Virtual Router Appliance Best Practices

Virtual Router Appliance (VRA) provides the high performance, ease of configuration, and maintenance advantages of running on a normal hardware server. The hardware is sized to handle the routing load for multiple VLANs, and can be ordered with redundant network links and redundant RAID arrays. For an overview of the benefits and functionality of the Virtual Router Appliances, please view the following article: http://knowledgelayer.softlayer.com/topic/virtual-router-appliance.

As the administrator of your network gateway, you are able to configure the gateway to enable the connectivity you want for the VLANs that you elect to route through the VRA. Some of the more prominent options include:

- Firewall incoming traffic from Internet for public VLANs.
- Zone policy between different VLANs
- NAT traffic between private VLANs and public VLANs
- Configure IPSec or GRE VPN tunnels and routing between them
- Shape traffic on VLANs

Terminology

- Configuration mode: Invoked with the use of the "**configure**" command, configuration mode occurs when configuration of the VRA system is performed.
- Operational mode: The initial mode upon logging into a VRA system where "**show**" commands can be run to query information on the running state of the system. The system can also be restarted from Operational mode.

Basics

The VRA system is configured using either the web GUI or a remote console session through SSH. By default, web GUI is disabled from the public Internet. This guide will focus on configuration using the remote SSH console. Configuring the VRA outside of the VRA shell and interface may produce unexpected results.

Accessing the device

To initially access the VRA, an SSH client is necessary. Most Unix-based operating systems, such as Linux, BSD, and Mac OSX typically have OpenSSH clients included with their default installations. Windows users will need to download an SSH client such as PuTTy for use with the system.

In the Customer Portal, you can find the IP addresses, login username, and password for the device. You will need to use the ‘vyatta’ account to login via SSH, as root access is disabled for SSH logins by default.

```
ssh vyatta@[IP address]
```

Once you've logged in, the SSH client will be connected to the system via the VRA shell. Please note that connections to private IPs will generally require connecting either from another machine you have, or by connecting to our backend management VPN. Also note that rather than editing the SSH configuration file, VRA installations can use the following command to enable root access via SSH:

```
set service ssh allow-root
```

Command exploration

The VRA command shell includes tab completion capabilities. If you are curious about what commands are available, you can press the tab key for a list showing the possible commands, along with short explanations of the commands. This works at the shell prompt and while typing a command. See the following example:

```
$show log dns [Press tab]
Possible completions:
  dynamic    Show log for Dynamic DNS
  forwarding Show log for DNS Forwarding
```

Operational mode / Configuration mode

The VRA's shell is a modal interface, with several modes of operation. The primary/default mode is "Operational", which is the mode presented upon login. In operational mode, you can view information and issue commands which impact the current run of the system, such as setting the date/time, and rebooting of the device.

The command "**configure**" will place the user into Configuration mode, where edits to the configuration can be performed. It is important to understand that configuration changes do not take place immediately; rather, they must be committed and saved. As commands are entered, they go into a configuration buffer. Once all the necessary commands are entered, you will need to run the "**commit**" command to make those changes live.

To save commands permanently in the configuration file it is necessary to run the "**save**" command after the "**commit**" command.

Operational mode commands can be run from configuration mode by starting the command with the word `run`. See the following example:

```
# run show configuration commands | grep name-server
set system name-server '10.0.80.11'
set system name-server '10.0.80.12'
```

The hash mark (#) indicates configuration mode. Beginning a command with the word "**run**" indicates to the VRA shell that an operational command is being presented. This example also shows the feature of being able to "grep" against the output of commands.

Sample Configuration

The configuration is laid out in a hierarchical pattern of nodes. Consider the following static route block:

```
protocols {
     static {
         route 10.0.0.0/8 {
             next-hop 10.60.63.193 {
             }
         }
         route 192.168.1.0/24 {
             next-hop 10.0.0.1 {
             }
         }
     }
 }
```

This static route block is generated by the commands:

```
set protocols static route 10.0.0.0/8 next-hop 10.0.0.1
set protocols static route 192.168.1.0/24 next-hop 10.0.0.1
```

This example illustrates that it is possible for a node (static) to have multiple child nodes. To remove the route for 192.168.1.0/24, the command `delete protocols static route 192.168.1.0/24` would be used. If 192.168.1.0/24 was left off of the command, then both of the route nodes would be marked for deletion. Remember, the configuration is not actually changed until the "**commit**" command is issued. To compare the current running configuration to any changes that are present in the configuration buffer, the "**compare**" command may be used. To flush the configuration buffer, use the "**discard**" command.

Users and Role-based Access Control (RBAC)

User accounts can be configured with three levels of access: Admin, Operator and Superuser. Operator level users are able to run show commands to view the running status of the system and issue reset commands to restart services on the device. Operator level permissions do not imply read-only access. Admin level users have full access to all configurations and operations for the device. Admin users can view running configurations, change configuration settings for the device, reboot the device, and shutdown the device. Superuser level users are able to execute commands with root privileges through the sudo command in addition to having admin level privileges.

The users can be configured for password, public key or both authentication styles. Public key authentication is used with SSH and allows the user to login using a key file on their system. To create an operator user with a password, run the following commands:

```
set system login user [account] authentication plaintext-password [password]
set system login user [account] level operator
commit
```

Note: Without the level specified a user is considered an Admin level user. The password (though entered in the clear) will display as encrypted in the configuration file.

Role-based Access Control (RBAC) is a method of restricting access to part of the configuration to authorized users. RBAC allows an admin to define the rules for a group of users that restrict which commands users of that group are allowed to run.

To perform RBAC you will first create a group assigned to the Access Control Management (ACM) rule set. Next, you will add a user to the group, creating a rule set to match the group to the paths in the system. Then, you'll configure the system to allow or deny the paths that are applied to the group.

By default, there is no ACM rule set defined in VRA, and ACM is disabled. If you want to use RBAC to provide granular access control, you need to enable the ACM and **MUST** add the following default ACM rules in addition to your own defined rules:

```
set system acm 'enable'
set system acm operational-ruleset rule 9977 action 'allow'
set system acm operational-ruleset rule 9977 command '/show/tech-
support/save'
set system acm operational-ruleset rule 9977 group 'vyattaop'
set system acm operational-ruleset rule 9978 action 'deny'
set system acm operational-ruleset rule 9978 command '/show/tech-
support/save/*'
set system acm operational-ruleset rule 9978 group 'vyattaop'
set system acm operational-ruleset rule 9979 action 'allow'
set system acm operational-ruleset rule 9979 command '/show/tech-
support/save-uncompressed'
set system acm operational-ruleset rule 9979 group 'vyattaop'
set system acm operational-ruleset rule 9980 action 'deny'
set system acm operational-ruleset rule 9980 command '/show/tech-
support/save-uncompressed/*'
set system acm operational-ruleset rule 9980 group 'vyattaop'
set system acm operational-ruleset rule 9981 action 'allow'
set system acm operational-ruleset rule 9981 command '/show/tech-
support/brief/save'
set system acm operational-ruleset rule 9981 group 'vyattaop'
set system acm operational-ruleset rule 9982 action 'deny'
set system acm operational-ruleset rule 9982 command '/show/tech-
support/brief/save/*'
set system acm operational-ruleset rule 9982 group 'vyattaop'
set system acm operational-ruleset rule 9983 action 'allow'
set system acm operational-ruleset rule 9983 command '/show/tech-
```

```
support/brief/save-uncompressed'
set system acm operational-ruleset rule 9983 group 'vyattaop'
set system acm operational-ruleset rule 9984 action 'deny'
set system acm operational-ruleset rule 9984 command '/show/tech-
support/brief/save-uncompressed/*'
set system acm operational-ruleset rule 9984 group 'vyattaop'
set system acm operational-ruleset rule 9985 action 'allow'
set system acm operational-ruleset rule 9985 command '/show/tech-
support/brief/'
set system acm operational-ruleset rule 9985 group 'vyattaop'
set system acm operational-ruleset rule 9986 action 'deny'
set system acm operational-ruleset rule 9986 command '/show/tech-
support/brief'
set system acm operational-ruleset rule 9986 group 'vyattaop'
set system acm operational-ruleset rule 9987 action 'deny'
set system acm operational-ruleset rule 9987 command '/show/tech-support'
set system acm operational-ruleset rule 9987 group 'vyattaop'
set system acm operational-ruleset rule 9988 action 'deny'
set system acm operational-ruleset rule 9988 command '/show/configuration'
set system acm operational-ruleset rule 9988 group 'vyattaop'
set system acm operational-ruleset rule 9989 action 'allow'
set system acm operational-ruleset rule 9989 command '/clear/*'
set system acm operational-ruleset rule 9989 group 'vyattaop'
set system acm operational-ruleset rule 9990 action 'allow'
set system acm operational-ruleset rule 9990 command '/show/*'
set system acm operational-ruleset rule 9990 group 'vyattaop'
set system acm operational-ruleset rule 9991 action 'allow'
set system acm operational-ruleset rule 9991 command '/monitor/*'
set system acm operational-ruleset rule 9991 group 'vyattaop'
set system acm operational-ruleset rule 9992 action 'allow'
set system acm operational-ruleset rule 9992 command '/ping/*'
set system acm operational-ruleset rule 9992 group 'vyattaop'
set system acm operational-ruleset rule 9993 action 'allow'
set system acm operational-ruleset rule 9993 command '/reset/*'
set system acm operational-ruleset rule 9993 group 'vyattaop'
set system acm operational-ruleset rule 9994 action 'allow'
set system acm operational-ruleset rule 9994 command '/release/*'
set system acm operational-ruleset rule 9994 group 'vyattaop'
set system acm operational-ruleset rule 9995 action 'allow'
set system acm operational-ruleset rule 9995 command '/renew/*'
set system acm operational-ruleset rule 9995 group 'vyattaop'
set system acm operational-ruleset rule 9996 action 'allow'
set system acm operational-ruleset rule 9996 command '/telnet/*'
set system acm operational-ruleset rule 9996 group 'vyattaop'
set system acm operational-ruleset rule 9997 action 'allow'
set system acm operational-ruleset rule 9997 command '/traceroute/*'
set system acm operational-ruleset rule 9997 group 'vyattaop'
set system acm operational-ruleset rule 9998 action 'allow'
set system acm operational-ruleset rule 9998 command '/update/*'
set system acm operational-ruleset rule 9998 group 'vyattaop'
set system acm operational-ruleset rule 9999 action 'deny'
set system acm operational-ruleset rule 9999 command '*'
set system acm operational-ruleset rule 9999 group 'vyattaop'
set system acm ruleset rule 9999 action 'allow'
set system acm ruleset rule 9999 group 'vyattacfg'
set system acm ruleset rule 9999 operation '*'
set system acm ruleset rule 9999 path '*'
```

It is recommended that you read and understand the Brocade 5600 RBAC User Guide first before you try to enable ACM rules. Inaccurate ACM rule setting can cause a device access deny, or can even cause a system to be inoperable.

To read the Brocade 5600 RBAC User Guide, please refer to the **Role-based Access Control** section in this document.

Routing VLANs

The VRA device is able to route multiple VLANs over the same network interface (i.e dp0bond0 or dp0bond1). This method is accomplished by setting the switch port into trunk mode and configuring virtual interfaces (VIFs) on the device. For example:

```
set interfaces bonding dp0bond0 vif 1432 address 10.0.10.1/24
set interfaces bonding dp0bond0 vif 1693 address 10.0.20.1/24
```

The previous commands create two virtual interfaces on the `dp0bond0` interface. The interface `dp0bond0.1432` would process traffic for VLAN 1432, while the interface `dp0bond0.1693` would process traffic for VLAN 1693.

Services

There are a variety of services running on VRA that can be configured, including:

```
vyatta@vrouter# set service
Possible Completions:
 > connsync        Connection tracking synchronization (conn-sync) service
 > dhcp-relay      Dynamic Host Configuration Protocol (DHCP) relay agent
 > dhcp-server     Dynamic Host Configuration Protocol (DHCP) for DHCP server
 > dhcpv6-relay    DHCPv6 Relay Agent parameters
 > dhcpv6-server   DHCP for IPv6 (DHCPv6) server
 > dns             Domain Name Server (DNS) parameters
 > flow-monitoring Flow-Monitoring traffic monitoring configuration
 > https           Enable/disable the Web server
 > lldp            LLDP settings
 > nat             Network Address Translation (NAT)
 > netconf         NETCONF (RFC 6241)
 > portmonitor     Portmonitor configuration
 > sflow           sflow configuration for dataplane
 > snmp            Simple Network Management Protocol (SNMP)
 > ssh             Secure Shell (SSH) protocol
 > telnet          Enable/disable Network Virtual Terminal Protocol (TELNET)
protocol
 > twamp           Two-Way Active Measurement Protocol
```

Noticeably, NAT is configured as part of service. Https controls access to WebGUI and API access. Connsync defines how two VARs can share connection tracking synchronization for things like stateful firewall failover.

Firewall

The VRA has the ability to process firewall rules on the device to protect the VLANs routed through the device. The firewalls in VRA can be broken into 2 steps: 1) defining one or more sets of rules, and 2) applying a set of rules to an interface or a zone. A zone consists of one or more network interfaces.

It is important to test firewall rules after creation to ensure that the rules you have applied work as intended. It is also important to verify that new rules do not restrict administrative access to the device. While manipulating rules on the `dp0bond1` interface, it is advisable to connect to the device via `dp0bond0`. Connecting to the console via the IPMI is an alternate option.

Stateless vs stateful

By default, the firewall is stateless, but it can be configured stateful if needed. A stateless firewall will need rules for traffic in both directions, while a stateful firewall is for connections where you only need rules for inbound traffic. To configure a stateful firewall, you need to specify related or established connections for outbound traffic.

To Make the firewall stateful globally, run the following commands:

```
set security firewall global-state-policy icmp
set security firewall global-state-policy tcp
set security firewall global-state-policy udp
```

Alternatively, you can set stateful in an individual firewall rule:

```
set security firewall name TEST rule 1 allow
set security firewall name TEST rule 1 state enable
```

Firewall rule sets

Firewall rules are grouped together into named sets of rules to make applying the same rules to multiple interfaces more simple. Each rule set has a default action associated with it. Consider the default action as a rule at the end of the set, which will accept or drop any traffic that was not otherwise resolved by the prior rules in the set. Consider the following example:

```
set security firewall name ALLOW_LEGACY default-action accept
set security firewall name ALLOW_LEGACY rule 1 action drop
set security firewall name ALLOW_LEGACY rule 1 source address network-group1
set security firewall name ALLOW_LEGACY rule 2 action drop
set security firewall name ALLOW_LEGACY rule 2 destination port 23
```

```
set security firewall name ALLOW_LEGACY rule 2 log
set security firewall name ALLOW_LEGACY rule 2 protocol tcp
set security firewall name ALLOW_LEGACY rule 2 source address network-group2
```

In the ruleset, `ALLOW_LEGACY`, there are two rules defined. The first rule drops any traffic sourced from an address group named network-group1. The second rule drops (discards) and logs any traffic destined for the telnet port (tcp/23) for traffic from the address group named network-group2. The default-action indicates that anything else is accepted.

Allowing datacenter access

IBM offers several IP subnets, which are used to provide services and support to systems running within the data center. For example, DNS resolver services are running on 10.0.80.11 and 10.0.80.12. Other subnets are used during the provisioning and support processes. You can find the IP ranges used in the datacenters at the following location: SoftLayer Network Addresses

Allowing access is accomplished by placing the proper `SERVICE-ALLOW` rules at the beginning of the firewall rule sets, with an action of "accept." Where precisely the rule set has to be applied depends on the routing and firewall design that is being implemented.

It is recommended that you place the firewall rules in the location that causes the least duplication of efforts. For example, allowing backend subnets inbound on dp0bond0 would require less work than allowing backend subnets outbound toward each VLAN virtual interface.

Per-interface firewall rules

One method for configuration the firewall on a VRA is to apply firewall rule sets to each interface. In this case an interface can be a data plane interface (`dp0s0`) or a virtual interface (`dp0bond0.303`). Each interface has three possible firewall assignments:

in	The firewall is checked against packets are entering via the interface. These packets can be traversing the VRA and destined for the VRA.
out	The firewall is checked against packets are leaving via the interface. These packets can be traversing the VRA or originating on the VRA.
local	The firewall is checked against packets that are destined directly for the VRA.

An interface can have multiple rule set applied on each direction. They are applied in the order of configuration. Note that it is not possible to firewall traffic originating from the VRA device itself using per-interface firewalls.

To assign the ALLOW_LEGACY rule set to the IN option for the `bp0s1` interface you would use the following configuration command:

```
set interfaces dataplane dp0s1 firewall in ALLOW_LEGACY
```

Control Plane Policing (CPP)

Control plane policing (CPP) provides protection against attacks on the VRA by allowing you to configure firewall policies that are assigned to your desired interfaces, and applying these policies to packets both entering and leaving the VRA.

CPP is implemented when the local keyword is used in firewall policies that are assigned to any type of VRA interface (i.e data plane interfaces or loopback). Unlike the filter rules applied for packets traversing through VRA, the default action of filter rules for traffic entering or leaving control plane is 'Allow'. You will need to add explicit drop rules if the default behavior is not desired.

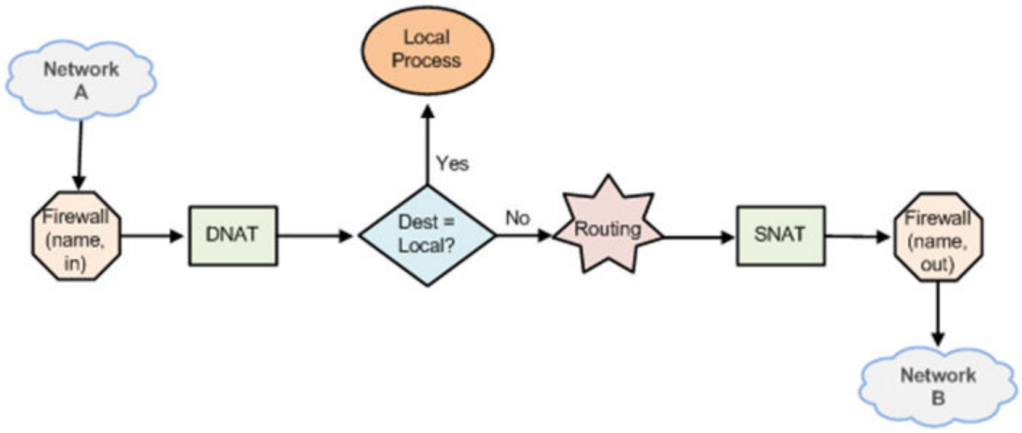
VRA provides a basic CPP rule set as template. You can merge it into your own configuration by running following command:

```
vyatta@vrouter# merge /opt/vyatta/etc/cpp.conf
```

After this rule set is merged, a new firewall rule set named 'CPP' will be added and applied to the lookback interface. It's recommend to modify this rule set suits your environment.

The following figure displays how firewall rules are applied while traffic flowing through VRA:

FIGURE 1 Traffic flow through firewall, NAT, and routing components



Zone firewalling

Another firewall concept in the VRA is zone-based firewalls. In a zone-based firewall operation, an interface is assigned to a zone (only one zone per interface). Firewall rule sets are also

assigned to the boundaries between zones, with the idea that all interfaces within a zone have the same security level and are allowed to route freely. Traffic is only scrutinized when it passes from one zone into another. Zones drop any traffic coming into them, which is not explicitly allowed.

An interface can either belong to a zone or have a per-interface firewall configuration; an interface cannot do both. Consider the following office scenario with 3 departments, where each department has its own VLAN:

Department A - VLANs 10 and 20 (interface `dp0bond1.10` and `dp0bond1.20`)
Department B - VLANs 30 and 40 (interface `dp0bond1.30` and `dp0bond1.40`)
Department C - VLAN 50 (interface `dp0bond1.50`)

A zone can be created for each department, and the interfaces for that department can be added to the zone. This is displayed in the following example:

```
set security zone-policy zone DEPARTMENTA interface dp0bond1.10
set security zone-policy zone DEPARTMENTA interface dp0bond1.20
set security zone-policy zone DEPARTMENTB interface dp0bond1.30
set security zone-policy zone DEPARTMENTB interface dp0bond1.40
set security zone-policy zone DEPARTMENTC interface dp0bond1.50
```

The "**commit**" command will populate each zone with different interfaces, and the default drop rules will discard any traffic trying to enter the zone from the outside of the zone. In the above example, VLANs 10 and 20 can pass traffic since they are in the same zone (DEPARTMENT A), but VLAN 10 and VLAN 30 cannot pass traffic because VLAN 30 is in a different zone (DEPARTMENT B).

The interfaces within each zone can pass traffic freely, and rules can be defined for interaction between the zones. A rule set is configured from the point of view of leaving one zone to another zone. The following command shows an example of how to configure a rule:

```
set security zone-policy zone DEPARTMENTC to DEPARTMENTB firewall ALLOW_PING
```

This command associates the transition from DEPARTMENT C to DEPARTMENT B, with the rule set named `ALLOW_PING`. Traffic entering the DEPARTMENT B zone from the DEPARTMENT C zone would be checked against this rule set. It is important to understand that this assignment from zone DEPARTMENT C, going into zone DEPARTMENTB, does not make any statement about the inverse. If there are no rules allowing traffic from zone DEPARTMENT B into zone DEPARTMENT C, then traffic (ICMP replies) will not get back to the hosts in DEPARTMENT C.

Session and packet logging

The VRA supports two types of logging:

1. Session logging. Use the `security firewall session-log` command to configure firewall session logging.

   For UDP, ICMP, and all non-TCP flows, a session transitions to four states over the lifetime of the flow. For each transition, you can configure the VRA to log a message. TCP flows have a greater number of state transitions, each of which can be configured to log.

2. Per packet logging. Include the keyword `log` in the firewall or NAT rule to log every network packet that matches the rule.

   Per-packet logging occurs in the packet-forwarding paths and generates large amounts of output. It can greatly reduce the throughput of the VRA and dramatically increase the disk space used for the log files. We recommend use per packet logging ONLY for the purpose of debugging. For all operational purposes, stateful session logging should be used.

High Availability (HA)

The VRA supports Virtual Router Redundancy Protocol (VRRP) as a high availability (HA) protocol. Your deployment of the devices should be in an active/passive manner, where one machine is the master and the other half of the HA pair is the backup device. All interfaces on both machines are configured to be members of the same "sync-group", so that if one interface experiences a fault, the other interfaces in the same group will also fault, and the device will stop being the master. The current backup will detect that the master is no longer broadcasting the keepalive/ heartbeat message, and will assume control of the VRRP virtual IPs and become the master.

VRRP is the most import part of configuration when you provision gateways. Given that the master broadcasts a heartbeat message that it is master, and the failover mechanism depends on the backup hearing these messages, it is vital that these heartbeat message are not blocked—otherwise, this deployment would not work effectively.

VRRP virtual IP (VIP) addresses

The VRRP virtual IP, or VIP, is the floating IP address changed from master to backup device when failover happens. When we deploy a VRA, it will have a public and private bonded network connection and real IPs is assigned on each interface. Additionally, you should be assigned a VIP on both interfaces, regardless of whether your device is a standalone or in an HA pair. Traffic that has a destination IP in subnets, in VLANs associated with the VRA, will be sent directly to these VRRP VIPs.

VRRP virtual IP addresses for any gateway group should not be changed, nor should the VRRP interface be disabled. These IP addresses are how traffic is routed to the gateway when

a VLAN is associated; therefore, if the IP address is not present, the traffic cannot be forwarded from SoftLayer BCR/FCR to the gateway itself.

VRRP group

A VRRP group consists of a cluster of interfaces or virtual interfaces that provide redundancy for a primary, or "master," interface in the group. Each interface in the group is typically located on a separate router. Redundancy is managed by the VRRP process on each system. The VRRP group has a unique numeric identifier, and can be assigned up to 20 virtual IP addresses.

The VRRP group id is assigned by SoftLayer, and should not be changed. When a new gateway group is provisioned behind a Front Customer Router (FCR)/Backend Customer Router (BCR) for the first time, it will receive a VRRP group of 1. Subsequent gateway group provisions will increment this value to prevent conflicts, so the next group will have group 2, followed by group 3, and so on. This group id value is calculated and assigned by the SoftLayer provisioning process. Altering this value risks colliding with other active groups, and master/master contention, which will likely cause an outage on both gateway groups.

If you migrate from a previous configuration, it's recommended that you double-check your configuration code to make sure that your VRRP group id is not statically assigned.

The VRRP group id is persisted in the SoftLayer database, so the same group id value will be used during an "OS reload" or "OS upgrade." Any user-modified VRRP group id will be overwritten with system-assigned value during an "OS reload."

VRRP priority

The first machine in a gateway group has a priority of 254, and this value is decremented for the next device provisioned. The priority should never be set to 255, as this defines the VIP "address owner," and can result in unintended behavior when the machine is brought online with a configuration that differs from the running active master.

VRRP preemption

Preemption should always be disabled on all VRRP interfaces, so that a new device or one that is being "OS reloaded" does not try to take over the cluster.

VRRP advertise interval

To signal that it is still in service, the master interface or vif sends "heartbeat" packets called "advertisements" to the LAN segment, using the IANA assigned multicast addresses for VRRP (`224.0.0.18` for IPv4 and `FF02:0:0:0:0:0:0:12` for IPv6). By default, the interval is set to 1. This value can be increased, but it is not recommended to use a value over 5. On a busy

network, it may require significantly more than one second for all VRRP notices to arrive on the backup device from the master, so a value of 2 can be used for high traffic networks.

VRRP synchronization group ("sync-group")

Interfaces in a VRRP synchronization group ("sync group") are synchronized such that, if one of the interfaces in the group fails over to backup, all interfaces in the group fail over to backup. For example, in many cases, if one interface on a master router fails, the whole router fails over to a backup router.

This value is different than the VRRP group, as it defines which interfaces on a device will fail over when an interface in that group registers a fault. It is recommended that all interfaces belong to the same sync-group, otherwise, some interfaces will be master and have gateway IPs, while others will be backup, and traffic will not forward properly anymore. Active/active configurations are not supported.

VRRP rfc-compatibility

Your sync group enables or disables RFC 3768 MAC addresses for VRRP on an interface. This should be enabled on the native interfaces, but not enabled on any configured VIFs. Some virtual switches (VMware, mostly) have problems with this being enabled, which will cause traffic to get dropped and not sent to the gateway IP from the hypervisor host machine. Leave this setting alone, and do not configure the setting for any VIFs.

VRRP authentication

If a password is set for VRRP authentication, the authentication type must also be defined. If the password is set and the authentication type is not defined, the system generates an error when you try to commit the configuration. Similarly, you cannot delete the VRRP password without also deleting the VRRP authentication type. If you do, the system generates an error when you try to commit the configuration. If you delete both the VRRP authentication password and authentication type, VRRP authentication is disabled.

The IETF decided that authentication is not to be used for VRRP version 3. For more information, refer to RFC 5798

VRRP version 2/3 support

VRA supports both VRRP version 2 (default) and version 3 protocol. Version 2 does not support the use of IPv4 and IPv6 together. However, in Version 3, you can use IPv4 and IPv6 at the same time.

High Availability VPN with VRRP

The VRA provides the ability to maintain connectivity through one IPsec tunnel using a pair of VRA with VRRP. When one router fails or is brought down for maintenance, the new VRRP master router restores IPsec connectivity between the local and remote networks.

When configuring a High Availability VPN with VRRP, you should consider that whenever a VRRP virtual address is added to a VRA interface, you must reinitialize the IPsec daemon. This is because the IPsec service listens only for connections to the addresses that are present on the VRA when the Internet Key Exchange (IKE) service daemon is initialized.

You will need to run the following command on the master and backup routers, so that when failover happens, IPsec daemons will be restarted on new master device after VIP transiting in, as shown in the following example:

```
vyatta@vrouter# set interfaces bonding dp0bond1 vrrp vrrp-group
1 notify ipsec
```

High Availability Firewalls with VRRP

When two devices are in an HA pair, care must be taken on your master device, so that you don't block access from the other device. Port 443 must be allowed between both devices for config-sync to work, and VRRP must be allowed to be sent and received, including the multicast range of 224.0.0.0/24.

Associated VLAN subnets with VRRP

For any vif configuration with VRRP you will need a virtual IP address (the first usable IP in the subnet, the gateway IP to which everything in that subnet will route) and also a real interface IP address for the vif on both devices. To conserve usable IPs, it is recommended that the real interface IPs use an out-of-band range, such as 192.168.0.0/30, where one device has the .1 and the other device has the .2 address. You only need one real address on each vif, and you can have multiple VRRP virtual IPs.

Connection Synchronization

When two VRA devices are in an HA pair, it can be useful to track stateful connections between the two devices, so that if a failover happens, the current state of all connections that are on the failed device are replicated on the backup device. This ensures that any current active sessions (like SSL connections) don't have to be rebuilt from scratch, which would result in a disrupted user experience. This is connection tracking synchronization.

To configure it, you need declare what the failover method is, what interface you will use to send the connection tracking information, and the IP of the remote peer:

```
set service connsync failover-mechanism vrrp sync-group SYNC1
set service connsync interface dp0bond0
set service connsync remote-peer 10.124.10.4
```

The other VRA will have the same configuration, but a different remote-peer.

Note that this option might not be for everyone. This configuration can saturate a link if there is a high number of connections coming in on other interfaces, and will compete with other traffic on the declared link.

“OS reloads” and configuration

The “OS reload” process, when initiated in the SoftLayer Customer Portal, will wipe away all configuration present on the device and restore it to the original version and configuration in which it was provisioned. Any changes made since the last load of the system will be wiped away, so if you have made changes, for example, to the VRRP group on another machine, the reloaded machine will likely have a conflict when it comes back up, because it will not have these changes present when it comes online after provisioning.

Configuration Synchronization

Two VRA in an HA pair need to have their configuration synchronized sufficiently so that both devices behave in a similar manner. This is done through "configuration sync-maps," allowing you to choose which portion of the configuration will be synchronized between both devices. When you make a change on one machine, it pushes the marked config over to the other device. **Note:** This process doesn't necessarily mean that the config on the other device is saved. Only running config changes are made.

Configuration that is unique to one system should not be synced to the other. Real IP addresses, MACs, should not be sync'd. The "system config-sync" configuration node itself and the "service https" node cannot be sync'd at all.

The following example config-sync, works like firewall rules:

```
set system config-sync sync-map TEST rule 2 action include
set system config-sync sync-map TEST rule 2 location security firewall
set system config-sync remote-router 192.168.1.22 username vyatta
set system config-sync remote-router 192.168.1.22 password xxxxxx
set system config-sync remote-router 192.168.1.22 sync-map TEST
```

The first two lines create the actual sync-map itself. Here, the configuration stanza for "security firewall" will be set in the sync-map. As a result, any changes made inside the config node will be pushed to the remote device. However, changes made to "security user" would NOT be sent,

because that doesn't match the rule. You can make the sync-map as specific or as general as you'd like.

The next three lines designate the remote router's config-sync user and password, IP, and what sync-map to push. Any changes that match the rules for TEST will go to the remote-router 192.168.1.22, using this login. Note that a REST call is made to perform this using the VRA API, so the HTTPS server needs to be running (and unblocked) on the remote router.

Config-sync happens when you commit a change, so watch out for any error messages that come from the remote device. If the configuration is out of sync, you'll have to fix it on the remote device to get it running again.

You can see any differences by running the command: `show config-sync difference`.

Configuration Backups

The configuration commands need to be backed up when there is a change to the system. This can be accomplished by running the operational mode command `show configuration commands` and then saving the output (for example by copying/pasting from the SSH session). This would be considered a minimum backup for the configuration.

A more complete backup involves generating a technical support archive for the system:

```
$ generate tech-support archive
Saving the archivals...
Saved tech-support archival at /opt/vyatta/etc/config/support/mpatr-vyatta-
one.tech-support-archive.2013-08-27-155554.tgz
```

The generated archive file can then be copied from the VRA device to the storage location of your choice. The archive contains backups of the configuration information, home directories and logging information. This archive provides a more complete backup of a system.

```
-rw-r--r--  1 michael  michael    7863 Aug 22 12:46 config.tgz
-rw-r--r--  1 michael  michael     112 Aug 22 12:46 core-dump.tgz
-rw-r--r--  1 michael  michael  716033 Aug 22 12:46 etc.tgz
-rw-r--r--  1 michael  michael    3698 Aug 22 12:46 home.tgz
-rw-r--r--  1 michael  michael    1092 Aug 22 12:46 root.tgz
-rw-r--r--  1 michael  michael    4204 Aug 22 12:46 tmp.tgz
-rw-r--r--  1 michael  michael   82976 Aug 22 12:46 var-log.tgz
```

Any notes that you create while configuring the device would also be good information to back up in a central location, accessible to all of your administration staff.

Solutions to Common Questions

Lost password

If there is access to the system, set a new password by running the following command:

```
set system login user [account] authentication plaintext-password [password]
```

If there is no access to the system, you can reboot the device and see a password recovery option on the GRUB menu for resetting the root user password.

Firewall lock out

The "**reboot at [time]**" construct can be useful when testing potentially dangerous firewall rules. If the rule works then use the command "`reboot cancel`" to cancel the reboot. If the rule locks out access, simply wait for the scheduled reboot to occur.

If there is no access to the system, then a reboot may be used to recover access. Upon rebooting, the system will read the configuration file, which would be unchanged by previous entries that were discarded by the reboot.

If there is access via IPMI, you can perform the following actions to recover access:

1. Disable the offending rule by
    a. `set security firewall name [firewall name] rule [rule number] disable`
    b. commit
2. Unhook the entire named rule set from the necessary interface. **Note:** Be VERY CAREFUL with this command, as incorrect use will wipe out your interface configuration.
    a. `delete interfaces dataplane [interface] firewall [type] [firewall name]`
    b. commit

Reference

A list of Brocade 5600 vRouter documents can be found here
http://www.brocade.com/content/dam/common/documents/content-types/technical-documentation-library/brocade-5600-vrouter-doc-library-index.pdf

Each PDF describes a specific function of the device in more detail than what is currently provided in this article. If there be a conflict between the information provided here, the Brocade documentation, unless otherwise specified, should be considered the authoritative source.