CacheGuard OS

Command Line Manual - Version UF-2.3.3

access	Manage remote accesses to the appliance
admin	Manage administration services and accesses
antivirus	Configure the antivirus
apply	Apply new settings
authenticate	Manage the Web access authentication
cache	Manage the persistent Web cache
cancel	Cancel new settings
clear	Clear the terminal screen
clock	Manage the internal clock's date & time
conf	Manage the whole configuration
countrylist	Display valid country codes
dhcp	Manage the DHCP server
dns	Display, add or delete DNS (Domain Name Service) servers
domainname	Set or get the local domain name
email	Configure the administrator email account and email addresses to use.
embedded	Manage embedded applications
end	Mark the end of a template or gateway configuration context
error	Display descriptions of error codes returned by commands
exit	Exit the administration login
file	Load, save or clear all files related to the configuration
firewall	Configure the firewall
guard	Manage the URL guarding (filtering)
ha	Manage the High Availability
halt	Halt the Operating System
help	Print command's usage and description
history	Display the command history list
hostname	Set or get the hostname
information	Display descriptions of information codes returned by commands
ip	Manage IP addresses and routing configurations
job	Print a report on the current running operation in background
keyboard	Set the key map for the console port

ldap	Perform LDAP actions
license	Display the CacheGuard-OS License Agreement
link	Manage L2 network interfaces
log	Manage Logs
manager	Configure a manager system
mode	Manage general features and functions
ntp	Manage the NTP configuration
password	Manage passwords
peer	Manage connected peer Web proxies
ping	Send ICMP packets to network hosts
port	Manage built-in service network listening TCP/UDP ports
qos	Manage the network QoS (Quality of Service)
reboot	Reboot the system
register	Manage the appliance S/N and license key registration
rweb	Manage the reverse Web mode (reverse proxy)
setup	Performs a basic startup configuration
sslmediate	Manage the SSL mediation
system	Manage the operating system
timezone	Set or get the local time zone
timezonelist	Display valid time zone codes
tls	Manage TLS (SSL) certificates and other components
traceroute	Trace the route to a host
transaction	Manage a set of commands as a single transaction
transparent	Manage the transparent Web proxy
tweb	Manage the transparent Web proxy
urllist	Manage URL lists
usleep	Suspend the execution of the calling thread for microseconds
vlan	Configure 802.1q VLANs (Virtual LANs)
vpnipsec	Manage IPsec VPN tunnels and networks
vrrp	Manage the VRRP configuration in HA mode
waf	Configure the Web Application Firewall (WAF)
warning	Display descriptions of warning codes returned by commands

access

NAME

access - Manage remote accesses to the appliance

SYNOPSIS

[1] **access** [**web** [**raz** | (**add** (**internal** | **auxiliary** | **vpnipsec** | **admin** | **antivirus** | **file** | **mon** | **rweb** | **web**) *<ip>* [*<network-mask>* [*<qos%>*]]) | (**del** *<ip>* *<network-mask>*)]]

[2] **access** [**admin** [**raz** | (**add** | **del**) (**internal** | **external** | **auxiliary** | **vpnipsec**) *<ip>* [*<network-mask>*]]]

[3] **access** [**mon** [**raz** | (**add** | **del**) (**internal** | **external** | **auxiliary** | **vpnipsec**) (*<ip>* | *<name>*)]]

[4] **access** [**file** [**raz** | ((**add** (**internal** | **external** | **auxiliary** | **vpnipsec**) (*<ip>* | *<name>*) [(**ftp** | **sftp**) *<login>* [*<password>*]]) | (**del** (*<ip>* | *<name>*)))]]

[5] **access** [**antivirus** [**raz** | (**add** (**internal** | **external** | **auxiliary** | **vpnipsec**) *<ip>* [*<network-mask>* [*<qos%>*]]) | (**del** *<ip>* *<network-mask>*)]]

[6] **access** [**manager** [**raz** | (**add** (**master** | **backup**) (**internal** | **external** | **auxiliary**) *<ip>* "*<manager-public-ssh-key>*") | (**del** (**master** | **backup**))]]

DESCRIPTION

This command is used to get or set access policies for traffic exchanged with the appliance itself (and not routed via). To configure access policies for routed traffic via the appliance refer to the command **firewall**.

The first [1] usage form is used to define allowed networks connected or routed via a given network interface to access the appliance as a gateway for Web browsing (ie as a Web proxy). Traffic bandwidths can also be customised for a network using the optional *<qos%>* parameter. The *<qos%>* value is a percentage of the ingress or egress bandwidth allocated to **web** traffic and should be an integer between 1 and 100. Ingress and egress bandwidth values to which the percentage is applied are as follows:

• For accesses allowed via the native **internal** network interface or 802.1q pseudo network interfaces in **vlan** mode (**web**, **rweb**...), the ingress and egress bandwidths to consider are defined with the command usage form **qos shape web internal**.

• For accesses allowed via the **auxiliary** network interface, the ingress and egress bandwidths to consider are defined with the command usage form **qos shape web auxiliary**.

• For accesses allowed via the **vpnipsec** virtual network interface, the ingress bandwidth to consider is defined with the command usage form **qos shape vpnipsec external ingress**. The egress bandwidth for accesses allowed via the **vpnipsec** virtual network interface can’t be customised.

If no *<qos%>* is given, the value of 100% is used by default.

If no **web** access entry is defined, all networks located behind all interfaces, except the **external** interface, are allowed to access the Web with a QoS of 100%. If at least one **web** access entry is defined, only explicitly defined networks are allowed to use the appliance as a forwarding proxy via the specified network interface. In case where the transparent mode is activated (see the command **mode**), the appliance will act as a transparent forwarding proxy only for defined transparent networks. Use the command **transparent** to define transparent networks.

When the **vlan** mode is activated (see the command **mode**), you have the possibility to specify an explicit 802.1q pseudo interface instead of the **internal** interface. Allowed 802.1q pseudo interfaces are **admin**, **antivirus**, **file**, **mon**, **rweb** and **web**. It is recommended to always use 802.1q pseudo interfaces even if the **vlan** mode is disabled. In this way, the activation of the **vlan** mode would not require to rewrite Web access rules. Finally, when the **vlan** mode is activated, the **web interface replaces the internal** interface in all Web access rules.

The <*qos%*> is defined for all IPs belonging to a Web access network. The effective <*qos%*> for a given IP depends on the number of concurrent traffic exchanged from that network. An automatic scheduling system manages concurrent traffic to equitably share the bandwidth allocated to a given network. In a concurrent environment the <*qos%*> limit may be surpassed when the load of other networks is under their <*qos%*> limits. This mechanism is called borrowing. The borrowing could be activated or deactivated. See the command **qos** for further information on the borrowing mechanism. Please note that there is no obligation to have a total of 100% even if this is a recommended configuration.

The second [2] usage form configures access policies for administrators using *ssh* or the Web administration GUI. Only networks defined with this command are allowed to remotely administrate the system via the specified network interface. The system's administration IP address can be the internal, external or the auxiliary IP address according to the administration topology defined with the **admin topology** command usage form. When the **vlan** mode is activated (see the command **mode**), the internal administration IP address is the IP address associated to the **admin** 802.1q pseudo device (see the command **vlan**).

It should be noted that if a user uses the appliance as an explicit Web gateway (proxy), to allow her/him to access the administration Web GUI, the appliance's internal IP (or the admin IP in VLAN mode) should be allowed to access the appliance itself as an administrator. Allowing the appliance itself to have access to the administration Web GUI is not a good idea because it can weaken the appliance security. Instead, an administrator can configure her/his Web browser to do not use the appliance as a Web proxy for the the appliance's internal IP address.

The third [3] usage form configures access policies for monitoring (management) servers using the SNMP protocol. Only SNMP managers defined with this command are allowed to access the appliance via the specified network interface. If no management access policy is defined, SNMP accesses are not allowed.

Operations like backing up the system or loading a URL list require access to a file server. Only file servers defined with this command are allowed to exchange data with the appliance. The fourth [4] usage form (with the **file** keyword) allows you to define access policies for file servers. A file server is represented by its IP address or network name. If no file access policy is defined, file transfers to and from the appliance are not allowed. Supported protocols for file servers are FTP, SFTP and TFTP. For **ftp** and **sftp** servers, if a login name is given, a mandatory password is then required. If no password is given on the command line, the password is requested in hidden mode (see also the command **password**). Please note that if the targeted FTP server supports SSL encryption and the CCC (Clear Command Channel) FTP command, the system will use SSL/TLS for the authenticating phase in order to encrypt the transmitted credentials (login/password).

The integrated antivirus can be used as a service offered to external systems such as an MTA (Mail Transfer Agent). The system's antivirus server IP address can be the internal, external or the auxiliary IP address according to the antivirus topology defined with the **antivirus topology** command usage form. The fifth [5] usage form is used to define allowed networks connected or routed via a given network interface to use the appliance as an antivirus service. If no antivirus access policy is defined, the antivirus can't be accessed as a service. Traffic bandwidths can also be customised for a network using the optional <*qos%*> parameter. The <*qos%*> value is a percentage of the ingress or egress bandwidth allocated to **antivirus** traffic and should be an integer between 1 and 100. Ingress and egress bandwidth values to which the percentage is applied are as follows:

• For accesses allowed via the native **internal** network interface or the 802.1q pseudo network interface called **antivirus** (in **vlan** mode), the ingress and egress bandwidths to consider are defined with the command usage form **qos shape antivirus internal**.

• For accesses allowed via the **auxiliary** network interface, the ingress and egress bandwidths to consider are defined with the command usage form **qos shape antivirus auxiliary**.

• For accesses allowed via the **vpnipsec** virtual network interface, the ingress bandwidth to consider is defined with the command usage form **qos shape vpnipsec external ingress**. The egress bandwidth for accesses allowed via the **vpnipsec** virtual network interface can't be customised.

If no <*qos%*> is given, the value of 100% is used by default.

When an appliance is installed as a *gateway* (as opposed to an appliance installed as a *manager*), it can be directly managed and administrated using the CLI and/or the Web GUI. In case you have a *manager* appliance, you have the possibility to configure gateways using that manager. Managers uses the SSH and SFTP protocols to exchange with gateways. In practice, managers upload all configuration files on gateways using the SFTP protocol and then remote execute commands on them using the SSH protocol (gateways act as SFTP nad SSH servers). Please note that to allow managers to access gateways, SSH administration should be activated on gateways (see the **admin** command for further information).

The sixth [6] usage form allows you to give access to allowed mangers to administrate and configure the system. You can allow only one master manager and optionally one backup manager to administrate and configure a gateway. To allow a manager to access the system, use the keywords **manager add** followed by the manager's role (**master** or **backup**), the logical interface (**internal**, **external** or **auxiliary**) from which the manager can access the system, the manager IP address and finally the textual representation of the managers's SSH public key. To get the public SSH key of a manager, you can use the **manager ssh show** command on that manager. To delete a manager form the list of allowed manager, use the

keywords **manager del** followed by the manager's role (**master** or **backup**). Finally you can use the keywords **manager raz** to erase the allowed manager list. Using the keywords **manager** without any other arguments allows you to print the list of all managers. To show a manager SSH key content, you can use the "**admin ssh key show** <*key-id*>" command where <key-id> can be **mmanager** (for master manager) or **bmanager** (for backup manager).

SEE ALSO

admin (1) **antivirus** (1) **apply** (1) **firewall** (1) **manager** (1) **mode** (1) **password** (1) **peer** (1) **qos** (1) **transparent** (1) **vlan** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

admin

NAME

admin - Manage administration services and accesses

SYNOPSIS

[1] **admin** [(**snmp** | **ssh** | **ssh password** | **wadmin** | **waudit**) [**on** | **off**]]

[2] **admin tls** [*<tls-id>*[:*<ca-id>*]]

[3] **admin** [**ssh** [**fingerprint** | **generate** [**on** | **off**] | (**key** [**raz** | (**add** | **del** | **show**) *<key-id>* | **load** *<key-id>* (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>*])]]

[4] **admin topology** (**internal** | **external** | **auxiliary** | **vpnipsec**) [**on** | **off**]

[5] **admin user** [**raz** | (**add** | **del**) *<admin-name>* [*<admin-password>*]]

[6] **admin snmp** [(**user** [*<user-name>*]) | (**community** [*<community-password>*]) | (**privacy** [*<privacy-password>*]) | (**udp** | **tcp** | **tls** [(**on** | **off**)]) | **engine** | (**mode** [(**on** | **off**)])]

[7] **admin snmp certificate** [**raz** | **load** (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>*]

[8] **admin snmp** [**trap** [**raz** | **add** (**v1** | **v2c**) *<receiver-server>* [*<port>* [*<community>*]] | **del** (**v1** | **v2c**) *<receiver-server>* [*<port>*] | **test**]]

[9] **admin snmp** [**trap** [**raz** | **add v3** *<receiver-server>* *<port>* *<user-name>* (**sha256** | **sha384** | **sha512**) (**des** | **aes**) [*<auth-password>* [*<privacy-password>*]] | **del v3** *<receiver-server>* *<port>* *<user-name>*]]

DESCRIPTION

The first [1] usage form allows you to activate or deactivate administration and management services. To activate a service, use its name (**snmp**, **ssh**, **wadmin** or **waudit**) followed by the keyword **on**. Use the keyword **off** to deactivate it. Available services are as follows:

• **snmp**: the SNMP agent with the possibility to send SNMP traps.

• **ssh**: the SSH server.

• **wadmin**: the Web administration GUI

• **waudit**: the Web auditing GUI

The usage form **ssh password** allows you to activate or deactivate the SSH password authentication.

Web administration and auditing GUI allow you to configure and audit the system using a Web browser. The auditing service allows you to have a live summary view of different available logs (virus, access...) and inspect Web requests on reverse websites (only for reverse websites when the **waf** mode is activated. The Web auditing service is for debugging purpose only and should not be activated in normal circumstances with the risk of weakening the system. The auditing is available at the URL *https://<admin-ip>:<wadmin-port>* where *<admin-ip>* and *<waudit-port>* are respectively the administration IP address and the **waudit** port number. The administration IP address maybe be the internal, external or auxiliary IP address of the appliance according to the configured administration topology (see below). When the **vlan** mode is activated the native internal IP address can no longer be used. Instead of that, the IP address associated to the administration 802.1q pseudo device can be used (see the command **vlan** for further information).

The second [2] usage form allows you set the TLS certificate to use for the Web GUI and the SNMP agent over TLS. The TLS certificate is specified by a giving a TLS server identifier. You can optionally specify an intermediate CA certificate by giving its identifier separated by a colon from the TLS server identifier. In this case the specified intermediate CA certificate will be used for the Web GUI.

The third [3] usage form allows you to:

• Show the fingerprint of the RSA and DSA keys used by the SSH server.

• Regenerate those RSA/DSA keys.

• Manage public SSH keys.

By importing a public SSH key, the owner of the private SSH key associated to the imported public SSH key, can login to the system as the admin (or restricted administrator) user without having to enter a password.

To show the SSH server fingerprints, use the keywords **admin ssh fingerprint**. To arm the regeneration of the SSH server keys, use the keywords **admin ssh generate on**. Use the keyword **off** (instead of **on**) to cancel the regeneration.

The keywords **ssh key** without any additional arguments allows you to display the list of defined public SSH keys (each key is identified by an identifier). The keywords **key raz** allows you to reset that list. Importing a public SSH key is done in two steps. First and empty public SSH key should be added using the keywords **ssh key add** followed by an identifier associated to that public SSH key. In a second step, the public SSH key content can be loaded from a trusted file server (refer to the **access** command to define trusted file servers). To load a public SSH key use the keywords **ssh key load** followed by the public SSH key identifier to load and the public SSH key location. The public SSH key location is specified by three arguments: the protocol to use (**ftp**, **sftp**, **tftp**) to load the public SSH key file, the IP address (or name) of the file server on which the public SSH key file is located and the public SSH key file name. The specified file must contain a valid RSA (or DSA) public key. To remove a public SSH key use the keywords **ssh key del** followed by the public SSH key identifier to delete. Finally the usage form **ssh key show** followed by an SSH key identifier allows you to show the content of a public key. Note that the system supports the SSH protocol version 2 only. Public SSH keys are activated after using the **apply** command. Also if you try to load an SSH key that already exists in the system, the loading is simply ignored.

Please note that SSH keys are not part of the configuration thus they are not saved when the configuration is saved with the **conf** command. To save SSH keys and all other loaded files you cane make a backup of the system using the **system** command. You have also the possibility to load/save SSH keys among other files related to the configuration using the **file** command.

The fourth [4] usage form allows you to define the access topology for administration services. The access topology defines logical network interfaces to which administrators (or SNMP manager systems) can connect. To allow administration on a logical network interface, use its name (**internal**, **external**, **auxiliary** or **vpnipsec**) followed by the keyword **on**. To deny the administration, use the keyword **off** instead. Activating the administration on the **vpnipsec** virtual network interface allows the administration on the internal network interface via an IPsec VPN tunnel.

The fifth [5] usage form allows you to add or remove unprivileged administrator users. Unprivileged administrators have read rights only (ie. they can only consult the configuration). Without any arguments, this command displays the list of unprivileged administrators. To add an unprivileged administrator use the keyword **add** followed by its user name. To delete an unprivileged administrator use the keyword **del** followed by the name of the unprivileged administrator to remove. A valid administrator name must begin with an alphabetic character followed by alpha numeric characters as well as the characters "_" and "-". To erase all unprivileged administrators, use the keyword **raz**.

In interactive mode, when a new unprivileged administrator is added, the privileged administrator (*admin* user) is invited to define a password for the added unprivileged administrator. The first time an unprivileged administrator is logged in, she/he is invited to modify her/hist password. The new defined password is then applied to the console as well as to the Web GUI interfaces. Please note that administrator passwords are not part of the configuration. Hence, they are not saved when the configuration is saved.

In a non interactive mode (for instance when a configuration file is loaded from a file server), if the added unprivileged administrator does not exist, a password is automatically generated for the added unprivileged administrator as follows: the admin user name (*<admin-name>*) followed by the character @ (at), the string "appliance", the the character - (dash) and the current year. As an example for the an unprivileged administrator added during the 2023 year and called *foo*, the automatically generated password would be *foo@appliance-2023*.

The sixth [6] usage form of the **admin** command allows you to configure the internal SNMP (Simple Network Management Protocol) agent. The keyword **community** allows you to set the community string for SNMP-v1 and SNMP-v2c. With SNMP-v3 the community string takes the role of the authentication password using SHA-256 hash function. The keyword **user** allows you to set the SNMP-v3 user name. When using SNMP-v3 the data portion of the message being sent could be encrypted using AES (Advanced Encryption Standard). The keyword **privacy** allows you to set the encryption password for the encryption algorithm. Note that the privacy encryption is not mandatory and the agent accepts requests without encryption. The keywords **udp**, **tcp** and **tls** allow you to activate or deactivate respectively SNMP over UDP, TCP and TCP tunnelled over TLS (for encryption). Please note that the SNMP agent listens on the following ports:

• SNMP over UDP: port 161

• SNMP over TCP: port 161

• SNMP TCP tunneled over TLS: port 10161

Please note that only trusted monitoring managers are allowed to access the SNMP agnet. Use the **access** command to define allowed SNMP managers to access the SNMP agent.

The SNMP agent supports TLS over TCP connections using mandatory client and server SSL certificates. The SSL server certificate is the same as the SSL server certificate used for the Web GUI (see the second usage form above). The seventh [7] usage form allows you to associate the defined SNMP-v3 user name (see above) to a client SSL certificate. The client certificate can be loaded from a file server. Only trusted file servers are allowed. Trusted file servers are defined with the **access** command.

The eighth [8] and ninth [9] usage forms of the command **admin** allow you to configure SNMP managers (SNMP trap receivers) to which SNMP traps and notifications are sent. The system uses TCP to send SNMP notifications (**and not UDP**). The system is able to send different SNMP versions traps and notifications. Supported version are: *v1*, *v2c* and *v3* respectively for SNMP-v1 traps, SNMP-v2c inform notifications and SNMP-v3 inform notifications. SNMP traps and notifications are sent to receivers specified by its IP addresses (or network names) and port numbers. For SNMP-v1 and SNMP-v2c if the port number is omitted, the port number is set to 162 (SNMP trap default port). To send SNMP-v1 traps and SNMP-v2c inform notifications a community string should be specified (a community string acts as a password for SNMP versions prior to v3). To send SNMP-v3 inform notifications the user name, the authentication hash function (**sha256**, **sha384** and **sha512**) and the encryption algorithm (**des** or **aes**) should be specified. According to the security level required by the SNMP-v3 receiver, an authentication password and possibly a privacy encryption password should be specified. If the receiver does not require those security levels just omit related parameters in the command. Please note that when passwords are specified they must be at least 8 characters long.

To check the connectivity with SNMP receivers you can send testing traps to all configured receivers by using the **admin snmp trap test** command. Please note that the new configuration should be applied using the command **apply** before being able to send testing traps.

The following is a brief description of some notifications sent by the system:

• During the installation, the system reserves the required space on HDDs to store different logs based mainly on users number and reverse websites. If a log file abnormally grows too quickly (maybe because the system is under a DoS attack) an SNMP trap is sent to notify that misbehaviour.

• During the installation, the system reserves required space for different filesystems according to the HDDs capacities so the system should never have a lack of space on disks. If for any reason (maybe an introduced bug) a filesystem’s free disk space falls below the threshold of 5%, an SNMP trap is sent to notify that misbehaviour.

• All network links are monitored so in case of a link up or down an SNMP trap is sent to notify that change.

• The load average of the system is continuously monitored and average loads for the past 5 and 15 minutes are calculated. If averages exceed the thresholds of 99% and 95% respectively for the past 5-minutes and 15-minutes, an SNMP trap is sent to notify that overload.

• All essential services are monitored so in case of a failure, disruption or lack of hardware resources to start enough related system processes to support the load an SNMP trap is sent to notify the disruption.

• A health checker service continuously examines all vital services and in case of a service failure, tries to restart it. In that case an SNMP trap is sent to notify that action. After the attempt to restart the service, another SNMP trap is sent to notify the result of that operation (failure or success). Finally if the High Availability mode is activated (see the command "mode ha") and the attempt to restart the service fails, an SNMP trap is sent to notify the failure. In this case all VRRP interfaces are shut down to explicitly remove the failed node from the pool of HA nodes.

• During URL lists auto loading if one or more URL list files can’t be loaded an SNMP trap is sent to notify the failure.

• If the antivirus mode is activated and the virus signature data base is outdated by more the one day an SNMP trap is sent to notify the dysfunction.

• If the hardware hosting system have HDDs with SMART (Self-Monitoring, Analysis and Reporting Technology) capapabilities, they are monitored and in case of failures on HDDs notifications are sent.

• If the system has been installed with software RAID capabilites, the RAID is monitored and in case of failures on HDDs notifications are sent.

• If a USB Ethernet adapter is plugged or unplugged the system sends an SNMP trap. A similar SNMP trap is sent during the appliance startup if a NIC is added to or removed from the system.

• If the IP routing table contains multi gateways routes, the system sends an SNMP trap in case of unavailability of those gateway.

The system supports known MIBs used to monitor Linux systems and also a dedicated MIB called

CACHEGUARD-MIB. You can find the ASN.1 MIB description of the CacheGuard MIB on the original installation CDROM or on the official CacheGuard website.

SEE ALSO

access (1) **apply** (1) **file** (1) **mode** (1) **password** (1) **system** (1) **tls** (1) **vlan** (1) **vrrp** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

antivirus

NAME

antivirus - Configure the antivirus

SYNOPSIS

[1] **antivirus** [**auto** [<*country-code*>]]

[2] **antivirus** [**extended** [**url** [<*URL*>]] | **vload** [(**on** | **off**)]]

[3] **antivirus whitelist signature** (**load** (**ftp** | **sftp** | **tftp**) <*file-server*> <*file-path*> | **clear**)

[4] **antivirus whitelist domainname** [(**add** | **del**) <*domain-name*> | **raz**]

[5] **antivirus** [**maxobject** [<*file-size*>]]

[6] **antivirus** [**pua** [(**on** | **off**)]]b

[7] **antivirus topology** (**internal** | **external** | **auxiliary** | **vpnipsec**) [**on** | **off**]

[8] **antivirus** (**update** | **create**) [**report** | **force** [**wait**]]

[9] **antivirus** [**report**]

DESCRIPTION

When the antivirus mode is activated, malware (viruses, trojans and worms) coming from the Web are eradicated by the system even before entering into your local networks. The command **antivirus** is used to configure and manage this antivirus. To activate the antivirus, use the command **mode antivirus on**.

The antivirus works in both forwarding (**web**) and reverse (Brweb) modes. In forwarding mode, it blocks all browsing accesses to malware objects while in reverse mode all attempts to upload malware on a protected Web server are blocked.

The antivirus detects MS Office macro viruses, mobile malware, and other threats. It supports 32/64-bit Portable Executable files and 32-bit ELF files. Additionally, it handles the following files:

• PE files compressed or obfuscated with the following tools: Aspack (2.12), UPX (all versions), FSG (1.3, 1.31, 1.33, 2.0), Petite (2.x), PeSpin (1.1), NsPack, wwpack32 (1.20), MEW, Upack, Y0da Cryptor (1.3).

• Almost every mail file format including TNEF (winmail.dat) attachments are supported.

• The most popular file formats like: MS Office and MacOffice files, RTF, PDF, HTML.

• Various obfuscators, encoders, files vulnerable to security risks such as: JPEG (exploit detection), RIFF (exploit detection), uuencode, ScrEnc obfuscation.

The antivirus scans not only simple files but looks inside archive and compression files. The following archive and compression formats are supported: Zip (+ SFX), RAR (+ SFX), Tar, Gzip, Bzip2, MS OLE2, MS Cabinet Files (+ SFX), MS CHM (Compiled HTML), MS SZDD compression format, BinHex, SIS (SymbianOS packages), AutoIt, NSIS.

Note that for performance reasons video/audio streaming contents are not checked by the antivirus in forwarding mode. In reverse mode all uploaded contents are checked.

Every 60 minutes the system automatically checks for virus signature DB updates and if necessary, downloads new virus signatures by connecting to regional servers using HTTPS. Updates are downloaded

from *db.<country-code>*.clamav.net (where the *<country-code>* is a two letter ISO 3166-1 alpha-2 code) or from *database.clamav.net*. The first [1] usage form allows you to set the regional update server name. To set the regional update server name use the keyword **auto** followed by your two letter country code. Use the command **countrylist** to get a list of valid country codes.

For a higher level of protection, extended antivirus signatures can be loaded into the system. The second [2] usage form allows you to configure the extended antivirus. Extended antivirus signatures can be loaded into the system using a supported file transfer protocol from a location that you should specify as a URL. Supported file protocols are **sftp**, **ftp** and **tftp**. The second usage form allows you to define that URL using the keywords **extended url** followed by a valid URL (*ftp://ftp.cacheguard.net/AV for instance). To allow the system to automatically load extended antivirus signatures from the specified URL, the URL host part should belong to the list of trusted file servers defined with the* **access** command. Additionally, if the file server is protected by username/password, it must be configured using the **password** command.

To load extended antivirus signatures, three methods can be used. The **load** method simply downloads signature files without any verifications. The **vload** (verify load) method allows you to secure downloads by verifying downloaded files contents. This is useful when you download signatures files from a file server managed by CacheGuard Technologies Ltd or one of its referenced partners. When using the **vload** method, a signature file is downloaded alongside the antivirus signatures file and the antivirus signatures file is verified using that signature file to assure that the downloaded antivirus signatures file has not been altered during its transfer. The signature file name has the same name as the downloaded antivirus signatures file followed by the extension *.sig*. If the *gateway* system is managed by a *manager* system (see the **manager** command), extended antivirus signatures can be pushed by the manager. In this case you can use the **push** method. The loading method can be set using the **extended method** keywords followed by the chosen method name (**load**, **vload**, **push**). Please note that if you modify the extended antivirus URL or its loading method, extended antivirus signatures are not updated during the **apply** operation. To effectively update extended signatures you should explicitly update the extended signatures by using the **antivirus update** usage form (see below) or wait for the next automatic update. If the chosen method is **push**, updates are done in two asynchronous phases: first, updates are pushed by the manager to the gateway and then they are asynchronously taken into account by the gateway.

In case where you encounter a false positive signature match, you should contact our support services to submit your case so we can study it and possibly fix it. Meanwhile, if your activity is blocked because of false positive matches, you have the possibility to bypass their checks with your own whit list of virus names. The third [3] usage form allows you to load a white list of virus signatures form a trusted file server. To load a white list from a file server use the keywords **whitelist signature load** followed by the protocol to use, the file server name or IP address and the white list file name. Only trusted file servers are allowed. Trusted file servers are allowed using the command **access**. A valid white list of virus signatures is a plain text file containing virus names (one virus name per line). To clear a previously loaded white list use the keywords **whitelist signature clear**. On a *manager* system, the white list of virus signatures is managed globally (ie. is not specific to a template or gateway context and can only be loaded outside a template or gateway context).

The fourth [4] usage form allows you to define a white list of domain names for which the antivirus at the Web gateway is bypassed in forwarding mode. You can use this usage form to white list websites such as *www.phishtank.com*.

The antivirus scans only files smaller than an upper limit. The fifth [5] usage form allows you to set this upper limit. By default the upper limit is 2048 kilobytes. To change this value use the keyword **maxobject** followed by the required size in kilobytes. Note that for optimal performance you should not leverage this value. The minimum and maximum authorized values are respectively 1024K and 24576 KB.

Additionally the antivirus may detect, Possibly Unwanted Applications (PUA). The sixth [6] usage form allows you to activate or deactivate the PUA detection mode. To activate the PUA checks use the keyword **pua** followed by the keyword **on**. To deactivate the PUA checks use the keyword **pua** followed by the keyword **off**. Detected PUA categories are as follows:

• Packed: This is a detection for files that use some kind of runtime packer. A runtime packer can be used to reduce the size of executable files without the need for an external unpacker. While this cannot be considered malicious in general, runtime packers are widely used with malicious files since they can prevent malware from detection by an antivirus product.

• PwTool: Password tools are all applications that can be used to recover or decrypt passwords for various applications like mail clients or system passwords. Such tools can be quite helpful if a password is lost, however, it can also be used to spy out passwords.

• NetTool: NetTools are applications that can be used to sniff, filter, manipulate or scan network traffic or networks. While a networkscanner can be an extremely helpful tool for admins, you may not want to see an average user playing around with it. Same goes for tools like netcat and the like.

• P2P: Peer to Peer clients can be used to generate a lot of unwanted traffic and sometimes it happens that copyrights are violated by downloading copyright protected content (music, movies),therefore we consider them unwanted.

• IRC: IRC Clients can be a productivity killer and depending on the client, can be a powerful platform for malicious scripts (take mIRC for example).

• RAT: Remote Access Trojans are used to remotely access systems, but can be used also by system admins, for example VNC or RAdmin.

• Tool: General system tools, like process killers/finders.

• Spy: Keyloggers, spying tools.

• Server: Server based badware like DistributedNet.

• Script: Known "problem" scripts written in Javascript, ActiveX or similar.

Please note that PUA detection may be too aggressive and lead to false positive matches.

The antivirus is mainly used by the integrated proxy to block malware in Web traffic. But it can also be used as a service offered to external systems such as an MTA (Mail Transfer Agent). Please refer to the commands **port** and **access** to configure the antivirus as a service for external systems. The seventh [7] usage form allows you to define the antivirus access topology. The antivirus access topology defines logical network interfaces to which external systems can connect to request the antivirus. To allow connections on an logical network internal interface use its name (**internal**, **external**, **auxiliary** or **vpnipsec**) followed by the keyword **on**. To deny connections, use the keyword **off** instead. Activating connections on the **vpnipsec** virtual network interface allows the connections on the internal network interface via an IPsec VPN tunnel.

The antivirus used by the present system is ClamAV. Please refer to the documentation of your external systems to get help on how to connect them to the antivirus.

The eighth [8] usage form allows you to perform an explicit **update**. To download and create the whole signature database use the keyword **create**. Updating and creating are asynchronous operations and are executed in background unless the **force wait** keywords are used. Note that you have to wait for the termination of other asynchronous commands before running these commands. **CAUTION**: in order to avoid to flood update servers, explicit **update** and **create** operations should be used moderately. Otherwise your system may be banned by some antivirus update servers. When the update (or create) operation is invoked, the user is invited to confirm its execution. The optional argument **force**, allows you to bypass this confirmation. To print a brief report on the **update** and **create** operations, use the keyword **report**. This report may produce some errors in different contexts. Meaningful errors are as follows:

[Antivirus signature base update (or create) context]:

• Error 58: can’t read databases from remote servers.

• Error 59: Remote servers are not fully synchronized (try again later).

• Error 101-109: can’t resolve remote servers names.

• Error 121: the Antivirus extended update program has been killed.

• Error 122: a downloaded Antivirus extended DB is not authentic. Use the command **antivirus update** to retry a signature update and get error details.

• Error 123: can’t download the Antivirus extended index file.

• Error 124: error(s) during the AV extended DB file(s). Use the command **antivirus update** to retry a signature update and get error details.

• Error > 124: multiple errors occurred. The error number is the sum of the above error numbers.

[Antivirus extended signature index update context]:

• Error 68: file not found on TFTP server.

• Error 78: the resource referenced in the URL does not exist.

• Error 101: the index file signature verification failed.

Finally the ninth [9] usage form displays a report on the status of the automatic antivirus signature updates.

CAUTION

• A Web Gateway as an antivirus is a network equipment that blocks malware coming from the Web and contributes to reinforce your security. Never deactivate local antivirus on your workstation or servers.

• If for some reason the antivirus service could not start (because for instance the gateway is disconnected from the internet), the Web access is blocked (the antivirus checking can’t be bypassed).

WAF

The present system is a Web gateway that protects against threats coming from the Web. When configured in forwarding mode (**mode web on**) and when the antivirus is activated, it protects the Web browsing from virus infections (as long as it is implemented in your network as a proxy or a transparent web proxy). When the system is implemented as a reverse proxy (**mode rweb on**) and WAF (**mode waf on**) in front of your Web servers, activating the **antivirus** allows you to scan all attempts to upload files onto your Web servers and instantly blocks malware before they reach the Web servers. Note that the only supported method to upload a file is the POST method with an encryption type of "multipart/form-data".

An HTML code that allows you to upload a file can be as follows:

<form enctype="multipart/form-data" method="post" action="/upload-file.html">

File name: <input type="file" /> <input type="submit" value="Upload" />

</form>

ANTIVIRUS UPDATES, QOS AND FIREWALL

1- If your system is placed behind a third party firewall, you should allow the following traffic in order to allow the antivirus signature updates:

• HTTPS (TCP 443) traffic form the system to the internet.

• Passive FTP (TCP 21) traffic form the system to *ftp.cacheguard.net* (commercial edition only).

2- If you plan to use the antivirus as a service for external systems and a third party firewall is implemented between those systems and the antivirus, you must allow the following traffic:
• TCP traffic from external systems to the system on its antivirus port (defined by the command **port**).

• TCP traffic from external systems to the antivirus on ports 61440 to 65535.

3- The QoS applied to signature update traffic that use HTTPS is the same as the QoS defined by "**qos shape web external**". The QoS applied to signature update traffic that use FTP is the same as the QoS defined by "**qos shape file**". See the command **qos** for further information.

EXTENDED ANTIVIRUS INDEX FILE

CacheGuard Technologies Ltd provides extended antivirus signatures available as a subscription service. However you can maintain your own antivirus signatures DB files on a file server and configure the system to automagically load them. To do so you must follow instructions below:

• Signatures DB files should be in gzip format and named with the **.gz** extension. They should be compatible with ClamAV.

• For each signatures DB file, and MD5 fingerprint file should be present on the file server and named with the **.md5** extension. For instance the MD5 file for the signatures DB file named *my-virus.gz* should be named *my-virus.gz.md5*. The MD5 file should contain the MD5 fingerprint (in lowercase) of the signatures DB file in a single line (and nothing else).

• An index file file named **index**, should be maintained and present on the file server. Each line in the index file must contain the following: <*signatures-db-file*>.gz <size-in-bytes> <md5-fingerprint>. For instance a line in the index file could be: my-virus.gz 86080 648c95f4c2f3f5b8730eea0a735300b7

• An archive compressed file in *tar.gz* format and named **pack.tar.gz** should be present on the file server. This archive file should include all previously listed files in a directory specified by the configuration command **antivirus extended url** <*URL*> (see above).

SEE ALSO

access (1) **countrylist** (1) **file** (1) **manager** (1) **mode** (1) **port** (1) **qos** (1) **rweb** (1) **waf** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

apply

NAME

apply - Apply new settings

SYNOPSIS

[1] **apply** [**check** | **force** [**wait**] | **report** | **cancel** [**force** [**wait**]]]

DESCRIPTION

When you use different commands to configure the system (or submit settings using the GUI) new settings are not immediately applied to the the running system but only program the system for a future new configuration considered as a whole. Most commands display both the running (current) and the new (net yet applied) configuration and the whole configuration can be displayed using the **conf** command.

To take effect, a new configuration should be applied to the system using the **apply** command. In the other hand the **apply** command replace the running configuration with the new configuration. The apply operation runs in background unless the **force wait** keywords are used. If you modify the new configuration during the apply operation, there are chances that your new configuration would be taken into account. You can use the **conf diff** command after the termination of the apply operation to display differences between the running and the new configuration.

Before being applied to the system, the **apply** command verifies the integrity of the new configuration as a whole to make sure that all new parameters are both consistent. After this step and if all goes well, the user is invited to confirm the apply operation. The optional argument **force** allows you to bypass this confirmation step. If the optional **check** keyword is used, the operation finishes after the integrity checks and possibly integrity errors are displayed. The optional **report** keyword allows you to display a system report on the last **apply** command.

Finally it is possible to cancel (or abort) a running **apply** operation and get the previous configuration before launching the **apply** command. To cancel the current running apply operation, use the **cancel** keyword. However, please note that the apply of the following settings can’t be cancelled: settings that depend on the content of external files (for instance custom WAF rules), generated SSL certificates and administrators passwords. Note that some sub-operations attached to the **apply** program cannot be cancelled immediately. **CAUTION**: aborting some operations such as downloading the antivirus signatures may let the system in an inconsistent state. That’s why the cancellation of an apply operation should always be followed by a new apply operation.

Please note that:

• You have to wait for the termination of other asynchronous commands before running the **apply** command.

• If you load a backup file (see the **system** command) to restore a system and at the same time you manually modify the new configuration using the CLI (or the GUI), the new configuration included in the backup file erases the manually modified new configuration.

The apply report (mentioned above) may produce some additional runtime errors in different contexts. Meaningful runtime errors are as follows:

[Antivirus signature base update context]:

• Error 58: can’t read databases from remote servers.

• Error 59: mirrors are not fully synchronized (try again later).

• Error 101-109: can’t resolve remote servers names.

[Antivirus extended signature index update context]:

• Error 6: couldn’t resolve host. The given remote host was not resolved.

• Error 7: failed to connect to host.

• Error 28: operation timeout. The specified timeout period was reached according to the conditions.

• Error 67: the user name, password, or similar was not accepted and the client failed to log in.

• Error 68: file not found on file server.

• Error 78: the resource referenced in the URL does not exist.

• Error 101: the index file signature verification failed.

[Antivirus white list integration context]:

• Error 11: can’t compile the antivirus white list.

• Error 13: can’t integrate the antivirus white list.

• Error 15: can’t reload the antivirus DB files.

[Appliance and license registration context]:

• Error 2: failed to initialize.

• Error 6: couldn’t resolve host. The given remote host was not resolved.

• Error 7: failed to connect to host.

• Error 22: HTTP page not retrieved. The requested URL was not found or returned another error with the HTTP error code being 400 or above.

• Error 26: read error. Various reading problems.

• Error 27: out of memory. A memory allocation request failed.

• Error 28: operation timeout. The specified timeout period was reached according to the conditions.

• Error 33: HTTP range error. The range "command" didn’t work.

• Error 34: HTTP post error. Internal post-request generation error.

• Error 35: SSL connect error. The SSL handshaking failed.

• Error 42: aborted by callback. An application told to abort the operation.

• Error 47: too many redirects. Hit the maximum amount when following redirects.

• Error 51: the peer’s SSL certificate or SSH MD5 fingerprint was not ok.

• Error 52: the service service didn’t reply anything, which here is considered an error.

• Error 53: SSL cryptographic engine not found.

• Error 54: cannot set SSL cryptographic engine as default.

• Error 55: failed sending network data.

• Error 56: failure in receiving network data.

• Error 58: problem with the local certificate.

• Error 59: couldn’t use specified SSL cipher.

• Error 60: peer certificate cannot be authenticated with known CA certificates.

• Error 61: unrecognised transfer encoding.

• Error 65: sending the data requires a rewind that failed.

• Error 66: failed to initialise SSL Engine.

• Error 75: character conversion failed.

• Error 76: character conversion functions required.

• Error 78: the resource referenced in the URL does not exist.

• Error 80: failed to shut down the SSL connection.

• Error 83: issuer check failed.

• Error 100: the registration service returned a non digit value code.

• Error 111-140: the appliance has sent an illegal appliance registration request.

• Error 141-143: an invalid email address has been used to register the appliance.

• Error 171-175: the appliance registration service is unavailable at this moment.

• Error 181: the appliance registration service returned an unknown state.

• Error 183: the transmitted OTP is not valid.

• Error 185: the OTP is transmitted by an IP address which is not allowed to register this appliance.

• Error 187: the appliance has never been registered.

• Error 189: an invalid password has been transmitted by an already registered appliance.

• Error 191: the appliance has been already registered.

• Error 199: the appliance registration service returned an unknown code.

• Error 201-205: the registration service returned non conform values.

• Error 209: can’t add the S/N account.

• Error 211-227: the appliance has sent an unauthorized license registration request.

• Error 241-245: the license registration service is unavailable.

• Error 251: can’t register a license key for a non registered appliance.

• Error 253: the appliance can’t be authenticated.

• Error 255: the license key is not intended to be install on this appliance for capacity incompatibility reasons.

• Error 257: the license key is revoked.

• Error 259: the license key has been already registered for another appliance.

• Error 261: can’t register the license key because no subscription exists for it.

• Error 265: the subscription associated to the license key has been canceled.

• Error 267: the subscription associated to the license key has been disabled.

• Error 268: the subscription associated to the license is pending.

• Error 269-289: errors due to registration service unavailability.

• Error 300: the registration service returned an unknown state value.

[Checking the RAM capacity]:

• Error 1: the RAM capacity of the appliance is not enough to simultaneously activate all configured features. This error is encountered because either some warnings have been ignored during the OS installation or the RAM capacity of the appliance has been reduced after the installation. To avoid this error you can either deactivate some RAM consuming features (like the caching, antivirus or compression) or upgrade the RAM capacity of your appliance. Also if you encounter this error because you activated the caching mode, you have the possibility to reinstall the OS and reduce the HDD capacity usage during the installation.

[Custom WAF rules compilation context]:

• Error 10: the maximum number of WAF rules per reverse website has been reached during a WAF rule compilation. In case the maximum number is reached, the compilation stops and rules limited to that maximum number are applied. Please note that this error should not occur in normal situation as the maximum number of WAF rules is verified during the WAF rules loading (see the command **waf**).

[License key checking context]:

• Error 11: the appliance is not yet registered and therefore does not have a S/N.

• Error 13: the specified license key is not valid.

[SSL Mediation exceptions list compiling context]:

• Error 11: can’t convert the domain name list to dump format.

• Error 13: can’t convert the domain name list to db format.

• Error 15: can’t dump a URL list in db format.

• Error 17: can’t convert a URL list from dump format to db format.

• Error 19: can’t convert an exceptions list form db format to a flat format.

• Error 21: can’t remove subdomains from the exceptions list.

[System restore operation context]:

• Error 11-21: backup file corrupted.

• Error 23-25: can’t restore the loaded backup file on the present system because the OS version of the backup differs from the OS version of the present system.

• Error 27: can’t restore the loaded backup file because the backup has been been made on a machine that its hardware configurations and/or OS installation parameters differ from the present system.

[System patch version matching context]:

• Error 11-13: internal error during version matching verification.

• Error 15: the patch is not adequate.

[System patch unpacking context]:

• Error 11: the patch is not in a compressed format.

• Error 21: the patch is not in an archive format.

• Error 27: patch signature verification failed.

• Error 41: CPU architecture mismatch.

[System patch applying context]:

• Error 11: internal error in pre installation program.

• Error 13: internal error during patched component installation.

• Error 15: internal error in post installation program.

[System patch machine retuning context]:

• Error 1-4, 7-255: unexpected errors that may leave the appliance in an inconsistent state. It is recommended to re-install the OS from scratch using the installation media.

• Error 5-6: these errors are due to little free space on disk(s) and may leave the appliance in an inconsistent state. It is recommended to re-install the OS from scratch using the installation media.

[TLS components loading context]:

• Error 11: can’t add the signed certificate to the certificates index file.

• Error 13: can’t extract the certificate information from the loaded certificate.

• Error 15: can’t generate the certificate configuration file for the loaded TLS component file(s).

• Error 17: can’t remove the certificate from the certificates index to update the certificate.

• Error 61: can’t install the certificate information file.

• Error 63: can’t install the certificate configuration file.

• Error 65: can’t install the private key file.

• Error 67: can’t install the certificate file.

• Error 101: can’t generate the configuration file for the loaded CSR file.

• Error 103: can’t remove the certificate from the certificates index to sign the generated certificate.

• Error 105: can’t update the certificates index to generate the signed certificate with the system’s CA certificate.

• Error 107: can’t generate the signed certificate with the system’s CA certificate.

• Error 109: can’t install the generated signed certificate.

• Error 111: can’t install the CSR file.

[System’s CA components loading context]:

• Error 21: can’t generate the CA certificate information file.

• Error 31: can’t convert a root CA certificate from PEM format to DER format.

• Error 61: can’t install the CA certificate information file.

• Error 65: can’t install the private key file.

• Error 67: can’t install the CA certificate file in PEM format.

• Error 69: can’t link the CA certificate file for hashing.

• Error 71: can’t install the CA certificate file in DER format.

[TLS server component generation context]:

• Error 11: can’t generate the certificate configuration file.

• Error 13: can’t generate the private key.

• Error 15: can’t generate the CSR for the certificate configuration.

• Error 17: can’t remove the certificate from the certificates index.

• Error 19: can’t update the certificates index to sign the server certificate with the system’s CA certificate.

• Error 21: can’t generate and/or sign a certificate with the system’s CA certificate.

• Error 23: can’t update the certificate reference.

• Error 61: can’t install the server certificate information file.

• Error 63: can’t install the server certificate configuration file.

• Error 65: can’t install the private key file.

• Error 67: can’t install the server certificate file.

• Error 69: can’t install the server CSR file.

[System’s CA components generation context]:

• Error 11: can’t generate the CA configuration file.

• Error 13: can’t generate the private key for the system’s CA certificate.

• Error 15: can’t generate CA certificate.

• Error 17: can’t convert the CA public certificate from PEM format to DER format.

• Error 19: can’t link the CA certificate file for hashing.

• Error 21: can’t initialise the certificates index.

• Error 61: can’t install the CA certificate information file.

• Error 63: can’t install the CA certificate configuration file.

• Error 65: can’t install the private key file.

• Error 67: can’t install the CA certificate file in PEM format.

• Error 69: can’t install the CA certificate file in DER format.

[Third party CA certificate loading context]:

• Error 13: can’t install the third party CA certificate file.

[Client certificates generation context]:

• Error 11: can’t generate the client certificate configuration file.

• Error 13: can’t generate the client private key.

• Error 15: can’t generate the client certificate.

• Error 17: can’t remove the client certificate from the certificates index.

• Error 19: can’t update the certificates index to sign the client certificate with the system’s CA certificate.

• Error 21: can’t sign the client certificate with the system’s CA certificate.

• Error 23: can’t generate the PKCS12 client certificate bundle.

• Error 25: can’t base64 encode the PKCS12 client certificate bundle.

• Error 27: can’t install generated TLS components (private key, certificate...).

[URL list building/updating context]:

• Error 7: URL list signature verification failed.

• Error 9: the URL list is not in a gzip compressed format.

• Error 13: not enough space left in the URL list directory.

• Error 15: can’t uncompress the URL list.

• Error 17: the uncompressed URL list is not an ASCII file.

• Error 19: can’t create the new URL list.

• Error 21: URL list update failed because it has never been created before.

• Error 23: can’t apply the URL list update.

GENERAL NOTES

In case where the new configuration to apply requires a DNS server restart, some name resolutions may fail during the apply operation. In this case you should wait for the apply operation end and run the apply command again.

SEE ALSO

cancel (1) **conf** (1) **system** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

authenticate

NAME

authenticate - Manage the Web access authentication

SYNOPSIS

[1] **authenticate** [(**web** | **rweb**) [**on** | **off**]]

[2] **authenticate** [mode [(**ldap** | **kerberos**) [**on** | **off**]]]

[3] **authenticate** [ldap [request [*'<user-base-dn>' '<login-attr>'* [*'<passwd-attr>'* [*'<ldap-filter>'* [*'<group-dn>'*]]]]]]

[4] **authenticate** [**ldap server** [(**raz** | (**add** | **del**) (**ldap** | **ldaps** | **sldap**) *<name> <ip>* [*<port>]*)]]

[5] **authenticate** [**ldaps** [**ca** [**raz** | **set** *<ca-id>*]]]

[6] **authenticate** [**ldap [binddn** [*'<dn>'* [(**off** | **on**) *<password>*]]]]

[7] **authenticate** [**ldap** [**test** [<login-name> [<password>]]]]

[8] **authenticate** [**kerberos** [(**encrypt** [(**des** | **aes**)]) | (**web** *<canonical-name>*) | (**hapassword** *<shared-password>*)]]

[9] **authenticate** [**kerberos** [**server** [(**raz** | (**add** | **del**) *<kerberos-server>*)]]]

[10] **authenticate** [**kerberos** [**rweb** [**add** *<site-name> <canonical-name>* | **del** *<site-name>* | **raz**]]]

[11] **authenticate** [**kerberos** [(**create** *<admin-user>* [*<admin-password>* [**wait**]]) | **report**]]

[12] **authenticate** [**ad [rdn** [<host-relative-dn>]]]

DESCRIPTION

This command is used to configure the authentication module. The authentication module allows restricting the Web (or rWeb) usage to authenticated users only.

The first [1] usage form allows you to define target authenticated users. The keyword **rweb** specifies users accessing reverse websites (see the command **rweb**) while the keyword **web** specifies forwarding users accessing all other websites. Forwarding users are located behind the appliance while end-uses of reverse websites can be located in front of the appliance (coming form the internet) as well as behind the appliance. Keywords **on** and **off** allow you respectively to activate or deactivate the authentication for a target.

The second [2] usage form allows you to set the authentication mode. Two authentication modes are supported: LDAP and Kerberos. With the LDAP authentication, the browser basic authentication is used. In this authentication mode, the user is asked to enter its login/password in a popup window (the user should have an account on the LDAP server). With the Kerberos authentication, no login/password is requested as the authentication is negotiated with the Kerberos domain controller (and the same credentials used to login to the client machine is then used - This is called SSO: Single Sign On).

While in most cases, only one of these two authentication modes is to be activated you, you have the possibility to activate both modes at. In this case, the authentication behaviour would then be as follows:

• Forwarding users accessing the Web are authenticated using the Kerberos protocol first. If the Kerberos authentication fails, the LDAP authentication is used.

• Reverse users accessing cloaked/protected websites are authenticated the same way as before if the WAF mode is deactivated. With the WAF mode being activated, two behaviours are possible: if the Kerberos mode is explicitly activated for the reverse website (see usage form number ten), only the Kerberos authentication is used. Otherwise the LDAP authentication is used.

Usage forms from the third to sixth [3][4][5][6] allow you to configure the LDAP authentication. To configure LDAP authentication, use the keyword **ldap** followed by the appropriate argument.

The third usage form allows you to set the LDAP request sent to LDAP servers. To set the LDAP request use the keyword **request** followed by:

• The base DN (Distinguished Name) under which users are located.

• The attribute name that contains the user login.

• The the attribute name that contain the user password (if not specified, the bind method is used).

• The LDAP search filter to locate users under the base DN.

• Optionally an LDAP group identified by its DN to which users should belong.

Empty values (’’) can be used as the *<passwd-attr>* and the *<ldap-filter>* arguments. If an empty value is specified as the *<passwd-attr>*, an LDAP binding is performed during the basic authentication phase instead of a comparison of the entered password against the value stored in the *<passwd-attr>* LDP attribute (this is the preferred method used by Microsoft® AD (Active Directory®). Please note that the login attribute in AD® is *sAMAccountName* (or *cn*). Note that there is a limitation in specifying a password attribute: When the **waf** mode is activated or the **antivirus** and **compress** modes are both activated, the given password attribute is ignored and the LDAP bind is used instead.

In case where an optional LDAP group is specified, members of that group should be identified by their DN and stored in the *member* attribute of that group. Please note that as LDAP DNs and filters contains the character ’=’ they must be enclosed in quotation marks to avoid being interpreted by the shell. For instance consider the following command:

authenticate ldap request *’ou=people,dc=example,dc=com’ ’uid’ ’’ ’objectClass=inetOrgPerson’ ’cn=web,ou=groups,dc=example,dc=com’*

This command allows you to authenticate users of the class *inetOrgPerson* registered under the object *ou=people,dc=example,dc=com* and identified by the LDAP attributes *cn*. In addition, this command specifies a group identified by the DN *cn=web,cn=groups,dc=example,dc=com* to which users should belong to be considered as properly identifed.

As another example to request a Microsoft® AD (Active Directory®) you can use the following command:

authenticate ldap requestt *’cn=users,dc=example,dc=com’ sAMAccountName*

The fourth [4] usage form allows you to manage LDAP servers. To add an LDAP server, use the keyword **add** followed by the LDAP protocol (**ldap**, **ldaps** or **sldap**), the LDAP server network name, IP address and optionally port number. If no port number is specified, standard port numbers are used (389 for ldap/sldap and 636 for ldaps). Allowed LDAP protocol types are: **ldap**, **ldaps** and **sldap**. The protocol type **ldap** stands for the clear (without encryption) LDAP protocol while **ldaps** and **sldap** are used for SSL/TLS encrypted LDAP protocols. To specify LDAP over SSL/TLS use the type **ldaps**. To specify SSL/TLS encryption within LDAP, use the type **sldap** (refers to *STARTTLS*). Note that the **sldap** type cannot be mixed with other types. That means if the LDAP servers list contains an **sldap** server, all other LDAP servers must be **sldap** servers. To delete an LDAP server replace the keyword **add** by the keyword **del** in the latter syntax. To erase all LDAP servers use the keyword **raz**.

For a higher level of security an SSL certificate transmitted by an LDAP**S** server can be verified against the CA certificate that has been used to sign that SSL certificate. The fifth [5] usage form allows you to activate this feature by specifying the CA certificate ID (*<ca-id>*) of the CA certificate to use for verification. To add and import a CA certificate see the command **tls**.

Usually LDAP servers require that a client (here the an appliance) authenticate itself before being able to send LDAP request. This authentication process is called binding. The sixth [6] usage form allows you to activate of deactivate the LDAP binding and configure the DN (Distinguished Name) and password to use for the binding process. To activate the binding, use the keyword **binddn** followed by the DN of a privileged user (usually the *administrator* user), the keyword **on** and the authentication password for that privileged user (on the LDAP server). The password specified here should not be longer that 64 characters. You can also use the command **password** to set the binding password. If no binding is required, use the keyword **off** instead of the keyword on (no password is required). Tip: the Bind DN for a Microsoft® AD (Active Directory®) is usually in the form of: *cn=administrator,cn=users,dc=example,dc=com*.

The seventh [7] usage form allows you to test the authentication against configured LDAP servers. Please note that as with any other commands the new configuration should be applied using the command **apply** before being able to test the authentication.

Usage forms eighth to twelfth [8][9][10][11][12] allow you to configure the Kerberos authentication against

an AD® server. The keyword **encrypt** allows you to set the encryption type to use during authentication negotiations. Allowed encryption types are **des** and **aes**. The keyword **web** allows you to set the name by which the appliance system is identified as a Web proxy. This should typically be the name that you use to set Web proxies at the client side (the Web browser). Please note that a canonical short name should be specified here (*proxy* for instance). See below the definition of a valid canonical name.

In an HA configuration (with two or more redundant nodes), in order to allow the service continuity (in case of a failure on a node), all nodes should use the same service name and share the same account password on Kerberos servers. In this case the used password should never expire or change (if you modify the account password on an HA node, think about modifying it on all other HA nodes). The keyword **hapassword** allows you to set the shared account password. Refer to the commands **ha** and **vrrp** to get help on how to configure the HA mode.

Please note that in a non HA configuration (with a stand alone node), the **hapassword** is not used as the system generates a random password and automatically changes it whenever it get older than 21 days.

To enable the Kerberos authentication mode, at least one Kerberos server (preferably two) should be specified. Kerberos servers can be specified by their DNS names or IP addresses. The present system is compatible with AD® domain controller. To use an AD® as a Kerberos server, the relative base DN of the present system should be specified in order to allow the AD® to identify it in its LDAP directory tree. To get the relative base DN of an LDAP object, you should remove the name specification prefix (cn=<*name*>) and the domain name specification part in an FDN (Full Distinguish Name). For instance if your domain name (see the **domainname** command) is *example.com* and if your system is identified by the FDN *cn=proxy,cn=computers,dc=example,dc=com* in your AD® LDAP directory tree, your relative base DN will be *cn=computers*. The **authenticate ad rdn** usage form allows you to specify that relative base DN.

The Kerberos authentication can also be activated for users of reverse websites provided that users to authenticate depend on the same Kerberos servers as the reverse proxy (**rweb** module). The Kerberos authentication is not activated for all reverse websites but only for reverse websites that are associated to a canonical name (short hostname). To associate a canonical name to a reverse website use the keywords **authenticate kerberos rweb add** followed by an existing reverse website name and a valid canonical name. See below the definition of a valid canonical name. To deactivate the Kerberos authentication for a reverse website use the keywords **authenticate kerberos rweb del** followed by the reverse website name. To deactivate the Kerberos authentication for all reverse websites use the keywords **authenticate kerberos rweb raz**.

It is important to note that the first time the Kerberos authentication mode is activated (after the **apply** operation), the Kerberos authentication should be initialised. During the initialisation process all LDAP objects associated to services provided by the proxy (*HTTP/proxy.example.com* in the example above) and the reverse proxy are created in AD®. Also the initialisation process allows your system to obtain a Kerberos tickets. To this end, it is necessary to use an AD® account with administrator permissions (user *administrator* for instance). The **authenticate kerberos create** <*admin-user*> usage form allows you to initialise the Kerberos authentication. The initialisation process runs in background unless the <*admin-password*> is specified on the command line and the **wait** keyword is used. This command requires that you interactively enter the password associated to the used administrator account. The given password here is not permanently saved and is removed after having obtained a Kerberos ticket. Please note that the Kerberos initialisation is an asynchronous operation and is executed in background. The **report** keyword allows you to display a report of the last Kerberos initialisation operation.

GENERAL NOTES

Because an LDAP filter or a distinguished name contains the equal character, it should be quoted with simple or double quotes. Otherwise the shell interpreter considers that as a variable setting.

KERBEROS NOTES

Since the security of Kerberos authentication is in part based upon the time stamps of tickets, it is critical to have accurately set clocks on Kerberos servers and the present system. For this doing, the usage of NTP servers is highly recommended (see the command **ntp**).

Regarding the Kerberos encryption type, if your AD is running on a Windows® server 2003 you should use the DES encryption type. For Windows® server 2008 and above the AES encryption type should be used (DES is an acronym of Data Encryption Standard while AES stands for Advanced Encryption Standard).

A canonical name is a string beginning with an alphanumeric character with a maximum length of 14 characters and containing a combination of alphanumeric and the dash ("-") characters. A canonical name should contain at least one alphabetic character and can’t begin or end with the dash character.

KERBEROS, AD® AND FIREWALL

• If your appliance is placed behind a third party firewall, you should allow the following traffic in order to allow the kerberos authentication against AD® servers:

• Kerberos traffic (TCP 88) form the appliance to the Kerberos servers (AD®).

- Kerberos password traffic (UDP 464) form the appliance to Kerberos servers (AD®).
- LDAP traffic (TCP 389) form the appliance to the AD® servers.

LIMITATIONS

The authentication is not supported in transparent mode.

SEE ALSO

access (1) **clock** (1) **domainname** (1) **ha** (1) **mode** (1) **ntp** (1) **password** (1) **rweb** (1) **tls** (1) **vrrp** (1) **waf** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

cache

NAME

cache - Manage the persistent Web cache

SYNOPSIS

[1] **cache [object** [*<min-size>* *<max-size>*]]

[2] **cache [bigobject** [**off** | **on** *<min-size>* *<max-size>*]]

[3] **cache** [**clear** [**report** | **force** [**wait**]]]

[4] **cache [loadurl** *<url>*]

[5] **cache [report** [**size** | **rsize** | **bigsize** | **meanobject** | **used** | **rused** | **hit**]]

DESCRIPTION

The first [1] usage form is used to modify the maximum cached object size in the persistent memory. Higher values save more bandwidth but may rapidly saturate the persistent cache memory. Lower values cache smaller objects and therefore increase performance. The size is specified in kilobytes.

The minimum size for cached object should be greater than or equal to 0 KB and less than or equal to the configured maximum size. The maximum size for cached object should be greater than or equal to 1024 KB and less than or equal to 262144 KB (256 MB). We suggest leaving these limits at their default values that are 1 KB for the minimum and 51200 KB (50 megabytes) for the maximum.

Objects smaller than or greater than those limits are not cached unless you activate the caching for big objects. When this type of caching is activated the system reserves an area its persistent cache to store objects bigger than the upper limit for common objects. This feature allows you to cache very big objects like smartphone OS without the disadvantage of having your total cache saturated by very big objects. The second [2] usage form allows you to manage big objects caching.

The third [3] usage form allows you to activate or deactivate the caching of very big objects. To activate this type of caching use the keyword **on** followed by the minim and maximum sizes for very big objects expressed in KB. The given minimum size should be greater or equal to the maximum object size for common objects and lesser than or equal to the size of the area reserved on the persistent cache for very big objects. To deactivate it, use the keyword **off**. When the caching for very big objects is deactivated, the cache is totally used to store common objects.

The size of the area reserved on the persistent cache for very big objects varies depending on the size of your hard drive(s) and other parameters given during the installation (the fifth usage form described later allows you to get this size).

In some circumstances you may need to clear the entire persistent cache. The third [3] usage form allows you to clear the entire persistent Web cache and the SSL certificate cache used in SSL mediation mode (see the command **mode**). The cache clearing operation runs in background unless the **force wait** keywords are used.

The fourth [4] usage form is used to force the system to reload a given URL. A valid URL value must be in the form:

<protocole>://<domainname>[/<path>] in which *<protocole> is* **http, https ftp.** See the command **domainname** for a valid domain name format. The last optional part *<path>,* is a path to a regular filename on the remote server.

This command is useful for automating the loading of specific URLs using the ssh programmed command and a remote task scheduler.

The third usage form allows you to clear the entire persistent Web cache. When this command is invoked,

the user is invited to confirm the clearing execution. The optional argument **force**, allows you to bypass this confirmation. The optional argument **report**, allows you to display a system report of the last **cache clear** command. Clearing operation is an asynchronous command and is executed in background. Note that you have to wait for the termination of other asynchronous commands before running this command.

The fifth [5] usage form allows you to display a **report** on the cache size and activity. The keyword **size** displays the total cache capacity in Kilo Byte while the keyword **used** displays the percentage of the main cache filled by Web objects. The keyword **bigsize** displays the cache size for big objects. The keyword **meanobject** displays the mean object size in the persistent cache. The keyword **hit** displays the main cache usage. It expresses how much data and requests are retrieved from the main cache compared to all data and requests passing through the system since the last cache initialization (or clearing).

In some particular configurations, reverse websites use a cache space other than the main cache space. This cache space is called the rWeb cache and should have been reserved during the appliance installation. The rWeb cache is activated in one of the following configurations:

* The WAF mode is activated.

* At least one rWeb site uses a sticky load balancing over more than one backend Web server.

* At least one rWeb site uses a load balancing method other than the round robin over more than one backend Web server.

Keywords **rsize** and **rused** display respectively the rWeb cache size and the the percentage of the rWeb cache filled by Web objects. Without any keyword, the command **cache report** displays all these values. The limitation is that the caching of very big objects is not possible in those configuration for reverse websites.

SEE ALSO

admin (1) **apply** (1) **mode** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

cancel

NAME

cancel - Cancel new settings

SYNOPSIS

[1] **cancel**

DESCRIPTION

This command is used to cancel a new configuration before its application with the command **apply.** The usage of different commands does not immediately affect the running system. A new configuration is applied only when the command **apply** is used. It is always possible to cancel all new settings by using this command.

SEE ALSO

apply (1) **conf** (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

clear

NAME

clear - Clear the terminal screen

SYNOPSIS

[1] **clear**

DESCRIPTION

This command clears your screen if this is possible.

SEE ALSO

exit (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

clock

NAME

clock - Manage the internal clock’s date & time

SYNOPSIS

[1] **clock** [*<YYYY/MM/DD-hh:mm:ss>*]

DESCRIPTION

This command is used to get or set the internal clock’s date & time. YYYY is a four digit year (2065 for instance), MM a two digit month (between 01 and 12) and DD a two digit day (01 for the first day of a month, 31 for the last day of March for instance). The time is specified by a two digit hour *hh* (between 1 and 23), a two digit minute *mm* and a two digit second *ss* (for example 12:00:00 specifies noon - 13:00:00 for 01:00 PM and 08:30:00 for 8:30 AM).

Date elements are separated by the character **/** ; time elements are separated by the character **:** and date and time are separated by the _ (underscore) character. An example of a complete date & time specification is *2065/04/10_02:30.23.* It specifies the following date & time: April 10, 2065 02:30:23 AM.

SEE ALSO

ntp (1) **timezone** (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

conf

NAME

conf - Manage the whole configuration

SYNOPSIS

[1] **conf** [**diff**]

[2] **conf load** (**ftp** | **sftp** | **tftp**) <*file-server*> <*file-path*>

[3] **conf save** (**ftp** | **sftp**| **tftp**) <*file-server*> <*file-path*> [**new**]

[4] **conf factoryreset**

[5] **conf manager** (**template** <*template-id*> | **ateway** <*gateway-id*>)

DESCRIPTION

Without any argument this command displays both the running and the new (not yet applied) logical configuration. It allows you to view all parameters at once before applying (using the command **apply**). When the optional argument **diff** is given, only the difference between the current and the new configuration is displayed. The first [1] usage form allows you to display the logical configuration.

In the second [2] usage form, the **load** argument allows you to load a configuration file located on a file server. Only trusted file servers are allowed. Trusted file servers are defined with the **access** command. The load usage form requires three mandatory arguments. The first argument is the protocol name (**ftp**, **sftp**, or **tftp**). The second argument is the name or IP address of the file server. The third argument is the configuration file name.

When **tftp** is used the configuration file must exist and be accessible on the file server. The file must contain a set of valid configuration commands. Empty lines or lines beginning with the character "#" are ignored. The loading of a configuration file does not affect the running configuration. The command **apply** must be used to apply the new loaded configuration.

The third [3] usage form is used to save the configuration in a file located on a trusted file server. This usage form requires the same arguments described for the load usage form. By default this command saves the running configuration (also called the current configuration). To save the new configuration (not yet applied) use the optional argument **new**. The saved file contains a list of configuration commands that can be loaded afterwards using the **conf load** usage form. Please note that when a setting is in the form of a list, the saved setting begins by a command that erase the whole list before adding new entries to the the new created list. This rule applies to all settings that are present in the system in two versions (the running and new versions). For lists that are present in the system only in one version (ie. the running and the new are the same), no erase command is saved. For instance, this is the case for templates on a *manager* system.

Also, please note that:

• Passwords used to connect to external services like SNMP managers or LDAP servers are always saved in an encrypted format. However as the encryption is done after the apply operation, parts of a new configuration that contain clear passwords are never saved.

• The fBconf command saves/loads the logical configuration only and not all files related to the configuration (such as SSL certificates or the antivirus signature base). To load or save files related to the configuration you can use appropriate commands or the **file** command to load or save all files in one operation.

The fourth [4] usage form (**factoryreset**) is used to make a factory reset of the configuration. When using this usage form, all parameters are set to their initial values. Please note that the administration passwords used to connect to the system are not part of the configuration and thus are not reset when using the 0

The fifth [5] usage form is only available on *manager* systems inside a template or gateway configuration context. This usage form allows you to quickly make a gateway configuration based on a template or another gateway configuration (the source configuration). Please note that this operation has an immediate effect on a (ie. does not require to use the **apply** command). You can refer to the **manager** command for further information. The built configuration inherit from all source configuration except from the following settings that normally are unique:

- The gateway Hostname (see the **hostname** command).
- Internal and Auxiliary IP addresses (see the **ip** command).
- All VRRP IPs (see the **vrrp** command).
- IP routes and Via Gateways (see the **ip** command).
- Transparent Web networks (see the **transparent** command).
- Access lists associated to Internal and Auxiliary interfaces (see the **access** command).
- All settings related to the embedded DHCP server (see the **dhcp** command).
- Firewall rules associated to Internal and Auxiliary interfaces (see the **firewall** command).
- IPsec VPN authentication PSK (Pre Shared Key) (see the **vpnipsec** command).
- Via Gateways and local networks for IPsec VPN tunnels (see the **vpnipsec** command).
- Via Gateways for reverse Web sites (see the **rweb** command).
- Share and HA peers (see the **peer** command).
- Client TLS objects.

SEE ALSO

access (1) **apply** (1) **file** (1) **manager** (1) **system** (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

countrylist

NAME

countrylist - Display valid country codes

SYNOPSIS

[1] **countrylist**

DESCRIPTION

This command displays all ISO 3166-1 alpha-2 country codes. You can use this command to find the code for a particular country. The country code retrieved here is to be used as an argument for commands such as "**antivirus country** <*country-code*>" or "**waf reputation country** <*country-code*>".

SEE ALSO

antivirus (1) **waf** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

dhcp

NAME

dhcp - Manage the DHCP server

SYNOPSIS

[1] **dhcp** [**range** [**raz** | (**add** | **del**) <*ip1*> <*ip2*>]]

[2] **dhcp** [**fixed** [**raz** | **add** <*hostname*> <*mac-address*> <*ip*> | **del** <*hostname*>]]

[3] **dhcp** [**peer** (**master** | **slave**) [**raz** | (**add** | **del**) <*peer-ip*>]]

[4] **dhcp report**

DESCRIPTION

CacheGuard integrates a built-in DHCP (Dynamic Host Configuration Protocol) server. This command allows you to configure the DHCP server. See the command **mode** to activate the DHCP server.

Without any argument, this command displays the DHCP configuration. To display only a specific DHCP feature give the DHCP feature name (**range**, **fixed** or **peer**).

The first [1] usage form allows you to add or delete an IP range. To add a DHCP IP range use the keywords **range add** followed by the lowest and highest IP addresses in a range. To delete a DHCP IP range use the keywords **range del** followed by the lowest and highest IP addresses in a range. To erase all ranges, use the keywords **range raz**.

Note that the DHCP server supports a limited number of IP address leases. The maximum number of supported IP address leases is the number of users configured during the installation phase.

The second [2] usage form allows you to associate a fixed IP address to a client. To associate a fixed IP address to a client use the keywords **fixed add** followed by the host name of the client, its MAC address and the fixed IP address. A valid MAC address is a sequence of 6 hexadecimal values between 00 and FF separated by the colon (":") character (for instance 00:01:02:03:04:05). To disassociate a fixed IP address from a client use the keywords **fixed del** followed by the host name of the client. To erase the list of fixed IP address use keywords **fixed raz**.

The third [3] usage form allows you to define a remote DHCP server as a failover peer for the local DHCP server. When using the DHCP failover feature, two CacheGuards must be configured as DHCP peer servers each for other (one must be configured as **master** and the other as **slave**).

To define a remote CacheGuard as a DHCP peer server for the local CacheGuard, use the keywords **peer add** followed by the role and the internal IP address of the remote CacheGuard (when using VLANs, give the IP address configured for the **peer** VLAN on the remote CacheGuard). The role is defined using the keywords **master** or **slave**.

To delete a DHCP peer server use the keywords **peer del** followed by the role and IP address of the remote CacheGuard. To erase all DHCP peer servers use the keywords **peer raz**.

The fourth [4] usage form allows you to display a report on active DHCP leases managed by the system.

LIMITATIONS

Please note that only one unique DHCP peer server may be configured.

SEE ALSO

apply (1) **mode** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

dns

NAME

dns - Display, add or delete DNS (Domain Name Service) servers

SYNOPSIS

[1] **dns** [**raz** | (**add** | **del**) <*ip*>]

[2] **dns** [**resolve** [(**on** | **off**)]]

DESCRIPTION

The first [1] usage form allows you to add (or delete) DNS servers. DNS servers allow the resolution of a domain name to an IP address. A DNS server is given by its IP address. The system has its own internal built in DNS server. To use the internal DNS server give the keyword **localhost** or the IP address *127.0.0.1* (the recommended configuration).

To add an DNS server use the keyword **add**. To remove one, use the keyword **del**. To erase all DNS servers, use the keyword **raz**.

The second [2] usage form allows you to arm the resolution of all names used in the configuration to IP addresses. When the resolution is armed all firewall and QoS rules are recompiled during the apply operation. To arm the resolution use the keyword **on**. To deactivate the resolution use the keyword **off**.

SEE ALSO

apply (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

domainname

NAME

domainname - Set or get the local domain name

SYNOPSIS

[1] **domainname** [<*name*>]

DESCRIPTION

This command is used to get or set the local domain name. The argument <name> must be a valid domain name. A valid domain name is a string beginning with an alphabetic character and containing a combination of alpha-numeric characters, the character "-" and the character ".".

SEE ALSO

apply (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

email

NAME

email - Configure the administrator email account and email addresses to use.

SYNOPSIS

[1] **email** [**ftp** [<*email-address*>]]

[2] **email** [**admin** [[<*email-address*>] [<*name*>]]]

[3] **email** [**account** [**raz** | (<*server*> | **auto**) <*port*> <*username*> [<*password*>]]]

[4] **email** [**send** <*email-address*> [<*name*> [<*subject*> [<*message*>]]]]

DESCRIPTION

This command is used to Configure the email server to send email from the system and set email addresses. The first [1] usage form is used to set or get the email address to use as the username when connecting to anonymous FTP servers.

The second [2] usage form is used to set or get the appliance administrator email address and name. The name must be a combination of alphanumeric characters and the space character. The administrator email address can be displayed in information and error Web pages generated by the system.

Some configuration files managed by the system (such as IPsec VPN client profile files) can be sent by email. Emails are sent from the administrator email address (set using the second [2] usage form). The third [3] usage form is used to set or get the administrator email account details to use when sending emails. Please note that only email servers that support StartTLS and PLAIN authentication can be used. To configure the administrator email account details, use the keyword **account** followed by the MTA server name, the TCP port on which the MTA is listening on (usually 587), the username associated to administrator email address (usually the same as the administrator email address) and the authentication password for that username.

If no password is given on the command line, the password is requested in hidden mode (see also the command **password**). To let the system to guess the MTA server name associated to the administrator email address, you can use the **auto** keyword as the MTA server name. To erase the MTA server name you can use the **raz** keyword.

You can use the fourth [4] usage form to send an email using the current (not new) administrator email account. This usage form can be very helpful to validate that the configured administrator email account is operational. To send a testing email you can omit all optional arguments.

SEE ALSO

apply (1) **password** (1) **vpnipsec** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

embedded

NAME

embedded - Manage embedded applications

SYNOPSIS

[1] **embedded** [**bevypn** [**off** [*<site-name>* [*<tls-id>*[:*<ca-id>*]]] | **on** *<site-name>* *<tls-id>*[:*<ca-id>*]]]

DESCRIPTION

This command allows you to manage embedded application. Embedded applications are applications that are built on top of a CacheGuard Gateway appliance and run on the appliance itself.

The first [1] usage form allows you to activate or deactivate embedded applications. To activate an embedded application, use the embedded application reference followed by the **on** keyword and appropriate arguments according to the embedded application to activate. Please note that the first time you activate an embedded application, you will need to register a license key that allows you to use it. You can refer to the **register** command for further information on how to register a license key. To deactivate an embedded application, use the embedded application reference followed by the **off** keyword.

To date, the only available embedded application is the **bevypn** (for *My VPN*) application. The **bevypn** application is an HTTPS Web application that allows you to easily create and manage VPN subscribers. To activate the **bevypn** embedded application, you must give it a public website name (could be an IP address) and associate it to a server TLS identifier. Please note that the chosen name should differ from reverse website names managed by the **rweb** command.

To use the **bevypn** application you must be authenticated through a login name and password. When you activate the application from a deactivated state, by default the password is set to the password used to login the administrator Web GUI. Afterwards, the password can be modified via the **bevypn** application itself. The login name to use is always *admin*. You can refer to the the **bevypn** documentation to get help on how to configure and use it.

LIMITATION

Embedded applications can’t be activated in HA mode.

SEE ALSO

apply (1) **ha** (1) **register** (1) **rweb** (1) B tls (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

end

NAME

end - Mark the end of a template or gateway configuration context

SYNOPSIS

[1] **end** [**cancel**]

DESCRIPTION

You can use this command on a **manager** system inside the following contexts to quit that context and return to the top level configuration contexts:

• Template configuration context.

• Gateway configuration context.

• Gateway execution context. In this case the optional **cancel** keyword allows you to cancel the execution of entered commands.You can alternatively press the CTRL+C keys to end and cancel the execution of entered commands.

Outside a template or gateway configuration context, this command has no effect.

SEE ALSO

manager (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

error

NAME

error - Display descriptions of error codes returned by commands

SYNOPSIS

[1] **error** [<*error-code*>]

DESCRIPTION

Use this command to display the description of all error codes return by commands. You can also display the description of a particular error by giving its code (<*error-code*>). This command can be very useful if for any reasons displayed errors are not fully displayed by commands.

SEE ALSO

manager (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

exit

NAME

exit - Exit the administration login

SYNOPSIS

[1] **exit**

DESCRIPTION

Exit the administration login.

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

file

NAME

file - Load, save or clear all files related to the configuration

SYNOPSIS

[1] **file load** (**ftp** | **sftp** | **tftp**) <*file-server*> [<*directory-path*>] [(<*exclusion*>)*]]

[2] **file save** (**ftp** | **sftp** | **tftp**) <*file-server*> [[<*directory-path*>] [(<*exclusion*>)*]]

[3] **file** (**clear** [(<*exclusion*>)*] | **report** | **exchange**)

[4] **file del** (**ftp** | **sftp**) <*file-server*> [<*file-path*>]

DESCRIPTION

When you load or save a logical configuration using the **conf** command, files related to that configuration such as TLS certificates are not loaded or saved. Normally, you will have to load or save them one by one using appropriate commands. For instance the **tls** command should be used to load or save TLS certificates. In some situation, it can be helpful to load or save all those files using a single command. That's the main purpose of the **file** command. The **file** command allows you to load or save all files related to the configuration in a one operation. To this end, the **file** command automatically executes all appropriates commands to load or save files related to the configuration. To make this work, files located on remote file servers should be named using a naming policy described in the**FILE NAMING POLICY** section below.

The first [1] usage form allows you to load all files related to the configuration from a remote file server. Only trusted file servers are allowed. Trusted file servers are defined with the **access** command. This usage form requires 2 mandatory arguments. The first argument is the protocol name (**ftp**, **sftp**, or **tftp**) and the second argument is the name or IP address of the file server. The optional third argument is the directory name on the file server from which files should be downloaded (see the**FILE NAMING POLICY** section below). If no directory name is specified, the default '/' directory is used.

The second [2] usage form allows you to save all files related to the configuration on a remote trusted file server. The save usage form requires the same arguments described for the load usage form. Concerned files are those that are installed and active in the system (after using the **apply** command). Files that are loaded into the system and not yet activated by the **apply** command are not saved.

Loaded or saved file types are given below. For each file type, the executed command is specified.

- Administrator public SSH keys: **admin ssh key load** <*key-id*> ...
- SNMP client certificate: **admin snmp certificate add** ...
- Custom WAF rules for rWeb websites: **waf rweb custom** <*site-name*> ...
- Manager SSH public key: **manager ssh public** ...
- Manager SSH public key: **manager ssh private** ...
- Server TLS certificates: **tls server** (**load** | **save**) <*tls-id*> **certificate** ...
- Server TLS private keys: **tls server** (**load** | **save**) <*tls-id*> **key** ...
- Client TLS certificates: **tls client** (**load** | **save**) <*tls-id*> **certificate** ...
- Client TLS base 64 encoded of the PKCS12: **tls client save** <*tls-id*> **pfx** ...
- Client TLS clear password: **tls client save** <*tls-id*> **password**...

• Client TLS PKCS12 (certificate+key): **tls client save** <*tls-id*> **pkcs12**...

• Client TLS private keys: **tls client save** <*tls-id*> **key** ...

• The TLS system CA certificate: **tls ca system certificate** ...

• The TLS system CA private key: **tls ca system key** ...

• TLS third party CA certificates: **tls ca third load** <*ca-id*> ...

• Antivirus white list signatures: **antivirus whitelist signature** ...

Please note that there are some limitations when using the **file** command. Limitations are as follows:

• Administrator public SSH keys can only be loaded.

• SNMP client certificates can only be loaded.

• Client TLS components other than certificates can only be saved.

• TLS third party certificate can only be loaded.

• The system's CA certificate can only be loaded in PEM format.

• URL list are not loaded or saved.

• Antivirus white list can only be loaded.

• Antivirus signatures are not loaded or saved.

In the third [3] usage form, the **clear** keyword allows you to clear all downloaded files into the system. Loading and saving files are performed in background. You can use the **report** keyword to print a report on the last operations on files.

Finally the **exchange** keyword allows you to display a report on the active file exchanges (files that are being exchanged).

When loading, saving or clearing files, you can optionally specify file types that you want to exclude from the loading or saving operation. File types that you can exclude form the operation and associated keyword are as follows:

• **antivirus.whitelist.signature**: antivirus white list signatures.

• **admin.snmp.certificate**: SNMP client certificates.

• **admin.ssh.key**: administrator public SSH keys.

• **tls.ca.system**: system CA certificate and private key.

• **tls.ca.third**: third party CA certificates.

• **tls.client**: client certificates and other associated components.

• **tls.server**: server certificates and other associated components.

• **waf.rweb.custom**: custom WAF rules for rWeb websites.

The fourth [4] usage form allows to delete a file on a file server when the used protocol supports it. This usage form can be very helpful to delete critical files such as client certificate passwords on a file server.

FILE NAMING POLICY

As no file names are specified when loading or saving files, a simple naming rule is used to identify loaded or saved files . The rule is this: each file is named using the sequence of the command name, keywords and the identifier that are normally used to load or save that file separated by the dot (".") character. For instance the pkcs12 file associated to a Client TLS identified by the *myId* identifier is named: *tls.client.pkcs12.myId*. In the same way the system CA private key is named *tls.ca.system.key*.

SEE ALSO

apply (1) **access** (1) **admin** (1) **antivirus** (1) **conf** (1) **job** (1) **system** (1) **tls** (1) **urllist** (1) **waf** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

firewall

NAME

firewall - Configure the firewall

SYNOPSIS

[1] **firewall** [(**external** | **web** | **rweb** | **antivirus** | **admin** | **mon** | **file** | **peer** | **auxiliary** | **vpnipsec**) [(**add** | **add:**<*rule-name*> | **insert:**<*rule-name*>) <*rule-name*> (**deny** | **allow**) [<*protocol*> [<*src-ip*> [/<*mask-prefix*>] [<*destination-zone*> [<*dst-ip*>[/<*mask-prefix*>] [<*dst-port1*>[**:**<*dst-port2*>] [(<*src-nat-ip*> | **nil**) [<*dst-nat-ip*> [<*dst-pat-port*>]]]]]]]]]

[2] **firewall** [(**external** | **web** | **rweb** | **antivirus** | **admin** | **mon** | **file** | **peer** | **auxiliary** | **vpnipsec**) [**move:(-**|**+**)<*rule-name*> <*rule-name*> | **move:**<*position*>]]

[3] **firewall** [(**external** | **web** | **rweb** | **antivirus** | **admin** | **mon** | **file** | **peer** | **auxiliary** | **vpnipsec**) [**del** <*rule-name*>]]

[4] **firewall** [(**external** | **web** | **rweb** | **antivirus** | **admin** | **mon** | **file** | **peer** | **auxiliary** | **vpnipsec**) [**raz**]]

[5] **firewall** [(**external** | **web** | **rweb** | **antivirus** | **admin** | **mon** | **file** | **peer** | **auxiliary** | **vpnipsec**) [((**on** | **off**) <*rule-name*>)+]]

[6] **firewall dos** [(**tcpflood** | **udpflood** | **webflood** | **rwebflood**) [**default** | **set** <*limit-per-second*> [<*limit-burst*>]]]

[7] **firewall dos** [(**piptcp** | **pipweb** | **piprweb** | **pipdns** | **pipocsp**) [**default** | **set** <*max-limit*>]]

DESCRIPTION

When the modes **router** and **firewall** are activated (see the command **mode**), the appliance acts as a stateful firewall filtering IP traffic crossing the system. The **firewall** command allows you to manage firewall rules. Please note that the command **firewall** allows you to filter traffic crossing the appliance but not destined to the appliance itself. To filter traffic destined to the appliance itself use the command **access**.

To the appliance, networks are divided to four main zones: the **external**, the **internal**, the **auxiliary** and the **vpnipsec** zones. Each zone is connected to the appliance via a network interface such that the external zone is connected via the **external** network interface and the auxiliary zone is connected via the **auxiliary** network interface. As the internal network interface may be configured with tagged vlans (see the command **mode**), possible internal zones are **web**, **rweb**, **antivirus**, **admin**, **mon**, **file** and **peer**. When the appliance works without tagged VLANs, these zones are not differentiated (and setting a firewall rule for one of them is applied to the others). When the IPsec VPN mode is activated (mode **vpnipsec on**), the **vpnipsec** zone refers to private networks inside connected IPsec VPN tunnels.

The firewall filters the traffic according to the incoming zone from which a new connection is sent. Each firewall rule is attached to a zone and can **allow** or **deny** incoming new connections from that zone and outgoing to another zones (<*destination-zone*>). A firewall rule should specify the source IP address, the destination IP address and possibly the destination port number. Optionally the rule may specify to NAT (Network Address Translation) the source and/or destination IP address of the traffic. In case of a NAT for the destination IP address the destination port may also be translated to another port number. The translation of destination port number is called PAT (Port Address Translation).

It is IMPORTANT to note that in rules, destination and source **NAT** are always applied **AFTER** the filtering.

Default rules are applied when no rules is specified for a zone. When at least one rule is specified for a given zone, any traffic other than those specified by that rule are denied by default and there is no need to add a deny rule at the end of a rule set. Default rules are base on the principals below:

• New connections incoming from the **external** zone and destined to the **internal**, **auxiliary** and **vpnipsec** zones are denied.

• New connections incoming from the **internal** zone (the **web** zone only if the **vlan** mode is activated) and destined to other zones are allowed.

• New connections incoming from the auxiliary zone are denied by default (to allow traffic you should explicitly specify rules).

• If the **vlan** mode is activated, new connections incoming from the **rweb**, **antivirus**, **admin**, **mon**, **file** and **peer** zones are denied by default. Otherwise, they are all considered as being the **internal** zone and therefore follow the principals applied to the **internal** zone.

• New connections incoming from the **vpnipsec** zone and destined to the **internal** zone (the **web** zone only if the **vlan** mode is activated) are allowed. Incoming connections from the **vpnipsec** zone and destined to other zones are denied by default.

The first [1] usage form allows you to add or insert a firewall rule to a firewall rule set. There are as many rule sets as there are network interfaces. Each firewall rule is identified by an identifier (<*rule-name*>) in a rule set. A <*rule-name*> must begin with an alpha character and may contains alpha numeric characters as well as the characters "_" and "-". To add a rule at the end of a all rules in a set, use the keyword **add**. To add a rule after a given rule, use the keyword **add:** followed by the rule name after which the new rule have to be inserted. To insert a rule before a given rule, use the keyword **insert:** followed by the rule name before which the new rule have to be inserted.

To allow a traffic use the keyword **allow**. To deny a traffic use the keyword **deny**.

Supported protocols are as follows:

• **ah** (IPsec AH - IP Authentication Header)

• **esp** (IPsec ESP - IP Encapsulating Security Payload)

• **etherip** (Ethernet within IP encapsulation)

• **fc** (Fiber Channel)

• **ftp_active** (Active File Transfer Protocol)

• **ftp_passive** (Passive File Transfer Protocol)

• **ftp_trivial** (TFTP or Trivial File Transfer Protocol)

• **sip** (Session Initiation Protocol for VoIP)

• **gre** (GRE tunneling)

• **icmp** (Internet Control Message Protocol - Ping)

• **igmp** (Internet Group Management Protocol - Multicast)

• **ipv6** (Internet Protocol Version 6)

• **mtp** (Multicast Transport Protocol)

• **ospfigp** (Open Shortest Path First IGP)

• **tcp** (Transmission Control Protocol)

• **tlsp** (Transport Layer Security Protocol)

• **udp** (User Datagram Protocol)

• **visa** (VISA Protocol)

The source or destination may be a single IP address or an IP address range using the CIDR (Classless Inter-Domain Routing) notation. CIDR notation uses a combination of an IP address and its associated network mask prefix separated with the '/' character. The network mask prefix is the number of (leftmost) '1' bits in the mask. For example 10.0.10.0/24 is a CIDR notation where the network mask 255.255.255.0 is applied to the 10.0.10.0 network. This notation represents the IP address range 10.0.10.0 - 10.0.10.255. The port specification may be a unique port number or a port range. A port range is specified by giving two port numbers separated by a colon (*:*) character (for instance *7510:7529*). The keyword **any** can be used to specify an undefined protocol, IP address or port number.

The second [2] usage form allows you to move a rule from one position to another in a firewall rule set. To move a rule before or after another denoted rule use the keyword **move:** followed by the sign **-** (for before) or **+** (for after), the rule name of the denoted rule and the rule name of the rule to move. Please note that white spaces are not allowed between the keyword **move:**, the signs **-** or **+** and the rule name of the denoted rule. To move a rule to an absolute position use the **move:** followed by the position number

and the rule name of the rule to move (the first position is the position number 1). Please note that white spaces are not allowed between the keyword **move:** and the position number.

The third [3] usage form allows you to delete a firewall rule identified by a *<rule-name>* in a rule set. The fourth [4] usage form allows you to erase all firewall rules in a rule set.

The fifth [5] usage allow you to activate or deactivate a firewall rule in a rule set. You can activate or deactivate several rules at the same time be giving several rule name preceded by **on** or **off**.

There are a bunch of built-in firewall rules that protect against known attacks such as Denial of Service (DoS) attacks. Built-in firewall rules and the tags that identify them in logs are described in the **INTERNAL RULES** section below. Regarding DoS attacks, it is important to properly tune the system in order to distinguish between high loads and attacks. During the OS installation, DoS limits are automatically configured according to the number of supported users. Usage forms [6] and [7] allow you to modify those limits if necessary. To modify the limits for a given DoS attack, use the keyword **dos** followed by the DoS attack name, the keyword **set** and the required limits. To apply default values, use the **default** keyword instead of fBset without any further arguments.

DoS attack for which limits can be configured are given below:

• **tcpflood**: protects against TCP SYN and RST flood attacks routed via the system (not destined to services running on the system). Two limit values could be specified for this DoS type: *<limit-per-second>* and *<limit-burst>*. The first value is mandatory and specifies the number of allowed TCP SYN/RST per second. The second value is optional and specifies the maximum number of TCP SYN/RST per second to reach before beginning to block TCP SYN/RST connections.

• **udpflood**: protects against UDP flood attacks routed via the system (not destined to services running on the system). Two limit values could be specified for this DoS type: *<limit-per-second>* and *<limit-burst>*. The first value is mandatory and specifies the number of allowed UDP requests per second. The second value is optional and specifies the maximum number of UDP requests per second to reach before beginning to block requests.

• **webflood**: protects against TCP SYN and RST flood and UDP flood attacks destined to services running on the system except the reverse web service (see the **rweb** command). Two limit values could be specified for this DoS type: *<limit-per-second>* and *<limit-burst>*. The first value is mandatory and specifies the number of allowed TCP SYN/RST or UDP requests per second. The second value is optional and specifies the maximum number of TCP SYN/RST or UDP requests per second to reach before beginning to block TCP SYN/RST connections or UDP requests. Even if the name of this DoS type includes the **web** word, it concerns non only Web services such as the forwarding Web proxy but also all other services used by protected end-users such as the embedded DNS and OCSP servers.

• **rwebflood**: protects against TCP SYN and RST flood attacks destined to the reverse web service (see the **rweb** command). Two limit values could be specified for this DoS type: *<limit-per-second>* and *<limit-burst>*. The first value is mandatory and specifies the number of allowed TCP SYN/RST per second. The second value is optional and specifies the maximum number of TCP SYN/RST per second to reach before beginning to block TCP SYN/RST connections. Setting this value to 0 allows you to disable this protection.

• **piptcp**: protects against an abusive number of TCP SYN incoming from the same source IP address and routed via the system. One limit value should be specified for this DoS type: *<max-limit>*. The *<max-limit>* is the maximum number of parallel TCP SYN incoming from the same source IP address. Setting this value to 0 allows you to disable this protection.

• **pipweb**: protects against an abusive number of new (TCP SYN) Web requests incoming from the same source IP address and destined to the system itself except the reverse proxy (see the **rweb** command). One limit value should be specified for this DoS type: *<max-limit>*. The *<max-limit>* is the maximum number of parallel TCP SYN incoming from the same source IP address. Setting this value to 0 allows you to disable this protection. Disabling this protection can be useful when the system is accessed by remote end-users having all the same IP address (for instance in a public cloud configuration).

• **piprweb**: protects against an abusive number of new (TCP SYN) requests on cloaked websites (rWeb websites) incoming from the same source IP address). One limit value should be specified for this DoS type: *<max-limit>*. The *<max-limit>* is the maximum number of parallel TCP SYN incoming from the same source IP address. Setting this value to 0 allows you to disable this protection. Disabling this protection can be useful when the system is accessed by remote end-users having all the same IP address.

• **pipdns**: protects against an abusive number of new (TCP SYN) DNS requests per source IP address. One limit value should be specified for this DoS type: *<max-limit>*. The *<max-limit>* is the maximum number of parallel TCP SYN incoming from the same source IP address. Setting this value to 0 allows you to disable this protection. Disabling this protection can be useful when the system is accessed by remote end-users having all the same IP address.

• **pipocsp**: protects against an abusive number of new (TCP SYN) OCSP requests per source IP address. One limit value should be specified for this DoS type: *<max-limit>*. The *<max-limit>* is the maximum number of parallel TCP SYN incoming from the same source IP address. Setting this value to 0 allows you to disable this protection. Disabling this protection can be useful when the system is accessed by remote end-users having all the same IP address.

If no <*limit-burst*> is specified, it is set to the same value as <*limit-per-second*>.

INTERNAL RULES

• **AllState**: TCP connection with all states set.

• **BruteForce**: SSH brute force attack.

• **ConLimit**: more than a specified number of TCP SYN from the same source IP (see the seventh usage form).

• **IllegalSyn**: synchronise packets with illegal state.

• **InvalidState**: TCP/UDP traffic with invalid states.

• **LoopbackIP**: 127.0.0.0/8 IP addresses.

• **MulticastIP**: multicast IP addresses other than 224.0.0.1 and VRRP in HA mode.

• **NullState**: TCP connection without any state.

• **Policy**: no defined rule matches found.

• **ReservedIP**: 240.0.0.0/4, 169.254.0.0/16 and 255.255.255.255/32 IP addresses.

• **ResetFlood**: TCP reset (RST) packet flooding (see the sixth usage form).

• **SelfSpoof**: spoofing of an IP address assigned to the appliance.

• **SmurfPing**: smurf DoS attack.

• **SynFlood**: TCP SYN (synchronise) packet flooding (see the sixth usage form).

• **TraceRoute**: route tracing.

• **UDPFlood**: UDP packet flooding (see the sixth usage form).

• **UpTime**: time left from the last reboot.

• **XMasTCP**: TCP connection with all states to detect the running OS.

• **ZeroIP**: 0.0.0.0/8 IP addresses.

SEE ALSO

access (1) **apply** (1) **mode** (1) **vlan** (1) **vpnipsec** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

guard

NAME

guard - Manage the URL guarding (filtering)

SYNOPSIS

[1] **guard [filter** [**ip** [**add** <*filter-name*> (**range** <*ip1*> [<*ip2*>] | **network** <*ip*> [<*network-mask*>]) | **del** <*filter-name*> | **raz**]]]

[2] **guard [filter [time** [**add** <*filter-name*> (**slot** <*[w-]hh:mm-hh:mm*> | **frame** <*yyyy/mm/dd-yyyy/mm/dd*> | **date** <*[yyyy]/[mm]/[dd][-hh:mm-hh:mm]*>) | **del** <*filter-name*> | **raz**]]]

[3] **guard [filter [ldap** [**add** <*filter-name*> '<*base-dn*>' <*login-attr*> '<*ldap-filter*>' | **del** <*filter-name*> | **raz**]]]

[4] **guard [policy** [**add** <*policy-name*> ((**ip** | **time** | **ldap**) <*filter-name*>)* | **del** <*policy-name*> | **raz**]]

[5] **guard [rule** [(**add** | **update** | **add:**<*policy-name*> | **insert:**<*policy-name*>) (<*policy-name*> | **default**) (**deny** | **allow**) (<*urllist-name*>)+ | **del** (<*policy-name*> | **default**) | **raz**]]

[6] **guard [rule** [**move:(-**|**+**)<*policy-name*> <*policy-name*> | **move:**<*position*>]]

[7] **guard [ip** [(**on** | **off**)]]

DESCRIPTION

The URL guarding (or filtering) allows you to have control over the Web browsing in your organisation. Two guarding methods are available: the blacklist guarding and the white list guarding. Guarding with blacklists denies access to predefined URL lists, allowing access to all other URLs. Guarding with white lists permits access to predefined URL lists only, blocking all other URLs.

The URL Filtering can be based on the source IP address, the access time and an LDAP request. To define a guard rule you need to create policies and filters first. A policy is the combination of several filters. A guard rule defines who (policies) can access what (URL lists).

The first [1] usage form allows you to define a filter based on source IP addresses of users use the keywords **filter ip add** followed by a filter name the keywords **range** or **network** and an IP specification. An IP **range** is defined by giving two IP addresses. So all IP addresses between the first and second IP addresses will match the defined filter. If the second IP address is omitted the filter will match the unique given IP address. A **network** is defined by giving a network IP and a network mask. If no network mask is given the value 255.255.255.0 will be used.

The second [2] usage form allows you to define a filter based on access time use the keywords **filter time add** followed by a filter name. Three types of access times are available: **slot**, **frame** and **date**. Using the keyword **slot** allows you to define a time slot. A time slot has the format *[w-]hh:mm-hh:mm* where *w* is an optional digit between 0 and 6 representing the day of the week (0 is Sunday and 6 is Saturday) and *hh:mm-hh:mm* represents a time slot between the first *hh:mm* and second *hh:mm* (*hh* is a number between 00 and 23 representing hours while *hh* is a number beween 00 and 59 representing minutes). Using the **frame** keyword allows you to define a date frame. A date frame has the format *yyyy/mm/dd-yyyy/mm/dd* where *yyyy*, *mm* and *dd* are numbers representing respectively the year (2000-2999), the month (01-12) and the day (01-31). Using the **date** keyword allows you to define special days. A date has the format *yyyy/mm/dd[-hh:mm-hh:mm]* where *yyyy*, *mm* and *dd* are numbers representing respectively the year (2000-2999), the month (01-12) and the day (01-31). If one of these numbers is omitted it will represent any value. For instance the date represented by "//01" represents the first day of every month. If the optional time slot part *[-hh:mm-hh:mm]* is given, the filter will define a time at the given date and time slot.

An LDAP filter allows you to define a guarding policy based on an LDAP request. This type of filter is only applicable if the authentication mode is activated and configured adequately (see the command **authenticate**). The third [3] usage form allows you to define a filter based on an LDAP request. To define

such a filter use the keyword **filter ldap add** followed by a filter name, the LDAP distinguished name of the target LDAP object, the LDAP attribute used to login users and an LDAP filter applied to returned objects. The LDAP server and bind configuration (if required) should be configured using the **authenticate** command. Because LDAP distinguished names and filters contains the character '=', they must be enclosed in (simple or double) quotation marks to avoid being interpreted by the shell.

A filter can be deleted using the keyword **del** followed by the filter name. Using the keyword **raz** allows you to erase all filters in a filter type (**ip time ldap**) list.

Once all required filters are defined you can combine them to create policies. In a combination of filters an OR logical operand is applied to filters of the same type while an AND logical operand is applied to filters of different types. The fourth [4] usage form allows you to manage policies. To create a policy use the keywords **policy add** followed by a policy name and a list of filters where each filter is represented by a filter type (**ip time ldap**) followed by a filter name. A policy can be deleted using the keyword **del** followed by the policy name. Using the keyword **raz** allows you to erase all policies. Note that deleting a filter removes that filter from the policies that use it. The system contains a default policy named **default** which contains no filters. Use this policy to define a default rule. A default rule is applied to users not caught by defined policies. If no default rule is defined, the default behaviour is to deny all accesses to all URLs.

Filter and policy names must begin with an alpha character and may contains alpha numeric characters as well as the characters "_", "-" and ".".

Finally you can define a guarding rule based on policies (who) and URL lists (what). Usage forms [5] and [6] allow you to manage guarding rules. To add a rule at the end of all rules use the keywords **rule add** followed by a policy name, the keyword **allow** or **deny** (depending on whether you want to create a blacklist guard or a white list guard) and a list of URL list name separated by a blank. A rule can be deleted using the keyword **del** followed by the associated policy name. Using the keyword **raz** allows you to erase all rules.

Please note that the order of guard rules is important as the system matches the first matching rule to allow or deny a URL for a client. To add a rule after a given rule, use the keyword **add:** followed by the rule name after which the new rule have to be inserted. To insert a rule before a given rule, use the keyword **insert:** followed by the rule name before which the new rule have to be inserted. Finally the keyword **update** allows you to modify a rule without changing its place in the rule list.

As an example completing the following commands allows you to define a policy applied to users with an IP address ranging from 172.18.2.10 to 172.18.2.100 who use the Web between 8:00AM and 5:00PM and belong the LDAP group "cn=worker,ou=groups,dc=example,dc=com". A guard URL list named WebMail is created to group sites that offer web-based email service. Then the defined policy is used to create a rule to deny the access to WebMail sites for those users.

guard filter add ip myNetwork **range** 172.18.2.10 172.18.2.100

guard filter add time myHours **slot** 08:00-17:00

guard filter add ldap myRequest 'cn=worker,ou=groups,dc=example,dc=com' memberUid 'objectclass=posixGroup'

guard policy add myPolicy ip myNetwork time myHours ldap myRequest

urllist add WebMail

urllist load create WebMail **ftp** ftp.cacheguard.net BL/WebMail

guard rule add myPolicy deny WebMail

apply

Note that in this example the ftp server "ftp.cacheguard.net" should have been defined as a trusted file server previously with the command**access**. Also that FTP server requires login/passowrd credentials that should be configured using the command **password**.

The sixth [6] usage form allows you to move a guard rule from one position to another in the list of guard rules. To move a rule before or after another denoted rule use the keyword **move:** followed by the sign **-** (for before) or **+** (for after), the rule name of the denoted rule and the rule name of the rule to move. Please note that white spaces are not allowed between the keyword **move:**, the signs **-** or **+** and the rule name of the denoted rule. To move a rule to an absolute position use the **move:** followed by the position number and the rule name of the rule to move (the first position is the position number 1). Please note that white spaces are not allowed between the keyword **move:** and the position number.

The seventh [7] usage form allow you to activate or deactivate the usage of IP addresses instead of a domain name by Web users (clients). To allow the usage of IP addresses, turn the guard IP off. To disallow the usage of IP addresses, turn the guard IP on. Please note that by disallowing the usage of IP address in URLs you can block the usage of networks such as TOR.

SEE ALSO

apply (1) **authenticate** (1) **mode** (1) **urllist** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

ha

NAME

ha - Manage the High Availability

SYNOPSIS

[1] **ha** [[**failover** | **active**] [**force**]]

DESCRIPTION

Without any arguments, this command displays the state of an appliance in HA mode. Possible states are as follows:

* **active**: the system is running as an active master or backup appliance.

* **failover**: the system has been put manually in this state or has failed for some reasons. Possible reasons are varied: hardware failures, network connectivity, software bugs...

To put manually the system in failover mode, use the keyword **failover**. To make an attempt to reactivate an HA system which is in failover mode use the keyword **active**. Without the optional argument **force** the user is invited to confirm the action. The optional argument **force** allows you to bypass this confirmation.

Pleas note that the **apply** command may activate a system in a failover state.

SEE ALSO

apply (1) **mode** (1) **vrrp** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

halt

NAME

halt - Halt the Operating System

SYNOPSIS

[1] **halt** [**force**]

DESCRIPTION

This command is used to shut down and then power off the system. It is only enabled for the *admin* user. During asynchronous operations (apply, log rotate...) this command is inoperable. You must wait until the end of these operations before calling the **halt** command (usually asynchronous commands have a **report** option that displays the execution statement).

Without the optional argument **force** the user is invited to confirm the halt action. The optional argument **force** allows you to bypass this confirmation.

Caution: It is highly recommended to use this command (or the Clean Power Button) to turn off the system. Turning the power supply directly off may damage your system.

SEE ALSO

reboot (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

help

NAME

help - Print command's usage and description

SYNOPSIS

[1] **help** [<*command-name*>]

DESCRIPTION

Without any argument the **help** command displays a brief description of all available commands (this manual). If a <*command-name*> is given as argument, the manual of the given command is displayed.

SYNTAX & TYPOGRAPHY

When typing commands in a terminal the <*TAB*> key allow you to complete the typing keyword and to display allowed arguments. Us it always to remember the syntax of commands.

In manuals, the syntax of command utilisation is given using some simple rules. When different choices are available the character **|** is used to separate choices. Parenthesis *(* and *)* are used to specify without any ambiguity the order of keywords and arguments. They are not part of the command. Optional arguments are given inside brackets *[* and *].*

Command names and keywords are given in **bold.** Arguments are in *italic.*

After reading an online command help, type "q" to quit the help command.

ALL COMMANDS

Here are a brief description of all available commands:

access - Manage remote accesses to the appliance

admin - Manage administration services and accesses

antivirus - Configure the antivirus

apply - Apply new settings

authenticate - Manage the Web access authentication

cache - Manage the persistent Web cache

cancel - Cancel new settings

clear - Clear the terminal screen

clock - Manage the internal clock's date & time

conf - Manage the whole configuration

countrylist - Display valid country codes

dhcp - Manage the DHCP server

dns - Display, add or delete DNS (Domain Name Service) servers

domainname - Set or get the local domain name

email - Configure the administrator email account and email addresses to use.

embedded - Manage embedded applications

end - Mark the end of a template or gateway configuration context

error - Display descriptions of error codes returned by commands

exit - Exit the administration login

file - Load, save or clear all files related to the configuration

firewall - Configure the firewall

guard - Manage the URL guarding (filtering)

ha - Manage the High Availability

halt - Halt the Operating System

help - Print command’s usage and description

history - Display the command history list

hostname - Set or get the hostname

information - Display descriptions of information codes returned by commands

ip - Manage IP addresses and routing configurations

job - Print a report on the current running operation in background

keyboard - Set the key map for the console port

ldap - Perform LDAP actions

license - Display the CacheGuard-OS License Agreement

link - Manage L2 network interfaces

log - Manage Logs

manager - Configure a manager system

mode - Manage general features and functions

ntp - Manage the NTP configuration

password - Manage passwords

peer - Manage connected peer Web proxies

ping - Send ICMP packets to network hosts

port - Manage built-in service network listening TCP/UDP ports

qos - Manage the network QoS (Quality of Service)

reboot - Reboot the system

register - Manage the appliance S/N and license key registration

rweb - Manage the reverse Web mode (reverse proxy)

setup - Performs a basic startup configuration

sslmediate - Manage the SSL mediation

system - Manage the operating system

timezone - Set or get the local time zone

timezonelist - Display valid time zone codes

tls - Manage TLS (SSL) certificates and other components

traceroute - Trace the route to a host

transaction - Manage a set of commands as a single transaction

transparent - Manage the transparent Web proxy

tweb - Manage the transparent Web proxy

urllist - Manage URL lists

usleep - Suspend the execution of the calling thread for microseconds

vlan - Configure 802.1q VLANs (Virtual LANs)

vpnipsec - Manage IPsec VPN tunnels and networks

vrrp - Manage the VRRP configuration in HA mode

waf - Configure the Web Application Firewall (WAF)

warning - Display descriptions of warning codes returned by commands

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

history

NAME

history - Display the command history list

SYNOPSIS

[1] **history**

DESCRIPTION

This command allows you to display the history of typed commands.

SEE ALSO

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

hostname

NAME

hostname - Set or get the hostname

SYNOPSIS

[1] **hostname** [<*name*>]

DESCRIPTION

This command is used to get or to set the host name. The argument <name> must be a valid host name. A valid host name is a string beginning with an alphanumeric character and containing a combination of alphanumeric characters, the dash ("-") and dot (".") characters.

SEE ALSO

apply (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

information

NAME

information - Display descriptions of information codes returned by commands

SYNOPSIS

[1] **information** [<*information-code*>]

DESCRIPTION

Use this command to display the description of all information codes return by commands. You can also display the description of a particular information by giving its code (<*information-code*>). This command can be very useful if for any reasons displayed informations are not fully displayed by commands.

SEE ALSO

manager (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

ip

NAME

ip - Manage IP addresses and routing configurations

SYNOPSIS

[1] **ip** [(**internal** [**.**<*vlan-id*] | **external** | **auxiliary**) [<*ip*> [<*network-mask*>]]]

[2] **ip route** [(**add** | **add:**<*network-ip*>/<*network-mask*>:<*gateway-ip*> | **insert:**<*network-ip*>/<*network-mask*>:<*gateway-ip*>) <*network-ip*> <*network-mask*> <*gateway-ip*> [<*balance-weight*> [<*pinged-gateway*>]]]

[3] **ip route** [**move:(-**|**+**)<*ip*>/<*network-mask*>:<*gateway-ip*> <*network-ip*> <*network-mask*> <*gateway-ip*> | **move:**<*position*>]

[4] **ip route** [**raz** | **del** <*network-ip*> <*network-mask*> <*gateway-ip*>]

[5] **ip via** [**raz** | (**add** | **del**) <*gateway-ip*> (**master backup**) [<*priority*>]]

[6] **ip name** [**raz** | (**add** | **del**) <*name*> <*ip*>]

[7] **ip vlan**

[8] **ip neighbour**

DESCRIPTION

The first [1] usage form is used to get or set the **internal**, **external** and **auxiliary** interface IP addresses. The internal interface is mainly used to connect clients or backend Web servers to the appliance and can be considered as a trusted and secure interface. The external interface is used to connect the system to the internet thus considered as an untrusted or non-secure interface. You can use the auxiliary interface for your specific needs (for instance to implement a DMZ or a Back Office zone). If a VLAN identifier is given for the internal interface, the command is applied to the 802.1q pseudo interface in the given VLAN. If no <*network-mask*> is specified the default network mask *255.255.255.0* is used.

The second [2] usage form allows you to add static IP routes. A static IP route is defined by the couple <*network-ip*> <*network-mask*> and followed by a gateway IP address. To set the default route use the keyword **default** instead of the couple <*network-ip*> <*network-mask*> . It is also possible to define the default route by giving the couple *0.0.0.0 0.0.0.0* as the network address.

To add a route at the end all existing routes, use the keyword **add**. To add a route after a given route, use the keyword **add:** followed by the route after which the new route should be inserted. To insert a route before a given route, use the keyword **insert:** followed by the route before which the new rule should be inserted. When adding after or inserting before a give route, the given route should be specified as a network IP address and a network mask separated by the character "/" followed by the character ":" and a gateway IP address.

The third [3] usage form allows you to move an existing route from one position to another in the routing table. To move a route before or after a given route use the keyword **move:** followed by the sign **-** (for before) or **+** (for after), the given route (<*network-ip*>/<*network-mask*>:<*gateway-ip*>) and the route to move (<*network-ip*> <*network-mask*> <*gateway-ip*>). Please note that white spaces are not allowed between the keyword **move:**, the signs **-** or **+** and the given route. To move a route to an absolute position use the **move:** followed by the position number and the route to move (the first position is the position number 1). Again white spaces are not allowed between the keyword **move:** and the position number.

The fourth [4] usage form allows you to delete a static route or completely erase the static route list.

If more than one gateway is specified for the same network to route, the traffic is balanced over those gateways. In this case the optional balance weight allows you to privilege a gateway over the others. The

balance weight is an integer between 0 and 100. The higher the weigh is, the higher the probability is that the traffic takes a path via a gateway. Specifying the value 0 means that the gateway is in standby and is only activated if all other gateways are unavailable. For the same network to route, only one gateway with the weigh value of 0 allowed. If no weigh is specified, the weigh is set to 50. In order to avoid IP spoofing, multiple gateways for the same network to route should belong to the same connected network. In case of the unavailability of a gateway, the related route is automatically removed from the routing table. If a failed route is reinstated, it is automatically added to the routing table and the overall traffic is balanced over it again.

All routing changes are notified if the monitoring mode (SNMP or other) is activated (see the **admin** command for further information). A gateway is marked unavailable if it can't reply to pings (ICMP protocol). In case where the optional <*pinged-gateway*> is specified, the gateway is marked unavailable if it can't route pings to that gateway. A pinged gateway (or server) should be a valid IP address (and can't be a name as ping results may be erroneous due to the non accessibility of DNS servers). It's a good compromise to specify as pinged gateway, the default gateway configured on the specified gateway (concerned by the route definition). In this case the unavailability of the WAN can be quickly detected while the connectivity test goes beyond the local gateway outage.

When the appliance is behind more than one external gateway (connected to the external interface) that source NAT the traffic with their own (distinct) IP addresses, you should explicitly specify the gateways from which external users can access exposed services via the external interface. For some essential services, you have the possibility to configure specific via gateways per source or destination IP addresses. Those services are the reverse proxy (**rweb** mode) and the IPsec VPN (**vpnipsec** mode). For other services the fifth [5] usage form allows you to globally specify via gateways to use. Via gateways can have two roles: the **master** role and the **backup** role. When all gateways are available, a master gateway with the highest priority is elected to route the traffic for technical services. The elected gateway is then activated for those services. Please note that at a given point, one and only one gateway is considered as active for technical services. In case of a failure on the active gateway, a backup gateway (with the highest priority) is then elected to be activated. In case where a faulty gateway becomes operational again, the process of electing and activating via gateways is performed again.

To add a master via gateway, use the keywords **via add** followed by the gateway IP address to use, the keyword **master** and optionally the priority associated to the specified gateway. To add a backup gateway, use the keyword **backup** instead of the **master** keyword. To delete a gateway, use the keyword **del** instead of **add**. To erase the list of all **via** gateways, use the keywords **raz**. The priority is a numeric value between 0 and 255. If no priority is specified, the priority is set to 110 for a master gateway and to 100 for a backup gateway.

In certain situations, it may help that the system overrides the name to IP resolutions provided by DNS servers. The sixth [6] usage form allows you to manage such overrides with a list of specific name to IP associations. To add an entry to that list, use the keywords **name add** followed by the name (<*name*>) to resolve and the specific IP address (<*ip*>) to associate to it. To delete an entry, use the keywords **name del** followed by the name (<*name*>) to remove from that list. To erase the list of specific name to IP associations use the keywords **name raz**. Please note that some internal names (like localhost) can't be override as well some IPs (like 127.0.0.1) can't be associated to a name.

The seventh [7] usage forms prints all IPs associated to 802.1q pseudo network interfaces.

The eighth [8] usage form (**ip neighbour**) allows you to display the ARP cache entries.

GATEWAY SELECTION AND ROUTE PERSISTENCE

In a multi gateway configuration, default routes are persistent while other routes are not. This means that when for a given connection a default gateway is selected (according to its weight), all traffic related to that connection will pass via the same initial selected gateway. Traffic passing via non default routes can sometimes pass via a gateway and sometimes via another gateway according to their associated weights. When the appliance is behind more than one external gateway (connected to the external interface) that source NAT the traffic with their own (distinct) IP addresses, listening services on the appliance (such as reverse websites) should be explicitly configured to use the right external public gateways. Related commands to configure such services are using the **via** keyword to define the master and backup gateways to use.

SEE ALSO

admin (1) **apply** (1) **domainname** (1) **hostname** (1) **link** (1) **rweb** (1) **vlan** (1) **vpnipsec** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

job

NAME

job - Print a report on the current running operation in background

SYNOPSIS

[1] **job**

DESCRIPTION

Some operations like the the apply or the cache clearing operations are executed in background. During those operations the configuration is locked and no other background operation can be executed. The **job** command without any arguments (first [1] usage form) prints a report on the current running operation in background.

SEE ALSO

antivirus (1) **apply** (1) **cache** (1) **file** (1) **ha** (1) **halt** (1) **log** (1) **reboot** (1) **system** (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

keyboard

NAME

keyboard - Set the key map for the console port

SYNOPSIS

[1] **keyboard**

DESCRIPTION

This command is used to set the key map for the console keyboard.

SEE ALSO

setup (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

ldap

NAME

ldap - Perform LDAP actions

SYNOPSIS

[1] **ldap search <filter>**

DESCRIPTION

You can use this command to search users or other objects on LDAP servers configured for the authentication (see the command **authenticate**). You can use this command to find relevant information about users in order to configure the authentication and/or LDAP filters used for the URL guarding. To use this command the LDAP authentication should be activated (see the command **mode**).

A mandatory LDAP filter should be specified with this command. Because an LDAP filter contains one or more equal characters, it must be enclosed in (simple or double) quotation marks to avoid being interpreted by the shell. For instance the filter '*objectClass=user*' display the list of all users.

SEE ALSO

authenticate (1) **guard** (1) **mode** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

license

NAME

license - Display the CacheGuard-OS License Agreement

START of CacheGuard-OS License Agreement

1. CacheGuard-OS LICENSE - Version 2.8

All CacheGuard Technologies (CACHEGUARD) software components and documentation whether on disk, in read-only memory, or on any other media (collectively, the ’CG-S’) are subject to the GNU General Public License v3. You should have received a copy of the GNU General Public License v3 along with CG-S. If not, refer to <http://www.gnu.org/licenses/>. The mere aggregation of CG-S and other open source software (as OSI definition) all together form CacheGuard-OS (collectively, the ’CG-OS’) which is licensed, not sold to you by CACHEGUARD. All copies of the CG-OS that you use are subject to this License.

2. PERMITTED USES and RESTRICTIONS

This License allows you to use each instance of CacheGuard-OS on **one** and only one logical machine (hardware or virtual machine) represented by a unique Serial Number and possibly one or more License Keys issued by CACHEGUARD for a predefined period of time and supported capacity. In case where CG-OS is installed as a forwarding *gateway* system, the number of users for which CG-OS is installed represents the supported capacity. In this case, you expressly acknowledge that theses users exist and are under your responsibility. In case where CG-OS is installed as a reverse *gateway* system (to protect your servers), the number of simultaneous users for which CG-OS is installed represents the supported capacity. In this case, you expressly acknowledge that those servers exist and are under your responsibility. In case where CG-OS is installed as a *manager* system, the number of managed gateways for which CG-OS is installed represents the supported capacity. THE SERIAL NUMBER AND LICENSE KEYS CANNOT BE CHANGED ONCE A COPY OS CG-OS IS INSTALLED AND REGISTRED.

By installing CG-OS on your own machine (hardware or virtual), your machine is bound to CG-OS for as long as CG-OS runs on that machine. The aggregation of your machine and CG-OS forms an appliance (collectively APPLIANCE). You consent that some technical information regarding that APPLIANCE (Ethernet address, installed number of users, version...) may be electronically transmitted to CACHEGUARD and registered for license controlling purposes.

CG-S is subject to the GPL v3. Hence you are allowed to edit and modify it for your personal use. If you modify CG-S or other Open Source Software components integrated into the CG-OS, you acknowledge: a) All your contributions are used at your own risk; b) No support will be granted by CACHEGUARD (we strongly recommend leaving the source code as is). You may not, except as permitted by applicable law, loan or create derivative commercial or non-commercial works from CG-OS. You are expressly not allowed to install CG-OS on a hardware of virtual machine and sell the resulted appliance and/or services without having obtained prior authorization from CACHEGUARD. In particular, reselling a hard drive image file of an installed APPLIANCE on a public or private cloud is subject to the present clause.

The License is valid only for the CG-OS version officially released and current at the date when is activated (the ’Initial Date’), and for the newer maintenance versions officially released within the subscription period starting on the Initial Date (the ’Initial Maintenance Period’). Your rights under this License will terminate automatically without notice from CACHEGUARD if you fail to comply with any term(s) of this License.

3. FUTURE VERSIONS

CACHEGUARD may from time to time develop new versions of CG-OS. Nothing in this Agreement shall obligate CACHEGUARD to develop any particular update or enhancement. CACHEGUARD may choose to implement new features and/or components of the CG-OS that require a new License. You are not obligated to use new licenses if you choose not to use those new features and/or components. Please refer to the "CacheGuard Support Services Agreement" for further information on this topic.

New versions of CG-OS may come with an update of the present License Agreement. By accepting the terms of the present License Agreement, you agree that you shall read and accept the terms of the updated License Agreement in case where you upgrade your installed CG-OS.

4. PERIODICALLY-LICENSED SERVICES

CG-OS operates as a service for a predefined supported capacity and a predefined period of time. Service renewal must be purchased from CACHEGUARD (or its official partners) to enable CG-OS after the end of each period of time.

5. DISCLAIMER of WARRANTY on CG-OS

You expressly acknowledge and agree that use of the CG-OS is at your sole risk. The CG-OS is provided ’AS IS’ and without warranty of any kind and CACHEGUARD EXPRESSLY DISCLAIMS ALL WARRANTIES AND/OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES AND/OR CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. CACHEGUARD DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE CG-OS WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE CG-OS WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN THE CG-OS WILL BE CORRECTED. FURTHERMORE, CACHEGUARD DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE CG-OS IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY CACHEGUARD OR A CACHEGUARD AUTHORIZED REPRESENTATIVE SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY. SHOULD THE CACHEGUARD PROVE DEFECTIVE, YOU (AND NOT CACHEGUARD OR A CACHEGUARD AUTHORIZED REPRESENTATIVE) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

6. LIMITATION of LIABILITY

UNDER NO CIRCUMSTANCES, INCLUDING NEGLIGENCE, SHALL CACHEGUARD BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO THIS LICENSE. SOME JURISDICTIONS DO NOT ALLOW THE LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THIS LIMITATION MAY NOT APPLY TO YOU. In no event shall CACHEGUARD’s total liability to you for all damages exceed the amount paid for this License.

7. Export Law Assurances

You may not use or otherwise export or reexport CG-OS except as authorized by the EU (European) law and the laws of the jurisdiction in which CG-OS were obtained. In particular, but without limitation, CG-OS may not be exported or reexported (i) into (or to a national or resident of) any EU embargoed country or (ii) to anyone on the consolidated list of persons, groups and entities subject to EU financial sanctions. By using CG-OS, you represent and warrant that you are not located in, under control of, or a national or resident of any such country or on any such list.

8. Controlling Law and Severability

If there is a local subsidiary of CACHEGUARD in the country in which CG-OS is purchased, then the local law in which the subsidiary sits shall govern this license. Otherwise, this License shall be governed by the laws of the FR (France). If for any reason a court of competent jurisdiction finds any provision, or portion thereof, to be unenforceable, the remainder of this License shall continue in full force and effect.

9. SSL mediation and Privacy

The SSL mediation feature integrated into CG-OS is to decrypt HTTPS traffic in order to cache and/or inspect its contents and possibly block unwanted contents. As HTTPS aims to give users privacy and security, its decrypting in the middle (before reaching the final client) may violate ethical norms and may be illegal in your jurisdiction. By activating the SSL mediation feature you expressly acknowledge that you understand what you are doing and accept that underlying consequences are fully under your responsibility.

10. Complete Agreement

This License constitutes the entire agreement between the parties with respect to the use of the CG-OS

and supersedes all prior or contemporaneous understandings regarding such subject matter. No amendment to or modification of this License will be binding unless in writing and signed by CACHEGUARD.

END of CacheGuard-OS License Agreement

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

link

NAME

link - Manage L2 network interfaces

SYNOPSIS

[1] **link** [**mac**]

[2] **link bond** [(**internal** | **external** | **auxiliary**) [**raz** | (**add** | **del**) <*eth-id*>]]

DESCRIPTION

The first [1] usage form is used to display some practical information (manufacturer, model...) related to physical NIC (Network Interface Cards). Use this command to identify physical NIC. The optional keyword "mac" allows you to display MAC address instead.

The second [2] usage form is used to get or set physical NIC bonding configurations. CacheGuard uses three logical network interfaces: the **internal** interface, the **external** interface and a third interface called **auxiliary**. Internal users and Web servers should access the internet through the internal interface while CacheGuard uses its external interface to access the internet. You can use the auxiliary interface for your specific needs (For instance to implement a DMZ or a Back Office zone).

Each logical network interface is associated to at least one physical interface. By default the **external** interface is associated to the *eth0* (ethernet interface "0") and the **internal** interface is associated to *eth1* (ethernet interface "1") . In order to increase the network availability it is possible to associate more than one physical interface to the same logical interface. To do this, CacheGuard uses NIC aggregation in "Active/Backup" mode.

The **bond** keyword allows you to add or delete physical NIC to logical NIC. When associating a list of physical NIC to a logical NIC, the first physical NIC is always used as the master (or active) NIC. All others are slave (or backup) interfaces. The argument <eth-id> is the physical NIC id.

For instance to add the ethernet NIC named *eth3* to the internal interface use the following command: **link bond internal add** *eth3*.

USB ETHERNET ADAPTERS

CacheGuard supports USB Ethernet adapters. To install your USB Ethernet adapter, just plug it in a USB port on your machine and wait for its detection and installation. If your USB Ethernet adapter is recognised by the system, you will be able to see it in the list of NIC on your machine using the present command. If more than one USB Ethernet adapters have to be installed, they should be installed one by one in order to let the system assign them a distinct a constant and unique Ethernet ID. To uninstall a USB Ethernet adapter, just unplug it from your machine and reboot your system.

SEE ALSO

apply (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

log

NAME

log - Manage Logs

SYNOPSIS

[1] **log** [**type** [(**web** | **rweb** | **firewall** | **guard** | **antivirus** | **avserver** | **waf**) [(**on** | **off**) [(**on** | **off**)]]]]

[2] **log** [**rotate** [**report**] | **force** [**wait**]]

[3] **log** [**save** (**web** | **rweb** | **firewall** | **guard** | **antivirus** | **avserver** | **waf** | **system**) <*serial*> (**ftp** | **sftp** | **tftp**) <*file-server*> <*file-path*>]

[4] **log syslog** [**raz** | (**add** | **del**) (**udp** | **tcp**) <*syslog-server*> [<*port*>] | **test**]

DESCRIPTION

Log reports gives you visibility into all traffic and key events happening in the system. To benefit from the logging the logging mode should be activated (see the command **mode**). Eight types of logs are generated by the appliance:

• **web** log: reports traffic managed by the forwarding and transparent proxy.

• **rweb** log: reports traffic managed by the reverse proxy.

• **guard** log: reports attempts to access non-authorized Web sites.

• **antivirus** log: reports attempts to access virus infected objects coming from the Web.

• **avserver** log: reports attempts to access virus infected objects coming from the external systems such as an MTA (Mail Transfer Agent).

• **waf** log: reports unauthorized requests blocked by the Web Application Firewall.

• **firewall** log: reports denied packets by the IP firewall.

• **system** log: reports low level system events (only for maintenance purpose).

Please note that an attempt to inject a virus on a Web server protected by the appliance (in reverse mode) is logged in the WAF log file and not in the antivirus log file (which is reserved for forwarding and transparent Web accesses).

All log types (except the **system** log) can be activated or deactivated. The first [1] usage form allows you to activated or deactivated log types. To activate a log type use the keyword **type** followed by the required log type specifier (**web**, **rweb**, **guard**, **antivirus**, **avserver**, **waf** or **firewall**) and the keyword **on**. To deactivate a log type use the keyword **off** instead. When a log type is activated, you can optionality activate of deactivate the remote logging for it (see the fourth usage form below). The final argument in this usage form allows you to activate (**on**) or deactivate (**off**) the remote logging (on syslog servers).

Without any arguments, this command displays a history of generated logs. Logs can be saved after their rotation and never when they are in use. To inspect live logs you can use the Web Auditing module (see the command **admin**).

The system keeps logs of *n* days of activity archived on separate files for each day. The number *n* is called the retention period and is set up during the appliance installation. Log rotation is normally an automatic daily operation. However it is always possible to force a log rotation by using the command **log rotate**. Use this if you want to download today's logs immediately without waiting for a daily log rotation. The log rotation operation runs in background unless the **force wait** keywords are used. Without the optional argument **force** the user is invited to confirm the log rotation operation. The optional **report** keyword allows you to display a report of the last manual log rotation.

Important notice: according to the number of users set up during the installation, an upper limit is fixed for log sizes and the required storage space is reserved for them. If a log report grows abnormally too fast a log rotation is forced without waiting for the daily log rotation. This prevents the system from being saturated. In the case of an advanced rotation an SNMP trap is sent to configure SNMP receivers.

A log rotation is an asynchronous operation (you are not blocked during its execution). Note that log rotation may fail if an **apply** operation or another asynchronous operation is running.

Accesses to non-authorized objects are logged in separate files while authorized forwarding and reverse Web accesses are saved each in a distinct file. A **web** or **rweb** log has the following format:

client-ip authuser [date] "*request*" *status bytes cache-status cache-peer-status* where:

• *client-ip*: the remote client IP address.

• *authuser*: the user name by which the user has authenticated himself (if the authentication mode is activated).

• *[date]*: date and time of the request in RFC3339 format (with the caveat that minutes and hours in the time offset are not separated by a colon).

• "*request*": the request line exactly as it came from the client.

• *status*: the HTTP status code returned to the client.

• *bytes*: the content-length of the object transferred (including headers).

• *cache-status*: the cache status (HIT, MISS...).

• *cache-peer-status*: peer cache status (HIT, MISS...).

• *User Agent*>: the Web browser type used by the user.

Please note that the two last information are not present in a **rweb** log.

The second [2] usage form allows you to save logs on a file server. The argument <*serial*> specifies the serial number of the saved log. The most recent log has the number 1. The older one has the number 2 and so on. If you want to save logs on a remote file server for archiving purpose, your backup cycle must be equal to or less than the logs retention period described below.

Logs can be saved on a trusted remote file server. Trusted file servers are defined with the command **access**. All logs are saved in gzip compressed files. The full file name including the extension *.gz* should be specified. A system log is saved in gzip tar (archive) format. The third [3] usage form allows you to save logs.

In addition to be locally saved, logs can be sent in real time to remote syslog servers. The fourth [4] usage form allows you to manage remote syslog servers. To add a remote syslog server use the keywords **syslog add** followed by its server IP address (or DNS name) and its listening port. To delete a remote syslog server use the **del** keyword instead of the **add** keyword. To erase the list of all syslog servers use the keywords **syslog raz**.

To check connectivities with syslog servers you can send testing syslog messages to all configured servers by using the keyword **test**. Please note that as with any other commands, the new configuration should be applied using the command **apply** before being able to send testing syslog messages.

SEE ALSO

access (1) **apply** (1) **mode** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

manager

NAME

manager - Configure a manager system

SYNOPSIS

[1] **manager ssh** [(**fingerprint** | **generate** | **show** | (**load** | **save**) (**private public**) (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>*)]

[2] **manager template** [**raz** | (**add** *<template-id>* [(**template** | **gateway**) *<source-id>*] | (**del** *<template-id>*)]

[3] **manager template** [**begin conf** *<template-id>*]

[4] **manager template** [**begin exec** [*<template-id>*]]

[5] **manager gateway** [**raz** | (**add** *<domain-id>* *<gateway-id>* *<gateway-ip>*) | (**del** *<gateway-id>*)]

[6] **manager gateway** [**begin conf** *<gateway-id>*]

[7] **manager gateway** [(**push** | **pull**) [*<domain-id>* [*<gateway-id>*]]]

[8] **manager gateway** [**begin exec** (**remote** | **local**) [*<domain-id>* [*<gateway-id>*]]]

[9] **manager template report exec**

[10] **manager gateway report** (**push** | **pull** | **exec** (**remote** | **local**))

[11] **manager sync** [(**role** [**alone** | **master** | **slave**]) | (**peer** [*<peer-ip>* ["*<peer-manager-public-ssh-key>*"]]) | **report**]

DESCRIPTION

This command is only available on a system installed as a *Manager* system (as opposed to a system installed as a *Gateway* system). A *Manager* system allows you to configure remote *Gateway* systems without having to directly connect to them.

A manager uses the SSH protocol to connect to remote gateways. In order to be automatically authenticated by remote gateways, a manger uses SSH private/public key pairs. In this operation mode, the manager's SSH public key should be exported to remote gateways using the **access** command on remote gateways. The first [1] usage form allows you to:

• Show the fingerprint of the active public RSA key used to connect to *Gateway* systems.

• Regenerate that RSA key.

• Show the active RSA key.

• Load or save the managers' SSH public and private SSH keys.

If you are managing a group of gateways that have almost the same configuration, you will notice that you have to execute the same command as many times as you have gateways in that group. To avoid that, you can use templates. A template is a set of commands forming a generic configuration that you can apply to a group of gateways. This ways, you can quickly configure a group of gateways that differ little by just executing commands that make differences.

Configuring a group of managed gateways with the aid of a template is done in four steps:

• First you create a template identified by a *<template-id>* and you configure it as if it was a gateway.

• Secondly you enrol (add) remote gateways on the manager and pull their current configuration from remote gateways.

• Then you can configure gateways by applying the template to them. At this step you can optionally make specific configuration for each gateway.

• And finally you can push gateway configurations made on the manager to remote gateways.

Please note that unlike most commands that require an apply to take effect (by using the **apply** command), the following operations on a manager have an immediate effect:

• Adding and deleting templates and managed gateways.

• Pushing and Pulling gateway configurations.

The second [2] usage form allows you to **add** a new template, delete (**del**) and existing template or erase (**raz**) the list of all existing templates. When adding a new template, you can set its configuration from a source template or gateway. To do so, you should use the keyword **template** or **gateway** followed by the source template or gateway identifier (<*source-id*>). If no source template or gateway is specified, a blank template based on the default factory configuration is created. Without any arguments, this usage form shows the list of all existing templates.

The third [3] usage form allows you to configure a template. To configure a template, you must specify the beginning of the template configuration by using the keywords **template begin conf** followed by the <*template-id*> to configure. Then you can use commands that you normally use to configure *Gateway* systems. When you finished configuring the template, you must use the **end** keyword to mark the end of the template configuration (and go back to the manager configuration level). Before ending a template configuration, please do not forget to validate it by using the **apply** command. Otherwise you will not be able to use it to configure a *Gateway* system.

The fourth [4] usage allows you to execute a set of commands on all templates or an identified template. You can use this usage form to modify settings related to entered commands on all templates without impacting other settings on templates.

Usage forms five to eight allow you to manage *Gateway* systems. The fifth [5] usage form allows you to enrol new remote gateways and **add** them to the list of managed gateways. During the gateway enrolment process, the *manager* should connect to the remote *gateway* (using the SSH protocol) in order to get some key information that uniquely identifies that remote *gateway. To allow this process working, the remote gateway* should accept connections from the *manager*. Please refer to the **access** command on remote *Gateway* systems to get information on how to accept connections from a *Manager* system. To add a gateway to the list of managed gateways, use the keywords **gateway add** followed by a domain group identifier (<*domain-id*>), a gateway identifier (<*gateway-id*>) and the remote gateway IP address. The gateway identifier uniquely identifies a gateway and it must begin with an alpha character and may contains alpha numeric characters as well as the characters "_" and "-". The domain group identifier allows you to place the enrolled gateway in a group identified by an identifier (<*domain-id*>). You can then apply operations like push or pull (see below) on a group of gateways that belong to a particular group (a group identifier syntax is the same as a gateway identifier syntax). To remove (and definitely forget) a gateway from the list of managed gateways use the keywords **gateway del** followed by the <*gateway-id*> to remove. To erase the list of all managed gateways use the keywords **gateway raz**.

The sixth [6] usage form allows you to make a configuration for an enrolled gateway. To configure a gateway, you must specify the beginning of the gateway configuration by using the keywords **gateway begin conf** followed by the identifier of the gateway (<*gateway-id*>) to configure. Then you can use almost all configuration commands to configure the gateway. In addition, you have the possibility to use the **conf template** <*template-id*> command to initialise the gateway configuration with settings in the specified template. When you finished configuring the gateway, you must use the **end** keyword to mark the end of the gateway configuration to go back to the manager configuration level. Before ending a gateway configuration, please do not forget to validate it by using the **apply** command. Otherwise you will not be able to push that configuration on the remote *Gateway* system.

The seventh [7] usage form allows you to pull configurations from or push configurations to remote gateways. It should be noted that to push or pull configurations to remote gateways, remote gateways should run the same OS version as the manager. To **push** a gateway configuration to a remote gateway, use the **gateway push** keywords followed by the domain and gateway identifiers (<*domain-id*> <*gateway-id*>) of that gateway. If no <*domain-id*> and <*gateway-id*> are specified, configuration of all gateways are pushed to remote gateways. If only a <*domain-id*> is specified, configuration of all gateways that belong to that <*domain-id*> are pushed to remote gateways. During a gateway **push** operation, the manager performs the following tasks in sequence and in the event of an error, the operation on the gateway in question is stopped.

• The local gateway configuration and all related files to that configuration are pushed to the remote gateway using SFTP.

• The **apply force** command is executed on the remote gateway (an **apply check** is automatically executed first).

When pushing a gateway configuration, you also push all files related to the configuration except the

following files:

• Private parts of TLS client objects (key, password...).

• All antivirus signatures files.

To pull the running configuration from a remote gateway use the keywords **gateway pull** followed by the domain and gateway identifiers (<*domain-id*> <*gateway-id*>) of that gateway. If no <*domain-id*> and <*gateway-id*> are specified, the configuration of all gateways are pulled. If only a <*domain-id*> is specified, the configuration of all gateways that belong to that <*domain-id*> are pulled. When pulling a gateway configuration, you also pull all files related to the configuration except the following files:

• URL list files (including **domains**, **urls** and **expressions**)

• All antivirus signatures files.

• The antivirus white list file.

The eighth [8] usage form allows you to execute a set of commands on managed gateways. You have the possibility to directly execute commands on remote gateways (without affecting gateway configurations stored on the *Manager* system) or execute commands on local gateway configurations stored on the *Manager* system. To mark the beginning of a remote execution session use the keywords **gateway begin exec remote** followed by the domain and gateway identifiers (<*domain-id*> <*gateway-id*>) of the target gateway. If no <*domain-id*> and <*gateway-id*> are specified, commands are executed on all enrolled gateways. If only a <*domain-id*> is specified, commands are executed on all gateways that belong to that <*domain-id*>.

Once the beginning of an execution session is marked, you can enter commands (one command per line) to execute on gateways. Entered commands are not executed until you mark the end of the execution session using the **end** keyword. If for any reasons you decide to do not execute specified commands, you can use the **end cancel** command (or alternatively press the CTRL+C keys). Please note that:

• A set of commands is executed in background and in parallel on specified remote gateways.

• Commands in a set are executed sequentially on a gateway and in case of an error on a command, following commands are ignored.

• As commands are executed in background, there is no possibility to read from the standard input. Hence, commands that require to read from the standard input return an error.

• Standard and error outputs are redirected to the execution report file (see the ninth usage form).

To execute commands on local gateway configurations stored on the *Manager* system, replace the **remote** keyword by the keyword **local**.

The ninth [9] and tenth [10] usage forms allow you to display a report on the last **push**, **pull** or **exec** operations.

Finally, the eleventh [11] usage form allows you to activate and configure the synchronisation mode between two peer managers. In this mode a peer manager should take the **master** role while the other should be configured as its **slave**. If you configure manager as a master manager, the other manager should be configured as a slave manager and vice-versa. When the synchronisation is activated, all modifications on a the master manager are automatically replicated to the slave manager. This way, is case of a failure on the master manager, the slave manager can be used as a backup manager. In order to preserve consistency between a master manager and a slave manger, operations like pulling a configuration from a remote gateways or modifying a template configuration are not allowed on a slave manager. To allow a slave manger to perform those operation its role should be modified to **master** or **alone**. However you can use a salve manager to execute commands on remote gateways.

Please note that both master and salves managers should be allowed on managed gateways. Master and slave managers on gateways are respectively referenced as **master** and **backup**. Automatic data replications between peer managers use the SSH protocol. That’s why the peer that acts as a slave manager should know the SSH public key of its master. In practice, each peer should know the SSH public key of the other to allow them to quickly swap their roles. It should also be noted that the peering relation between a master and a slave managers can only be established if both run the same OS version.

To activate the synchronisation mode on a peer manager, use the keywords **sync role** followed by the **master** or **slave** keywords. To disable the synchronisation mode use the keywords **sync role alone**. To allow peers to be synchronised, each peer should know the IP address and the SSH public key of the other. To set them on a peer, use the **sync peer** keywords followed by the IP address of the remote peer and its SSH public key.

On a master manager you can display a report on the latest synchronisation operation. To do so use the **sync report** keywords.

URL LISTS PUSH TO REMOTE GATEWAYS

Loaded full or update URL list files present on the manager at the time of the push operation are pushed to remote gateways. If URL lists are configured to be automatically updated on the manager, the **manager** command will always push the latest URL lists alongside other configurations to remote gateways. URL lists files can be full lists as well as updates depending on chosen configuration. Please refer to the**urllist** command for further information.

PUSH OF EXTENDED ANTIVIRUS SIGNATURES TO REMOTE GATEWAYS

Extended antivirus signatures can be loaded on a *Manager* system and automatically pushed to managed *Gateway* systems whenever they are updated on the manager. Please note that the automatic push of extended antivirus signatures can only be activated on a commercial installation of a manager system. Please refer to the **antivirus** command for further information.

SEE ALSO

access (1) **antivirus** (1) **apply** (1) **conf** (1) **end** (1) **error** (1) **file** (1) **information** (1) **urllist** (1) **warning** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

mode

NAME

mode - Manage general features and functions

SYNOPSIS

[1] **mode** [**router** | **dns** | **dhcp** | **snat** | (**tweb** | **transparent**) | **tnat** | **vlan** | **ftppassive** | **ha** | **qos**) [(**on** | **off**)]

[2] **mode** [((**forward** | **web**) | (**rweb** | **reverse**) | **anonymous** | **guard** | **waf** | **antivirus** | **sslmediate** | **firewall** | **authenticate** | **ocsp** | **vpnipsec** | **cache** | **compress** | **log**) [(**on** | **off**)]]

DESCRIPTION

This command is used to set or get general appliance modes. There are two mode categories: network related modes and feature related modes. Network related modes are listed in the first [1] usage form and feature modes in the second [2] usage form. To activate a mode just use the mode specifier keyword followed by the keyword **on**. To deactivate a mode use the keyword **off** instead. You can find below the description of each mode.

In a common implementation, the appliance is used as a gateway router to access the internet (or external insecure zones). In such a network topology all traffic is routed via the appliance. You can use the **router** keyword to set the router mode.

The appliance may act as a caching-only DNS (Domain Name Server). The **dns** keyword is used to turn on/off the access to the embedded DNS. Only clients located behind the internal and auxiliary interfaces can access the DNS.

The appliance may act as a DHCP server for connected devices to the internal network interface. The **dhcp** keyword is used to turn on/off the embedded DHCP server. Refer to the **dhcp** command to configure the DHCP server.

The **snat** mode is used to activate or deactivate the appliance's Source NAT (Network Address Translation) mode for traffic exchanged via the external network interface. When the **snat** mode is activated, the source IP address of all (including Web traffic) outgoing traffic from the external network interface are translated to the external IP address of the appliance (see the command **ip** to set the external IP address). When the **snat** mode is deactivated, traffic other than Web traffic outgoing from the external network interface use their real IP addresses. To deactivate the source NATing of Web traffic see the **tnat** mode below.

To isolate different types of traffic passing through the internal network interface you can activate the VLAN mode. The **vlan** keyword refers to the VLAN mode. When using VLANs, the real internal network interface can no longer be used. You can set VLANs using the **vlan** command.

When the appliance tries to connect to external FTP servers, it may use the passive or active FTP mode. The **ftppassive** mode is used to activate or deactivate the passive FTP mode. The **ftppassive** keyword refers to the passive FTP mode.

The High Availability mode provides continuity of service in case of a failures on an appliance. The HA mode requires two or more combined appliances to make a virtual appliance based on redundant appliances. To activate or deactivate the HA mode use the **ha** keyword. When the HA mode is activated, feature services listen on VRRP IPs (and not on real IPs). Refer to the **vrrp** command to configure VRRP IPs.

The Quality of Service mode allows you to share the available bandwidth between different types of traffic based on policies you configure. The **qos** keyword refers to the QoS mode. When the QoS mode is activated you can use the **qos** command to configure the QoS.

The appliance embeds a Web proxy that allows you to securely browse the Web. To activate or deactivate the embedded Web proxy, use the **web** (or **forward**) keyword. The Web proxy is only available to clients located in the internal area (behind the internal network interface). In this mode, clients are protected

against threats coming from the external zone (in front of the external network interface). The Web proxy can be used in explicit or transparent mode. In explicit mode, clients should configure their browsers to use the appliance as a proxy by specifying its internal network interface and web port (see the **ip** and **port** commands for furhter information).

In transparent mode, the proxy transparently intercepts Web traffic even if clients do not explicitly select to use it. To activate or deactivate the transparent mode use the **tweb** (or **transparent**) keyword. Please note that the transparent mode does not operate when the **authenticate** mode is activated. The **tnat** mode is used to activate or deactivate the appliance's Source NAT (Network Address Translation) mode for Web traffic (only when the embedded proxy is activated). When the **tnat** is activated, the source IP address of Web traffic outgoing traffic from the external network interface are translated to the external IP address of the appliance (see the command **ip** to set the external IP address). When the **tnat** is deactivated, Web traffic outgoing from the external network interface use their real IP addresses. To configure the source NATing of traffic other than Web traffic, see the **snat** mode above.

You can implement the appliance in front of your Web server/applications to protect them agains direct accesses. In this case, Web clients are located in the external zone (in front of the external network interface) while Web servers/applications are located in the internal zone (behind the internal network interface). This is the reverse of a the forwarding mode, thus the designation reverse web or simply **rweb** mode. In **rweb** mode, the appliance acts as a reverse proxy to which you can associate content filtering with the **waf** mode (see below).

The **anonymous** mode hides some HTTP headers to make requests and responses anonymous. Hidden headers are: "From", "Referer", "Server" and "Link".

The **guard** mode is used to allow or deny access to defined websites for Web users. The **guard** mode is based on black or white lists of domain names, URL or regular expressions (commonly named URL). See the command **guard** to manage the guard policies. The guarding feature is only available when the appliance is configured in forwarding proxy mode (**web** mode) and allows you to control access to requested URLs. To control the content of Web requests (GET and POST methods) in reverse mode (**rweb** mode) activate the **waf** mode (see below).

The **waf** keyword is used to turn on/off the Web Application Firewall used in reverse mode (**rweb** mode) to protect Web servers. When this mode is activated, the system inspects all inside requests and filters unwanted and/or malicious requests. See the command **waf** to manage the filtering policy.

The **antivirus** keyword is used to turn on/off the antivirus mode. In this mode, the system inspects all Web traffic in forwarding mode (**web** mode) and blocks malware objects (viruses , trojans, worms). You can also combine this mode with the **waf** and **rweb** modes to block all attempts to upload malware onto your protected web servers. See the command **antivirus** to manage the malware filtering policies. Note that activating the antivirus clears the persistent cache.

You can use the **sslmediate** keyword to activate or deactivate the SSL mediation. The SSL mediation feature allows you to decrypt HTTPS traffic at the gateway point in order to cache, inspect its contents and possibly block unwanted contents. When the SSL mediation mode is turned off the HTTP CONNECT method is used to establish point-to-point tunnels to connect Web users to HTTPS servers across the system. Without the SSL mediation the system fully respects Web users privacy without decrypting the content of HTTPS traffic. The downside of having the SSL mediation off is that as the HTTPS traffic is encrypted unwanted contents like viruses can reach Web users without giving the opportunity to the system to block it. Also because of the HTTPS protocol encrypted objects can't be cached by the system.

When the SSL mediation mode is turned on the system decrypts HTTPS traffic, inspects its content and re-encrypts it before forwarding to the final client. In the process of re-encrypting the traffic the system uses a dynamically generated SSL certificate signed by its own CA (Certificate Authority) certificate. In this case clients should trust that CA certificate by importing it into their Web browsers. The CA certificate of the system is available at : *http://<internal-ip-address>* (or *http://<web-ip-address>* if the **vlan** mode is activated). **CAUTION**: please note that as HTTPS aims to give users privacy and security, its decrypting in the middle (before reaching the final client) may violate ethical norms and should be used with caution.

By default, the appliance acts as a *state full* firewall allowing only those connections coming from the internal area (incoming from the internal network interface) and going to the external area (outgoing from the external network interface). In certain cases, you may want to deactivate the firewall mode but please note that the deactivation of the firewall mode exposes your infrastructure to networks attacks. To turn on/off the Firewall mode use the **firewall** keyword.

Web accesses may be controlled by an external authentication system. The keyword **authenticate** allows you to turn on/off this feature. When the authentication is activated, only authenticated Web users are allowed to access the Web. Note that the **authenticate** mode does not operate in **transparent** mode. See the command **authenticate** for further information.

The **ocsp** keyword is used to turn on/off the embedded OCSP server. OCSP stands for Online Certificate Status Protocol. It's a protocol used for obtaining the revocation status of an X.509 digital certificate. When this mode is activated, the system acts as an HTTP OCSP responder for certificates signed by the system's certificate CA. The OCSP server listens on the external IP address and the OCSP port configured with the command **port**. Further configurations can be set using the **tls** command.

The **vpnipsec** keyword is used to turn on/off the IPsec VPN server. VPN stands for Virtual Private Network and IPsec for Internet Protocol Security. An IPsec VPN allows you to authenticate and encrypt the packets of data between 2 networks over an IP network to provide secure encrypted communications. You can build a persistent IPsec VPN between 2 sites and/or allow remote workers to access your internal infrastructures via a IPsec VPN server. See the **vpnipsec** command for further information.

The **cache** keyword is used to turn on/off the caching mode. The caching mode saves browsed Web objects in an internal cache memory, allowing their use in future requests instead of looking for them on internet. This method allows you to save bandwidth and in some cases improves performance.

To save the internal bandwidth consumption the compress mode can be activated. This is especially interesting when clients and the appliance are connected using a low bandwidth WAN. Compression may reduce the size of an textual files (HTML, CSS, JavaScript... by 80%). Note this mode requires large CPU resources. Use the keyword **compress** to set the compression mode.

The appliance may log all allowed Web accesses (in forwarding or reverse mode) as well as denied accesses to unauthorized contents (virus, blacklisted URLS...). The **log** keyword allows you to turn on/off this feature. You can refer to the **log** command to configure the logging.

SEE ALSO

access (1) **antivirus** (1) **apply** (1) **authenticate** (1) **dhcp** (1) **ip** (1) **guard** (1) **log** (1) **peer** (1) **port** (1) **sslmediate** (1) **tls** (1) **transparent** (1) **vlan** (1) **vpnipsec** (1) **vrrp** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

ntp

NAME

ntp - Manage the NTP configuration

SYNOPSIS

[1] **ntp** [**raz** | (**add** | **del**) <*ntp-server*> | **update**]

DESCRIPTION

The internal clock can be maintained in sync with internet standard time servers called NTP servers. This command allows you to manage NTP servers for the system. To leverage the synchronism precision more than one NTP server can be configured.

To add an NTP server use the keyword **add** followed by the NTP server name or IP address. To remove one, use the keyword **del** instead. To erase all NTP servers, use the keyword **raz.**

In some cases the current system time may differ radically from the network time. This may happen for instance with a suspended virtual machine. In such a situation ntp takes a long time to synchronise the system time with the network time. To immediately update the system time, use the keyword **update**.

Like in any other configuration command, a new NTP server configuration is applied after invoking the command **apply**. If NTP servers are unreachable during the apply process, the initialisation of the NTP service fails and an error is generated, but the system will automatically continue its attempts to restart the NTP service as soon as NTP servers become reachable.

SEE ALSO

apply (1) **clock** (1) **timezone** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

password

NAME

password - Manage passwords

SYNOPSIS

[1] **password** [**console** | (**login** | **wadmin**) [<*current-password*> [<*new-password*>]]]

[2] **password file** [((**add** (**ftp** | **sftp**) <*file-server*> <*login*> [<*password*>]) | (**del** <*file-server*>) | **raz**)]

[3] **password ldap** [<*password*>]

[4] **password snmp** (**community** | **privacy**) [<*password*>]

[5] **password kerberos** [<*shared-password*>]

[6] **password email** [<*password*>]

DESCRIPTION

The first [1] usage form is used to update the administrator's authenticating token. To update the console or the SSH login password, use the keyword **console**. To update the Web GUI password use the keyword **wadmin**. To update both the console and Web GUI passwords use the keyword **login**. If no keyword is specified, the **login** keyword is selected by default. The length of a valid password should be at least 16 characters and less than 33. It is highly recommended to use a pass phrase that includes alphanumeric and special characters.

Please note that administrator passwords are not part of the configuration and thus are not saved when the configuration is saved.

The second [2] usage form is used to manage file server logins and passwords (see the **access** command).

The third [3] usage form is used to set the password to access LDAP servers. LDAP servers are used to authenticate users (see the **authenticate** and **mode** commands). This is useful only if access to LDAP servers is protected by a password. Classically this is the LDAP administrator's password. Do not confuse this password with passwords required by users for authentication.

To set the DN (Distinguished Name) of the top LDAP node which requires a password use the command **authenticate**.

The fourth [4] usage form is used to set the SNMP agent community password. The minimum SNMP password length is 8 characters. To set the SNMP v3 user name of the SNMP agent use the command **admin**.

The fifth [5] usage form allows you to set the shared password to use for the kerberos account associated to the present system. This setting is only used if the system is in an HA configuration (with two or more redundant nodes). Please note that in an HA configuration, the same password should be used on all HA nodes.

The sixth [6] usage form is used to set the password for the administrator email account. The administrator email account is used by the system to send some configuration files by email (see the **email** command).

In all cases if no password is specified, you will be prompted to enter the password in hidden mode. Note that to take effect, the **apply** command must be used after password modifications (except for administration passwords).

SEE ALSO

access (1) **admin** (1) **authenticate** (1) **apply** (1) **email** (1) **mode** (1) **waf** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

peer

NAME

peer - Manage connected peer Web proxies

SYNOPSIS

[1] **peer** [(**share** | **ha** | **previous**) [**add** <*ip*> [<*qos*>]]]

[2] **peer** [**next** [**add** <*ip*> [<*port-number*>]]]

[3] **peer** [(**share** | **ha** | **next** | **previous**) [**raz** | **del** <*ip*>]]

DESCRIPTION

A peer appliance is a remote Web proxy interacting with the local Web proxy. There are three types of peer:

* Share peer,

* HA (High Availability) peer

* Chained peers (next and previous)

A share peer is a remote proxy that is requested from the local proxy when it does not have a requested object (by a client) in its cache. Several share peers may be defined to share their caches. In this case, each peer must be configured adequately with regards to the others. Share peers work together in order to deliver Web object to end-users from their caches rather than searching them on the internet and hence optimise the bandwidth usage.

An HA (High Availability) peer is a specific share peer, the difference being that Web objects retrieved from a share peer are not cached on the requesting proxy while Web objects retrieved from a HA peer are cached. HA peers should be used when peers are configured in a HA mode. In this case, when a master HA node fails, the newly elected master contains all Web objects as in the failed system (see the **mode** and **vrrp** commands for further information).

Share and HA peers works in parallel to provide caching performances and availability to local clients. It is also possible to serialise (or chain) peers to provide caching performances and high availability to remote end-users. To chain a proxy with a peer, one must be configured as a next peer while the other is configured as a previous peer (to allow the other to query it). A next peer is queried when it receives a request from an end-user. When several peers are chained, each system can participate to secure and optimise the traffic. For instance one can be configured to cache the traffic while a second one performs traffic shaping and a third one filters the traffic.

It is important to note that when at least on previous peer is defined, only end-users (clients) that are explicitly defined with the command **access web** are allows to connect to the local web proxy (see the command **access**). Also, note that, next peers are applicable only if the **rweb** mode is not activated (see the command **mode**).

The first [1] usage form allows you to add a **share**, **ha** or **previous** peer to the system. To add a share peer, use the **share add** keywords followed by the internal IP address of the remote share peer. The optional <*qos*> value allows you to customise the shaping of the traffic exchanged with the added peer. The <*qos*> is a percentage of the total bandwidth allocated to peers by the **qos** command. It should be an integer between 1 and 100. If no <*qos*> is given, the value of 100% is used by default. To add an HA or previous peer, use the **ha** or **previous** keywords instead of the **share** keyword.

A peer always uses its external IP address to query the internet or next peers. That’s why previous peers should be added using their external IP addresses.

The second [2] usage form allows you to add a next peer. To add a next peer, use the **next add** keywords

followed by the internal IP address of the remote next peer. By default the system connect to the next peer using the proxy port (see the **port** command). If the next peer is listening on another port, the optional <*port-number*> argument can be used to specify that port.

The third [3] usage form allows you to delete a peer or completely erase a list of peers.

SSL MEDIATION LIMITATION

The current OS version does not support the SSL mediation between peers so HTTPS traffic use the HTTP CONNECT method to establish point-to-point tunnels to connect Web users to HTTPS servers across the system without passing by peers.

SEE ALSO

access (1) **apply** (1) **mode** (1) **port** (1) **qos** (1) **vrrp** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

ping

NAME

ping - Send ICMP packets to network hosts

SYNOPSIS

[1] **ping** (<*name*> | <*ip*>)

DESCRIPTION

Ping uses the ICMP protocol’s mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. Use this command to check the network connectivity and name resolution by CacheGuard.

SEE ALSO

traceroute (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

port

NAME

port - Manage built-in service network listening TCP/UDP ports

SYNOPSIS

[1] **port** [(**wadmin** | **proxy** | **thttp** | **thttps** | **antivirus** | **ocsp** | **isakmp** | **natt** | **httppeer** | **htcppeer** | **dhcp** | **waudit**) [*<port-number>*]]

DESCRIPTION

Use this command to get or set IP ports for built-in services. This command requires two arguments: a port type and a valid port number. Valid port types are given in the above synopsis. A valid port number is an integer between 1024 and 49151.

The **wadmin** port is associated to the administration Web GUI. The administration Web GUI allows you to configure the appliance with a Web browser using HTTPS. The administration Web GUI is reachable at the URL *https://<hostname>.<domainname>:<wadmin-port>* where *<hostname>*, *<domainname>* and *<wadmin-port>* are respectively the host name, the domain name and the **wadmin** port.

When the proxy is in forwarding mode (**mode web on**) clients (Mozilla, Netscape, Internet Explorer...) using the proxy must use the **proxy** port configured here for the following protocols: HTTP, SSL, and FTP. In transparent mode (**mode tweb on**) all Web traffic (destined to the port 80) are transparently caught by the proxy on its **thttp** (transparent http). When the SSL mediation mode is activated in transparent mode (see the commands **sslmediate** and **mode**), the **thttps** port is used to transparently intercept HTTPS traffic. Clients should not use explicitly the **thttp** and **thttps** ports.

When the antivirus is configured as a service open to external systems such as an MTA (Mail Transfer Agent), the **antivirus** configured here is used for communications between external clients and the integrated antivirus. See the commands **antivirus** and **access** for further information.

When the OCSP mode is activated (**mode ocsp on**) the embedded OCSP server is configured to listen on the system’s external IP address on the **ocsp** port configured here. See the commands **mode** and **tls** for further information.

When the IPsec VPN mode is activated (**mode vpnipsec on**) the IKE (Internet Key Exchange) server is configured to listen on the system’s external IP address on the **isakmp** (ISAKMP: Internet Security Association and Key Management Protocol) and **natt** (NAT Transversal) ports configured here. See the commands **mode** and **vpnipsec** for further information.

The **httppeer** and **htcppeer** ports are used for peer intercommunications. For Share and HA Peers, peer ports must be the same in all peers to work together. The port **httppeer** is used to connect to a Next Peer (in this case the Next Peer must have its **proxy** port set to the httppeer of its Previous Peer. The protocol associated to the **httppeer** is HTTP which stands for Hyper Text Transfer Protocol. The protocol associated to the **htcppeer** is HTCP which stands for Hyper Text Caching Protocol.

When two appliances act as DHCP failover peer servers for each other, the **dhcp** port configured here is used for communications between those peers.

The couple *<hostname>.<domainname>* must be resolved to the appliance administration IP address in your network. The administration IP address is the internal interface IP address or IP address set for the administration 802.1q pseudo device in **vlan** mode.

The administration Web GUI is only available when the **wadmin** administration mode is activated. See the command **admin.**

The **waudit** port is associated to the Web traffic auditing module (see the command **admin**).

Please note that all port numbers must be unique.

SEE ALSO

apply (1) **access** (1) **antivirus** (1) **admin** (1) **dhcp** (1) **domainname** (1) **hostname** (1) **ip** (1) **mode** (1) **rweb** (1) **sslmediate** (1) **vlan** (1) **vpnipsec** (1) **waf** (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

qos

NAME

qos - Manage the network QoS (Quality of Service)

SYNOPSIS

[1] **qos [bandwidth** [(**internal** | **external** | **auxiliary**) [[(**ingress** | **egress**)] <*bandwidth*>]]]

[2] **qos [shape** [(**web** | **antivirus** | **file** | **default**) [(**internal** | **external** | **auxiliary**) [(**ingress** | **egress**) [<*qos*>]]]]]

[3] **qos [shape** [**tweb** [(**internal** | **auxiliary**) [(**ingress** | **egress**) [<*qos*>]]]]]

[4] **qos [shape** [**rweb** [(**internal** | **external**) [(**ingress** | **egress**) [<*qos*>]]]]]

[(] **qos [shape** [**peer internal** [(**ingress** | **egress**) [<*qos*>]]]]

[6] **qos [shape** [**vpnipsec** [**external** [(**ingress** | **egress**) [<*qos*>]]]]]

[7] **qos [shape** [**router** [((**add** | **add:**<*rule-name*> | **insert:**<*rule-name*>) <*rule-name*> (**internal** | **external** | **auxiliary**) <*protocol*> <*src-ip*>[/<*mask-prefix*>] <*src-port*> <*ingress-qos*> <*dst-ip*>[/<*mask-prefix*>] <*dst-port*> <*egress-qos*> [<*dscp*>]) | (**del** <*rule-name*>) | **raz**]]]

[8] **qos [shape** [**router** [**move:(-**|**+**)<*rule-name*> <*rule-name*> | **move:**<*position*>]]]

[9] **qos [borrow** [(**internal** | **external** | **auxiliary**) [(**ingress** | **egress**) [**on** | **off**]]]]

[10] **qos [report** [(**all** | **admin** | **gateway** | **etc**) | (**web** | **tweb** | **rweb** | **peer** | **antivirus** | **file** | **router** | **default**)] [**diff**]]

DESCRIPTION

The QoS (Quality of Service) controller allows you to manage the network bandwidth used by different types of traffic exchanged with or via to the appliance. The bandwidth management is done using different technologies such as prioritising, shaping and scheduling the network traffic. All those technologies combined together allows you to protect your business critical traffic from being penalized by other traffic in your networks. To this end, network traffic are classified by category and each category or class is configured to receive the adequate network bandwith. The QoS controller is deactivated by default. To activate it use the command **mode**.

The QoS configuration is done at different levels. To begin with, the overall incoming (**ingress**) and outgoing (**egress**) bandwidth limits should be configured for each logical network interface. The bandwidth limits are normally set according to the maximum supported bandwidth by connected networks. The network traffic can then be shaped to limit the bandwidth usage for each class of traffic. In addition, concurrent traffic in the same class are placed in multiple queues that are scheduled to equitably receive and transmit data. This way extensive usages by some clients (users or machines) in a class do not penalise others.

We distinguish the following traffic categories and classes:

• Management traffic used to administrate and monitor the appliance itself: unlike other class of traffic, management traffic is not shaped but is directly queued and managed with the highest level of priority. This means that as long as there data in the management queue, no other traffic is treated. This QoS policy ensure that the appliance is always reachable and can be monitored and administrated even in case where the network is overloaded. The QoS can’t be modified for management traffic. The following traffic are classified as management traffic: SSH, Web GUI, SNMP and SysLog. Management traffic is placed in a class named **admin**.

• Technical low level and basic service traffic destined to orinitiated from the appliance: ARP, DHCP, ICMP,

VRRP, NTP, OCSP and LDAP traffic are all placed in this category with a limited bandwidth that is automatically configured by the system. The **etc** designate this class of traffic.

• Technical traffic exchanged with the appliance itself: 2 classes of traffic are in this category which are **file** and **peer**. The class **file** designates file exchange traffic such as URL blacklists downloads and backup traffic. The class **peer** refers to traffic with shared or HA peer appliances via the internal network interface (see the command **peer** to manage peer appliances).

• Web traffic treated by the appliance before being routed. We distinguish 3 classes of Web traffic: **web**, **tweb** and **rweb** traffic: **web** stands for Web traffic exchanged with clients using the appliance as an explicit forwarding proxy ; **tweb** stands for Web traffic transparently intercepted by the forwarding proxy ; **rweb** stands for Web traffic exchnaged between clients and Web servers via the appliance implemented as a reverse proxy. Web traffic includes HTTP on port 80, HTTPS on port 443 and DOMAIN (for name resolutions) on port 53.

• Traffic exchanged between the appliance used as an antivirus service and clients like an MTA (Mail Transfer Agent). This class of traffic is named **antivirus**.

• Traffic exchanged in IPsec VPN tunnels established between the appliance via its external interface and remotre peers. The class named **vpnipsec** designates this traffic. ESP, ISAKMP and IPsec NAT transversal (ESP encapsulation with UDP) traffic are placed in this category.

• Traffic that are just routed via the appliance without being intercepted or inspected. This class of traffic is only allowed when the router mode is activated (see the command **mode** to activate the router mode).

• Any other traffic which is not classified in one of the above traffic types is classified in a class called **default**.

The **qos** command uses the terms **ingress** and **egress** respectively for incoming traffic and outgoing traffic from a logical network interface. The *<qos>* value represents the shaping to apply to a traffic. It may be a percentage of the total bandwidth limit configured for a logical network interface or a bandwidth value expressed in kbps (kilo bit per second). If the *<qos>* value ends with the character '%' it is considered as a percentage (integer between 1 and 100). Otherwise it is considered as a kbps value. If you use percentages, real used values are always calculated to get values in kbps. In other words, values given in percentages are not dynamic and are not calculated whenever bandwidths are modified in runtime.

Please note that there is always a gap between the perceived bandwidth at an application level and the defined *<qos>* value. This gap is due to protocol headers and other technical data in the traffic. In practice perceived bandwidths are from 6% to 9% less than specified *<qos>* values.

The first [1] usage form allows you to define the total bandwidth limit for each logical network interface for incoming and outgoing traffic. All bandwidths are given in kbps.

The second to sixth [2][3][4][5][6] forms allow you to shape the traffic destined to the appliance itself and allocate different bandwidths to different types of traffic. Having defined global QoS with this command, a customisation is possible network by network with other commands like **access**, **transparent**, **rweb** and **peer**. Traffic shaping is subject to the following classification rules:

• If a network is declared to have **web** (forwarding) access and **transparent** access at the same time, the QoS defined for the transparent access (using the **transparent** command) has a higher priority than the QoS defined for the **web** access (using the **access web** command).

• Incoming IPsec VPN traffic via the external network interface pass twice in the QoS controller: first as an encrypted **vpnipsec** traffic type and then as an unencrypted and classified as other traffic types (**web**, **file**, ...).

• Reverse websites traffic (see the **rweb** command) requested by clients using the appliance as an explicit proxy are shaped as being **web** traffic (and not **rweb** traffic).

The seventh [7] usage form allows you to configure the appliance to shape routed traffic (not destined to the appliance itself). The traffic shaping for routed traffic is configured with QoS rules. A QoS rule is associated to a logical network interface and specifies the part of the total bandwidth to reserve according to the used protocol, the source and destination of a traffic. Logical network interfaces to which a QoS rule can be associated are: **internal**, **external**and **auxiliary**. You may notice that it's not possible to associate a QoS rule to the **vpnipsec** virtual network interface as the traffic shaping of encrypted egress traffic is not supported by the system. However, as a routed traffic via an IPsec VPN tunnel ends up to pass via a logical network interface, you can manage its shaping with a QoS rule associated to that logical network interface.

A traffic rule is uniquely identified by rule a name (*<rule-name>*). A valid rule name should begin with an alpha character and may contains alpha numeric characters as well as the characters "_" and "-". To add a QoS rule at the end of all rules, use the keyword **add**. To add a rule after a given rule, use the keyword **add:** followed by the rule name after which the new rule should be inserted. To insert a rule before a given rule, use the keyword **insert:** followed by the rule name before which the new rule is to be inserted.

Source and destination of a traffic are identified by their respective IP address and port. The keyword **any** can be used to specify an undefined protocol, IP address or port number. The default *<mask-prefix>* is 32

(to specify a single remote machine). Supported protocols are:

• **tcp** (Transmission Control Protocol)

• **udp** (User Datagram Protocol).

The *<ingress-qos>* specifies the shaping for forth traffic (from source to destination) that come in from the given network interface while the *<egress-qos>* specifies the shaping for back traffic (from destination to source) that go out from the given network interface.

Please note that the following convention should be considered when defining QoS rules:

• An ingress (incoming) traffic comes from the source and goes to the destination while the related egress (outgoing) traffic comes from the destination and goes to the source (in the opposite sense).

• As NAT in firewall rules operate after IP packets come into the appliance, QoS rules are applied to non NATed IPs for incoming traffic. For IP packets that go out from the appliance, the NAT is already done. Hence QoS rules are applied to NATed IPs.

Please note that when all *<qos>* values are expressed as percentages, there is no obligation to have a total of 100% even if this is a recommended configuration when no shaping rules are defined for routed traffic. When a *<qos>* is expressed as a kbps value, it should be less than or equal to the defined bandwidth limit for the given logical network interface. The command **apply** verifies the integrity of *<qos>* values configured here.

Finally when defining QoS rules for routed traffic, you have the possibility to set a DSCP value for IP packets. DSCP stands for Differentiated Services Code Point and is a 6-bits value in the IP header. If your appliance is connected to a network that supports the classification of traffic based on the DSCP filed you can set this value. Please note that the DSCP field is meaningful only if all network devices between the source and destination take it into consideration. For instance the DSCP field is meaningful in an MPLS network while it is useless on the internet. A valid DSCP value is an integer between 0 and 63.

The eighth [8] usage form allows you to move a rule from one position to another in the list of shaping rules for routed traffic. To move a rule before or after another denoted rule use the keyword **move:** followed by the sign **-** (for before) or **+** (for after), the rule name of the denoted rule and the rule name of the rule to move. Please note that white spaces are not allowed between the keyword **move:**, the signs **-** or **+** and the rule name of the denoted rule. To move a rule to an absolute position use the **move:** followed by the position number and the rule name of the rule to move (the first position is the position number 1). Please note that white spaces are not allowed between the keyword **move:** and the position number.

In a concurrent environment the *<qos>* limit may be configured to be surpassed when the load of other networks is under their configured *<qos>* limits. This mechanism is called borrowing (a traffic type borrows its available bandwidth to other traffic types). The ninth [9] usage form allows you to activated or deactivated the borrowing mechanism. Deactivating the borrowing allows you to affect strict bandwidth to a traffic type while its activation allows you to share the available bandwidth more flexibly.

The tenth [10] usage form allows you to display a report on the network traffic (in kilobit) managed by the QoS controller since the last QoS configuration modification. This can help you to test your QoS configuration and validate that the bandwidth is managed as expected. If a traffic type is specified, the report covers only that type of traffic. If the optional **diff** keyword is specified, the differential network traffic (in kilobit) since the last report display is displayed (instead of the network traffic since the last QoS configuration modification).

Traffic types that can be specified with this usage form are as follows:

• **all**: the total traffic exchanged by the appliance (includes traffic destined to or initiated from the appliance as well as traffic routed via the appliance).

• **admin**: administration and monitoring traffic destined to or initiated from the appliance itself.

• **gateway**: all traffic destined to or initiated from the appliance itself except administration and monitoring traffic.

• **etc**: technical low level and basic service traffic destined to or initiated from the appliance itself.

• **web**, **tweb**, **rweb**, **peer**, **antivirus**, **file**, **router**, **default** : these are traffic types that can be shaped (descriptions are given above in the **qos shape** usage form).

LIMITATIONS

The QoS controller classifies the network traffic according to the used protocol (TCP or UDP), the source/destination IP address and the source/destination port. In certain circumstances, the appliance may not shape the traffic as requested because of an ambiguity in the configuration. For instance if an external FTP server and an antivirus client (such as an MTA) share the same IP address, the traffic can be classified as antivirus traffic as well as file traffic. The reason is that both FTP and antivirus traffic use dynamic ports, thus creating ambiguity.

When the **vlan** mode is activated (see the **mode** and **vlan** commands), the DSCP is only set for the **web** VLAN.

SEE ALSO

access (1) **firewall** (1) **apply** (1) **mode** (1) **peer** (1) **port** (1) **rweb** (1) **transparent** (1) **vlan** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

reboot

NAME

reboot - Reboot the system

SYNOPSIS

[1] **reboot** [**force**]

DESCRIPTION

Use this command to reboot the system. Without the optional argument **force** the user is invited to confirm the reboot action. The optional argument **force** allows you to bypass this confirmation.

SEE ALSO

halt (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

register

NAME

register - Manage the appliance S/N and license key registration

SYNOPSIS

[1] **register** [**appliance** (**new** <*email*> [<*otp*>]) | (**old** [<*otp*>])]

[2] **register** [**license** <*license-key*>]

[3] **register** [**embedded** [(**bevypn**) [<*license-key*>]]]

[4] **register** [**clear**]

DESCRIPTION

The appliance registration process associates an appliance to a S/N (Serial Number) that uniquely identifies that appliance. The registering is free and allows you to purchase optional services and/or activate commercial installations. Note that a S/N is bound to the first MAC address of an appliance. Hence your MAC address should not be modified after having registered your appliance.

Without any arguments, the **register** command shows the appliance registration/subscription state and gives you useful information about the appliance registration/subscription. As many commands, to take effect, the register command requires that you apply it by using the **apply** command.

The first [1] usage form allows you to register an appliance in order to get its unique S/N. If you register an appliance for the first time, use the keywords **appliance new** followed by a registration email address and a registration OTP (One Time Password). To get a registration OTP please go to the menu option *[GENERAL] > [Main Settings] > [Registration & Subscription]* of the Web GUI, select "First Registration" and click on the "Get OTP" button. Alternatively you can get the registration URL by using the **register** command without any arguments. You must provide a valid and active email address to properly register the appliance. Otherwise the registration fails. In case where no OTP is specified, you will be prompted to enter the OTP in hidden mode.

If the appliance has been previously registered and for any reason (an OS reinstallation for instance) its S/N is no longer locally assigned to it, you can recover its S/N by re-registering it. To recover the S/N of an already registered appliance, use the keywords **appliance old** followed by a re-registration OTP. To get a re-registration OTP, please go to the menu option *[GENERAL] > [Main Settings] > [Registration]* of the Web GUI, select "Recover S/N" and click on the "Get OTP" button. Alternatively you can get the re-registration URL by using **register** command without any arguments. Please note that during the re-registration process the previously used registration email address is asked and the OTP is sent to that email. In case where no OTP is specified, you will be prompted to enter the OTP in hidden mode.

Please note that to register, the appliance should be connected to the internet and have access to the registration services (https://appliance.cacheguard.net and https://os.cacheguard.net) during the whole registration process.

For commercial installations, the registration and subscription is mandatory. After having registered and appliance and subscribed for a given period of time, a license key is assigned to the appliance. The license key is used to activate a subscription on an appliance for the first time. Following the initial subscription period, a subscription renewal is required to continue to use the appliance. In case where the renewal in time is omitted, the appliance enters into a suspended mode. In suspended mode, Web traffic via the Web gateway (forwarding Web proxy) is blocked and the system configuration is locked (can no longer be modified). To reactivate a suspended appliance, you should proceed with its subscription renewal. After getting the renewal confirmation, the appliance is automatically reactivated during the night following the renewal confirmation date. The reactivation can also be manually done by re-registering the appliance license key (see the second usage form below). To avoid any service interruption, it is highly recommended to renew a subscription before its expiry date.

The second [2] usage form allows you to activate a subscription on an appliance using a license key. To activate a subscription use the keywords **license** followed by the license key.

The third [3] usage form allows you to register license keys for embedded applications. Embedded applications are applications that are built on top of a CacheGuard appliance and run on the appliance itself. You can activate an embedded application using the **embedded** command but when you activate it for the first time, you need a license key that you should register using the **register** command. To register a license key for an embedded application, use the **embedded** keyword followed by the embedded application reference and its activation license key.

The fourth [4] usage form allows you to deregister an already registered appliance. You may need to deregister an appliance in case where you need to replace its first hardware NIC (Network Interface Card) and then need to get a new S/N for it. Please note that a S/N and a license key are valid on a unique machine only. Refer to the CacheGuard-OS License Agreement for further information (see the **license** command). Deregistering an appliance also deregister all registered embedded applications. To deregister an appliance, use the keyword **clear**.

MAC ADDRESS AND VIRTUAL MACHINES

A hardware NIC is identified by a MAC address that is universally unique and guarantees a successful appliance registration. If you install the OS on a VM (Virtual Machine), you should ensure that its first NIC is unique at least in your environment. If the chosen MAC address is already registered for a another appliance, the registration fails. In this case, we suggest that you modify your VM’s MAC address and try to register it again.

SEE ALSO

apply (1) **embedded** (1) **system** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

rweb

NAME

rweb - Manage the reverse Web mode (reverse proxy)

SYNOPSIS

[1] **rweb** [site [**raz** | **add** *<site-name>* [(**http** | **https** *<tls-id>*[:*<ca-id>*]) [*<public-ip>* [*<qos>*]]] | **del** *<site-name>* [(**http** | **https**) [*<ip>*]]]]

[2] **rweb** [**rhttp** [*<site-name>* [(**on** | **off**)]]]

[3] **rweb** [**host** [*<site-name>* [**raz** | (**add** (**rweb** | **vpnipsec** | **external**) (**http** | **https**) *<ip>* [*<port>* [*<balance-weight>* [*<qos>*]]]) | (**del** (**rweb** | **vpnipsec** | **external**) (**http** | **https**) *<ip>* [*<port>*])]]]

[4] **rweb** [**balancer** [*<site-name>* [(**robin** | **traffic** | **pending**) [**nosticky** | **sticky** [(**insert** | **use**) [*<cookie-name>*]]]]]]

[5] **rweb** [**via** [*<site-name>* *<public-ip>* [**raz** | (**add** | **del**) *<gateway-ip>* (**master backup**) [*<priority>*]]]]

[6] **rweb [standby** [*<site-name>* [**off** | **on** *<url>*]]]

DESCRIPTION

This command is used to configure the reverse Web (or reverse proxy) mode. The reverse Web proxy acts as a virtual Web server between Web clients and Web servers and allows you to do not expose real Web servers to Web clients. Websites for which the system acts as a reverse Web proxy, are called reverse websites and protected (or cloaked) real Web servers are called hosts in the system.

The first [1] usage form allows you to add, delete or erase websites. To add a website use the keyword **add** followed by the website name, the used protocol (**http** or **https**), a TLS server identifier (only for HTTPS) and the external IP address (accessed by Web clients) for the defined website. An optional final argument may be used to customise the QoS (Quality of Service) for the given reverse website. For HTTPS websites, you have the possibility to specify a TLS server identifier and optionally an intermediate CA certificate identifier separated by the colon (:) character.

To delete a website use the keyword **del** followed by the website name, the protocol and the associated external IP address. If no protocol and external IP address are given, it is assumed that the used protocol is HTTP and the IP address is the external IP address of the appliance. To erase all defined websites, use the keyword **raz**. Without any arguments, the first usage form displays the list of all reverse websites.

The website external IP address can be the external IP address of the appliance, an external VRRP IP address (in HA mode) or an IP address belonging to the external network. In the latter case the IP address is added to the external network interface device as an alias IP address. Distinct HTTP website names can always share the same external IP address but sharing the same external IP address by distinct HTTPS website names is only supported when the WAF mode is activated (see the **mode** command). If activating the WAF mode is not required, an alternative solution can be the usage of wildcard or SAN certificates for HTTPS websites that share the same external IP address. It is important to note that only Web clients that support TLS SNI (Server Name Indication) can properly access distinct HTTPS websites that share the same external IP address.

If the same website is configured to support both HTTP and HTTPS, all HTTP traffic will be redirected to HTTPS by default. In this case, it is important notice that in order to avoid endless redirection loops, real Web servers should not froward HTTP traffic to HTTPS in turn. In case where real Web servers are configured to redirect HTTP traffic HTTPS and that redirection can’t be deactivated, the HTTP to HTTPS redirection can be deactivated for a website (see the second usage form).

The *<qos>* value in this usage form represents the percentage of the total bandwidth defined for reverse

Web traffic allocated by the command **qos** (see the usage form **qos shape rweb external**). The <*qos*> value here should be an integer between 1 and 100. If no <*qos*> is given, the value of 100 is used by default. Please note that the QoS configured here is based on the given external IP address and protocol only (and not the given website name). So if two reverse websites share the same external IP address and protocol, the applied QoS will be the QoS configured first. To configure different QoS for distinct websites that share the same IP address and protocol, you have the possibility to associate them to distinct real Web servers and configure different QoS to access those real Web servers (see the third usage form).

A reverse website acts as a virtual website that forward its traffic to hosts (or real Web servers) using HTTP or HTTPS. For HTTPS reverse websites, if the used protocol to forward the traffic to hosts is HTTP, the appliance acts as an SSL offloader. In this case, as communications between the appliance and real hosts are in unencrypted format (HTTP), it is highly important to use an end to end secure network to interconnect them. In addition, real hosts should be placed in a network that benefit from a restricted access. Recommended implementation are as follows:

• Real hosts are placed in the internal network directly connected to the appliance. In this case, the appliance should be exclusively used as a reverse proxy (**web** and **web** are disabled) and the internal network should be exclusively used to connect real Web servers.

• The **vlan** mode is activated and real hosts are placed in the **rweb** VLAN directly connected to the appliance. In this case, the **rweb** VLAN should be exclusively used to connect real Web servers.

• The **vpnipsec** mode is activated, the IPsec VPN is in site to site mode (the IPsec VPN **access** mode is disabled) and real hosts are placed in a remote private network connected via a site to site IPsec VPN tunnel established with the appliance. In this case, that remote private network should be exclusively used to connect real hosts.

In any case, real hosts can only be accessed via the following network interfaces:

• The internal network interface (called **rweb** in this context).

• The **rweb** network interface in case where the **vlan** mode activated.

• The **vpnipsec** network interface in case where the **vpnipsec** mode is activated and the IPsec VPN is in site to site mode (the IPsec VPN **access** mode is disabled).

• The **external** network interface (with the limitation that real hosts associated to HTTPS reverse websites are configured to use HTTPS only).

The second [2] usage form allows you to activate or deactivate the HTTP to HTTPS redirection for reverse websites that support both HTTP to HTTPS protocols. To activate the HTTP to HTTPS redirection for reverse websites use the keyword **rhttp** followed by the reverse website name and the keyword **on**. To deactivate the redirection, use the keyword **off** instead. Without any arguments, the second usage form displays the state of HTTP to HTTPS redirection for all HTTPS reverse websites.

The third [3] usage form allows you to associate real hosts to virtual websites. To associate a host to a virtual website, give the website name followed by the keyword **add**, the network interface via which the host is accessed (**rweb**, **vpnipsec** or **external**), the used protocol (**http** or **https**), the IP address of the real host and the port number on which the Web application on that host is listening (80 for HTTP and 443 for HTTPS by default). If more than one host is associated to the same website, the website load is distributed over those hosts. In this case, the optional balance weight allows you to balance the load more or less on a host. The balance weight should be an integer between 1 and 100. By default this value is set to 50. Traffic bandwidths can also be customised for a host using the optional <*qos%*> parameter. The <*qos%*> value is a percentage of the ingress or egress bandwidth allocated to **rweb** traffic and should be an integer between 1 and 100. Ingress and egress bandwidth values to which the percentage is applied are as follows:

• For hosts accessed via the native **internal** network interface or the 802.1q pseudo network interface called **rweb** (in **vlan** mode), the ingress and egress bandwidths to consider are defined with the command usage form **qos shape rweb internal**.

• For hosts accessed via the external network interface, the ingress and egress bandwidths to consider are defined with the command usage form **qos shape rweb external**.

• For hosts accessed via the **vpnipsec** virtual network interface, the ingress bandwidth to consider is defined with the command usage form **qos shape vpnipsec external ingress**. The egress bandwidth for hosts accessed via the **vpnipsec** virtual network interface can’t be customised.

If no <*qos%*> is given, the value of 100% is used by default. If the same host represented by an IP address and port number is used for more than one reverse website name, the effective QoS will be the highest defined QoS.

Please note that a real host can exclusively be accessed via a unique network interface. This means that if a host is added with an IP address that already exists via a distinct network interface, the existing host is automatically removed. It is also important to note that real hosts should host website names to which they are associated and in case where the used protocol is HTTPS, they should use valid certificates. In this context, a valid certificate is a certificate signed by a CA (Certificate Authority) or by the system’s CA

certificate (see the **tls** command).

To delete a host use the keyword **del** followed by the network interface via which the host is accessed (**rweb** or **vpnipsec**), the used protocol (**http** | **https**), the IP address of that host and optionally the port number on which the Web application on that host is listening (80 for HTTP and 443 for HTTPS by default). To erase all hosts associated to a reverse website use the keyword **raz**. Without any arguments, the third usage form displays the list of all hosts.

The fourth [4] usage form allows you to configure the load balancing policy over hosts. To configure the load balancing policy for a given website you must specify a load balancing method and the stickiness of connections. Three load balancing methods are available:

• **robin**: load balance real Web servers in a round-robin fashion based on the number of requests.

• **traffic**:load balance real Web servers according to the generated traffic. With this method, real Web servers generating less traffic are requested more.

• **pending**: load balance real Web servers according to the number of pending requests. With this method, real Web servers having more pending requests are requested less.

The stickiness allows the same Web client to be always redirected to the same host. This ensures that the application context running on a host is preserved for a given Web client. Sticky connections are based on HTTP session cookies. Session cookies are either generated by the Web application running on host or are inserted by the appliance into the Web traffic. By default the session cookie is inserted by the appliance. Please note that if your real Web servers use a deterministic algorithm to redirect the traffic in turn to other servers, you probably do not need to activate the stickiness here.

To activate the stickiness based on an inserted cookie, use the keyword **sticky** followed by the keyword **insert** and the cookie name prefix. The default prefix for an inserted cookie name is *WGICPATHID*. The inserted cookie name for a given website will be the concatenation of the cookie prefix and the range number of that website. To activate the stickiness based on a cookie generated by the Web application running on real Web servers use the keyword **sticky** followed by the keyword **use** and the cookie name. The default cookie name is *JSESSIONID*. To completely deactivate the stickiness use the keyword **nosticky**.

The default load balancing method is "**robin nosticky**". Without any arguments, the fourth usage form displays the list of all load balancing policies.

In a load balancing configuration, an automatic health checking is performed on real Web servers. If one Web server fails, it is removed from the load balancing pool and will be automatically restored to the load balancing pool when the failure is fixed. Real Web servers must be accessible on their listening IP address and port 80.

When the appliance is behind more than one external gateway (connected to the external interface) that source NAT the traffic with their own (distinct) IP addresses, you should explicitly specify the gateways from which external users can access a reverse website. The fifth [5] usage form allows you to specify (via) gateways to use for a given website. Via gateways can have two roles: the **master** role and the **backup** role. When all gateways are available, a master gateway with the highest priority is elected to route Web traffic for a given website (identified by its <*site-name*> and <*public-ip*>). The elected gateway is then activated for that website. Please note that at a given point, one and only one gateway is considered as active for a given website. In case of a failure on the active gateway, a backup gateway (with the highest priority) is then elected to be activated. In case where a faulty gateway becomes operational again, the process of electing and activating via gateways is performed again.

To add a master via gateway to a given website, use the keyword **via** followed by the name and public IP address of that website (<*site-name*> <*public-ip*>), the keyword **add**, the gateway IP address to use, the keyword **master** and optionally the priority associated to the specified gateway. To add a backup gateway, use the keyword **backup** instead of the **master** keyword. To delete a gateway, use the keyword **del** instead of **add**. To erase the list of all **via** gateways for a given website, use the keywords **raz**. The priority is a numeric value between 0 and 255. If no priority is specified, the priority is set to 110 for a master gateway and to 100 for a backup gateway. Without any arguments, the fifth usage form displays the list of all via gateways.

The sixth [6] usage form allows you to temporarily put a reverse website in standby mode. When a reverse website is in standby mode, all requests to it are redirected to an information page provided by the appliance or to a given URL. The standby mode may be useful during maintenance period. If a URL is given it should be in the form: (http|https|ftp)://<domain-name>[/<URI>] where an URI may contain an alphanumeric or any of the following characters: -._~:/?#[]@!$&()*+,;=. Any other character needs to be encoded with the percent-encoding. A percent-encoding is in the form *%(a-fA-F0-9)(a-fA-F0-9)* (use %27 for the quote character). Without any arguments, the sixth usage form displays the list of all reverse websites in standby mode.

HTTP HEADERS

The "X-Forwarded-Proto" and "X-Forwarded-For" headers are added to HTTP requests sent to real Web servers in order to indicate the initial protocol and source IP address used by the end-user.

LIMITATION

The QoS defined here for a HTTP website is based on the IP address associated to that website. To define distinct QoS for two HTTP websites, distinct IP addresses must be configured for those HTTP websites.

The TLS SNI for reverse websites is supported only if the WAF mode is activated or if at least one reverse website is configured with sticky load balancing on more than one real Web server.

SEE ALSO

access (1) **apply** (1) **mode** (1) **qos** (1) **system** (1) **tls** (1) **vlan** (1) **vpnipsec** (1) **waf** (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

setup

NAME

setup - Performs a basic startup configuration

SYNOPSIS

[1] **setup**

DESCRIPTION

This command performs a basic startup configuration by:

* Phase 1: Asking you pertinent questions to set up your basic configuration.

* Phase 2: Executing online commands with your inputs.

* Phase 3: Applying the configuration with the command **apply**.

This command is automatically executed when you first connect to the system.

SEE ALSO

access (1) **admin** (1) **apply** (1) **dns** (1) **ip** (1) **port** (1) **mode** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

sslmediate

NAME

sslmediate - Manage the SSL mediation

SYNOPSIS

[1] **sslmediate** [**transparent** [(**on** | **off**)]]

[2] **sslmediate** [(**expired** | **premature** | **selfsigned**) [(**on** | **off**)]]

[3] **sslmediate policy** [(**allow** | **deny**)]

[4] **sslmediate exception** [**domainname** [(**add** | **del**) *<domain-name>* | **raz**]]

[5] **sslmediate exception** [**urllist** [(**add** | **del**) *<urllist-name>* | **raz**]]

DESCRIPTION

The SSL mediation allows you to decrypt HTTPS traffic at the gateway point in order to cache, inspect its contents and possibly block unwanted contents. The **sslmediate** command is used to configure the SSL mediation at the gateway point.

The first [1] usage form allows you to activate or deactivate the transparent SSL mediation. When the transparent mode is turned off the proxy IP address and its port number must be set at the client side (Firefox, Chrome...) for HTTPS. When the transparent mode is turned on this setting is not required as long as HTTPS traffic are routed via the appliance.

The second [2] usage form allows you to configure the tolerance of the system to allow or deny some errors in original certificates transmitted by HTTPS servers. The following errors can be allowed or denied at the gateway level:

• **expired**: the certificate has expired and is no longer valid.

• **premature**: the certificate is not yet valid because its validity period is in the future.

• **selfsigned**: the certificate is self signed or an intermediary certificate used to sign it is self signed (not signed by a know CA: Certificate Authority).

To allow an error use the keyword **on**. To deny an error use the keyword **off**.

The third [3] usage form allows you to configure the SSL mediation policy. In this way the gateway can bypass the SSL mediation for some websites (**deny** policy) or only act as an SSL mediator for some given websites (**allow** policy). The fourth [4] and fifth [5] usage form are used to define domain name exceptions to be treated or bypassed by the SSL mediation. Please note that exception lists can’t contain a domain name and one of its sub domains at the same time. In such a case sub domains are simply ignored.

To add a domain name to the list of exceptions use the keywords **domain add** followed by the domain name. To add a URL list of domain names (see the command **urllist**) use the keywords **urllist add** followed by the URL list name (as defined by the command **urllist**). To delete an entry use keyword **del** instead on **add**. To completely erase the exception list use the keyword **raz**.

CAUTION: please note that as HTTPS aims to give users privacy and security, its decrypting in the middle (before reaching the final client) may violate ethical norms and should be used with caution.

SEE ALSO

apply (1) **mode** (1) **tls** (1) **transparent** (1) **urllist** (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

system

NAME

system - Manage the operating system

SYNOPSIS

[1] **system** [**soft** [**check**] | **role** | **hard** | **machine** | **architecture** | **cpu** | **memory** | **disk** | **raid** | **serial** | **uuid** | **end**]

[2] **system patch** (**aload** | (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>*)

[3] **system backup** [(**create** [**report**] | **clear** | **wait**)]

[4] **system backup** [(**save** | **load**) (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>*]

[5] **system report** [**load** | **memory** | **disk** | **raid** | **link** | **gateway** | **connection** | **service** | **vpnipsec** | **antivirus** | **counter** [**web** | **rweb** | **firewall** | **guard** | **antivirus** | **avserver** | **waf**] [**raz**]]

DESCRIPTION

The first [1] usage form displays information about the actual system. In this usage form the optional arguments are as follows:

* **soft**: software OS (Operating System) name and version. If the optional **check** keyword is specified, the system checks for available updates.

* **role**: the system role (can be Gateway or Manager).

* **hard**: this is a string consisting of fields that gives the appliance model configured during its installation. Details are as follows:

  - OS|VE|WH: (OS installation, VE: Virtual Edition, HW: Hardware),
  - US<number>: number of USers in forwarding mode,
  - GR<number>: GuaRd blacklist records number,
  - RU<number>: number of Userse in Revers mode,
  - RW<number>: number of Reverse Websites,
  - RC<number>: Reverse Cache size in MB,
  - LR<number>: Logs Rotation period,
  - UL<number>: maximum size for UpLoaded files in MB,
  - PC<number>: Persistent Cache (0:off, 1:on),
  - WL<number>: persistent Web access logging (0:off, 1:on),
  - RL<number>: persistent rWeb access logging (0:off, 1:on).

* **machine**: machine manufacturer and product name.

* **architecture**: the installed CPU architecture (32 or 64 bits).

* **cpu**: CPU information.

* **memory**: total RAM capacity.

* **disk**: disks information.

* **raid**: RAID configuration.

* **serial**: the serial number.

* **uuid**: the appliance Universally Unique IDentifier.

* **end**: subscription period end of the system.

Please note that your appliance should be connected to the internet and have access to registration services (HTTP/HTTPS) in order to allow its subscription renewal.

The second [2] usage form allows you to load an OS patch from a file server or automatically from an official Web server. To automatically download a patch from an official Web server use the keyword **aload**. If you want to explicitly download from a file, only trusted file servers are allowed. Trusted file servers are defined with the command **access**. The explicit download form requires three mandatory arguments. The first argument is the protocol name (**ftp**, **tftp** or **sftp**). The second argument is the name or IP address of the file server. The third argument is the patch file name. The **apply** command must be used to apply a loaded patch. Note that some patches require a system reboot. In this case the system is automatically rebooted.

The third [3] and fourth [4] usage forms allow you to make a full system backup and restore. A system backup contains all necessary files to restore a crashed system on a new machine. Note that a system backup should only be restored on a freshly installed OS. Compared to a saved configuration (see the command **conf**), a system backup includes not only the logical configuration but also all data uploaded to the system like the antivirus signatures, URL lists, SSL certificates or custom WAF rules. Note that cached objects and archived logs are not part of the backup.

It is important to note that the OS version and tuning parameters (chosen during the installation) in the backup should be the same as the OS version and tuning parameters chosen for the new machine. In addition the CPU architecture and number of network interface devices on the new machine should be identical to the machine on which the backup has been made. Otherwise the restore operation can fail. As a restoration operation may fail for various reasons, it is highly recommended to test and validate backups prior to rely on them.

CAUTION: as the S/N of a registered appliance is bound to the MAC address of its first NIC (eth0), the first NIC of the new machine should have the same MAC address as the crashed machine. If changing a MAC address on the new machine is not an option, please contact our support services.

Without any argument, the **system backup** command displays the last system backup date and time. A system backup should be first created prior to save it on a file server. To create a system backup, use the keyword **create**. This will launch the system backup operation. The backup operation runs in background unless the **wait** keyword is used. Depending on your configuration, a system backup could take between 10 seconds and several minutes. The optional keyword **report** allows you to display the last system backup report. To delete a previously created or loaded system backup use the keyword **clear**.

A system backup can only be save on a trusted file servers. Trusted file servers are defined with the command **access**. The **save** usage form requires three mandatory arguments. The first argument is the protocol name. Possible values for this argument are **ftp**, **sftp** and **tftp**. The second argument is the name or IP address of the file server. The third argument is the system backup file name. To restore a previously saved system backup use the **load** usage form. To take effect, the command **apply** should be used after having loaded a system backup. Please note that loading a system backup automatically clear any previousely loaded or created backup.

The fifth [5] usage form allows you to display a **report** on the system activity and its health. It checks critical software and hardware health components and informs you about potential malfunction. Without any optional argument, this command displays a complete report on all components. If an optional argument is given, only a report related to that component is displayed.

The keyword **load** displays a report on the combined CPU and disk IO utilization over 1, 5, and 15 minutes. It is an indication of the the system load. Zero means there is no load, 100 means the system is fully loaded and a value greater than 100 means the system is overloaded.

The keyword **memory** displays a report on the RAM and swap memory size.

The keyword **disk** displays a report on disks activity in your configuration. This report includes the disks' average i/o time in milliseconds and the disks' i/o time averaged over the last 1 minute. If disks support SMART (Self-Monitoring, Analysis and Reporting Technology), the report also includes the health status of the corresponding disks and if possible the lifetime percentage for SSDs (100% means the SSD has 100% life). Note that the health status is not always available for disks in a hardware raid array. Refer to your vendor-specific health checking systems to monitor those disks.

If the system has been installed with software RAID support, the keyword **raid** displays a report on the installed RAID.

The keyword **link** displays the status of Ethernet links. Associated IP addresses to each Ethernet interface are also displayed. You can use this command to check floating IP addresses configured with the command **vrrp**.

The keyword **gateway** displays the status of connected gateways. Please note that only gateways that have been manually added by the **ip** command are monitored (ie. automatically added gateways in public cloud environment are not monitored).

The keyword **connection** displays the number of active (established) TCP connections per appliance network interface. In addition the total number of all TCP connections in all states (established, syn-sent, syn-recv, fin-wait-1, fin-wait-2, time-wait, closed, close-wait, last-ack, closing and listening) is displayed.

The keyword **service** displays the status of all critical software components. Note that critical software components differ according to your current system configuration.

The keyword **vpnipsec** displays the status of site to site IPsec VPN tunnels.

The keyword **antivirus** displays the status of the last automatic antivirus signatures update.

Finally the keyword **counter** displays the total number of blocked or allowed contents until the last log rotation. Please note that a counter is available only if its related logging is activated (see the command **log**). The optional argument **raz** allows you to reset a counter. Counter calculations are made since the system installation or the last reset operation. The real number of blocked or allowed contents can differ from values given by counters as some logs can be deactivated for a period of time and then be reactivated.

SEE ALSO

access (1) **antivirus** (1) **apply** (1) **conf** (1) **file** (1) **ip** (1) **job** (1) **log** (1) **mode** (1) **reboot** (1) **register** (1) **rweb** (1) **transparent** (1) **vrrp** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

timezone

NAME

timezone - Set or get the local time zone

SYNOPSIS

[1] **timezone** [<*zone-code*>]

DESCRIPTION

This command is used to get or to set the local time zone code. Use the command **timezonelist** to get a list of valid timezone codes.

SEE ALSO

clock (1) **ntp** (1) **timezonelist** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

timezonelist

NAME

timezonelist - Display valid time zone codes

SYNOPSIS

[1] **timezonelist**

DESCRIPTION

This command displays time zone codes. Use this command to find your time zone code. The time zone code retrieved here is to be used as an argument for the command **timezone.**

SEE ALSO

timezone (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

tls

NAME

tls - Manage TLS (SSL) certificates and other components

SYNOPSIS

[1] **tls server** [(**add** | **del**) *<tls-id>* | **revoke** *<tls-id>* [*<reason>*] | **raz**]

[2] **tls server** [**generate** *<tls-id>* [[**sign**] [**ocsp**]]]

[3] **tls server** [(**load** | **save**) *<tls-id>* (**key** | **certificate** | **csr**) (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>* [*<days>* [**ocsp**]]]

[4] **tls server** [**show** *<tls-id>* (**certificate** | **csr** | **key**) | **fingerprint** *<tls-id>*]

[5] **tls server [days** [*<days>*]]

[6] **tls ca system** [**generate** | **fingerprint** | **show** (**certificate** | **key** | **der**)]

[7] **tls ca system** [(**load** | **save**) (**key** | **certificate** | **der**) (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>*]

[8] **tls ca third** [**add** *<ca-id>* [**on** | **off**] | **del** *<ca-id>* | (**show** | **fingerprint**) *<ca-id>* | **load** *<ca-id>* (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>* | **raz**]

[9] **tls client** [(**add** *<tls-id>* [*<days>* [**no**][**ocsp**] [*<private-key-size>*]]) | (**del** *<tls-id>* [**private** [**cancel**]]) | **raz**]

[10] **tls client** [(**generate** *<tls-id>* [**cancel**]) | (**show** (**key** | **certificate**) | **fingerprint**) *<tls-id>* | **revoke** *<tls-id>* [*<reason>*]]

[11] **tls client** [**save** *<tls-id>* (**key** | **certificate** | **pkcs12** | **pfx** | **password**) (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>*]

[12] **tls client** [**load** *<tls-id>* **certificate** (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>*]

[13] **tls client** [**days** [*<days>*]]

[14] **tls ocsp** [**host** [(*<ip>* | *<name>*)] | **days** [*<days>*] | **tls** [**raz** | **set** *<tls-id>*]]

[15] **tls report**

DESCRIPTION

This command allows you to manage TLS (SSL v3) objects. A TLS object regroups different components that can be as follows: a private RSA key, an X.509 certificate and a CSR (Certificate Signing Request). We distinguish different types X.509 certificates: CA certificates, server certificates and client certificates. You can generate a TLS object or import its components from a file server. Prior to using a TLS object, it should be created in the system. A TLS object is created in two stages: an empty object represented by a unique identifier is created first and then TLS components are generated or loaded from the file server.

Usage forms from one to five [1][2][3][4][5] allow you to manage server certificates and associated TLS object components. To create a new server TLS object use the keyword **server add** followed by a unique TLS object identifier (*<tls-id>*). To delete a TLS object use the keyword **server del** followed by the TLS object identifier to remove. To erase all TLS objects use the keyword **server raz**. To revoke a certificate use the keyword **server revoke** followed by the TLS object identifier to revoke and an optionally revoke reason (refer to the **CERTIFICATE REVOKE REASONS** section below for allowed reasons). Please note that you can only revoke a certificate that has been signed with the system's CA certificate (see below).

The second usage form allows you to generate server TLS object components. To generate server TLS components use the keyword **server generate** followed by the identifier of a previously added TLS object.

During the process of generation you will be asked questions related to the generated certificate. The first information to provide is the common name for the generated certificate. If more than one name is given, a SAN (Subject Alternate Names) certificate will be created. If a name contains the character "*" wildcard certificate will be created. The wildcard "*" can replace any allowed charters in a domain name. For instance the name "*.*example.com*" will create a certificate for all "*example.com*" sub domains. Provided names should be separated by a blank. If the optional **sign** argument is specified the certificate is signed by the system's CA (see below). Finally the optional **ocsp** argument can be used to specify the usage of OCSP in the generated server certificate.

The third usage form allows you to save/load private RSA keys, X.509 certificates (in PEM format) and CSR on/from a file server. Only trusted file servers are allowed. Trusted file servers are defined with the **access** command. Please note that:

• Loaded files should be in PEM format.

• Private keys should be in an unencrypted format.

In case where a CSR is loaded, a certificate is generated and it is signed with the system's CA certificate (see below). In this case you can't use the generated certificate by the present system as its associated private RSA key won't be known. The generated certificate can then be saved on a file server and used on other systems. The generated certificate will be valid for the number of days specified as the last argument. If no number of days is specified, the certificate will be valid for the number of days specified by the usage form **days** (the fifth usage form). Finally the optional **ocsp** argument can be used to specify the usage of OCSP in the generated server certificate.

The fourth usage form allows you to show a certificate or a CSR. For security reasons a private RSA key can't be shown so this usage form shows only its fingerprint. The keyword **fingerprint** followed by a *<tls-id>* prints the SHA256 and SHA1 fingerprints of a certificate. Prior to trust and use a saved TLS object component it is highly recommended to compare its fingerprint (SHA1, SHA256) against the fingerprint of the same component displayed by the system using the CLI (with the **tls** command) or the Web GUI.

Note that the generated TLS components are not part of the configuration and thus are not saved when saving the configuration with the command **conf**. To save the configuration including TLS objects you can backup the system using the command **system**.

The sixth [6] and seventh [7] usage forms allow you to load and save the system's CA certificate and private RSA key. Please note that the system's CA certificate can be saved in both PEM and DER formats but it can be loaded only in PEM format. The system's CA certificate is used to sign server and client certificates generated by the present system. It is also used by the SSL mediation module (see the command **sslmediate**) to sign dynamically generated SSL certificates for browsed HTTPS websites. The system's CA certificate is available in PEM and DER format at: *http://<internal-ip-address>* (or *http://<web-ip-address>* if the **vlan** mode is activated). Please note that:

• Loaded files should be in PEM format.

• The private RSA key should be in an unencrypted format and not protected by a password.

In case where the system's CA certificate is an intermediate CA certificate (signed with another CA certificate), it is best practice to specify an OCSP URL for it. Otherwise some browsers refuse to import the intermediate CA certificate. The system's CA certificate is identified in the system by the certificate ID **system** (you can't use this ID for other CA certificates).

The eighth [8] usage form allows you to add and import third parties CA certificates. Third parties CA certificates can be used for Web browsing when the SSL mediation mode is activated and/or for other purposes such as VPN peers authentication. When adding a CA certificate the last optional argument allows you to turn **on** or **off** the usage of the added CA certificate for Web browsing. Trusted CA root certificates are regularly updated but in case where a CA root certificate is missing in the system, you have the possibility to manually add it with this usage form. Imported CA certificates can be a root or an intermediate certificates. Intermediate CA certificates are useful to verify server certificates in case where a server does not properly send the full certificate chain (including all intermediate CA certificates).

Usage forms ninth to thirteenth [9][10][11][12] allow you to manage cient SSL certificates signed by the system's CA root certificate. Client certificates are used to authenticate VPN peers and clients. To add a new SSL client certificate, use the keyword **add** followed by a unique TLS object ID. To delete an existing SSL client certificate use the keyword **del**. You can optionally delete private components only by specifying the optional **private** keyword. Private components are **key**, **pkcs12**, **pfx** and **password**. Keeping the public component part (**certificate**) allows you to revoke the certificate and publish its revocation using the embedded OCSP server. To erase all SSL client certificates (and its associated TLS components) use the keyword **raz**.

To revoke an existing SSL client certificate use the keyword **revoke**. To generate SSL client certificates and other related TLS components the **apply** command should be used. You can regenerate an existing client certificate by using the keyword **generate** followed by its TLS object ID. To cancel the regeneration use the keyword **cancel** as the last argument. Main TLS components are the signed **certificate** itself and its related private RSA **key**. Other components are **pkcs12**, **pfx** and **password**. The **pkcs12** component is a file storing both the private RSA key and the signed client certificate protected by an automatically generated strong password. The **pfx** component is the base 64 encoded form of the **pkcs12** component.

Client TLS components are generated using the following rules:

• The certificate common name is formed by concatenating the client certificate ID, the character "." (dot) and the system's domain name (see the command **domainname**).

• The default size of the RSA private key is 2048 bits.

• The password is in the form XXXXXXXX-XXXXXXXX-XXXXXXXX-XXXXXXXX where X is an alphanumeric character.

The eleventh [11] usage form allows you to save TLS client components on a trusted file server in order to distribute them to clients. If you completely delete a TLS client object (including its public certificate part), you will not be able to revoke the certificate. That's why you have the possibility to load the certificate component using the twelfth [12] usage in order to be able to revoke it. Please note that if the loaded client certificate is different than the existing client certificate in the system, all private components associated to that certificate are purged from the system.

A new generated client certificate will be valid for the number of days specified by the optional argument *<days>* (in the ninth usage form). If the number of days is not specified, the client certificate will be valid for the number of days configured with the thirteenth [13] usage form (**client days**). The optional **ocsp** and **noocsp** arguments (in the ninth usage form) can be used to specify the usage or no usage of OCSP in the generated client certificate. Finally the last optional argument *<private-key-size>* allows you to specify the size of the generated private RSA key in number of bits. This size should be greater than or equal 512 and less than or equal to 4096. If you regenerate an existing client certificate (using the tenth usage form), it will be valid for the same number of days as the number of days specified during its first generation (and will use the same OCSP specification if any). In case where the existing certificate uses OCSP, the regenerated certificate will be updated using the OCSP URI in effect during the regeneration. In case where the existing certificate does not use OCSP, the regenerated certificate will do the same.

The fourteenth [14] usage form allows you to configure the embedded OCSP server. OCSP stands for Online Certificate Status Protocol. It's a protocol used for obtaining the revocation status of an X.509 digital certificate. You can use this OCSP server to revoke server and client certificates signed by the system's CA certificate. To set the OCSP host name use the keyword **host** followed by an OCSP network name or IP address. To set the validity days for an OCSP response use the keyword **days** followed by a number of the required validity days. OCSP responses are signed/checked using a [private RSA key/SSL certificate] pair. The keyword **tls** is used to set the TLS object to use to that end. If no TLS object is specified (by using the **raz** keyword), the system's CA TLS object is used. If another TLS object is to be set (by using the keyword **set**) the TLS object to use should be associated to a certificate that has been signed bt the system's CA certificate. To activate the embedded OCSP server use the command **mode ocsp on**.

The fifteenth [15] usage form (**report**) allows you to display a report on the status of certificates with respects to their expiration dates.

CERTIFICATE REVOKE REASONS

When revoking a signed certificate a reason can be specified. Allowed revoking reasons are as follows:

• **keyCompromise**: the token or disk location where the private RSA key associated with the certificate has been compromised and is in the possession of an unauthorized individual. This can include the case where a laptop is stolen, or a smart card is lost.

• **CACompromise**: the token or disk location where the CA's private RSA key is stored has been compromised and is in the possession of an unauthorized individual. When a CA's private RSA key is revoked, this results in all certificates issued by the CA that are signed using the private RSA key associated with the revoked certificate being considered revoked.

• **affiliationChanged**: the user has terminated his or her relationship with the organization indicated in the Distinguished Name attribute of the certificate. This revocation code is typically used when an individual is terminated or has resigned from an organization. You do not have to revoke a certificate when a user changes departments, unless your security policy requires different certificate be issued by a departmental CA.

• **superseded**: a replacement certificate has been issued to a user, and the reason does not fall under the previous reasons. This revocation reason is typically used when a smart card fails, the password for a token is forgotten by a user, or the user has changed their legal name.

• **cessationOfOperation**: if a CA is decommissioned, no longer to be used, the CA's certificate should be revoked with this reason code. Do not revoke the CA's certificate if the CA no longer issues new certificates, yet still publishes CRLs for the currently issued certificates.

• **unspecified**: no reason has been specified.

• **cancel**: this is not intrinsically a revoke reason but if the revoke operation has not yet been applied, you can cancel it by using this keyword.

SEE ALSO

access (1) **admin** (1) **apply** (1) **file** (1) **mode** (1) **domainname** (1) **port** (1) **rweb** (1) **sslmediate** (1) **system** (1) **vpn** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

traceroute

NAME

traceroute - Trace the route to a host

SYNOPSIS

[1] **traceroute** (<*name*> | <*ip*>) [**icmp** | **udp**]

DESCRIPTION

You can use this command to trace the route to a host specified by its IP address or name. This command uses the ICMP protocol by default. You can use the UDP protocol on the 33434 port by using the **udp** optional argument.

SEE ALSO

ping (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

transaction

NAME

transaction - Manage a set of commands as a single transaction

SYNOPSIS

[1] **transaction** (**open** | **close** | **show** | **commit** [**log**])

DESCRIPTION

For internal usage. Use with caution!

SEE ALSO

apply (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

transparent

NAME

transparent - Manage the transparent Web proxy

SYNOPSIS

[1] **transparent** [**raz** | (**add** (**internal** | **auxiliary** | **vpnipsec**) <*ip*> [<*network-mask*> [<*qos*>]]) | (**del** (**internal** | **auxiliary** | **vpnipsec**) <*ip*> <*network-mask*>)]

DESCRIPTION

When the transparent mode is activated, the system transparently intercepts incoming Web traffic and manage it the same way as it has destined to the system's Web proxy (in **web** mode). This command is used to limit the transparent mode to specified Web traffic only. If no transparent traffic is defined, the system acts as a transparent Web gateway (Web proxy) for all Web traffic incoming from the **internal** (**web** in **vlan** mode), **auxiliary** and **vpnipsec** interfaces. If at least one transparent Web traffic or one Web access network (see the command usage form **access web**) is defined, the system acts as a transparent Web gateway only for specified Web traffic. A transparent Web traffic is identified by two parameters: the network interface via which it enters into the system and and it's source IP address.

Traffic bandwidths can also be customised for transparent networks by setting the optional <*qos*> parameter. The <*qos%*> value is a percentage of the ingress or egress bandwidth allocated to **tweb** (transparent Web) traffic and should be an integer between 1 and 100. Ingress and egress bandwidth values to which the percentage is applied are as follows:

• For Web traffic exchanged via the via the native **internal** network interface or the 802.1q pseudo network interface called **web** (in **vlan** mode), the ingress and egress bandwidths to consider are defined with the command usage form **qos shape tweb internal**.

• For Web traffic exchanged via the **auxiliary** network interface, the ingress and egress bandwidths to consider are defined with the command usage form **qos shape tweb auxiliary**.

• For Web traffic exchanged via the **vpnipsec** virtual network interface, the ingress bandwidth to consider is defined with the command usage form **qos shape vpnipsec external ingress**. The egress bandwidth for Web traffic exchanged via the **vpnipsec** virtual network interface can't be customised.

If no <*qos%*> is given, the value of 100% is used by default.

SEE ALSO

access (1) **apply** (1) **mode** (1) **qos** (1) **vlan** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

tweb

NAME

tweb - Manage the transparent Web proxy

SYNOPSIS

[1] **tweb** [**raz** | (**add** (**internal** | **auxiliary** | **vpnipsec**) <*ip*> [<*network-mask*> [<*qos*>]]) | (**del** (**internal** | **auxiliary** | **vpnipsec**) <*ip*> <*network-mask*>)]

DESCRIPTION

See the **transparent** command manual.

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

urllist

NAME

urllist - Manage URL lists

SYNOPSIS

[1] **urllist** [**raz** | (**add** | **del**) *<urllist-name>* | **update** | **report**]

[2] **urllist** [(**load** | **vload**) [(**create** | **update** | **auto**) *<urllist-name>* (**ftp** | **sftp** | **tftp**) *<file-server>* *<base-file-name>* [(**domains** | **urls** | **expressions**)*]]]

[3] **urllist** [**clear** *<urllist-name>* [(**domains** | **urls** | **expressions**)*]]

[4] **urllist auto** [*<urllist-name>* [**off** | (**on** (**load** | **vload**) (**create** | **update**) (**daily** | **weekly**) (**push** | (**ftp** | **sftp** | **tftp**) *<file-server>* *<base-file-name>*))]]

DESCRIPTION

This system allows you to filter the access to URLs and domain names according to policies and rules (see the **guard** command). The command **urllist** allows you to manage those lists of URLs and domain names. A URL list is represented by a unique name in the system. First, an empty URL list must be created. The first [1] usage form allows you to create, delete or update URL lists. To create an empty URL list use the keyword **add** followed by a URL list name. A URL list name must begin with an alpha character and may contains alpha numeric characters as well as the characters "_" and "-". To delete a URL list use the keyword **del** followed by the URL list name to delete. The keyword **raz** allows you to delete all URL lists.

The system can be configured to periodically update URL list contents (see later). In the first usage form the keyword **update** allows you to explicitly update URL list contents without having to wait for the periodic automatic updates. This could be useful for instance with a suspended virtual machine having a non updated system time (in this case do not forgetgorget to update the system time prior to updating URL list contents). Using the keyword **report** allows you to display the latest report of the automatic URL list content updating. During such an update if a URL list can't be loaded, an error is occurred. For error explanations see the section **DOWNLOAD ERROR CODES** below.

The second [2] usage form allows you to load URL lists from files located on a file server. Only trusted file servers are allowed. Trusted file servers are defined with the command **access**. To load and build a URL list from scratch use the keyword **load** followed by the keyword **create**. The content of a URL list to load should be defined in three compressed (gzip format) text files having the same base name and three different extensions. The first file has the *.domains.gz* extension and contains a list of domain base names (one domain base name per line). Only domain **base** name should be specified (for instance "example.com" is a valid domain base name while "www.example.com" is not). The second file has the extension *.urls.gz* and contains a list of URLs (one URL per line). A URL is in the form *<domain-name>/<loaction>*. Please note that the protocol part (for instance "http://") of the URL should not be specified in that file. Finally the third file has the extension *.expressions.gz* and contains a list of regular expressions (one regular expression per line). See the section **REGULAR EXPRESSIONS** below.

When defining a domain name, do not specify the prefixes www, web and ftp. For a URL definition, protocol parts (http://, ftp://...), prefixes (www, web, ftp) and ports (:80, :443...) should not be specified. An expression is a regular expression as described in regex (see the regex (5) manual on a UNIX system). Note that complex regular expressions require large CPU resources.

A **domain** or **url** diff file contains lines beginning with "-" (minus) or "+" (plus). To add a domain or a URL precede it with the character "-". To delete a domain or a URL precede it with the character "+". The *<base-file-name>* parameter is the base file name (without extensions *.domains.gz*, *.urls.gz* or *.expressions.gz*) of a defined URL list. File(s) must exist and be accessible on the file server. The optional last arguments specify which URL list types to load. URL list types are: **domains**, **urls** and **expressions**. If

no URL list types are specified, **domains** and **urls** URL list types are loaded.

To update an existing URL list use the keyword **update**. In this case downloaded files are diff files. Only **domains** and **urls** files can be updated. If more than one update file is loaded for the same URL list, updates well be successively applied to that URL list. In this case the order in which update files are loaded is important.

To automatically download all updates since the last create or update operation, use the keyword **auto**. In this case downloaded files are diff files and should be named as follows: *<base-file-name>.<yyyymmdd>.(domains | urls).gz* where *<yyyymmdd>* is the date (*yyyy* is the year, *mm* is the month and *dd* is the day). In the case where the URL list has never been loaded before and the update mode is used, the create mode is used (the URL list is entirely loaded from scratch).

The **vload** (verify load) option allows you to secure downloads. This is useful when you download URL lists from a file server managed by CacheGuard Technologies Ltd or one of its referenced partners. When using the **vload** method, a signature file is downloaded alongside the URL list file and the URL list file is verified using that signature file to assure that the downloaded URL list file has not been altered during its transfer. The signature file name has the same name as the downloaded URL list file followed by the extension *.sig*. You can print a list of all downloaded files by using the keyword **load** (or **vload**) without any other options.

Please note a URL list file in gzip compressed format can't be larger than than 128 MB. In case where **domains**, **urls** and **expressions** files are all loaded (separately or at the same time), if one of them exceeds the allowed limit, all of them are deleted and the current load operation is immediately cancelled. To load files larger than the allowed limit, you have the possibility to split them into smaller pieces.

The third [3] usage form allows you clear the content of a URL list without completely deleting that list. You can select what parts of the URL list should be cleared by specifying the **domains**, **urls expressions** keywords. If no part name is specified, all parts are cleared.

The fourth [4] usage form allows you to program the system to automatically download and apply URL list contents. This can be done **daily** or **weekly**. To activate the automatic updating for a URL list use the **auto on** keywords followed by download specifications. To download the complete URL list use the **create** keyword. In case where a complete URL list is downloaded, files to download should be named as follows: *<base-file-name>.(domains | urls).gz*. Please note that both domains and urls files should exist on the remote file server. To only download updates (diff files) use the **update** keyword. Because downloading and rebuilding a complete URL list can be quite time and bandwidth consuming, always prefer the **update** mode rather than the **create** mode. The **update** mode allows the periodical download of all updates since the last **create** or **update** operation. This way, in case of an unavailability of a file server, not update files are lost. To allow the system to automatically download all update files since the last automatic update, files should be named as follows: *<base-file-name>.<yyyymmdd>.(domains | urls).gz* where *<yyyymmdd>* is the date (*yyyy* is the year, *mm* is the month and *dd* is the day).

Automatic downloads can be done using the **ftp**, **sftp** or **tftp** protocols. In case where the system is managed by a remote *manager* system, the manager can be configured to download URL lists and push them to all gateways that it manages. In this case, you must specify the **push** method as the used protocol. The **AUTO UPDATES ON A MANAGER SYSTEM** section below gives more information regarding automatic URL lists updates on a *manager* system.

To deactivate the automatic updating for a URL list use the keywords **auto** followed by the keyword **off**.

DOWNLOAD ERROR CODES

• Transfer error 1: Unsupported protocol. This build of the program has no support for this protocol.

• Transfer error 2: Failed to initialize.

• Transfer error 3: URL malformed. The syntax was not correct.

• Transfer error 5: Couldn't resolve proxy. The given proxy host could not be resolved.

• Transfer error 6: Couldn't resolve host. The given remote host was not resolved.

• Transfer error 7: Failed to connect to host.

• Transfer error 8: FTP weird server reply. The server sent data the program couldn't parse.

• Transfer error 9: FTP/SFTP access denied. The server denied login or denied access to the particular resource or directory you wanted to reach. Most often you tried to change to a directory that doesn't exist on the server. To save on an SFTP server always specify the full target path.

• Transfer error 11: FTP weird PASS reply. The Program couldn't parse the reply sent to the PASS request.

• Transfer error 13: FTP weird PASV reply, The Program couldn't parse the reply sent to the PASV request.

• Transfer error 14: FTP weird 227 format. The Program couldn't parse the 227-line the server sent.

• Transfer error 15: FTP can't get host. Couldn't resolve the host IP we got in the 227-line.

• Transfer error 17: FTP couldn’t set binary. Couldn’t change transfer method to binary.

• Transfer error 18: Partial file. Only a part of the file was transferred.

• Transfer error 19: FTP couldn’t download/access the given file, the RETR (or similar) command failed.

• Transfer error 21: FTP quote error. A quote command returned error from the server.

• Transfer error 22: HTTP page not retrieved. The requested url was not found or returned another error with the HTTP error code being 400 or above. This return code only appears if -f/--fail is used.

• Transfer error 23: Write error. The Program couldn’t write data to a local filesystem or similar.

• Transfer error 25: FTP couldn’t STOR file. The server denied the STOR operation, used for FTP uploading.

• Transfer error 26: Read error. Various reading problems.

• Transfer error 27: Out of memory. A memory allocation request failed.

• Transfer error 28: Operation timeout. The specified timeout period was reached according to the conditions.

• Transfer error 30: FTP PORT failed. The PORT command failed. Not all FTP servers support the PORT command, try doing a transfer using PASV instead!

• Transfer error 31: FTP couldn’t use REST. The REST command failed. This command is used for resumed FTP transfers.

• Transfer error 33: HTTP range error. The range "command" didn’t work.

• Transfer error 34: HTTP post error. Internal post-request generation error.

• Transfer error 35: SSL connect error. The SSL handshaking failed.

• Transfer error 36: FTP bad download resume. Couldn’t continue an earlier aborted download.

• Transfer error 37: FILE couldn’t read file. Failed to open the file. Permissions?

• Transfer error 38: LDAP cannot bind. LDAP bind operation failed.

• Transfer error 39: LDAP search failed.

• Transfer error 41: Function not found. A required LDAP function was not found.

• Transfer error 42: Aborted by callback. An application told the program to abort the operation.

• Transfer error 43: Internal error. A function was called with a bad parameter.

• Transfer error 45: Interface error. A specified outgoing interface could not be used.

• Transfer error 47: Too many redirects. When following redirects, the program hit the maximum amount.

• Transfer error 48: Unknown TELNET option specified.

• Transfer error 49: Malformed telnet option.

• Transfer error 51: The peer’s SSL certificate or SSH MD5 fingerprint was not ok.

• Transfer error 52: The server didn’t reply anything, which here is considered an error.

• Transfer error 53: SSL cryptographic engine not found.

• Transfer error 54: Cannot set SSL cryptographic engine as default.

• Transfer error 55: Failed sending network data.

• Transfer error 56: Failure in receiving network data.

• Transfer error 58: Problem with the local certificate.

• Transfer error 59: Couldn’t use specified SSL cipher.

• Transfer error 60: Peer certificate cannot be authenticated with known CA certificates.

• Transfer error 61: Unrecognised transfer encoding.

• Transfer error 62: Invalid LDAP URL.

• Transfer error 63: Maximum file size exceeded.

• Transfer error 64: Requested FTP SSL level failed.

• Transfer error 65: Sending the data requires a rewind that failed.

• Transfer error 66: Failed to initialise SSL Engine.

• Transfer error 67: The user name, password, or similar was not accepted and the program failed to log in.

• Transfer error 68: File not found on TFTP server.

• Transfer error 69: Permission problem on TFTP server.

• Transfer error 70: Out of disk space on TFTP server.

• Transfer error 71: Illegal TFTP operation.

• Transfer error 72: Unknown TFTP transfer ID.

• Transfer error 73: File already exists (TFTP).

• Transfer error 74: No such user (TFTP).

• Transfer error 75: Character conversion failed.

• Transfer error 76: Character conversion functions required.

• Transfer error 77: Problem with reading the SSL CA certificate (path? access rights?).

• Transfer error 78: The resource referenced in the URL does not exist.

• Transfer error 79: An unspecified error occurred during the SSH session.

• Transfer error 80: Failed to shut down the SSL connection.

• Transfer error 82: Could not load CRL file, missing or wrong format.

• Transfer error 83: Issuer check failed.

REGULAR EXPRESSIONS

An expression list file contains regular expressions with one regular expression per line. The command guard uses Posix regular expressions. The Posix Regular Expression language is a notation for describing textual patterns. Of most interest is:
. Matches any single character (use "." to match a ").
[abc] Matches one of the characters ("[abc]" matches a single "a" or "b" or "c").
[c-g] Matches one of the characters in the range ("[c-g]" matches a single "c" or "d" or "e" or "f" or "g". "[a-z0-9]" matches any single letter or digit. "[-/.:?]" matches any single "-" or "/" or "." or ":" or "?".).
? None or one of the preceding ("words?" will match "word" and "words". "[abc]?" matches a single "a" or "b" or "c" or nothing (i.e. "")).
***** None or more of the preceding ("words*" will match "words" and "wordsssssss". ".*" will match anything including nothing).
+ One or more of the preceding ("xxx+" will match a sequence of 3 or more "x").
(expr1|expr2) One of the expressions, which in turn may contain a similar construction ("(foo|bar)" will match "foo" or "bar". "(foo|bar)?" will match "foo" or "bar" or nothing (i.e. "")).
**$** The end of the line ("(foo|bar)$" will match "foo" or "bar" only at the end of a line).
\x Disable the special meaning of x where x is one of the special regex characters **".?*+()^$[]{}\"** (\. will match a single ".", "\\" a single "\" etc.)

AUTO UPDATES ON A MANAGER SYSTEM

On a *manager* system, the automatic URL lists update can be activated depending on the installed license level. The activation is normally possible for non-free installation only. This feature on a manager allows you to periodically download URL lists from one or more remote file servers and then automatically push them on all remote gateways. Please refer to the **manager** command for further information.

If on a manager, an automatic URL list download is configured to download updates (diff files and not the complete content) and after a successful download from a remote file server, its push to a remote gateway fails (because of an unavailability of that gateway at the time of the push operation), the next time (the day or week after) the system will download and push the complete content for that URL list.

SEE ALSO

access (1) **apply** (1) **file** (1) **guard** (1) **manager** (1) **sslmediate** (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

usleep

NAME

usleep - Suspend the execution of the calling thread for microseconds

SYNOPSIS

[1] **usleep** <*us*>)

DESCRIPTION

Suspends execution of the calling thread for <us> microseconds. Valid microseconds are digit between 1 and 9999999999999999.

SEE ALSO

apply (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

vlan

NAME

vlan - Configure 802.1q VLANs (Virtual LANs)

SYNOPSIS

[1] **vlan** [(**admin** | **antivirus** | **peer** | **file** | **mon** | **rweb** | **web**) [<*vlan-id*>]]

DESCRIPTION

This command allows you to define 802.1q VLANs (Virtual LANs). VLANs allow you to increase the network security by isolating each type of access crossing the same physical NIC (Network Interface Card) in a Virtual LAN (VLAN). When using VLANs an associated pseudo virtual NIC is implicitly defined for each defined VLAN.

A VLAN is identified by a <vlan-id> (VLAN identifier). This is a numeric value between 0 and 4095. By default, all access types are associated to the VLAN 0. Different access types may use the same VLAN. This **vlan** command allows you to define new VLANs and associate them to an access type. Valid access types are as follows:

• **admin**: access from remote administrators (SSH and Administration GUI). You can use this VLAN to connect privileged workstations having administration rights.

• **antivirus**: access to the antivirus as a service from external systems. You can use this VLAN to isolate all request/responses with the embedded antivirus.

• **peer**: access to/from peer appliances access. You can use this VLAN to connect all peer appliances.

• **file**: access to file server (FTP, TFTP and SFTP). You can use this VLAN to connect file servers.

• **mon**: monitoring (SNMP) access access to/from SNMP managers. You can use this VLAN to connect monitoring servers.

• **rweb**: access to backend Web servers (and/or Web application servers). You can use this VLAN for any other servers having the same security level as for Web servers.

• **web**: Web access from transparent and non transparent users. You can use this VLAN for all users having to access the internet.

VLANs are all associated to the **internal** network interface. The **external** network interface is associated to a unique type of access which is the access to the external untrusted world (internet). The **auxiliary** network interface is also associated to a unique type of access. You can use the **auxiliary** network interface as per your requirement (to connect the Back Office zone or the DMZ for instance).

Note that when using VLANs, the native internal network interface (ie. **internal**) is no longer available. To use VLANs activate the **vlan** feature with the command **mode**.

SEE ALSO

apply (1) **ip** (1) **mode** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

vpnipsec

NAME

vpnipsec - Manage IPsec VPN tunnels and networks

SYNOPSIS

[1] **vpnipsec** [**authenticate** [**psk** (*<pre-shared-key>* | **auto**) | (**tls** | **eaptls**) *<tls-id>* [(**dn** | **fqdn**)]]]

[2] **vpnipsec** [**access** [(**off** | **on**) [*<ike-encryption* [*<ike-integrity>* [*<ike-diffie-hellman>* [*<esp-encryption>* [*<esp-integrity>*]]]]]]]

[3] **vpnipsec** [**access** [**conf** (**android** | **apple** | **linux** | **windows**) (*<client-tls-id>* | **any**) (**show** | **save** (**ftp** | **sftp** | **tftp**) *<file-server>* *<file-path>* | **email** *<email-address>*) [*<name>*]]]

[4] **vpnipsec** [**access** [**authenticate** [(**psk** | **tls** | **eaptls**)]]]

[5] **vpnipsec** [**access** [**access** [**raz** | (**add** | **del**) *<ike-id>*]]]

[6] **vpnipsec** [**site** [**add** *<vpn-id>* *<peer-vpn-address>* (**psk** *<pre-shared-key>* | **tls** (**certificate** *<tls-id>* | **dn** *'<distinguished-name>'* | **fqdn** *<domain-name>*)) [*<ike-encryption*] [*<ike-integrity>* [*<diffie-hellman>* [*<esp-encryption>* [*<esp-integrity>* [*<remote-isakmp-port>* [*<remote-nat-transversal-port>*]]]]]]]]

[7] **vpnipsec** [**site** [**raz** | **del** *<vpn-id>*]]

[8] **vpnipsec** [**network** [(**access** | **site** *<vpn-id>*) [(**raz** | ((**add** | **del**) (**local** | **remote**) *<network-ip>* [*<network-mask>*]))]]]

[9] **vpnipsec** [**via** [(**access** | **site** *<vpn-id>*) [**raz** | (**add** | **del**) *<gateway-ip>* (**master backup**) [*<priority>*]]]]

[10] **vpnipsec** [**to** [*<vpn-id>* [**raz** | (**add** | **del**) *<peer-vpn-address>*]]]

[11] **vpnipsec** [**nat** (**public** [**raz** | (**add** | **del**) *<local-public-ip>*]) | (**role** [*<vpn-id>* [(**active** | **passive** | **raz**)]])]

[12] **vpnipsec** [**report**]

DESCRIPTION

VPN stands for Virtual Private Network and IPsec for Internet Protocol Security. An IPsec VPN allows you to authenticate and encrypt the data packets between private networks over a public IP network (ie internet) to provide secure encrypted communications. You can build a persistent IPsec VPN between 2 sites or allow remote workers to access your internal infrastructures via an IPsec VPN server. To use the IPsec VPN server on the present system you should activate it first (use the command **mode**). Then the **vpnipsec** command can be used to create and manage IPsec VPNs.

We distinguish two IPsec VPN modes: site to site VPNs and remote access VPNs. A site to site (or inter site) VPN allows you to build a permanent secure tunnel between two sites. With such a tunnel, computers in both sites can communicate with each other in a secure way as they were on the same location whereas in reality they can be separated by several thousands of kilometers. To build a site to site IPsec tunnel you need two VPN servers: a local VPN server and the remote (or peer) VPN server. As IPsec is a standard protocol, you can build a site to site VPN using VPN servers provided by distinct manufacturers/editors.

A remote access VPN is a central VPN server to which remote workers can connect via secure tunnels built on top of the internet. With such tunnels remote workers can access computers protected by the VPN

server in a secure way as they were on the same location. To build a remote access IPsec VPN you need a central IPsec VPN server while each remote worker connect the central VPN server using an IPsec VPN client. This system supports native IPsec VPN clients provided by most devices and OS in the market. In case where native VPN clients would not work, alternative third party IPsec VPN clients such as strongSwan® can be used. The **vpnipsec** command allows you to build site to site as well as remote access IPsec VPNs.

The **vpnipsec** command allows you to build IPsec VPNs without having extensive knowledge in cryptography and IPsec protocols. However having some basic knowledge on IPsec principals can help to better understand the configuration to set. The **IPSEC VPN IN SHORT** section below aims to explain IPsec in short. Please note that you should chose between the site mode and the remote accesse mode: if you activate the remote access mode you can no longer use site to site IPsec VPNs. To use site to site IPsec VPNs the remote access mode should be deactivated.

The present system use its external interface to establish IPsec VPN with peers or remote clients. That's why there is no need to specify the local IP address to set an IPsec VPN on this system. You can set the system's external network interface and IP address using the commands **link** and **ip**.

The first [1] usage form allows you to configure the authentication method used on the local IPsec VPN server. To connect to the local IPsec VPN server, remote peers/clients must then use the authentication method configured with this usage form. Currently three authentication methods are supported: the PSK (Pre Shared Key) method (**psk**), the SSL certificate based method (**tls**) and the EAP-TLS (Extensible Authentication Protocol TLS) method (**eaptls**). To activate the pre shared key method use the keyword **psk** followed by a word composed by a mix of at least 32 alphabetic, numeric or the dash (-) characters. To automatically generate a strong pre-shared-key you can use the keyword **auto**. To activate the SSL or EAP-TLS based methods use respectively the keyword **tls** or **eaptls** followed by the TLS identifier of the SSL certificate to use (refer to the **tls** command to import, export or generate SSL certificates). Please note that only one authentication method can be activated at the same time. When using the **tls** or **eaptls** authentication methods, remote peers/clients must identify the local IPsec VPN server by using the **dn** (distinguished name) or **fqdn** (fully qualified domain name) of the specified SSL certificate. The last argument of this usage form allows you to specify the identifier type that should be used by remote IPsec VPN peers/clients. If no identifier type is specified the distinguished name (**dn**) is used.

It is important to note that:

• In case where the **fqdn** based identification is used, remote peers/clients should connect the IPsec VPN server using its **fqdn** and not its IP address (which is the appliance external IP address). Otherwise, the connection won't establish.

• The **dn** to use by remote IPsec VPN peers/clients should be the local SSL certificate subject. You can get the subject of an SSL certificate using the **tls** command.

• The **fqdn** to use by remote IPsec VPN peers/clients should be a subject alternative name of the local SSL certificate. This means that when using the **fqdn**, the local SSL certificate should be a SAN certificate. You can get the subject alternative name of an SSL certificate using the **tls** command.

The second [2] usage form (**vpnipsec access**...) allows you to deactivate, activate and configure the remote access mode. To deactivate the remote access mode use the keyword **off**. To activate the remote access mode use the keyword **on** followed by optional IPsec settings. A brief description of these settings is given in the **IPSEC VPN IN SHORT** section below.

The third [3] usage form (**vpnipsec access conf**...) allows you to automatically generate configuration profiles (or connection scripts) to use on remote IPsec VPN client systems. The generated configuration can be displayed (**show** keyword), saved (**save** keyword) on a trusted file server or be sent to an email address. Trusted file servers are defined with the **access** command. The email account used to send the email is configured using the **email** command. Supported IPsec VPN client systems are as follows (for each system, a brief description on how to use them is given):

• The strongSwan® App on Android® (**android** keyword): save the generated profile on a file having the *.sswan* extension, place it on Web server and then import it into the strongSwan® App from that Web server. Please refer to the strongSwan® App documentation for further information.

• Apple® (MacOS® or iPhone® / iOS®) (use the **apple** keyword): save the generated profile on a file having the *.mobileconfig* extension and then import it into the Apple® device. Please refer to the Apple® documentation for further information.

• Microsoft® Windows® (**windows** keyword): run the generated PowerShell script a Windows® system. With Windows®, the client certificate should be imported separately as a machine certificate into the Windows® system. Please refer to the Windows® documentation for further information.

In case where the runtime authentication method is based on a PSK, the IPsec VPN server address used in the generated configuration is set as follows (in order of priority):

• The first IP address in the NAT public IP address list (see the eleventh [11] usage form below).

• If the HA mode is enabled (see the **mode** and **vrrp** commands), the external master VRRP IP address having the highest priority. If no master VRRP IP address is defined, the external backup VRRP IP address

having the highest priority.

• If the HA mode is disabled, the appliance external IP address (see the **ip** command).

In case where the runtime authentication method is based on a certificate (**tls** or **eaptls** authentication methods), the IPsec VPN server address used in the generated configuration is set to canonical name (CN) part of the subject in that certificate.

The fourth [4] usage form is used to specify the client side authentication method to use in remote access mode. Allowed authentication methods at the client side are **psk**, **tls** and **eaptls**. When using the **psk** method , the pre shared key to use at the client side should be the same as the pre shared key at the server side. When using the **tls** or the **eaptls** methods, the IKE (or EAP) identifier to use by the client should be the canonical name (CN=) part of the subject specified in its SSL certificate

In remote access mode, when remote clients use the **tls** or **eaptls** authentication methods at their side, you have the possibility to limit client accesses to IKE (or EAP) identifiers that are explicitly allowed. The fifth [5] usage form is used at that end. In case where the access list defined with this usage form is empty, all authenticated clients are allowed to connect to the IPsec VPN server. Otherwise, only IKE (or EAP) identifiers that are part of that access list are allowed to connect. Normally, the IKE (or EAP) identifier is the CN part of the client certificate subject. In case where the remote client is a Windows® machine and the used authentication method at its side is **tls**, the IKE identifier is the complete client certificate subject (and not only its CN part).

The sixth [6] usage form (**vpnipsec site**...) allows you to create site to site IPsec VPNs. A site to site IPsec VPN is identified by a VPN identifier (<*vpn-id*>). A <*vpn-id*> must begin with an alpha character and may contains alpha numeric characters as well as the characters "_" and "-". To establish an IPsec tunnel between two IPsec VPN servers each peer should authenticate the other first. Currently two authentication methods are supported: **psk** and **tls**. To add a site to site IPsec VPN use the keyword **add** followed a VPN identifier (<*vpn-id*>), the remote VPN server address (<*peer-vpn-address*> (IP address or network name) and an authentication method required from the remote IPsec peer. In case where the remote VPN server IP is set to 0.0.0.0 (or the keyword **any**), any authenticated remote VPN server would be allowed to establish a site to site IPsec VPN with the local system. Each authentication method required from a remote peer needs its specific parameters to specify. Those parameters are described below.

In case where the authentication method required from the remote peer is **psk**, the used pre shared key must be between 32 and 64 characters.

In case where the authentication method required by the peer is based on an SSL certificate (**tls** authentication method), the SSL certificate advertised by the remote peer should be a SAN certificate and must be trusted by the local system. To trust an SSL certificate, the CA certificate that has been used to sign it should be present on the local system. The remote peer can also advertise a self signed certificate. In this case that SSL certificate should be present as a server certificate on the local system. You can import and trust CA and server SSL certificates using the **tls** command. The remote peer identification can be either DN (Distinguished Name) based, FQDN (Fully Qualified Domain Name) based or self signed certificate based:

• For a DN based identification, use the **dn** keyword followed by the distinguished name of the advertised SSL certificate. The distinguished name to use should be the subject part of the SSL certificate. It’s a list of assertions (*key* = *value*) separated by a comma (the whole distinguished name should be placed between quotes).

• For an FQDN based identification, use the keyword **fqdn** followed by the fully qualified domain name of the advertised SSL certificate. The **fqdn** of an SSL certificate is the last assertion part of its subject and should also be specified as a subject alternative name in the SSL certificate.

• For a self signed certificate based identification, use the **certificate** keyword followed by the <*tls-id*> of that self signed certificate. It is important to note that if the identification method required by the remote peer is FQDN, the remote peer address must be specified as network name and not an IP address. In this case, the specified network name should match the FQDN of the self signed certificate. Otherwise, the IPsec tunnel won’t establish.

With the sixth [6] usage form, some optional arguments can be specified. In case where they are not specified default values are used. Default values for optional arguments are given below.

Default values for optional arguments are as follows:

• <*ike-encryption* used by the remote peer: **aes256**

• <*ike-integrity*> used by the remote peer: **sha256**

• <*diffie-hellman*> used by the remote peer: **modp2048**

• <*esp-encryption*> used by the remote peer: **aes256**

• <*esp-integrity*> used by the remote peer: **sha256**

• <*remote-isakmp-port*> used by the remote peer: 500

• <*remote-nat-transversal-port*> used by the remote peer: 4500

The seventh [7] usage form is used to delete a site to site IPsec VPN or completely erase all site to site IPsec VPNs in the system.

With an IPsec VPN a local network can SECURELY communicate with a remote network via an untrusted public network such as the internet. The eighth [8] usage form allows you to specify the local and remote networks to automatically route via the IPsec tunnel. With a site to site IPsec VPN at least one remote network should be specified. In remote access mode if no remote network is specified the default 172.17.0.0/16 network is used. This means that remote workers will get a virtual IP address in the 172.17.0.0/16 network. In all cases, if no local network is specified, the internal connected network (connected to the **internal** interface (or the **web** interface in VLAN mode)) is routed via the IPsec tunnel. If no <*network-mask*> is specified the default network mask *255.255.255.0* is used. To specify the default route you can either use the network "0.0.0.0 0.0.0.0" or the keyword **default**.

When the routing configuration uses more than one external gateway (connected to the external interface) that source NAT the traffic with their own (distinct) IP addresses, the IPsec VPN configuration requires some tweaks in order to allow the system to properly establish a VPN tunnel. The ninth [9] usage form allows you to explicitly specify (via) gateways to use for a given IPsec VPN. Via gateways can have two roles: the **master** role and the **backup** role. During the first IPsec VPN establishment, a master gateway with the highest priority is elected to route IPsec VPN traffic for a given IPsec VPN (specified by its <*vpn-id*>). The elected gateway is then activated for that IPsec VPN. Please note that at a given point, one and only one gateway is considered as active for a given IPsec VPN. In case of a failure on the active gateway, a backup gateway (with the highest priority) is then elected to be activated. In case where a faulty gateway becomes operational again, the process of electing and activating a new gateway is performed again.

To add a master via gateway to a given site to site IPsec VPN, use the keyword **via site** followed by the given IPsec VPN identifier (<*vpn-id*>), the keyword **add**, the gateway IP address to use, the keyword **master** and optionally the priority associated to the specified gateway. To add a backup gateway, use the keyword **backup** instead of the **master** keyword. To delete a gateway, use the keyword **del** instead of **add**. To erase the list of all **via** gateways for a given IPSec VPN, use the keywords **raz**. To add or delete gateways in IPsec access mode, replace the couple "**site** <*vpn-id*>" by the keyword **access**. The priority is a numeric value between 0 and 255. If no priority is specified, the priority is set to 110 for a master gateway and to 100 for a backup gateway.

For a site to site IPsec tunnel where the remote IPsec peer is accessible via multiple gateways (with distinct public IP addresses), remote backup IP addresses or network names should be specified on the local system. Otherwise, in case of a failure on the remote master address (IP or network name), the IPsec tunnel can’t be established. The tenth [10] usage form allows you to manage remote backup addresses for a site to site IPsec tunnel. To add a remote backup address to a site to site IPsec tunnel, use the keyword **to** followed by VPN identifier (<*vpn-id*>), the keyword **add** and the remote peer backup address. To delete a remote peer backup address, use the keyword **del** (instead of the **add** keyword). To erase the list of all remote peer backup addresses associated to a site to site IPsec tunnel, use the keyword **to** followed by the VPN identifier (<*vpn-id*>) and the keyword **raz**.

In a configuration where the appliance is behind NAT (its external IP address is translated into one or more public IP addresses) and the used authentication method (on the local appliance) is PSK, all public NAT IP addresses should be specified on the local system in order to allow remote **active** (see below) peers to be properly authenticated. The eleventh [11] usage form allows you to manage those NAT IP addresses. To add a NAT IP address use the keywords **nat public add** followed by the NAT IP address. To delete a NAT IP address, use the keyword **del** (instead of the **add** keyword). To erase the list of all public NAT IP addresses use the keywords **nat public raz**.

In a site to site IPsec VPN configuration where both local and remote peers are behind NAT, one peer should take the passive role while the other should act as active (the one that initiates first the connection). In a Hub and Spoke architecture, where the central hub and spokes are all behind NAT, the central hub must act as passive while spokes should take the active role. The eleventh [11] usage form allows you to configure the role of remote peers. To configure a remote peer as passive for a site to site IPsec VPN, use the keywords **nat role** followed by its identifier (<*vpn-id*>) and the keyword **passive** (in this case the local peer takes the **active** role). To configure a remote peer as active for a site to site IPsec VPN, use the keywords **nat role** followed by its identifier (<*vpn-id*>) and the keyword **activee** (in this case the local peer takes the **passive** role). When configuring such a site to site IPsec VPN, please take care to do not select the same roles for both peers. If no NAT role is specified for a site to site IPsec VPN, the active role is assumed for it. Using the keyword **raz** allows you to restore the default behaviour for a NAT role.

The twelfth [12] usage form allows you to display a report on the status of configured IPsec VPNs. To display the report use the keyword **report**.

IPSEC VPN ROUTING

The **vpnipsec** command automatically adds required routes for IPsec VPN traffic (and you shouldn’t add them with the **ip** command). The **apply** command checks the compatibility of all routes to install and in case of conflicts it refuses to apply the new configuration. When managing routes and networks with the **ip**

and **vpnipsec** commands please carefully consider the following rules:

• IPsec VPN remote networks and networks in the main routing table (created using the **ip** command) should be distinct (this is also valid for the default network).

• An IPsec VPN local network can be either the default network, a network connected to the internal or auxiliary interface or a non connected network on condition that a route is manually added (using the **ip** command) for that network via a gateway connected to the internal or auxiliary interface.

• At least one remote network should be specified for a site to site IPsec VPN.

• An IPsec VPN remote network can’t be a network connected to the internal, external or auxiliary interface.

IPSEC VPN, ACCESS POLICIES AND THE FIREWALL

IPsec VPN remote networks are subject to security policies as follows:

• In case where the internal network is routed in an IPsec tunnel, remote networks can ping the system’s **internal** network interface (**web** network interface in **vlan** mode) via that that tunnel.

• In case where the internal network is routed in an IPsec tunnel, remote networks can potentially access the system’s embedded proxy, transparent proxy, DNS server and the antivirus. Accesses are controlled by the **access** command (see the the **access** command for further information).

• In case where no rule is defined in the **vpnipsec** firewall rule set, all new connections incoming from the **vpnipsec** zone and destined to the **internal** zone (the **web** zone in **vlan** mode) are allowed (see the the **firewall** command for further information).

IPSEC VPN IN SHORT

An Ipsec VPN tunnel between two peers is built in two phases: in the phase 1, the two participants negotiate the security mechanisms to use in order to establish the VPN tunnel in the phase 2. Negotiation phases use a protocol called IKE (Internet Key Exchange). Please note that the system supports IKE version 2 only. One of the most important step in the phase 1 is the authentication of a peer by the other. With IKEv2 the two peers can use distinct authentication methods. Currently the system supports three authentication methods: the pre shared key (**psk**), the X.509 SSL certificate based (**tls**) and the EAP-TLS (**eaptls**).

Once the two peers has authenticated each other, they should agree on the encryption algorithm, the integrity algorithm and the strength of the key to use in the DH (Diffie Hellman) key exchange process in phase 1. At the end of the phase 1 the two peers will have a shared key that they can use in phase 2 to securely communicate with each other to agree on the encryption algorithm, the integrity algorithm and the strength of the key to use in the DH key exchange process for future transited data between the two peers. With the present system the DH to use in phase 2 is the same as the DH in phase 1.

The present system supports the following encryption algorithms:

• **aes128**: 128 bit AES-CBC

• **aes192**: 192 bit AES-CBC

• **aes256**: 256 bit AES-CBC

• **aes128ctr**: 128 bit AES-COUNTER

• **aes192ctr**: 192 bit AES-COUNTER

• **aes256ctr**: 256 bit AES-COUNTER

Supported integrrity algorithms are as follows:

• **sha1**: SHA1 160 160 HMAC

• **sha256**: SHA2 256 128 HMAC

• **sha384**: SHA2 384 192 HMAC

• **sha512**: SHA2 512 256 HMAC

The Diffie-Hellman key-exchange groups that you can use with the present system are as follows:

• *Regular Modular Prime Groups:*

• **modp1536**: group 5 (1536 bits)

• **modp2048**: group 14 (2048 bits)

• **modp3072**: group 15 (3072 bits)

• **modp4096**: group 16 (4096 bits)

• **modp6144**: group 17 (6144 bits)

• **modp8192**: group 18 (8192 bits)

• *NIST Elliptic Curve Groups:*

• **ecp192**: group 25 (192 bits)

• **ecp224**: group 26 (224 bits)

• **ecp256**: group 19 (256 bits)

• **ecp384**: group 20 (384 bits)

• **ecp521**: group 21 (521 bits)

• *Brainpool Elliptic Curve Groups:*

• **ecp224bp**: group 27 (224 bits)

• **ecp256bp**: group 28 (256 bits)

• **ecp384bp**: group 29 (384 bits)

• **ecp512bp**: group 30 (512 bits)

Algorithms are ordered from the weakest to the strongest from the security perspective. Please note that the stronger the algorithm is, the more computation it requires and hence a more powerful CPU.

IKE uses ISAKMP (an UDP protocol) for the key exchange so an IPsec VPN client or peer should know the IPsec VPN server IP address an ISAKMP port to connect to. Because an IPsec VPN server may use a NATed IP address, IPsec packets can be encapsulated in UDP packets. In IKE this is called NAT-Traversal. The default ISAKMP and NAT-Traversal ports are respectively 500 and 4500. In most cases you do not need to modify default ports but because some internet providers may block those ports you have the possibility to modify them. ISAKMP and NAT-Traversal ports for a remote IPsec VPN peer can be set when adding an IPsec VPN while local ISAKMP and NAT-Traversal ports can be modified using the command **port**.

An IPsec VPN uses the AH (Authentication Header) or the ESP (Encapsulating Security Payload) protocols. AH provides a mechanism for authentication only while ESP provides data encryption in addition. The present system supports the ESP protocol only.

CLIENT DEVICES IN ACCESS MODE SPECIAL NOTES

At the time of writing, in remote access mode, the following restrictions are applied regarding Apple® & Android® devices:

• The Android® native IPsec VPN client is not compatible with the present system. To establish IPsec tunnels between Android® devices and the present system we suggest to use an IPsec VPN client such as strongSwan® App (www.strongswan.org).

• To establish IPsec tunnels between Apple® devices and the present system, the authentication method to use can be "**psk**" or "**tls fqdn**" (**dn** identifier based is not supported). Also please note that only SAN certificates are supported with Apple® devices.

• The Windows® native IPsec VPN client supports the **tls** based authentication method only (**psk** based authentication authentication is not supported).

SEE ALSO

access (1) **apply** (1) **domainname** (1) **email** (1) **firewall** (1) **ip** (1) **link** (1) **mode** (1) **port** (1) **qos** (1) **system** (1) **tls** (1) **vlan** (1) **vrrp** (1)

AUTHOR

CacheGuard Technologies Ltd *<www.cacheguard.com>*

Send bug reports or comments to the above author.

COPYRIGHT

vrrp

NAME

vrrp - Manage the VRRP configuration in HA mode

SYNOPSIS

[1] **vrrp** [(**internal** | **external** | **auxiliary** | **web** | **rweb** | **antivirus**) [**add** <*vrrp-ip*> (**master** | **backup**) [<*priority*> [<*vrrp-id*>]]]]

[2] **vrrp** [(**internal** | **external** | **auxiliary** | **web** | **rweb** | **antivirus**) [**del** <*vrrp-ip*>]]

[3] **vrrp** [(**internal** | **external** | **auxiliary** | **web** | **rweb** | **antivirus**) [**raz**]]

DESCRIPTION

This command allows you to set the VRRP (Virtual Redundancy Router Protocol) IP configuration when using the HA (High Availability) mode (see the **mode** command to activate the HA mode). One or more VRRP IP addresses can be associated to each logical network interface (**external**, the **internal** and the **auxiliary**). VRRP IPs can be also associated to the **web**, **rweb** and **antivirus** 802.1q pseudo network interfaces when using the 802.1q VLAN mode (see the **mode** command to activate the VLAN mode).

The appliance implements the VRRP v2 to assure a High level of Availability. When using VRRP, two (or more) appliance utilise the same virtual IP (VRRP IP) address while each appliance has its own real IP address. Virtual IP addresses are then used by clients accessing the virtual appliance (made by two or more real appliances). For each VRRP IP address, one appliance is configured as the master and other appliances act as backups. In case of the failure of the master appliance, a backup appliance is then elected as the new master.

A classical configuration uses two appliances with two VRRP IP address configured on both. The first VRRP IP address is configured as master on the first appliance while this VRRP IP address is configured as a backup VRRP IP address on the second appliance. The second VRRP IP address is then a backup address on the first appliance and master on the second appliance. In such a configuration, users, backend web servers and any other equipments connected to the appliance should use VRRP IP addresses configured on appliances.

To use both appliances equally, different methods are available. One method consists of using a round robin DNS (appliances are addressed by using a unique name configured on a local DNS and this name is associated to both VRRP IP addresses). Another method uses WPAD (Web Proxy Auto Discovery) script, which automatically configures settings for clients (this script should share the load between both appliances). Finally, a Switch L4 can be used to load balance the traffic on both appliances.

The first [1] usage form allows you to associate a VRRP IP to a logical network interface. To define a VRRP IP address for a network interface, give the network interface name followed by the keyword **add** , the VRRP IP address and its initial state (**master** or **backup**). Note that the same VRRP IP address must be defined on all real appliances creating the virtual appliance but only one appliance must declare that VRRP IP address as master.

The optional <*priority*> argument is a numeric value between 0 and 255 specifying the priority during the master election phase. The appliance with the highest priority value is then elected as the Master. By default, the priority is set to 110 for a master VRRP IP and to 100 for a backup VRRP IP. Take care to define different priorities for the same VRRP IP on each appliance.

The priority value is also used to bind together all logical network interfaces (**internal**, **external**, **auxiliary**, **web**, **rweb** or **antivirus**) so in case of a failure on one of them all associated VRRPs having the same priority will move together. This avoid to have inconsistencies situations where an appliance is active for one configured VRRP IP and passive for others.

A VRRP identifier could be specified as the last argument. When more than one virtual network equipment share the same LAN, the <*vrrp-id*> allows you to specify to which virtual equipment belongs a real

network equipment. The <*vrrp-id*> must be a numeric value between 0 and 255. If no <*vrrp-id*> is given, the last byte of the VRRP IP address is used as the <*vrrp-id*>.

Usage forms two [2] and three [3] allow you delete a VRRP IP or erase the list of all VRRP IPs associated to a logical network interface.

SHARED VLAN-ID RULES

In case where a pseudo network interface shares the same <*vlan-id*> of another pseudo network interface and no VRRP IP address is specified for it, the following rules are applied:

• The pseudo network interface takes the VRRP IP addresses specified for the other pseudo network interfaces having the same <*vlan-id*>.

• If more than one pseudo network interface share the same <*vlan-id*>, the pseudo network interface takes the VRRP IP addresses of the preferred pseudo network interface. Preferences are in the following order: **web**, **rweb** and **antivirus**.

SEE ALSO

apply (1) **mode** (1) **vlan** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

waf

NAME

waf - Configure the Web Application Firewall (WAF)

SYNOPSIS

[1] **waf** [(**score** [(**response** | **request**) [<*score-nb*>]]) | **level** [<*level-nb*>]]

[2] **waf** [**generic** [(**dliis** | **dljava** | **dlphp** | **dlsql** | **java** | **lfi** | **nodejs** | **php** | **rce** | **rfi** | **sf** | **sqli** | **xss**) (**on** | **off**)]]

[3] **waf** [**reputation** [(**country** [**raz** | (**add** | **del**) (<*country-code*>)+]) | (**rbl** [**set** <*api-key*>])]]

[4] **waf** [**limit** [(**response** | **request** | **assertions** | **name** | **value** | **arguments** | **files**) <*limit-value*>]]

[5] **waf** [(**errors** | **imethods**) [(**allow** | **deny**)]]

[6] **waf** [**rweb** [**generic** [<*site-name*> <*generic-filter*> (**on** | **off**)]]]

[7] **waf** [**rweb** [**custom** [<*site-name*> [**clear** | (**load** | **save**) (**ftp** | **sftp** | **tftp**) <*file-server*> <*file-path*> [**new**]]]]]

[8] **waf** [**rweb** [**audit** [<*site-name*> [**on** | **off**]]]]

[9] **waf** [**rweb** [**denyurl** [(**set** <*site-name*> [<*url*>]) | **raz** [<*site-name*>]]]]

[10] **waf** [**rweb** [**errors** [**raz** | **add** <*site-name*> (**allow** | **deny**) | **del** <*site-name*> | **show** [<*site-name*>]]]]

[11] **waf** [**bypass rule** [**raz**] | **rweb bypass** [**rule** [<*site-name*> [**raz** | (**add** | **del**) (<*rule-id*>)+]]]]

[12] **waf** [**bypass application** [**raz**] | **rweb bypass** [**application** [<*site-name*> [**raz** | **set** (**cpanel** | **dokuwiki** | **drupal** | **nextcloud** | **wordpress** | **xenforo**)]]]]

[13] **waf** [**dos** [<*requests-limit*> <*duration-period*> <*blocking-period*>]]

DESCRIPTION

The WAF (Web Application Firewall) blocks malicious requests and prevent data leakages to protect Web applications against unwanted or threatening accesses. This command is used to manage the WAF to protect websites cloaked by the reverse proxy in **rweb** mode (see the **rweb** command). To protect Web applications, you can use generic WAF rules or design your own customised WAF rules.

Generic WAF rules are classified by groups called filters (a filter is composed of WAF rules). Each blocking rule in a generic filter has a score point (an integer) and whenever a rule is matched in a Web traffic, its score point is added to a global score. Once a threshold score is reached for a given Web traffic, the Web traffic is blocked by the WAF. The WAF distinguish two types of traffic: requests and responses. The first [1] usage form allows you to configure the scoring system: use the keyword **score** followed by the keywords **request** or **response** to set score thresholds for Web requests and responses. A valid score threshold is an integer between 1 and 25.

In order to eliminate potential false positive matches (see below), you can temporarily set score thresholds to their highest values and audit all possible requests and responses using the Auditing module (see the**admin** command). Once all false positive matches have been eliminated, you can restore score thresholds to acceptable values in your environment. Recommended score thresholds are 5 and 4 respectively for Web requests and responses.

Rules in generic filters are classified by levels. The higher the WAF level, the more restrictive (paranoic) rules are activated. When you activate the WAF for the first time, it is recommended to use the lowest WAF level (1) in order to avoid false positive matches (see below). The first [1] usage form allows you to configure the WAF level. To configure the WAF level, use the keyword **level** followed by the required level.

A valid level is an integer between 1 and 4.

The second [2] usage form allows you to activate or deactivate global generic WAF filters applicable to all websites. Generic rules are provided by OWASP (https://owasp.org/). Please note that generic rules may produce false positive matching. In this case, you can review your Web application for adjustment instead of deactivating the related generic rule. You can use the Auditing module to inspect Web requests and rule matching (see below).

Here is a brief description of generic WAF rules:

• **dliis**: rules to prevent from IIS applications Data Leakage.

• **dljava**: rules to prevent from Java applications Data Leakage

• **dlphp**: rules to prevent from PHP applications Data Leakage.

• **dlsql**: rules to prevent from SQL requests Data Leakage.

• **java**: rules to protect Java Applications.

• **lfi**: rules to protect against Local File Inclusion attacks.

• **nodejs**: rules to protect Node.js Applications.

• **php**: rules to protect PHP Applications.

• **rce**: rules to protect against Remote Code Execution attacks.

• **rfi**: rules to protect against Remote File Inclusion attacks.

• **sf**: rules to protect against Session Fixation attacks.

• **sqli**: rules to protect against SQL Injection attacks.

• **xss**: rules to protect against Cross-site Scripting attacks.

You can find more information about OWASP CRS at https://owasp.org/

The third [3] usage form allows you to block Web requests according to the client reputation. Two types of reputation are available: reputation based on countries and reputation based on IP Real Time Blacklists (RBL). RBL are provided by the Project Honey Pot. To benefit from IP blacklists you must have an API key. Please refer to *https://www.projecthoneypot.org/* to get a free API key. Country based reputation use data provided by MaxMind (*https://www.maxmind.com/*). Country data are automatically updated on a monthly basis.

To block all Web requests coming from a particular country use the keywords **reputation country add** followed by the ISO 3166-1 alpha-2 code of that country. To block multiple countries you can specify a list of country code separated by a blank or use this command usage form several times. You can use the keyword **del** to disable the blocking for countries. The keyword **raz** allows you to disable all country based blockings. To activate the reputation based on RBL use the keywords **reputation rbl set** followed by your private API key. You can deactivate this feature by using an ampty string as the API key.

The fourth [4] usage form allows you to specify limits for all Web requests and responses. These limits are applied globally to all responses and requests. The following limits can be configured:

• **response**: maximum response body size in KB.

• **request**: maximum request body size in KB, excluding the size of any files being transported in the request.

• **assertions**: maximum number of assertions (attributes=values) or arguments in a request (the separator should be '&').

• **name**: maximum length for an argument name in a request.

• **value**: maximum length for an argument value in a request.

• **arguments**: total arguments (names and values) length limit in a request.

• **files**: maximum size in KB for combined uploaded files. This value can't be greater than the value given during the installation for uploaded files.

The fifth [5] usage form allows you to globally configure some WAF policies. The keyword **errors** allows you to expose or rewrite original error pages sent by websites. By an error page we mean a page sent with an HTTP status code other than 200. To allow the exposure of original error pages use the keyword **allow**. To rewrite original error pages use the keyword **deny**. The keyword **imethods** allows you to allow or deny insecure HTTP methods. Insecure HTTP methods are "PUT", "PATCH", "DELETE", "CONNECT" and "TRACE". To allow insecure HTTP methods use the keyword **allow**. To deny insecure HTTP methods use the keyword **deny**.

For security reasons, even if you globally allow insecure HTTP methods, you should bypass some OWASP rules using the **waf rweb bypass** usage form and then create custom rules to allow them. Please note that by default, insecure HTTP methods are not allowed and if you decide to globally allow them, you should know what you are doing. Otherwise you expose your Web applications to severe threats.

The sixth [6] usage form allows you to activate or deactivate generic filters for a specific website. A website that no specific generic filters are defined for inherits global filters.

The seventh [7] usage form allows you to load/save custom rule files from/to a file server. Custom rules allow you to have restrictive controls on allowed or denied requests on a website. The "GET", "HEAD", "POST", "PUT", "PATCH", "DELETE", "CONNECT", "OPTIONS" and "TRACE", HTTP methods can be filtered by defining regular expressions for allowed or denied requests.

The process of custom rule definition is very easy. To do so, just create your rule file for a specific website and then load it into the system. Because regular expressions may be complex, this process allows you to use your favourite text editor (vi, emacs...) to edit the rules file.

In the custom rule file, each rule definition begins with a line having the following syntax:

rule <*rule-name*> (**allow** | **deny**) <*method*>

Where <*method*> is a valid HTTP method (**get**, **head**, **post**, **put**, **patch**, **delete**, **connect**, **options**, **trace**) in lower-case or a list of valid HTTP methods separated by the pipe charcter.

Followed by optionnal lines having the following syntax:

(**uri** | **body** | **ip**) "<*regular-rexpression*>"

Keywords and arguments must be separated with blanks or tabulations delimiters.

If a line begins with the **rule** statement, three mandatory arguments must be specified:

The first argument is the rule identifier. It allows you to identify the rule in the Auditing module (see below). A rule identifier must be a combination of alphanumeric characters and the characters "_", "-" or "." (the dot) and begin with an alphanumeric character.

The second argument specifies the action (**allow deny**) and finally the third argument is the HTTP method in lowercase. Optional lines begins with the following keywords:

• **uri**: The part after the "/" character in a URL (can be used with POST and GET methods)

• **body**: Arguments in the body of a POST (can be used only in conjunction with a **uri** statement and POST method)

• **ip**: the source IP address of the client making the Web request

followed by a regular expression between quotation marks. Note that a quotation mark in a regular expression **should not** be preceded by a back slash character here. Regular expressions are based on PCRE (Perl Compatible Regular Expression).

For instance to allow only the GET on "/" and the POST on "/cgi-bin/set-phone.cgi" with arguments "name=<string> phone=<numbers>" the custom rule file looks like:

```
rule r10 allow get
uri "^/$"

rule r20 allow get
uri "^/cgi-bin/get-phone\.cgi"

rule r30 allow post
uri "^/cgi-bin/set-phone\.cgi$"
body "^name=[[:print:]]*\&phone=[[:digit:]]*$"
ip "^192\.168\.155\.254$"

rule r40 deny get|post
uri "^/cgi-bin/set-phone\.cgi$"

rule r1000 deny get|head|post|put|patch|delete|connect|options|trace
```

After a custom rule file is loaded, its syntax is verified and the loading is confirmed only if no errors are detected. Note that generic filters are always applied (when activated) *before* custom rules.

The eighth [8] usage form allows you to manage the **audit** mode for a website. Auditing allows you to inspect HTTP/S request contents and facilitate the filtering rule design process. To activate the audit mode for a website use the keyword **on**. To deactivate the audit mode use the keyword **off**. Without specifying a state (**on** or **off**), this command prints the audit mode for a website (or for all websites if no website is specified).

Caution: Note that the auditing feature is for debugging purpose only and in normal

circumstances, it should not be activated.

When the audit mode is activated for a website and when the administration audit mode is turned on (see the **admin** command), all related Web request contents can be inspected at the URL *https://<admin-ip>:<wadmin-port>*.

In audit mode, an HTTP request header and body (for the POST method), the matched filtering rule and the resulting state (allowed or denied) are logged. In this way, the security staff can easily design custom filtering rules for managed websites.

Note that the audit mode is helpful during the filtering rule design process and should not be activated on a production appliance. Auditing consumes lots of hardware resources and, in terms of security, is not recommended for production appliances.

When the audit mode is deactivated for a website, all auditing data for that website is lost.

By default when access is denied, a generic deny page is displayed. The ninth [9] usage form allows you to set a custom deny page to redirect to a website. To set a deny URL page for a website, use the keyword **denyurl** followed by the keyword **set**, the website name and the URL you want to redirect to. To reset to the default behaviour use the keyword **raz** followed by the website name. To reset to the default behaviour for all websites give no website name. When no optional arguments are used, this usage form displays the current settings. If a URL is given it should be in the form: (http|https|ftp)://<domain-name>[/<URI>] where an URI may contain an alphanumeric or any of the following characters: *-._~:/?#[]@!$&()*+,;=*. Any other character needs to be encoded with the percent-encoding. A percent-encoding is in the form *%(a-fA-F0-9)(a-fA-F0-9)* (use %27 for the quote character).

Note: When both the **waf** and the **antivirus** modes are activated, the system filters all attempts to upload malware files on protected Web servers (usually files are uploaded using the post method and the *multipart/form-data* encryption type).

The tenth [10] usage form allows you specifically set the exposure or rewriting of original error pages for a website. If no specific rule is defined for a website the global behaviour is used (see the fifth usage form). Keywords **add**, **del**, **show** and **raz** allows you respectively add a rule, delete a rule, show a rule and erase all rules. To allow the exposure of original error pages use the keyword **allow**. To rewrite original error pages use the keyword **deny**.

OWASP CRS may generate false positive matches. The term false positive refers to legitimate Web traffic denied by the WAF. To avoid this situation you can review your Web application source code in order to bypass false positive matches. But in most cases this is not an option because you don’t own the application or just don’t want to modify things. In this case you have the possibility to bypass the rule causing the false positive match. To do so you can use the eleventh [11] and twelfth [12] usage forms. To bypass an OWASP rule for a given website use the keywords **rweb bypass rule** followed by the website name, the keyword **add** and one or more rule IDs to bypass. To remove a bypass statement use the keyword **del** instead of the keyword **add**. To remove all bypass statements for a given website use the keyword **raz**. To remove all bypass statements for all websites use the keyword **bypass rule** followed by the keyword **raz**. Please note that the rule ID is a numerical value and is different than the rule name given for custom rules. You can find the ID of the rule causing false positive match by using the Auditing module. See the command **admin waudit** for further information on the Auditing module.

Some well-known applications such as WordPress or Drupal are known to produce false positive matches. To bypass all known rules that produce false positive matches for a given application you can use the twelfth [12] usage form. Supported applications are as follows:

• cPanel

• DokuWiki

• Drupal

• Nextcloud

• WordPress

• XenForo

The WAF has the ability to protect Web applications against DoS (Denial of Service) by blocking clients that make requests (excluding static files) above a certain limit. The thirteenth [13] usage form allows you to tune the DoS protection by specifying how many requests is allowed by a client (<*requests-limit*>) in a given duration (<*duration-period*>) and how long block (<*blocking-period*>) that client. Periods should be specified in seconds. For instance values *120 300 180fR would mean that if a client makes more than 300 requests in 120 seconds, he will no longer be able to submit requests for 180 seconds. Allowed values for arguments are as follows: <requests-limit>* is an integer between 1 and 1000000, <*duration-period*> is an integer value between 1 and 86400 and <*blocking-period*> is an integer value between 1 and 604800.

SEE ALSO

admin (1) **antivirus** (1) **apply** (1) **countrylist** (1) **domainname** (1) **file** (1) **hostname** (1) **ip** (1) **mode** (1) **port** (1) **rweb** (1) **vlan** (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT

warning

NAME

warning - Display descriptions of warning codes returned by commands

SYNOPSIS

[1] **warning** [<*warning-code*>]

DESCRIPTION

Use this command to display the description of all warning codes return by commands. You can also display the description of a particular warning by giving its code (<*warning-code*>). This command can be very useful if for any reasons displayed warnings are not fully displayed by commands.

SEE ALSO

manager (1)

AUTHOR

CacheGuard Technologies Ltd <*www.cacheguard.com*>

Send bug reports or comments to the above author.

COPYRIGHT