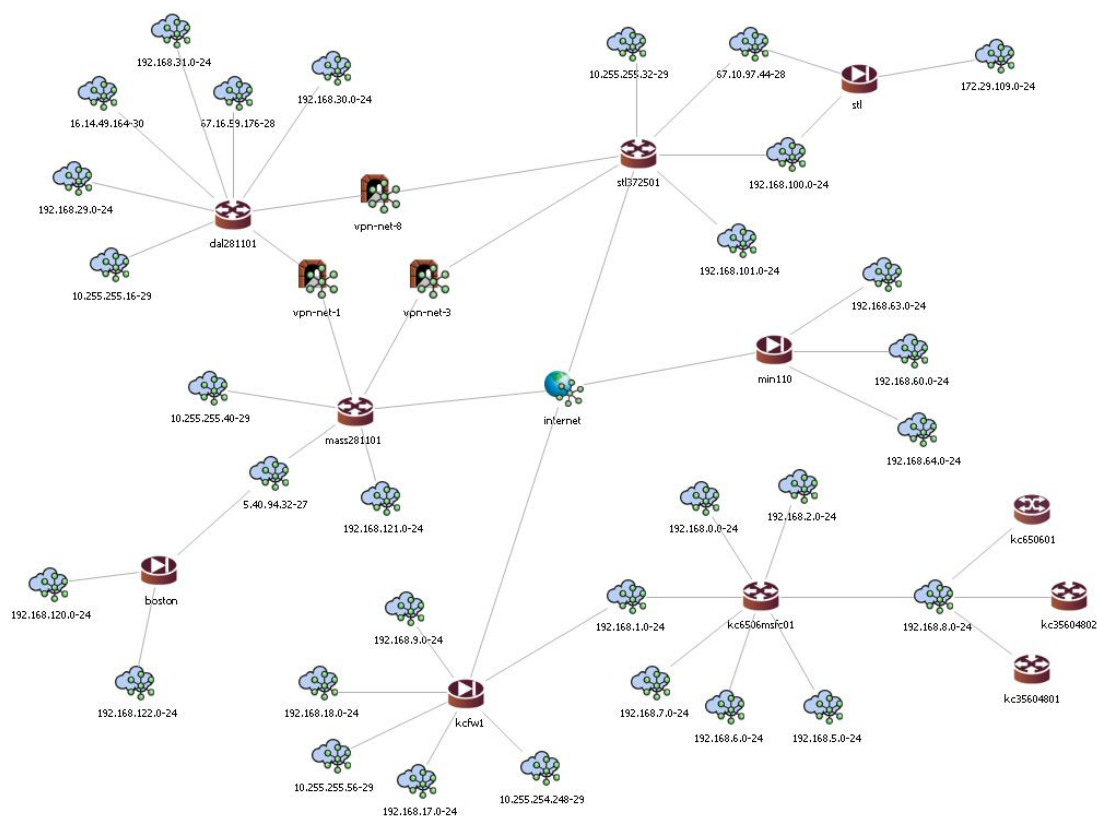


Effective Solutions for Check Point™ Rule Cleanup

White Paper



Effective Solutions for Check Point™ Firewall Rule Cleanup *Using Athena FirePAC*

Abstract

Check Point™ firewall rules that provide access to a wide array of services in a large network, while at the same time securing the critical assets from attacks, tend to become very large in size and redundant in functionality. As the rules are added in a single flat rule base, the rule base quickly becomes large and administrators become hesitant to modify existing rules. Instead they often add new rules for fear of causing an adverse impact on existing service availability. Over time, the rule base becomes bloated, requiring not only more effort in making changes but also having an adverse impact on firewall performance. It is therefore essential to clean up the rule base and reduce its size. This paper presents some techniques to cleaning up the rule base *along with an effective solution that addresses these automatically for you using Athena FirePAC for Firewall Rule Cleanup.*

Introduction

All firewalls protecting enterprise networks with an already complex web of inter-connections will inevitably grow more complex with time. Rules must be added in order to provide network access and protect against attacks. Ideally, these rules would be added to the firewall in an organized manner. Furthermore, rules would be organized and enhanced to suit specific business purposes. Unfortunately, that is not reality. Firewall administrators change, and as new people transition into the role, rules are added in an ad hoc manner without realizing that the new rules are redundant and not needed in the first place. Moreover, as the rule bases become large, firewall administrators become hesitant to modify existing rules and instead add new rules for fear of causing an adverse impact on existing service availability. This makes the problem even worse and the job of administrators very difficult if they have to respond to issues raised during firewall or PCI audits. Here are some items that administrators should address to prevent the rule bases from becoming unmanageable and redundant:

- New generalized rules are added that replace a number of more specific rules that already exist in the firewall. This typically happens when specific rules are added initially to the firewall to allow services to specific hosts or subnets and then more general rules are added when the business scope expands to other networks or services or much larger subnet or services (sometimes “any” network or service). When this happens, the previous specific rules become redundant and need to be cleaned up.
- New rules are added without realizing that one or more rules preceding or succeeding the new rule already handle the functionality being addressed by the new rule. Depending on where the rule is added, the new rule might never be triggered. This happens when there are multiple rules in the rule base that each cover portions of the new rule and together completely cover the new rule. As a general practice, before adding new rules, existing rules should be queried to see if they can be modified to satisfy the change request. Change requests do not happen in a vacuum, they are made to serve a business purpose that probably already exists and is being enhanced.
- New rules are added as a special case of one or more subsequent rules to exhibit special behavior (often temporarily). These special cases include enabling or disabling logging only for specific hosts or services instead of the much larger networks or services being handled by the subsequent rules; i.e., performing application inspection for specific services involving specific assets, tracking quality of service attributes, and requiring user authentication for specific services or assets. Sometimes special cases are created at the beginning of the ruleset for the most used traffic to increase firewall performance. Some of these rules are temporary in nature, sometimes added to track usage or do some testing; however these are not cleaned up even when the reason for adding these in the first place is no longer relevant.
- Rules become stale when the business reason for adding the rules goes away. These rules are not used anymore but remain in the firewall impacting maintenance and firewall performance. These rules can be identified and cleaned up by tracking their usage in the firewall either by logging these rules or by looking at rule usage hit counts. Attention must be paid to those rules that might be used only during a specific time of the year and will not be in the firewall logs unless logs are captured and analyzed for a sufficient time period. Either way, rules should be removed after confirming with the business owner of the rules.

Identifying redundant rules that are never triggered

The redundant rules that can be safely removed without affecting the firewall behavior are rules that are never triggered because preceding rules cover them and match first. Identifying redundant rules that are completely covered by one rule is a little easier than finding rules that are covered by more than one preceding rule(s). In the first case, you can compare the two overlapping rules that are in question side by side and verify if the first rule covers the second rule completely. In the second case, each of the preceding rules will each cover only portion of the redundant rule but together they cover the redundant rule completely. This requires a more detailed analysis of all the overlaps that exist between the rules. Check Point™ policy verification catches only some of the simple overlaps and a lot of the redundancies are still left in place.

Here is the process for identifying these redundant rules that are not caught by Policy verification:

1. Understand the Source, Destination and Service elements used in each of the rule(s) in the rule-base. If object groups are used or if multiple objects are used, you need to understand the complete expanded combinations in the rule.
2. Find out rule by rule in the rule base, the rules that overlap with each other on all the Source, Destination and Service elements of the rule. For each overlap found for a rule,
 - a. If the overlap with another rule higher up in the rule sequence of the access list is such that it completely covers this rule, then this rule is never triggered and can be removed.
 - b. If there is no single rule that completely covers this rule, but there are multiple rules higher up in the rule sequence that together cover this rule, then this rule can be removed.

Examples: In the following rule base: rule 4 is covered by rule 3, rule 5 is covered by rules 1 and 2 and rules 6 and 7 cover rule 8. Rules 3, 5 and 8 are redundant and can be safely removed without affecting firewall behavior.

The address object group DMZ_svrs contains only hosts from the object groups DMZ_app and DMZ-web. Similarly object group inside_servers contains some of the hosts from the internal_nets object group.

Please note that Check Point™ Policy verification does not complain about these redundancies during Policy installation

NO.	SOURCE	DESTINATION	VPN	SERVICE	ACTION	TRACK	INSTALL ON	TIME
1	Net_10.10.1.0	* Any	* Any Traffic	TCP telnet	accept	- None	* Policy Targets	* Any
2	Net_10.10.1.0	* Any	* Any Traffic	TCP ssh	accept	- None	* Policy Targets	* Any
3	internal_nets	* Any	* Any Traffic	UDP udp-high-ports	accept	- None	* Policy Targets	* Any
4	internal_nets	* Any	* Any Traffic	UDP udp-5901	accept	- None	* Policy Targets	* Any
5	Net_10.10.1.0	* Any	* Any Traffic	TCP telnet-ssh	drop	- None	* Policy Targets	* Any
6	internal_nets	DMZ_app	* Any Traffic	TCP ftp TCP ssh TCP http	accept	- None	* Policy Targets	* Any
7	internal_nets	DMZ_web	* Any Traffic	Web	accept	- None	* Policy Targets	* Any
8	inside_servers	DMZ_servers	* Any Traffic	TCP http	accept	- None	* Policy Targets	* Any
9	* Any	* Any	* Any Traffic	* Any	drop	- None	* Policy Targets	* Any

Identifying redundant rules that are covered by succeeding rules

Identifying redundant rules that are covered by one or more succeeding rules is a little more difficult and attention needs to be paid to the rule options before removing them. Some of these rules might have been added as a special case of succeeding rules for special processing; for example to enable or disable logging or other tracking options, user or other forms of authentication etc. It may be that some of the special cases were needed for a temporary time period only but never cleaned up. So all these cases should be reviewed and any special cases that no longer require the special processing should be removed.

Here is the process for finding and removing these rules.

1. Find out rule by rule in the rule base, the rules that overlap with each other on all the Source, Destination and Service elements of the rule.
2. For each overlap found for a given rule, the rule can be removed
 - a. If there is a rule below in the rule sequence that is the first rule that completely covers this rule and have the same rule action and rule options.
 - b. If there are one or more rules in the rule base that together cover this rule and have the same rule action and rule options. If any of the rules that overlap this rule have a different action, then you cannot remove this rule.

Examples: In the following rule base: Rules 1 and 2 are covered by rule 5. Rule 3 is a special case of rule 4 with a special processing of sending a snmp trap to an snmp server. Rule 6 is covered by rules 7 and 8. Rules 1, 2 and 6 can be safely removed without affecting firewall behavior. However rule 3 can only be removed if snmp traps are not needed for the service udp-5901.

The address object group DMZ_svrs contains all the hosts present in the object groups DMZ_app and DMZ-web while the service group Web contains http and https as members. Similarly object group inside_servers contains some of the hosts from the object group internal_nets.

Please note that Check Point™ Policy verification does not complain about these redundancies during Policy installation

NO.	SOURCE	DESTINATION	VPN	SERVICE	ACTION	TRACK	INSTALL ON	TIME
1	Net_10.10.1.0	* Any	* Any Traffic	TCP telnet	accept	- None	* Policy Targets	* Any
2	Net_10.10.1.0	* Any	* Any Traffic	TCP ssh	accept	- None	* Policy Targets	* Any
3	internal_nets	* Any	* Any Traffic	UDP udp-5901	accept	SnmpTrap	* Policy Targets	* Any
4	internal_nets	* Any	* Any Traffic	UDP udp-high-ports	accept	- None	* Policy Targets	* Any
5	Net_10.10.1.0	* Any	* Any Traffic	TCP telnet-ssh	accept	- None	* Policy Targets	* Any
6	inside_servers	DMZ_servers	* Any Traffic	TCP http	accept	- None	* Policy Targets	* Any
7	internal_nets	DMZ_app	* Any Traffic	TCP ftp TCP ssh TCP http	accept	- None	* Policy Targets	* Any
8	internal_nets	DMZ_web	* Any Traffic	Web	accept	- None	* Policy Targets	* Any
9	* Any	* Any	* Any Traffic	* Any	drop	- None	* Policy Targets	* Any

Identifying unused rules

Identifying rules that become stale because the business purpose for that rule went away might not be straight forward because of absent documentation on the firewall rules and the transitioning of the firewall administrator and/or the business owner of the rule. Identifying these rules then requires analysis of usage data present in the firewall logs. The Check Point™ firewall logs all traffic that is allowed by a rule if the Tracking option on the rule is set to Log option. So the firewall rules need to be enabled with the log option first before collecting the usage data from the firewall. No deduction can be made about rules that do have the tracking option set to Log. Once the usage data is collected for a reasonable time period, then the usage data can be analyzed to find rules that do have Tracking set to Log and have zero usage. These rules should then be disabled with appropriate documentation and can be removed after monitoring for any service availability complaints.

Check Point™ firewall stores in the logs, the unique Rule UID of the rule that is responsible for the traffic getting denied or allowed. This Rule UID can be used to collect all used rules even as rules are being modified, added or deleted. Any rule that is not used and do not have the logging enabled can then be determined as an unused rule.

How does Athena FirePAC help?

Athena FirePAC automates the cleanup and optimization of firewall configurations. Using FirePAC, administrators can isolate more rules for removal than any other solution. FirePAC performs a thorough analysis of the rule overlaps and dependencies to identify every possible rule relationship. Whether your firewall has a few hundred or thousands of rules, FirePAC guarantees you will reduce your maintenance burden by at least 10-30% above and beyond the redundancies identified by Check Point™ Policy Verification. Specifically, Athena FirePAC allows you to identify:

1. All rules that can be safely removed without affecting firewall behavior. These include rules that are covered by one or more preceding or succeeding rules.
2. Additional candidates for removal based on special processing options in the rules.
3. The most used and unused rules from firewall logs or access list hit count data.
4. All unused objects that are not referred in any rules.
5. Optimized rule order, which takes the rule order dependencies into account to move the most used rules as far as possible up the rule set.

See an example FirePAC rule cleanup and optimization report here at:

<http://www.athenasecurity.net/pdf/ExampleCP-02-FirewallCleanup.pdf>

Who is Athena Security?

Over 300 companies use Athena products to clean up the firewall rules and reduce the risks to critical hosts by eliminating the vulnerabilities, non-compliances and errors in firewall infrastructure. Athena FirePAC is an affordable, easy to use firewall analysis tool for large or small enterprises. It confirms that each firewall is configured to behave correctly. FirePAC performs safe, offline analysis on the rule base to predict how data flows through the firewall to reach critical hosts. Install it on your desktop in seconds, and generate reports that reveal exactly how your firewall is working. See more at <http://www.athenasecurity.net>