Stateful Inspection Firewall

To secure all network connections, Juniper Networks devices use a dynamic packet filtering method known as Stateful inspection. Using this method, the firewalls collect information on various components in a packet header— source and destination IP addresses, source and destination port numbers, and packet sequence numbers. The device then maintains the state of each TCP session or UDP pseudo-session traversing the firewall, performing TCP reassembly when necessary to ensure proper interpretation of the communication session. When a responding packet arrives, the firewall will compare the information reported in its header with the state of its associated session stored in the inspection table. If they match, the responding packet is allowed to pass the firewall. If the two do not match, the packet is dropped. Juniper Networks firewall can secure a network by inspecting, and then allowing or denying, all connection attempts that require crossing an interface from and to that network.

By default, the Juniper Networks firewall denies all traffic in all directions. Using centralized, policy-based management, enterprises can create a series of security policies that will control the traffic flow from network to network by defining the kinds of traffic permitted to pass from specified sources to specified destinations at scheduled times. At the broadest level, all types of traffic can be allowed from any source in security zones to any destination in all other zones without any scheduling restrictions. At the narrowest level, policies can be created that allows only one kind of traffic between a specified host in one zone and another specified host in another zone during a scheduled period of time.

Stateful inspection is more secure than other firewall technology such as packet filtering because it opens smaller "holes" through which traffic can pass. For example, instead of permitting any host or program to send any kind of TCP traffic on port 80, a Stateful inspection firewall ensures that packets belong to an existing session. Furthermore, it can authenticate the user when the session is established, determine whether the packets really carry HTTP, and enforce granular constraints at the application layer (e.g., filtering URLs to deny access to black-listed sites).