

Rule Organization For Stateful Inspection Firewalls

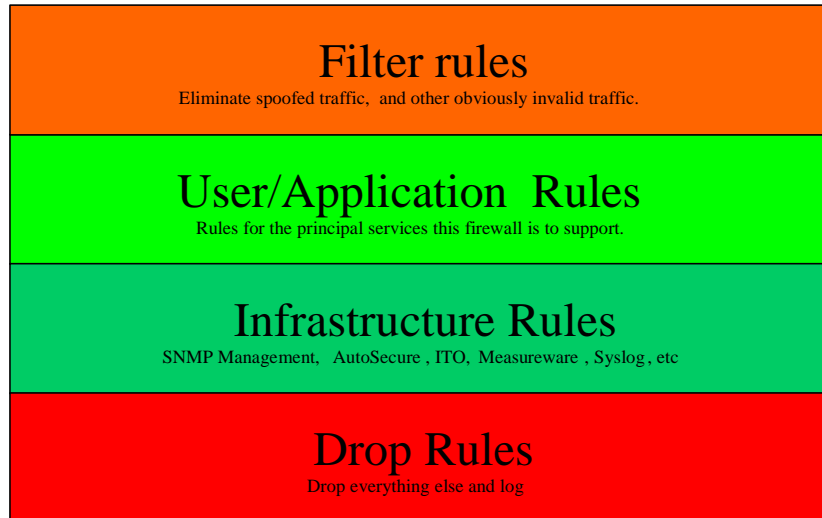
A fair number of popular firewalls in use today are stateful inspection firewalls. The most popular example is Firewall-1, but IPTables, Contivity Stateful Firewall, and many others also use stateful inspection.. With Stateful Inspection, packets are intercepted at the network layer for best performance (the same as packet filters such as routers) and filtering decisions are based on context information determined from the state of the connection.

Stateful Inspection Firewalls use a first match methodology. This means that they compare packets to the ruleset starting from the first rule in the ruleset and compare the packet to each subsequent rule in the ruleset until there is a match, or the end of the ruleset is reached. Although each comparison does not add significant amount of latency to the transaction, it is still true that the more comparisons performed the more latency is added to a packet. Because of this it is important to pay attention to a couple of things:

- Order of rules is extremely important. Because stateful inspection uses a first match algorithm it is important for efficiency and latency reasons that the rules that will match most frequently be placed near the beginning of the rule set when possible.
- Because the firewall is first match, in general rules should be written with the most specific near the beginning of the rule set and the least specific near the end. Improper ordering of rules can cause serious security problems. Placing a less specific rule too high in the rule set may result in a security hole caused by the less permissive rule being matched before the more specific rule.

There are two general philosophies which are commonly used for stateful inspection firewalls. The first philosophy is to optimize the ruleset so that the rules which are matched the most are closest to the top. This philosophy optimizes the absolute performance of the firewall, but possibly at the expense of important applications. An alternate philosophy is to optimize the rules so the applications you want to have the least latency are closest to the top. This philosophy optimizes application latency, but at the expense of overall firewall efficiency.

My general approach is to use a hybrid of the two philosophies. It divides applications into classes. Precedence is given to the important applications, while at the same time trying to optimize the firewalls efficiency as much as possible.



The diagram above illustrates this approach. Filter rules are placed at the top to eliminate traffic that is never valid. This includes traffic like spoofed traffic, and traffic that should never pass your perimeter (like Netbios). This traffic is either as a result of nefarious purposes of the originator, or more often than not is just the usual invalid noise that always seems to happen on any network. This traffic is discarded up front thus consuming the minimum amount of overhead on the firewall. The ingress portion of this traffic should be dealt with on the screening border, but you will probably still need to eliminate some egress traffic here.

Following the filter rules are the user/application rules. These are close to the top because they are the traffic that is most important to the purpose of your firewall. I usually like to optimize the user/application rules so that the traffic which needs to execute the fastest gets precedence within the rules block. This is a little harder on firewall performance, but ensures your latency sensitive applications are least impacted.

The next block is the infrastructure rules. These are traffic required for management or administrative purposes. and the traffic which is required to ensure your network is healthy and happy. This would include connections for device monitoring, and NTP traffic to the Internet to ensure the clocks are all synchronized. Even though these may be the largest proportion of traffic on your firewall, these rules are not latency sensitive, so can afford to wait a little longer. It is probably a good idea to optimize these rules so that the highest frequency traffic gets precedence within the block, thus reducing the overall impact on the firewall.

The remaining block contains the drop rules. If network traffic makes it this far in the rule set it will be denied. The traffic that remains generally falls into one of three categories:

- On every network there is a certain amount of noise, such as router broadcasts, which although it is annoying, is not a sign of suspicious activity.

Depending on your security policy, this traffic can usually just be dropped without logging.

- The second type of traffic is traffic that the security analysts consider interesting. This is traffic that shows the signature of the latest exploit, or is interesting for other reasons. This traffic is discarded and an alert generated.
- The remaining traffic is everything left. We drop this and log it. Although this traffic is usually harmless, it is useful to analyze because it usually points out misconfigured or improperly hardened systems that are generating noise. Once in a while this traffic may show an attack against your network.

Remember, the order in which the firewall rules appear is important, for efficiency reasons, security reasons, and to preserve the administrator's intent. Pay careful attention to specificity of rules within the blocks. Placing a less specific rule too high in the block may result in a security hole caused by a less permissive rule being matched before the more specific rule.

References

1. Thompson, Rick, "GIAC Practical: Implementation of Firewall Filters", August 14, 2000. http://www.sans.org/infosecFAQ/firewall/fw_filters.htm
2. Wanner, Rick, "Practical Project for GCFW", May 2001 http://www.giac.org/practical/Rick_Wanner_GCFW.doc,