[Figure 1: Main Flow 4](#_Toc176443155)

[Figure 2: Rule Processing Flow 5](#_Toc176443156)

[Figure 3: Input Data 6](#_Toc176443157)

[Figure 4: Rule 1 Conversion 7](#_Toc176443158)

[Figure 5: Rule 2 Conversion 8](#_Toc176443159)

[Figure 6: Rule 3 Conversion 9](#_Toc176443160)

[Figure 7: Rule 4 Conversion - part 1 10](#_Toc176443161)

[Figure 8: Rule 4 Conversion - part 2 11](#_Toc176443162)

[Figure 9: Rule 5 Conversion 12](#_Toc176443163)

[Figure 10: Partial sample of rule 1 conversion with flow IDs included 13](#_Toc176443164)

# Firewall Rule Processing Workflow

## Overview

This document describes how firewall rule requests are processed and organized to make them easier to implement across a network. The process takes raw firewall rule requests and transforms them into a structured format that network administrators can use to configure firewalls efficiently. It helps bridge the gap between those requesting firewall changes and the network administrators implementing them, ensuring that security policies are implemented completely and correctly across the network.

## Steps in Processing Firewall Rules

### 1. Expanding Rule Requests

When someone requests a firewall rule, they typically specify:

- Source IP addresses

- Destination IP addresses

- The type of traffic allowed (e.g., ports or services)

- Any additional comments or justifications

The first step in processing these requests is to expand them based on the network topology. This means:

- Identifying all the firewalls that the traffic might pass through

- Creating separate entries for each firewall that needs to be configured

This expansion ensures that we don't miss any firewalls that need to be updated to allow the requested traffic.

### 2. Grouping Similar Rules

After expanding the rules, we group them based on similarities. This grouping helps reduce redundancy and makes the rules easier to manage. We group rules that have:

- The same source and destination networks

- The same types of traffic allowed

- The same path through the network

By grouping similar rules, we can reduce the number of individual firewall configurations needed, which saves time and reduces the chance of errors.

### 3. Consolidating Gateway Information

In many networks, traffic might pass through multiple firewalls or gateways. To make implementation easier, we consolidate information about these gateways. This means:

- Identifying all the gateways that a particular rule affects

- Combining this information into a single entry

This consolidation gives network administrators a clear view of which devices need to be configured for each rule or group of rules.

### 4. Formatting for Easy Reading

The final step is to format the processed rules in a way that still represents what the user requested and intended:

- Add backing group headings even when the groups have been split across multiple rules

- Add back in the description for each IP address or network that the user originally provided

- Clearly listing all source and destination IP addresses

- Providing clear descriptions of the traffic being allowed

- Including any relevant comments or justifications

## Benefits of This Process

This workflow provides several benefits:

1. \*\*Completeness\*\*: By expanding rules across all relevant firewalls, we ensure no devices are missed.

2. \*\*Efficiency\*\*: Grouping similar rules reduces redundancy and saves time in implementation.

3. \*\*Clarity\*\*: Consolidating gateway information and formatting the output makes it easier for administrators to understand and implement the rules.

4. \*\*Accuracy\*\*: By automating much of this process, we reduce the chance of human error in interpreting and implementing complex rule requests.

## Operation and Sample Input/Output

The following sections provide flowcharts that illustrate the logical flow. A contrived example is also provided to show how rules are transformed from the user request to the end result.

### Main Flow



Figure 1: Main Flow

### Rule Processing Flow



Figure 2: Rule Processing Flow

### Sample Input Data and Subsequent Rule Expansion and Transformation



Figure 3: Input Data



Figure 4: Rule 1 Conversion



Figure 5: Rule 2 Conversion



Figure 6: Rule 3 Conversion



Figure 7: Rule 4 Conversion - part 1



Figure 8: Rule 4 Conversion - part 2



Figure 9: Rule 5 Conversion



Figure 10: Partial sample of rule 1 conversion with flow IDs included



Figure 11: Output Options



Figure 12: Input User Request from Excel



Figure 13: Add or Modify Customers and Topologies



Figure 14: Re-render Graphviz Diagrams

## Defining Network Topologies

The location within the network topology of the subnets that will cover the requested rules needs to be defined for at least one topology. If no matching network is found and there is no default route defined (the subnet matching is done on longest match) the IPs will be excluded from the final rule set. This will be outputted to the user to notify them that these networks need to be defined.

The topology can be defined using YAML format and/or by providing a route dump from the firewalls or Routers. Below is the example being used in YAML format:

FW1:  
 - 10.1.1.0/24  
 - 10.1.2.0/24  
FW2:  
 - 10.2.1.0/24  
 - 10.2.2.0/24  
FW3:  
 - 10.3.1.0/24  
 - 10.3.2.0/24  
FW4:  
 - 10.4.1.0/24  
 - 10.4.2.0/24  
AZURE:  
 - 10.100.1.0/24  
 - 10.100.2.0/24  
AWS:  
 - 10.200.1.0/24  
 - 10.200.2.0/24  
TRANSIT\_FW:  
 - 10.0.1.0/24  
 - 10.0.2.0/24

Below is an example of the route dump the can be provided and loaded in alongside the YAML file above.

FW1  
10.1.101.0 10.1.6.1 255.255.255.0 UGHD 0 0 0 bond1.6  
10.1.102.0 10.1.6.1 255.255.255.0 UGHD 0 0 0 bond1.6  
10.1.103.0 10.1.6.1 255.255.255.0 UGHD 0 0 0 bond1.6  
FW2  
10.2.101.0 10.2.6.1 255.255.255.0 UGHD 0 0 0 bond1.6  
10.2.102.0 10.2.6.1 255.255.255.0 UGHD 0 0 0 bond1.6  
10.2.103.0 10.2.6.1 255.255.255.0 UGHD 0 0 0 bond1.6

Below is the network topology diagram which is expressed as a text source file below. The intention is to use a network diagram in text form that the script can analyse to determine the network but which can also be displayed graphically to assist the user.

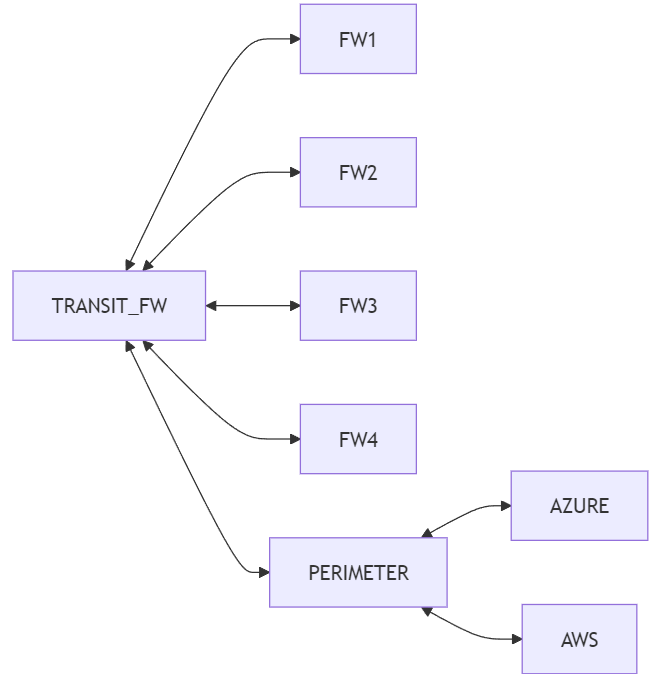


Figure 15: Sample Topology Diagram

The text that represents this diagram is below, the script reads this to obtain a map of the network.

flowchart LR  
ZONE1 <--> FW1  
TRANSIT\_FW <--> FW1  
TRANSIT\_FW <--> FW2  
TRANSIT\_FW <--> FW3  
TRANSIT\_FW <--> FW4  
TRANSIT\_FW <--> PERIMETER  
PERIMETER <--> AZURE  
PERIMETER <--> AWS

These examples are for one topology only each topology specified will need its own subnet definition, optional route dump and network topology diagram source file.