Best Practices Guide: **Vyatta Firewall**

SOFTWARE-BASED NETWORKING & SECURITY FROM VYATTA

February 2013

VYATTA®
A Brocade® Company

INTRODUCTION

Vyatta Network OS is a software-based networking and security solution that delivers advanced Layer 3 routing, VPN, Stateful firewall, NAT, QoS and more. Vyatta's software form-factor enables it to operate on any bare metal x86 server or as a virtual machine on any hypervisor.

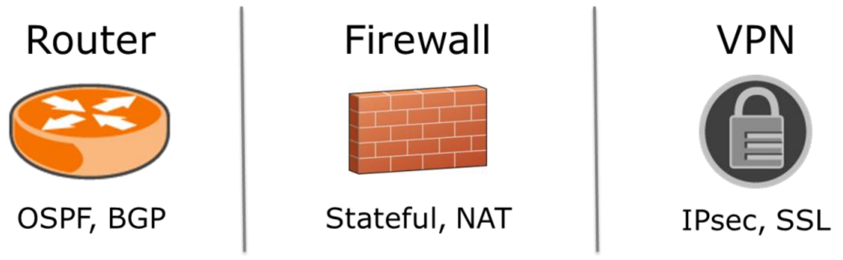

Figure 1: Vyatta delivers enterprise-class Layer 3 routing and security

Vyatta is deployed in a wide range of deployment scenarios for its comprehensive routing, rich security feature set, and its flexibility and ease of deployment. The level of flexibility that Vyatta offers its customers results in a wide range of tunable firewall, system security, and packet handling options. When configured for the level of security desired, the tunable options can be used to improve the working performance of the network.

Vyatta does not ship with a default security policy, so the firewall and security related options must be configured else all traffic will be forwarded unfiltered. The recommendations provided in this document are geared toward a high security and maximum performance deployment scenario. However, they may not be suitable for each and every deployment.

This document serves to establish a generic set of guidelines thereby enabling the user to choose the correct combination of options and values in order to maximize both security and connectivity in their Vyatta environment.

NOTE: The hardware, system resources, and unique device drivers play a large role in how the system performs under load. Configurations may need to be modified to account for the available resources.

Finally, Vyatta recommends that any changes to the configuration and settings be tested first in a non-production environment prior to applying to a production network.

For a more detailed description and configuration examples, refer to Vyatta Configuration and User Guide.

VYATTA STATELESS VERSUS STATEFUL FIREWALL

Vyatta supports both a stateless and stateful firewall configuration.

STATELESS FIREWALL

A stateless firewall considers every packet in isolation. Packets can be accepted or dropped according to only basic ACL criteria such as the source and destination fields in the IP or TCP/UDP headers. A stateless Vyatta router does not store connection information and has no requirement to look up every packet's relation to previous flows, both of which consume small amounts of memory and CPU time. Raw forwarding performance is therefore best on a stateless system. Vyatta recommends keeping the router stateless for best performance if you do not require the features specific to statefulness.

Use cases for a stateless Vyatta firewall include core routers or transit providers where fast delivery of packets is more important than stateful filtering or translation. Note that the following caveat holds true when using stateless firewall rules:

- When firewall rules are applied to multiple interfaces used in a flow, traffic must be explicitly accepted in both directions. There must be a rule on the interface or zone facing the initiator, and a rule which will allow traffic with the inverted source and destination on the interface or zone facing the responder.

STATEFUL FIREWALL

A stateful firewall keeps a table of previously seen flows, and packets can be accepted or dropped according to their relation with previous packets. As a general rule, stateful firewalls are generally preferred where application traffic is prevalent.

NOTE: The Vyatta router does not track the state of connections with default configuration.

Vyatta's firewall remains stateless until one of the following conditions have to been met:

- Firewall is configured and any rule has a "state" parameter set
- A "firewall state-policy" is configured
- NAT is configured
- The transparent web proxy service is enabled
- A load-balancing configuration is enabled

VYATTA FIREWALL RULES CONFIGURATION BEST PRACTICES

Firewall rules configured on Vyatta firewall rules are not stateful by default. The firewall has a default drop rule that is active when a default action is not specified by the user. To configure stateful firewall implies that all incoming and outgoing traffic for a specific session is allowed. The stateful firewall ruleset includes commands for established, new, and related as shown in the example below

```
firewall {
        name clienttoserver {
                default-action drop
                rule 5 {
                action accept
                destination {
                        port 80
                }
                protocol tcp
                state {
                        established enable
                        new enable
                        related enable
                }
                }
        }
        name servertoclient {
                default-action drop
                rule 5 {
                action accept
                state {
                        established enable
                        related enable
                }
        }
}
```

The general recommendations and best practices to follow when configuring Vyatta firewall are listed below:

- Prior to configuring any rules that include NAT and firewall, it is imperative that the user understands the interactions of firewall, NAT, and routing in Vyatta. The section titled 'Interaction Between Firewall, NAT, and Routing' under the firewall chapter in the Vyatta User Guide explains the order of packet processing in Vyatta. Understanding the order of operations is critical to achieving the desired behavior when multiple elements of the configuration are working together.
- Careful ordering of firewall and NAT rules ensures that the most frequently-matched rules are at the top of the firewall or NAT configuration. Rules are evaluated in sequence, and earlier matching means less evaluation.
- Rule numbering should be used to allow for easier addition and deletion of rules.
- Use logging only as needed for both firewall and NAT since logging affects performance and disk space.
- Combine multiple rules into a single rule if possible to simplify configuration and improve performance. For example, multiple ports 80, 443, and tcp can be combined into a single rule rather than multiple rules.
- Use the firewall groups for simplify configuration and ease of management.
- Use an appropriate name and description to describe the firewall or NAT rule which makes it easier for troubleshooting.
- Firewalls should be named according to the zones or interface contexts to which they will be applied. For example, a firewall named "INTERNET-IN" (interface-based) or "INTERNET-TO-DMZ" (zone-based) is better than a firewall named "WEB-SERVERS." Since only one firewall can be applied per interface or pair of zones, naming a firewall for the services it initially allows will become deceptive when rules for additional services are added.

FIREWALL GROUPS

The use of firewall groups can improve performance when common actions must be performed on a large number of elements. Firewall groups provide ease of configuration and manageability by allowing users to create a set of common elements. For example, it can be used to define a group of web server IP addresses that will perhaps have the same set of actions applied within a firewall rule. Adding or removing a web server from the network would result in a single line of configuration change to add or remove the web server's associated IP address from the group. A second advantage is that firewall groups can also be referenced in multiple locations, which allows the user to match on the same group within multiple firewall rule sets and within multiple rules within each rule set.

Firewall groups use hash tables (for address groups and network groups) and a bitmap index for port groups. This reduces memory consumption and improves processing efficiency when working with a large number of elements such as IP networks, addresses, or ports. It should be noted that the benefits of hash tables and in-memory bitmaps may not be realized with small number of elements or for dynamically changing elements within the group.

Because firewall logging and statistics gathering is performed on a per-rule basis, firewall rules that reference a group as a match condition will combine logging and statistical information for all members of the group under a single rule. If a more granular level of logging and statistical measurement is required, groups may not be an appropriate option.

Firewall groups should be used in cases where more than five common elements exist that will have the same set of actions applied within a firewall rule or set of firewall rules.

FIREWALL RULE SETS

A firewall rule set must be applied to an interface before it can match and filter packets. This behavior simplifies firewall configuration management by allowing the user to create, delete and modify firewall rule sets without affecting existing traffic.
On the Vyatta system, there are two methods of applying a firewall rule set to an interface so that it can begin matching and filtering packets:

a. Interface-based firewall

b. Zone-based firewall

In general, the choice of using one or the other depends on the number of interfaces and the number of rules. Interface-based firewall can be used if firewall rules are being applied to one or two Vyatta interfaces. Zone-based firewall is recommended for more than two Vyatta interfaces as it simplifies the configuration drastically.

INTERFACE-BASED FIREWALL

The interface method of applying a firewall rule set allows the user to apply a firewall directly to a system interface. Packets flow into, out of, and directly to the system via a system interface. Therefore**,** when applying a firewall to an interface, the direction of packet flow must also be specified. In the Vyatta configuration, these directions are defined as "in", "out" and "local".

On each system interface the "in" direction refers to traffic flowing into the system interface from outside of the system. The "out" direction refers to traffic flowing from the system interface to an external device. The "local" direction refers to traffic that is destined to the Vyatta system itself via the system interface.

A total of three firewall rule sets can be applied per system interface, one in the "in" direction, one in the "out" direction and one in the "local" direction. When the interface-based method is used for applying firewall rule sets, firewalls are most commonly applied in the "in" and "local" directions as this protects hosts behind the system from packets initiated from outside the system as well as the system itself. In more secure environments, the "out" direction or the zone based method of applying firewall rule sets may be used.

The interface-based method of applying firewalls to system interfaces is recommended for use in environments where connectivity takes precedence over security. For instance, the primary function of a BGP edge router may be to forward traffic as quickly as possible. In this type of environment, it's common for security mechanisms to be placed behind the edge routing devices. So, a simple interface-based firewall policy may be used to protect the system itself from outside access while forwarded traffic is allowed unfiltered. Applying firewalls to interfaces is also generally simpler to configure than a zone based firewall. In environments where security is not a major concern such as those where a firewall is only required on a single interface facing the Internet, an interface-based firewall may be the best option.

ZONE-BASED FIREWALL

When using the zone-based method of applying a firewall rule set, a zone policy must first be defined. The zone policy is used to group system interfaces into zones that share a common security policy. For instance, a corporate router may have an external, internal and DMZ zone while a service provider edge router may have a zone per customer VLAN interface.

A zone policy is created by defining a zone and placing interfaces into the zone. Once an interface is placed into a zone, a firewall rule set can no longer be placed directly on the interface. When a zone policy is in use, filtering takes place between zones rather than per interface direction.

By default, no traffic is allowed into or out of a zone. A firewall rule set must be applied to allow traffic to and from each and every zone.

The "local" zone is a special zone reserved for all traffic destined for the system itself. When this zone is defined, no traffic will be allowed to the local zone from any other zone unless the zone policy is configured to allow it. This is especially useful in the service provider edge environment where customer facing interfaces should not be allowed access to the system itself.

NOTE: With the interface-based model, a firewall would need to be applied in the local direction on every customer interface to prevent access.

While a zone policy appears to be more complex to configure than the interface-based method of applying firewall rule sets, it can be easier to manage going forward. This is especially true if interfaces are constantly being added to a zone. Only small changes are needed to provision a new customer interface into the existing zone policy framework once it has been properly designed and configured.

Some best practices to keep in mind when using zone policies are:

- Take precaution when defining a local zone to allow access from the interface that is in use for configuration and management purposes. It is possible to lock oneself out of the router by incorrect zone configuration.
- Any interface that is not included in a zone is an unfiltered/unsecured interface. All interfaces, even those that are not currently in use, should be placed into a zone. This will ensure that if these interfaces are used in the future, they are not inadvertently excluded from the zone policy.
- Virtual interfaces must also be included in a zone policy if they must pass traffic to an existing zone. Wild card values such as "pptp+" and “l2tp+” can be used for dynamic interfaces such as those that are generated by remote access VPN connections. Firewalls can only be applied to these interfaces within a zone policy as they do not exist in the configuration.

- No filtering can be performed within a zone. This means that if for instance, traffic must be filtered between interface eth0 and interface eth1, they must be in different zones. Only group interfaces within in the same zone that are allowed unfiltered access to all interfaces within that zone.
- Using a naming convention (described below), for zones and their associated firewall rule sets, minimizes confusion and organizes the firewall and zone-policy configuration. This helps with the process of building and configuring the zone-policy and firewall, as well as with troubleshooting potential zone or firewall issues. When naming zones and their associated firewall rule sets, the following should be kept in mind:
    - Zone names should be used to describe the zone they are associated with such as "external," "internal", "dmz" etc. This will help system users easily understand the purpose of each zone with just a quick glance at the zone policy configuration.
    - In addition to descriptively naming each zone, the firewall rule sets used to filter traffic between zones should be named in such a way that clearly describes their purpose. For example, the firewall rule set filtering traffic from the "external" zone to the "internal" zone should be named "external-internal." This naming convention also makes it clear which firewall rule set to apply when configuring the zone policy.

VYATTA FIREWALL GLOBAL SYSTEM OPTIONS

The options listed in this section are global system firewall options that affect system behavior toward certain types of packets. These options are designed to protect against common attacks that could potentially reduce the ability or prevent the system from forwarding packets. Although some of the options listed in this section have an effect on how the system behaves toward forwarded traffic, in general, these options are exposed with the intention of protecting the system itself from malicious traffic.

all-ping

This option is enabled by default such that the system will craft and send an IPv4 echo reply message in response to an IPv4 ICMP echo request message received on any interface that are destined for one of its own IPv4 addresses. Disabling this option causes the system to ignore all IPv4 ICMP echo request (ICMP type 8) messages that are destined for one of its own IPv4 addresses.

This option is useful in environments where the security policy demands that the Vyatta system not respond to pings from any interface. Performance and configuration management wise, it is simpler and more efficient to set 'all-ping' to disable than to block IPv4 ICMP echo request messages on all system interfaces.

If even a single system interface must respond to IPv4 ICMP echo request messages, then this option should be left at its default value, enabled. ICMP can be selectively blocked per system interface using the Vyatta firewall.

Enabling or disabling this option has no effect on ICMP messages that are being forwarded by the Vyatta system to remote networks or connected hosts.

broadcast-ping

This option is disabled by default such that the system will ignore IPv4 ICMP echo request and IPv4 ICMP timestamp (ICMP type 13) messages that are destined for a broadcast or multicast address.

Most recent host and network operating systems ignore broadcast pings by default. Broadcast pings can be used to generate "smurf attacks." Smurf attacks are a type of denial of service attack that can be triggered by ICMP echo reply packets flooding a network in response to a broadcast ICMP echo request. It is recommended to leave the 'broadcast-ping' option disabled unless there is a specific need for enabling it.

Enabling or disabling this option has no effect on ICMP messages that are being forwarded by Vyatta to remote networks or connected hosts.

ip-src-route and ipv6-src-route

This option is disabled by default such that the system will drop IPv4 and IPv6 packets that contain source route information in the IP header. Allowing the system to process IP header source route information can override standard destination-based route processing making the system vulnerable to IP spoofing attacks. The attacker can manipulate the source route information to redirect responses to spoofed destination addresses back to itself. It is recommended to leave the “ip-src-route” and “ipv6-src-route” options disabled unless there is a specific need for enabling them.

When this option is disabled, all packets received by the system that are populated with source routing information whether locally destined or destined for a remote network or connected host, will be dropped.

Note: The source routing referred to and affected by this option should not be confused with and is NOT in any way related to policy-based routing or the default system routing table.

log-martians

This option is enabled by default such that the system will generate a log message when it receives a packet that is considered to contain an invalid source or destination address. Such packets can arise from IP address spoofing or other denial of service attacks, but are also commonly seen with Layer 2 misconfiguration issues. Specifically, martian packets are commonly seen and logged when two or more system interfaces are connected to the same VLAN. In this case, packets seen on interface X that are destined for interface Y will be logged as martian packets. These log messages will be seen when viewing the default system logs.

This option has minimal impact on performance and in most cases should be left as enabled so that potential attacks or network configuration issues can be logged and identified if and when they arise.

When this option is enabled, the system will log any packet it receives that it considers a martian packet regardless of whether the packet is locally destined or is to be forwarded.

Receive-redirects and ipv6-receive-redirects

This option is disabled by default such that the system will ignore ICMP redirect (ICMP type 5) messages. The purpose of an ICMP redirect message is for network gateways or routers to notify hosts of a shorter path to a destination. When a host receives an ICMP redirect message, it should then update its own routing table based on the information received in the ICMP redirect message. As ICMP redirect messages are used to update routing information, they can also easily be used in denial of service attacks. It is recommended to leave the “receive-redirects” and “ipv6-receive-redirects” options disabled unless there is a specific need for enabling them.

Enabling or disabling this option has no effect on ICMP messages that are being forwarded by Vyatta to remote networks or connected hosts.

send-redirects

This option is enabled by default such that the system will send ICMP redirect (ICMP type 5) messages to directly connected hosts under the following conditions:

- A packet is received on and subsequently routed out the same interface.
- The source address of a packet is in the same subnet as the next hop for the packet’s destination.

  For example, host A sends a packet to its default gateway (the Vyatta router on eth0) and the next hop for the packet's destination turns out to be reachable via the same subnet/Ethernet segment. This could occur in cases where more than one router exists on the same segment and is often seen when topology issues or Layer 2 configuration issues exist.

This option has minimal impact on performance and in most cases should be left as enabled so that potential attacks or network configuration issues can be identified if or when they arise. Vyatta will only send ICMP redirect messages to directly connected hosts on the same Layer 2 segment.

source-validation

By default, this option is disabled. To enable source-validation, an option of either “loose” or “strict” must be selected.

The 'source-validation' option controls what is commonly referred to as reverse path filtering. Reverse path filtering is the process of validating whether or not the source address of a received packet is expected on the interface it came in on. Most ISPs perform strict reverse path filtering to ensure that only the IP addresses they've allocated to a particular link are able to send traffic through that link.

The “strict” option will cause the system to drop any packet it receives on an interface if the return path is not via that same interface. The strict option will not work in topologies where asymmetric routing conditions exist. That said, setting source-validation to strict is not recommended when dynamic routing is in use.

The “loose” option will cause the system to drop any packet it receives on an interface if the source address of the packet is not reachable via any other system interface. There are cases where even enabling source-validation in loose mode can cause connectivity issues in certain topologies.

Source-validation can be used to protect against IP source address spoofing attacks however, if enabled in dynamic routing environments, it can cause the system to drop valid traffic. Because of this, this option is not recommended for use in complex routing environments. Whether enabled in 'loose' or 'strict' mode, once enabled, all interfaces on the system will be affected by the change.

syn-cookies

This option is enabled by default such that the system will use TCP SYN cookie functionality to mitigate SYN flood attacks by decreasing the likelihood of overloading the local TCP connection queue.

It is important to be aware that enabling SYN cookies on Vyatta only enables this feature for TCP packets whose destination address is an address on the Vyatta system itself. Vyatta does not act as a SYN proxy for forwarded traffic or for packets whose destination address is an address behind the Vyatta system. Additional information regarding how SYN cookies work can be found in the Vyatta Firewall documentation.

This option has little to no impact on performance and should be left as enabled to help mitigate potential TCP SYN flood attacks directed at Vyatta itself. Any denial of service attack directed at Vyatta has the potential to impede or reduce forwarding performance while the attack is occurring.

CONNECTION TRACKING

When connection tracking is enabled, all packets that the system forwards, sends and receives will have an associated conntrack table entry. The system creates an entry in the conntrack table when it sees a connection for the first time and considers the connection to be valid. An example of a valid new connection could be a TCP SYN packet. Whereas a TCP FIN packet not belonging to an existing connection would not be considered valid and would not generate an entry.

Most conntrack table entries are stored in the "main" table. This table contains entries for all new and existing valid connections. Connections that are considered to be "related" to a connection in the main conntrack table, such as an FTP data stream associated with a passive FTP connection, are created and stored in the "expect" table. IPv4 and IPv6 connection entries are stored in separate tables.

The system's current conntrack table entries can be viewed by running the 'show conntrack table [ipv4|ipv6]' command in operational mode. On a live system, this command will have several pages of output. Each line of the output contains a conntrack table entry that will look similar to the following:

```
3720128944 192.168.50.13:61140 74.125.224.67:80 tcp [6] TW 8
```

The above is a conntrack table entry that is associated with a TCP connection. The first value in the entry is the "connection ID". This value is used by Vyatta to identify and manipulate conntrack entries using operational mode commands. For instance, conntrack table entries can be identified and deleted based on the connection ID that the user specifies in the 'delete conntrack' command. The next value consists of the source IP address and source port combination. This information is used by the system to identify return traffic and perform the appropriate NAT translations if required. The source portion of the entry is followed by the destination information. The destination information is also listed in the form of IP address:port. The next value is the protocol in use by the connection in alpha and numeric form. The connection above is a TCP connection which is protocol number 6. For TCP connections, information on the TCP state is displayed next. This connection is in a state of time wait or "TW" for short. The final value displayed for the entry above is the current timeout interval in seconds. At the time the entry above was viewed, the timeout for the entry was 8 seconds. This value is subject to change if the connection sees more packets within 8 seconds at which point the timeout interval will be reset and begin to decrement again until more packets are seen or the connection times out.

SYSTEM PERFORMANCE CONSIDERATIONS AND TUNING

TUNING SYSTEM WIDE CONNTRACK RELATED PARAMETERS

Several connection tracking related values are configurable and may require adjustment in certain topologies and on systems that will be handling heavy loads. Each of the values below is set with a default value that may or may not be adequate for a particular environment. It is important to understand what behavior each value affects and how adjusting each value will affect the underlying system.

The recommendations provided below are meant to be used as a guideline for selecting the appropriate values. This depends on the topology of the network, the prevailing traffic patterns, and the desired average packets per second throughput.

1. **Connection Tracking Table Sizes**

    - *system conntrack expect-table-size*

    The default value of the system ‘conntrack’ expect-table-size is 2048. This value indicates the quantity of expect table entries the system will store.

    As described in the Connection Tracking Overview, the expect table is used to store expected or related connections. Once a related connection becomes established, it graduates into the main table. Under normal operating conditions, entries remain in the expect table for a very short time. Because of this, it is unlikely to exceed the default expect table size unless the system is handling large amounts of traffic that will generate related connections such as a VoIP provider edge device.

Vyatta recommends only increasing the expect table size when the majority of traffic will consist of "expected" connections such as RTP voice streams associated with SIP sessions. If it does become necessary to increase the expect table size, it should be to a value that is a power of 2. For example, the default value could be doubled to 4096 or quadrupled to 8192

When considering conntrack table size modifications, it's best to be conservative as allocating a value that is too large could cause the system to exhaust its available memory.

- *system conntrack table-size*

The default value of the system conntrack table-size is 262144 . This value indicates the number of conntrack table entries the system will store.

The main conntrack table tracks every single connection the system receives. Unlike the expect table, Vyatta recommends increasing the conntrack table size on most systems to a value of 2 to the 20th (1048567) unless the system is being run in a memory constrained environment (e.g. a VM with only 1GB of RAM or less). If the system has the minimum or below the minimum recommended memory, the conntrack table-size should be left as the default value to prevent potential memory exhaustion.

As with the expect table adjustment recommendations, the amount of traffic the system will be handling and the available memory must be known before an informed decision can be made on what value the conntrack table size should be set to. When increasing the conntrack table size, ensure that the selected value is a power of 2.

- *system conntrack hash-size*

The default value of the system conntrack hash-size is 4096.

The system conntrack hash-size defines the size of the hash table used to manage actions such as lookups that are performed on the conntrack table. For efficiency, the conntrack hash-size should be set to a value that is 1/8th the size of the conntrack table-size. For the recommended conntrack table-size value mentioned above (2 to the 20th), the associated hash-size value should be 131070.

When the number of standard or related connections exceeds either the main conntrack or conntrack expect table, the system will begin dropping new connections. As mentioned earlier, if the conntrack table size is set too high, it can consume all available memory resulting in the termination of vital system processes. This can occur when the system itself or a device it is forwarding traffic to is the subject of a DDoS attack. When selecting conntrack table size values, it is very important to select a reasonable value that corresponds to the amount of traffic the system will be handling. Not meeting or exceeding the correct table size values for a specific environment can result in major to complete loss of functionality.

2. Connection Tracking TCP Specific Adjustments

- *system conntrack tcp half-open-connections*

The default value for the system conntrack tcp half-open-connections is 512. This value defines the queue size for IPv4 TCP connections that are in a "half open" state.

The system conntrack tcp half-open-connections option works in conjunction with the firewall syn-cookies feature described in an earlier section of this document. This option, as well as the syn-cookies option only applies to IPv4 TCP packets. If the firewall syn-cookies option is disabled, this option becomes extraneous.

TCP half open connections are connections that have not yet received the third packet in the TCP 3-way handshake. Connections are in a SYN-RCVD state during this phase of the connection establishment. A TCP SYN packet has been received and the system has replied with a corresponding SYN-ACK, but has not yet received a reply to the SYN-ACK. This state will occur with every new TCP connection and is expected under normal operating conditions. However, TCP SYN flood attacks may create an excessive number of invalid half open connections and quickly fill up the receiving system's connection queue leaving it unable to accept new (valid or invalid) connections for some period of time.

As with the firewall syn-cookies feature, the system conntrack tcp half-open-connections only affects TCP connections that are terminating on the Vyatta system. This option has no effect on forwarded TCP packets. In nearly all cases, Vyatta will be primarily forwarding TCP packets and will only be terminating TCP connections for a small amount of services that are unlikely to stay in a half-open state for a long period of time. For this reason, it is recommended to leave the system conntrack tcp half-open-connections at the default value of 512 unless for some reason, the system will be terminating large amounts of TCP connections.

- *system conntrack tcp loose*

This option is a boolean value and is enabled by default. That is, by default, the system will assume that an IPv4 TCP packet that appears to be part of an existing connection but that does not already exist in the conntrack table is a valid packet that is part of an established connection. Modifying this option affects the behavior of the global and per rule "state established" and/or "state invalid" match conditions.

When the system conntrack tcp loose option is enabled, the system will consider a TCP ACK packet that is not associated with an existing conntrack table entry to be part of an established connection. It will then add the connection to the conntrack table and any additional packets associated with that same connection will also be considered as established.

When the system conntrack tcp loose option is disabled, the system will consider a TCP ACK packet that is not associated with an existing conntrack table entry to be an invalid connection. If a global state policy or per rule "state invalid" match condition is defined, these packets will match the state policy or the per rule match condition.

In environments where connectivity takes priority over security, it is recommended to leave the system conntrack tcp loose option as enabled. This allows the system to accept connections that still exist after recovering from a failure such as a system crash or unexpected power cycle. It is also a requirement for asymmetric routing environments where the return path for a connection is different than the initiating path.

In secure environments where asymmetric routing does not exist, the system conntrack tcp loose option may need to be disabled. Disabling this option will reduce the system's vulnerability to certain DDoS attacks and malicious scans such as TCP ACK scans that could otherwise bypass the firewall by posing as valid connections. It is important to remember that disabling this option may cause the system to drop valid connections under certain conditions.

- *system conntrack tcp max-retrans*

The default value for the system conntrack tcp max-retrans is 3. This value is used to set a timeout for stale IPv4 TCP connections. Once the system sees ‘n’ (value of system conntrack tcp max-retrans) TCP retransmissions for a tracked connection without seeing a corresponding TCP ACK packet from the remote end of the connection a timeout of 5 minutes will be set for that connection. Once the timeout is reached, the connection will be removed from the conntrack table.

This value allows the system to more efficiently and quickly remove stale TCP connections from the conntrack table to make room for new, valid connections. If within the 5 minute tcp max-retrans timeout an ACK packet is received from the remote end of the connection, the default timeout for the TCP connection (5 days) will be reset.

It is recommended to leave this setting at the default value of 3 unless there is reason to expect significant delays with valid TCP sessions.

3. Logging Connection Tracking Events

By default, connection tracking events such as the creation, modification and deletion of an entry from the conntrack tables are not logged. For debugging and troubleshooting purposes, the system can be configured to send conntrack related log messages to the system log file. The conntrack related logging options exist under the 'system conntrack log' sub-tree in the Vyatta CLI and GUI. For example, to enable logging for the creation of new TCP connections in the conntrack table, configure the following:

- *set system conntrack log tcp new*

After committing the above, messages such as the following will be begin to display in the system log file:

```
log-conntrack:     [NEW] tcp      6 120 SYN_SENT src=192.168.50.10
dst=74.125.224.97 sport=60587 dport=80 [UNREPLIED] src=74.125.224.97
dst=76.74.103.41 sport=80 dport=60587 id=3718230264
```

Logging can be enabled for the creation (new), modification (update), and deletion (destroy) for TCP, UDP, ICMP and "other" (all other protocols).

Only enable conntrack event logging if there is a strong business requirement for accounting, or forensics or for temporary troubleshooting purposes. This is not recommended under normal operating conditions due to its impact on performance and disk space consumption. Most production systems will see hundreds or even thousands of conntrack related log messages per second if conntrack logging is enabled. Logging conntrack related events adds additional processing overhead to each connection being logged creating a potentially severe impact on forwarding performance. Use this capability with caution when operating in a production environment.

4. Disabling System ALGs (Conntrack Helper Modules):

Vyatta recommends disabling all the conntrack helper modules for system ALGs unless absolutely required. This improves performance by reducing the code path necessary to traverse the helper modules.

system conntrack modules ftp disable

system conntrack modules gre disable

system conntrack modules h323 disable

system conntrack modules nfs disable

system conntrack modules pptp disable

system conntrack modules sip disable

system conntrack modules sqlnet disable

system conntrack modules tftp disable

SIP ALG Tuning Options:

system conntrack modules sip enable-indirect-media

system conntrack modules sip enable-indirect-signalling

system conntrack modules sip port

SUMMARY

Vyatta Network OS provides stateful firewall, NAT, VPN and advanced routing capabilities within one software image. The same software can be deployed on any bare metal x86 server or as a virtual machine where it acts as a guest operating system on top of any hypervisor. In addition to the Vyatta specific firewall configuration best practices documented here, it is necessary to adhere to certain general principles while configuring firewalls:

Keep the firewall configuration as simple as possible to ensure that the firewall is configured to meet the security policies of the organization. This makes it easy to monitor and troubleshoot in the event of a security breach.

Ensure that the firewall functionality is running on a hardened and routinely patched operating system. Deploying Vyatta's router with appropriate firewall rules can help prevent security attacks.

Any and all changes to the firewall configuration should be pre-tested in a non-production environment to achieve the desired security policy. Change-management practices for the firewall should be put in place to ensure that all changes are logged for future reference and record keeping. Regular backups of the configuration will prevent any loss of configuration due to router malfunctions.

Firewall logging and alerting should be enabled for monitoring and tracking purposes but this should be done in a manner so as to not degrade performance of the Vyatta system. Monitor the logs regularly and investigate any log entries that may be suspicious.

Vyatta offers a variety of configuration options to fine tune the firewall operation and to provide maximum security. Vyatta's firewall not only protects physical servers and hosts but can also provide the same security in a virtual environment. Vyatta software can be installed on inexpensive hardware to provide a proof-of-concept platform for testing prior to deploying in a production network.

To learn more about Vyatta's firewall or Vyatta software in general, please visit Vyatta's website or contact the Vyatta sales team at sales@vyatta.com. Additional references:

http://www.vyatta.com/learn/resourcelibrary

http://www.sans.org/reading_room/whitepapers/firewalls/deploying-vyatta-core-firewall_33493