Fortigate debug and diagnose commands complete cheat sheet

Author: Yuri Slobodyanyuk, admin@yurisk.info

NOTE	To enable debug set by any of the commands below, you need to run **diagnose debug enable**. This is assumed and not reminded any further.
NOTE	To disable and stop immediately any debug, run **dia deb res** which is short for **diagnose debug reset**.
NOTE	All debug will run for 30 minutes by default, to increase use `diagnose debug duration <minutes>`, setting to 0 means unlimited by time. Reboot will reset this setting.

Security rulebase debug (diagnose debug flow)
CPU, and Memory
Session stateful table
High Availability Clustering debug
IPSEC VPN debug
SSL VPN debug
Static Routing Debug
Interfaces
NTP debug
SNMP daemon debug
BGP
Admin sessions
Authentication
Fortianalyzer logging debug
SD-WAN verification and debug
Virtual Fortigate License Status

Security rulebase debug (diagnose debug flow)

Table 1. Security rulebase diagnostics with `diagnose debug flow`

Command	Description
diagnose debug flow filter	Show the active filter for the flow debug
diagnose debug filter clear	Remove any filtering of the debug output set

Command	Description
diagnose debug flow filter <filtering param>	Set filter for security rulebase processing packets output. You can set multiple filters - act as AND, by issuing this command multiple times. Parameters: `vd` - id number of the vdom. When entering the vdom with `edit vdom`, this number is shown first. `vd-name` - limit debug to specific VDOM by its name. Fortigate translates the name to VDOM ID (`vd`). `proto` - Protocol number. `addr` - IP address of the packet(s), be it a destination or/and a source. `saddr` - IP source address of the packet(s). `daddr` - IP destination address of the packet(s). `port` - Source or/and destination port in the packet(s). `sport` - Source port of the packet(s). `dport` - Destination port of the packet(s). `negate <parameter>` - negate the match, i.e. match if a packet does NOT contain `<parameter`. Where `parameter` is one of the above: `vd`, `addr`, `saddr`, `port`, `sport`, `dport`
diagnose debug filter6 <parameter>	Same as `diagnose debug filter` but for IPv6 packets. The rest of matching and conditions remain of the same syntax.
diagnose debug flow show function-name enable	Show function names responsible for each step in processing.
diagnose debug flow trace start [number]	Actually start the debug with optional `number` to limit number of packets traced.

General Health, CPU, and Memory

Table 2. General Health, CPU, and Memory loads

Command	Description
get sys stat	Get statistics about the Fortigate device: FortiOS used, license status, Operation mode, VDOMs configured, last update dates for AntiVirus, IPS, Application Control databases.
get sys performance stat	Show real-time operational statistics: CPU load per CPU, memory usage, average network/session, uptime.
diagnose debug crashlog read	Display crash log. Records all daemons crashes and restarts. Some daemons are more critical than others.
diagnose debug crashlog clear	Clear the crash log.
get hardware memory	Show memory statistics: free, cached, swap, shared

Session stateful table

Command	Description
get system session status	Show current number of sessions passing the Fortigate. Run inside the VDOM in multi-vdom environment to get number of connections/sessions for this specific VDOM.

High Availability Clustering debug

Table 3. HA Clustering related debug and verification

Command	Description
get sys ha status	Show general status and statistics of the clustering - health status, cluster uptime, last cluster state change, reason for selecting the current master, configuration status of each member (`in-sync/out-of-sync`), usage stats (average CPU, memory, session number), status (`up/down`, `duplex/speed`, `packets received/dropped`) for the heartbeat interface(s), HA cluster index (used to enter the secondary member CLI with `exe ha manage`).
diagnose sys ha dump-by group	Print detailed info per cluster group, shows actual uptime of each member in `start_time`, as well monitored links failures, status.

Command	Description
diagnose sys ha checksum cluster	Shows configuration checksum for each cluster member separated in individual VDOMs and *global*. In properly synchronized cluster all member checksums should be identical, look at `all` value.
diagnose sys ha checksum recalculate	Force cluster member to recalculate checksums, often will solve the out of sync problem. No adverse effects. Run on each cluster member.
diagnose sys ha checksum show < ***VDOM*****/global>**	Print detailed synchronization status for each configuration part. Use after seeing `out-of-sync` in **diagnose sys ha checksum cluster** to know which part of configuration causes members to be out-of-sync. Need to run on each cluster member and compare, long output - use `diff` /`vimdiff/Notepad++ Compare plugin` to spot the differences.
diagnose sys ha checksum show < ***VDOM*****/global> <settings part name>**	Show exact setting inside the settings tree that causes out-of-sync. Use output from **diagnose sys ha checksum show** (see above) for *settings part name*. E.g. if `diagnose sys ha checksum show root` indicates that *firewall.vip* is out-of-sync, running `diagnose sys ha checksum show root firewall.vip` will give checksums of each VIP in the root domain to compare with those of secondary member.
diagnose debug app hatalk -1	Enable heartbeat communications debug. It shows in real time if members are talking over sync interfaces. The output will look like `state/chg_time/now=2(work)/1610773657/1617606630`, where the desired `state` is *work*, *chg_time* is last cluster state/failover date in epoch, and *now* is the last time communication occurred on heartbeat interface(s), also in epoch.
diag debug application hasync -1	Real time synchronization between members. As only things that changed get synchronized after 1st sync is established, may take time to produce output. See next.
execute ha synchronize stop **diag debug enable** **diag debug application hasync -1** **execute ha synchronize start**	Stop, enable debug, then start again HA synchronization process, will produce lots of output.

Command	Description
exe ha manage ? **exe ha manage <id>**	First show index of all Fortigate cluster members, then enter any secondary member CLI via its index.

IPSEC VPN debug

Table 4. IPSEC VPN Debug

Command	Description
diagnose vpn ike log-filter <parameter>	Filter VPN debug messages using various parameters: • `list` Display the current filter. • `clear` Delete the current filter. • `name` Phase1 name to filter by. • `src-addr4`/`src-addr6` IPv4/IPv6 source address range to filter by. • `dst-addr4`/`dst-addr6` IPv4/IPv6 destination address range to filter by. • `src-port` Source port range • `dst-port` Destination port range • `vd` Index of virtual domain. -1 matches all. • `interface` Interface that IKE connection is negotiated over. • `negate` Negate the specified filter parameter.
diagnose debug application ike -1	Enable IPSec VPN debug, shows phase 1 and phase 2 negotiations (for IKEv1) and everything for IKEv2. "-1" sets the verbosity level to maximum, any other number will show less output.
diagnose vpn ike gateway flush name <vpn_name>	Flush (delete) all SAs of the given VPN peer only. Identify the peer by its Phase 1 name.
diagnose vpn tunnel list [name <Phase1 name>]	Show operational parameters for all or just specific tunnels: Type (dynamic dial up or static), packets/bytes passed, NAT traversal state, Quick Mode selectors/Proxy Ids, mtu, algorithms used, whether NPU-offloaded or not, lifetime, DPD state.
diagnose vpn ike gateway list	Show each tunnel details, including user for XAuth dial-up connection.

Command	Description
get vpn ipsec tunnel details	Detailed info about the tunnels: Rx/Tx packets/bytes, IP addresses of the peers, algorithms used, detailed selectors info, lifetime, whether NAT Traversal is enabled or not.
get vpn ipsec stats tunnel	Short general statistics about tunnels: number, kind, number of selectors, state
get vpn ipsec tunnel summary	Short statistics per each tunnel: number of selectors up/down, number of packets Rx/Tx.
get vpn ipsec stats crypto	Crypto stats per component (ASIC/software) of the Fortigate: encryption algorithm, hashing algorithm. Useful to see if unwanted situation of software encryption/decryption occurs.

SSL VPN debug

Table 5. SSL VPN client to site/Remote Access debug

Command	Description
get vpn ssl monitor	List logged in SSL VPN users with allocated IP address, username, connection duration.
diagnose debug app sslvpn -1	Debug SSL VPN connection. Shows only SSL protocol negotiation and set up. That is - ciphers used, algorithms and such, does NOT show user names, groups, or any client related info.

Static Routing Debug

Table 6. Static and Policy Based Routing debug & diagnostics

Command	Description
get router info kernel	View the kernel routing table (FIB). This is the list of resolved routes actually being used by the FortiOS kernel. `tab` Table number, either 254 for unicast or 255 for multicast. `vf` Virtual domain index, if no VDOMs are enabled will be 0. `type` 0 - unspecific, 1 - unicast, 2 - local , 3 - broadcast, 4 - anycast , 5 - multicast, 6 - blackhole, 7 - unreachable , 8 - prohibited. `proto` Type of installation, i.e. where did it come from: 0 - unspecific, 2 - kernel, 11 zebOS module, 14 - FortiOS, 15 - HA, 16 - authentication based, 17 - HA1 `prio` priority of the route, lower is better. `pref` preferred next hop for this route. `Gwy` the address of the gateway this route will use `dev` outgoing interface index. If VDOMs enabled, VDOM will be included as well, if alias is set it will be shown.
get router info routing-table all	Show RIB - active routing table with installed and actively used routes. It will not show routes with worse priority, multiple routes to the same destination if unused.
get router info routing database	Show ALL routes, the Fortigate knows of - including not currently used.
get router info routing-table details <route>	Show verbose info about specific route, e.g. `get router info routing-table details 0.0.0.0/0`
get firewall proute	Get all configured Policy Based Routes on the Fortigate.

Interfaces

Table 7. Interafces of all kinds diagnostics

Command	Description
get hardware nic <inerface name>	Hardware info of the interface: MAC address, state (up/down), duplex (full, half), Rx/Tx packets, drops.
diagnose hardware deviceinfo nic <nic name>	Same as above.
get hardware npu np6 port-list	Show on which interfaces the NPU offloading is enabled.
diagnose npu np6lite port-list	Same as above but for NP6-lite.
fnsysctl ifconfig <interface name>	Gives the same info as Linux `ifconfig`. The only way to see the MTU of the interface.
diagnose ip address list	Show IP addresses configured on all the Fortigate interfaces.
diagnose sys gre list	Show configured GRE tunnles and their state.
diag debug application pppoed -1 **dia debug application pppoe -1** **dia debug applicaiton ppp -1**	Enable all ADSL/PPPoE-related debug.
execute interface pppoe-reconnect	Force ADSL re-connection.

NTP debug

Table 8. NTP daemon diagnostics and debug

Command	Description
diag sys ntp status	Current status of NTP time synchronization. Shows all NTP peers and their detailed info: reachability, stratum, clock offset, delay, NTP version.
execute date	Show current date as seen by Fortigate.
exec time	Show current time as seen by Fortigate.

SNMP daemon debug

Table 9. SNMP daemon debug

Command	Description
diagnose debug application snmpd -1	ENable SNMP daemon messages debug.
show system snmp community	Show SNMP community and allowed hosts configuration

BGP

Table 10. BGP debug

Command	Description
diagnose ip router bgp level info **diagnose ip router bgp all enable**	Set BGP debug level to INFO (the default is ERROR which gives very little info) and enable the BGP debug.
exec router clear bgp all	Disconnect all BGP peering sessions and clear BGP routes in BGP table and RIB. Use with care, involves downtime.
get router info bgp summary	State of BGP peering sessions with peers, one per line.
get router info bgp network <prefix>	Detailed info about <prefix> from the BGP process table. Output includes all learned via BGP routes, even those not currently installed in RIB. E.g. `get router info bgp network 0.0.0.0/0`. The <prefix> is optional, if absent shows the whole BGP table.
get router info routing-table bgp	Show BGP routes actually installed in the RIB.
get router info bgp neighbors	Detailed info on BGP peers: BGP version, state, supported capabilities, how many hops away, reason for the last reset.
get router info bgp neighbors <IP of the neighbor> advertised-routes	Show all routes advertised by us to the specific neighbor.
get router info bgp neighbors <IP of the neighbor> routes	Show all routes learned from this BGP peer. It shows routes AFTER filtering on local peer, if any.
get router info bgp neighbors <IP of the neighbor> received-routes	Show all received routes from the neighbor BEFORE any local filtering is being applied. It only works if `set soft-reconfiguration enable` is set for this peer under `router bgp` configuration.
diagnose sys tcpsock | grep 179	List all incoming/outgoing TCP port 179 sessions for BGP.

Admin sessions

Table 11. Admin sessions management

Command	Description
get sys info admin status	List logged in administrators showing `INDEX` value for each session

Command	Description
execute disconnect-admin-session <INDEX>	Disconnect logged in administrator by the session INDEX.

Authentication

Table 12. Authentication in all kinds LDAP, Radius, FSSO

Command	Description
diagnose debug app fnbamd -1	Enable debug for authentication daemon, valid for ANY remote authentication - RADIUS, LDAP, TACACS+.
diagnose test authserver ldap <LDAP server name in FG> <username> <password>	Test user authenticaiton on Fortigate CLI against Active Directory via LDAP. E.g. test user `Tara Addison` against LDAP server configured in Fortigate as `LDAP-full-tree` having password `secret`: `diagnose test authserver ldap LDAP-full-tree "Tara Addison" secret`.
diagnose debug authd fsso list	List logged in users the Fortigate learned via FSSO
diagnose debug authd fsso server-status	Show status of connections with FSSO servers. Note: it shows both, local and remote FSSO Agent(s). The local Agent is only relevant when using Direct DC Polling, without installing FSSO Agent on AD DC, so it is ok for it to be `waiting for retry ⋯ 127.0.0.1` if you don't use it. The working state should be `connected`.

Fortianalyzer logging debug

Table 13. Verify and debug sending logs from Fortigate to Fortianalyzer

Command	Description
get log fortianalyzer setting	Show active Fortianalyzer-related settings on Fortigate.
config log fortianalyzer	Complete Fortianalyzer configuration on CLI, as GUI configuring is usually not enough for it to work.
get log fortianalyzer filter	Verify if any log sending filtering is being done, look for values of `filter` and `filter-type`. If there are any filters, it means not all logs are sent to FAZ.

Command	Description
exec log fortianalyzer test-connectivity	Verify that Fortigate communicates with Fortianalyzer. Look at the statistics in `Log: Tx & Rx` line - it should report increasing numbers, and make sure the status is `Registration: registered`.
exec telnet <IP of Fortianalyzer> 514	Test connectivity to port 514 on the Fortianalyzer. If pings are allowed between them, you can also try pinging.
diagnose sniffer packet any 'port 514' 4	Run sniffer on Fortigate to see if devices exchange packets on port 514. Click in GUI on `Test Connectivity` to initiate connection.

SD-WAN verification and debug

Table 14. SD-WAN verification and debug

Command	Description
diagnose sys sdwan health-check (6.4 and newer) **diagnose sys virtual-link health-check** (5.6 up to 6.4)	Show state of all the health checks/probes. Successful probes are marked `alive`, failed probes are marked `dead`. Also displays `packet-loss, latency, jitter` for each probe.
diagnose sys sdwan member **diagnose sys virtual-wan-link member**	Show list of SD-WAN zone/interface members. Also gives each interface gateway IP (if was set, 0.0.0.0 if not), `priority`, and `weight` both by default equal `0`, used with some SLA Types.
diagnose sys sdwan service **diagnose sys virtual-wan-link service**	List configured SD-WAN rules (aka `services`), except the Implied one which is always present and cannot be disabled, but is editable for the default load balancing method used. Shows member interfaces and their status `alive` or `dead` for this rule.
diag sys sdwan intf-sla-log <interface name> **diag sys virtual-wan-link intf-sla-log <interface name>**	Print log of <interface name> usage for the last 10 minutes. The statistics shown in bps: `inbandwidth`, `outbandwidth`, `bibandwidth`, `tx bytes`, `rx bytes`.
diag netlink interface clear <interface name>	Clear traffic statistics on the interface, this resets statistics of the SD-WAN traffic passing over this interface. Needed, if, for example, you changed SD-WAN rules, but not sure if it's already active. E.g. `diag netlink interface clear port1`.

Command	Description
diagnose firewall proute list	List ALL Policy Based Routes (PBR). SD-WAN in Fortigate, after all, is implemented as a variation of PBR. This command lists manual (classic) PBR rules, along with SD-WAN created via SD-WAN rules. **Important**: Manually created PBR rules (via `Network → Policy Routes` or on CLI `config route policy` always have preference over the SD-WAN rules, and this command will show them higher up.

Virtual Fortigate License Status

Table 15. Verify status of VM Fortigate License

Command	Description
get sys status | grep -i lic	Get status o fthe license (valid for Hardware Fortigate as well as VM). The corect status is `Valid`.
diagnose debug vm-print-license	Show detailed info on VM Fortigate license status: allowed CPUs and memory, date of license activation, license expiration date (if set), serial number.
diagnose hardware sysinfo vm full	Show license data as seen by FortiGuard: status (should be `valid=1`), last time it was checked (`recv`), answer code, should be `code: 200`, `code: 401` is for duplicate license found, `code: 502` is for VM cannot connect to FortiGuard, and `code: 400` is for invalid license.