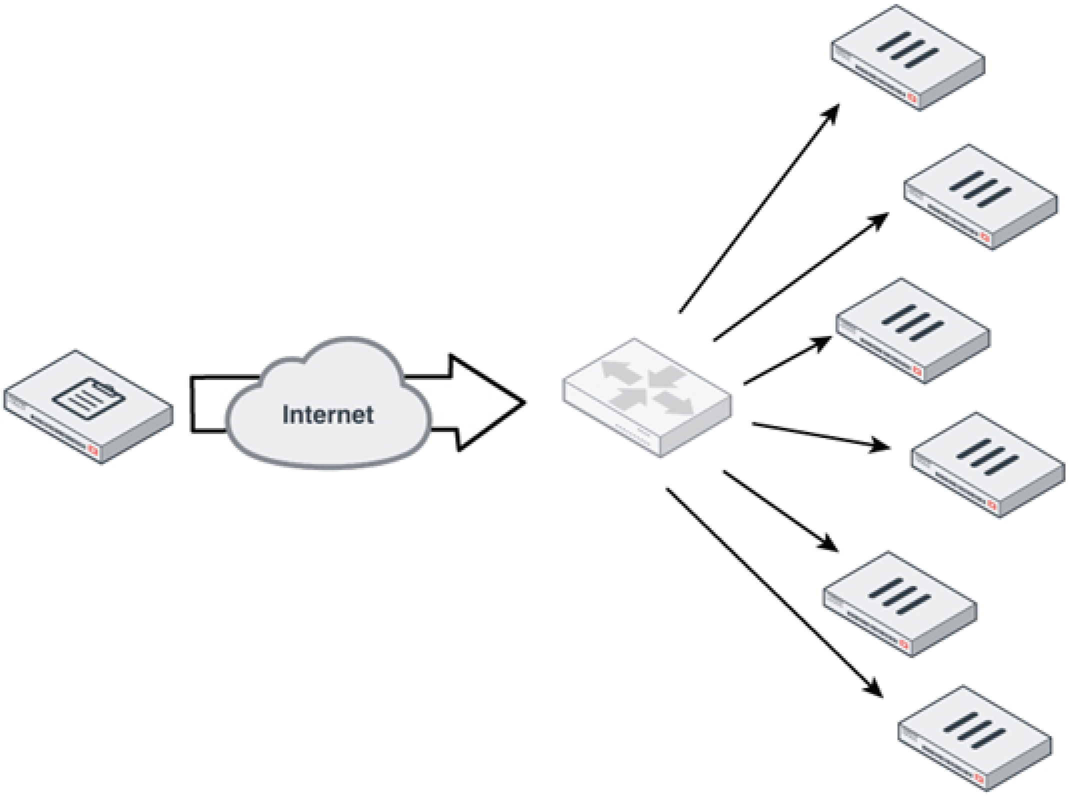
- FortiManager on a routable public IP address / FortiGate behind NAT
- FortiManager is behind NAT / FortiGate on routable public IP address
- Both units have routable public IP addresses
- Mixed topology

Scenario 1: FortiGate has public IP address, FortiManager is behind NAT



In this scenario, the FortiManager administrator must configure the FortiGate's IP address of hostname during the Add Device operation. Once complete, the FortiManager will initiate a connection to the FortiGate to perform authentication. Once authentication is successful, the FortiGate is immediately registered on the FortiManager and the unit can begin management functions on the FortiGate.

Scenario 2: FortiManager on a routable public IP address / FortiGate behind NAT

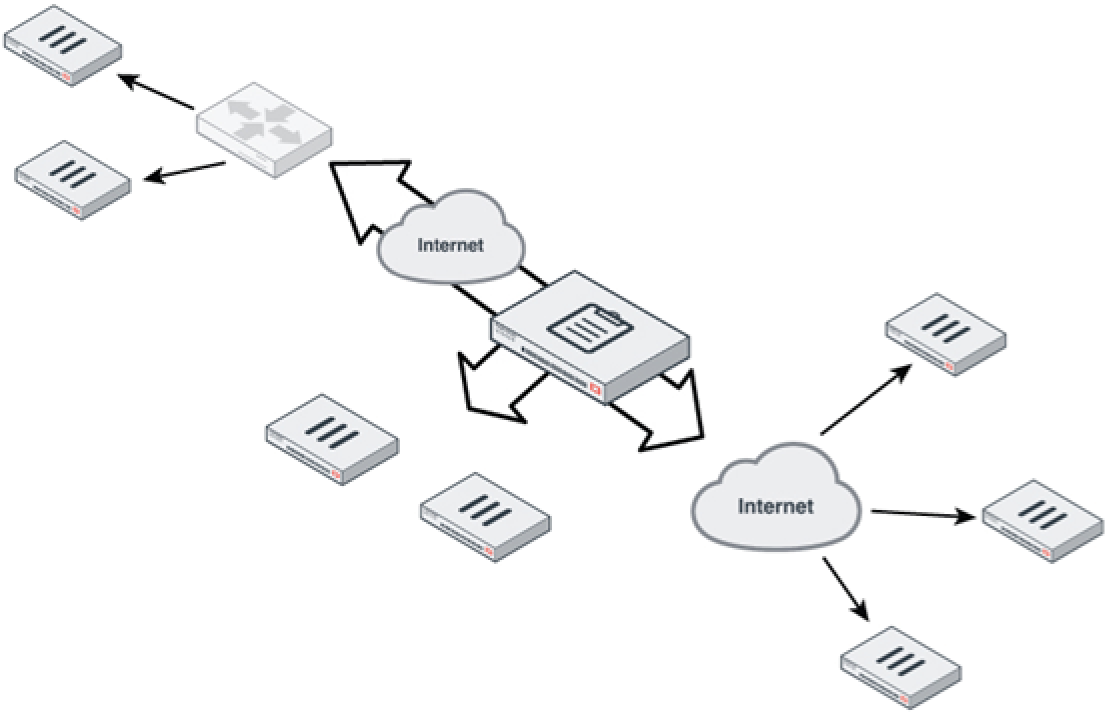


In this scenario, the FortiGate administrator must configure the IP address (or hostname) of the FortiManager on the FortiGate or via a virtual IP address mapped to the FortiGate unit. After this is configured, the FortiGate will automatically attempt to connect to the FortiManager because it is directly routable. Once the connection is active, authentication will be attempted and if successful, the FortiGate and FortiManager can begin communication. After this occurs, the FortiGate will show up on the FortiManager as an unregistered device. Once the FortiManager administrator registers the FortiGate, the FortiManager can begin performing management functions.

Scenario 3: Both FortiManager and FortiGate have public IP addresses

In this scenario, the FortiManager administrator can use either method indicated in scenarios 1 or 2 to complete device registration.

Scenario 4: Mixed topology



In this scenario, a FortiManager may be on a private network and is able to manage FortiGate units with publicly available IP addresses. It will also be able to manage FortiGate units located on its own private network. For FortiGate units residing behind NAT on another network, the FortiManager will be able to manage them once properly configured as described in the previous scenarios.

Scenario 5: Both devices behind NAT

In this case, the FortiManager and FortiGates are on different private networks. In order to configure the devices to allow management traffic to pass between them, a Virtual IP must be set up and configured on one side.

FGFM built-in recovery

When a connection between a managed FortiGate unit and a FortiManager is broken, the protocol has a built-in failsafe recovery.

Making changes to the FortiGate management IP address

Occasionally a FortiGate device will need to make a change to its IP address. If the IP address change allows the FortiGate to recreate its path to the FortiManager unit, functionality will remain unchanged. However, if that change breaks connectivity, after 15 minutes the FortiGate unit will revert to its last known good configuration in an attempt to restore connectivity.

The 15 minute timer is hardcoded and cannot be configured or disabled.

FGFM recovery logic

For each install:

- The FortiManager sends the following to the FortiGate:
  - a listing of the set commands needed to apply the configurations changes
  - a listing of the unset commands that would revert the configuration changes
- The FortiGate uses the following logic when applying changes:
  - apply the set commands, using memory only, nothing written to a configuration file
  - test the fgfm connection to the FortiManager
  - if the connection goes down, it applies the unset commands
  - retest the fgfm connection
  - if connection remains down, the FortiGate unit reboots to recover the previous configuration from its config file

The final step above is optional and can be enabled and disabled via the CLI using the following command:

```
config system dm
   set rollback-allow-reboot {enable |disable}
end
```

Example

The following scenario is an example of an installation configuration from a FortiManager 400E unit to a FortiGate 60E device that tries to change the FortiGate’s management IP (WAN1) and causes the fgfm connection to break down.

The original configuration on the FortiGate:

```
config system interface
   edit “wan1”
      set vdom “root”
      set ip 192.168.48.81 255.255.255.0
      set allowaccess ping https ssh snmp http telnet fgfm
      set type physical
   next
end
```

If you were to change the FortiGate WAN1 interface's IP address to 192.168.49.81/24 and then attempt to install the configuration change, the fgfm connection would break.

To enable the following viewing, you must log in to the FortiGate CLI with the administrative account and enter the following debug commands:

```
# diagnose debug enable
# diagnose debug cli 8
```

You will see the following log on the FortiGate CLI during install:

```
0: get sys status
0: get system mgmt-csum
0: config system interface
0: edit “wan1”
0: set ip 192.168.49.81 255.255.255.0
0: next
0: end
```

The configuration change will break the fgfm connection, causing the FortiGate unit to attempt to reconnect for 900 seconds. If the FortiGate cannot reconnect, it will rollback to its previous configuration.

You will see the following log when the FortiGate performs the rollback:

```
0: config system interface
0: edit “wan1”
0: set ip 192.168.48.81 255.255.255.0
0: next
0: end
0: config system interface
0: edit “modem”
-23: unset type
0: next
0: end
0: config system central-management
0: end
```