TECHNICAL WHITE PAPER - **APRIL 2018**

VMWARE NSX DISTRIBUTED FIREWALL:

Policy Rules Configuration

Table of Contents

Introduction

NSX Distributed Firewall (DFW) provides capability to enforce firewalling function directly at the vNIC layer of a virtual machine (VM). It is the core component of a micro-segmentation security model where east-west traffic can now be viewed, inspected and enforced at near line rate processing, helping prevent many types of lateral attacks.

This technical white paper gives details about how to easily configure and create and consume DFW rules. The first part will cover performing creation using tools such as the DFW Rule Table and Service Composer, the extended capabilities with Layer 7 context-awareness, and will finish showing automated recommendation capabilities in NSX 6.4 with the Application Rule Manager.

The Distributed Firewall now provides Layer 7 context-awareness capabilities to allow admins the ability to control security policies for east-west communication in the data center using context beyond the standard Layer 2-4 methods. Application Rule Manger provides admins with recommendations for micro-segmentation rule sets based on the traffic flows observed during monitoring, further simplifying the day 2 operations with NSX.

We assume reader has already some knowledge of NSX objects, DFW and Service Composer functions. Please refer to the appropriate collateral found in the reference section for more information on individual NSX components.

DFW Object Grouping Model

It's important to understand the methodologies that can be used to build NSX DFW policies. Not all policies apply for the same situation or configuration of the applications, infrastructure, or networking of the organization. The following section takes each of these methodologies and highlights appropriate usage.

Methodologies

The following methodologies are the typical use cases for building security rule sets within the NSX DFW.

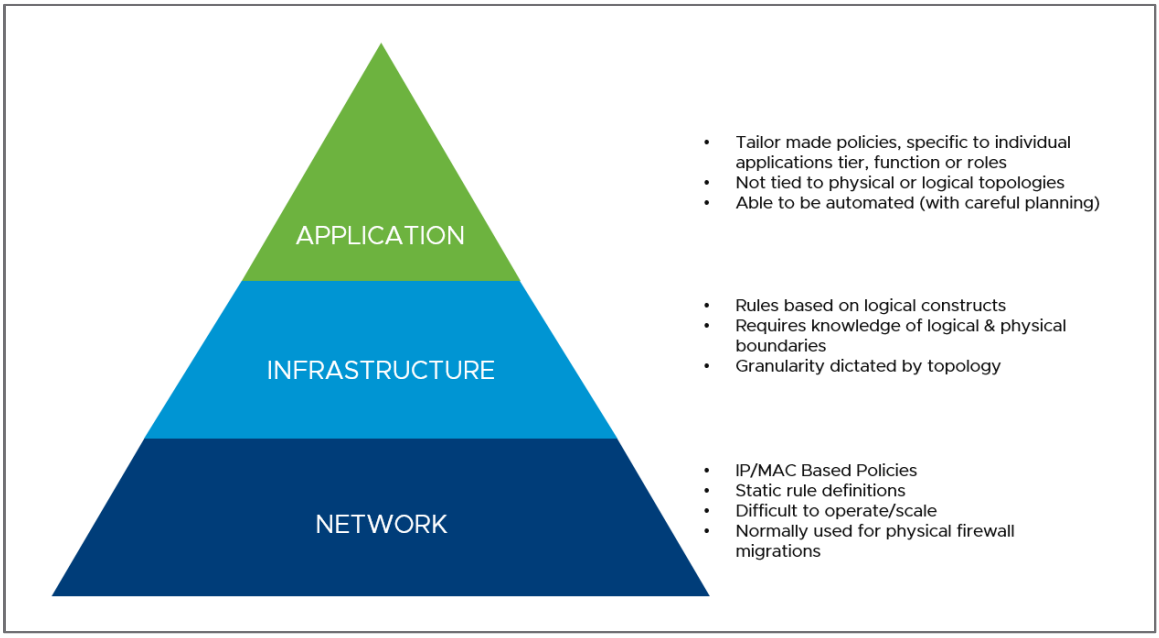


Figure 1 - Micro-segmentation Methodologies

Application

In this approach, grouping is based on the application type (e.g: VMs tagged as "Web_Servers"), application environment (e.g: all resources tagged as "Production_Zone") and application security posture. The advantage of this approach is the security posture of the application is not tied to either network constructs or infrastructure. Security policies can move with the application irrespective of network or infrastructure boundaries. Policies can be templated and reusable across instances of same types of applications and workloads. You can use variety of mechanisms to group. The security team needs to be aware of only the application needed to be secured based on the policies. The security policies follow the application life cycle, i.e. comes alive when the application is deployed and is destroyed when the application is decommissioned.

When not to use this: If the environment is static without mobility and infrastructure functions are properly demarcated. You do not need to use application-based policies.

Application-based policy approach will greatly aid in moving towards a Self-Service IT model. The Security team needs to be only aware of how to secure an application without knowing the underlying topology. Concise and reusable security rules will require application awareness. Thus a proper security posture can be developed via application based policies.

Infrastructure

In this approach, grouping is based on infrastructure like vCenter clusters, logical switches, distributed port groups, etc. An example of this would be, clusters 1 to cluster 4 are earmarked for PCI applications. In such a case, grouping can be done based on cluster names and rules can be enforced based on these groups. Another example would be, if you know which logical switches in your environment are connected to which applications. E.g. App Tier Logical switch contains all VMs pertaining to application 'X'. The security team needs to work closely with the vCenter administration team to understand logical and physical boundaries.

When not to use this: If there are no physical or logical boundaries in your environment then this type of approach is not suitable. Also, you need to be very careful where you can deploy your applications. For example, if you would like to deploy a PCI workload to any cluster that has adequate compute resources available; the security posture cannot be tied to a cluster but should move with the application.

Network

This is the traditional approach of grouping based on L2 or L3 elements. Grouping can be based on MAC addresses or IP addresses or a combination of both. NSX supports this approach of grouping objects. The security team needs to aware of networking infrastructure to deploy network-based policies. There is a high probability of security rule sprawl as grouping based on dynamic attributes is not used. This method of grouping works great if you are migrating existing rules from a different vendor's firewall.

When not to use this: In dynamic environments, e.g. Self-Service IT; Cloud automated deployments, where you are adding/deleting of VMs and application topologies at a rapid rate, MAC addressed based grouping approach may not be suitable as there will be delay between provisioning a VM and adding the MAC addresses to the group. If you have an environment with high mobility like vMotion and HA, L3/IP based grouping approaches may not be adequate either.

NSX Security Consumption Model

NSX Consumption model simplifies provisioning, audit, troubleshooting of security. Security policies are tied to the workload and not the infrastructure. Troubleshooting can now start at the workload instead of starting at

infrastructure components. End-Users and Cloud Admins can now define a Security Policy once and apply to a group of workloads, Security Groups.

Figure 2 - NSX Consumption Model

Security Groups

Security Groups is an extremely important concept. It allows you to abstract out the grouping of workloads from the underlying infrastructure topology. This allows a Security Policy to be written for a workload or for a zone like PCI zone, DMZ or production zone. Extremely powerful granular DFW rules can be written or can be combined with other security controls like Intrusion Prevention (IPS), Next Generation Firewall (NGFW), Vulnerability Scanner, Anti-Virus (AV).

A Security Group is a logical construct that allows grouping into a common container of the following elements

- Static Criteria – vCenter/NSX Objects
- Dynamic Criteria – VM Properties

Static Criteria includes both inclusion criteria and exclusion criteria. Static Inclusion provides capability to manually include particular objects into the Security Group. Static Exclusion provides capability to manually exclude particular objects from the Security Group. For dynamic inclusion criteria, Boolean logic can be used to create groups between various criteria.

A Security Group constructs a logical grouping of VMs based on this calculation:

The following are selection criteria of Security Groups:

Table 1 - vCenter Objects used for Security Groups

vCenter/NSX Object	Description
Cluster	All VM/vNIC within this ESXi cluster will be selected.
Datacenter	All VM/vNIC within this Datacenter cluster will be selected.
Distributed Port Group	All VM/vNIC connected to this DVS port-group will be selected.
IP Sets	Selected IP Sets container will be used. IP Sets contains individual IP address or IP subnet or range or IP addresses.
Legacy port group	All VM/vNIC connected to this VSS port-group will be selected.
Logical Switch	All VM/vNIC connected to this Logical Switch (or VXLAN) segment will be selected.
Resource Pool	All VM/vNIC defined within the Resource Pool will be selected.
Security Group	All VM/vNIC defined within the Security Group will be selected.
vApp	All VM/vNIC defined within the vApp will be selected.
Virtual Machine	All VM/vNIC will be selected.
vNIC	This particular vNIC instance will be selected.
MAC Sets	Selected MAC sets container will be used. MAC sets contains a list of individual MAC addresses.
Active Directory Group	AD Group.

Table 2 - VM Properties used for Security Groups

VM Property	Description *
VM Name	All VMs that contain the string as part of their name.
Security Tags	All VMs that are applied with specified NSX Security tags.
Computer Name	All VMs that contain the string as part of their computer name.

Computer Operating System	All VMs that are running a particular operating system of the VM.
Entity	All VMs that belong to a particular vCenter Object as listed in the table above.

Note: *Limited regex available on all the VM Properties to create Security Groups.

Security Groups can be written using more complicated selection criteria expression statements using the selection criteria below. A Security Group can be comprised of multiple Security Groups. Additionally, a VM can be part of multiple Security Groups.

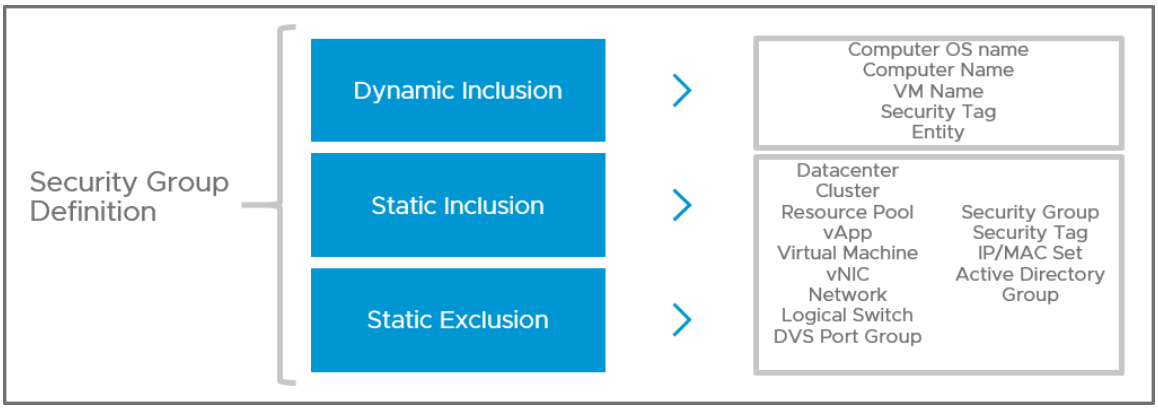


Figure 3 - Security Groups Definition

Examples of Security Groups as expressions:

- Dynamic inclusion criteria can state all VMs with name starting by "WEB-VM" to be included in Security Group named SG-ALL-WEB-VM.
- Dynamic inclusion criteria states all VMs containing the name "APP-VM" *and* having a Security Tag "PCI Zone" to be included in Security Group named SG-ALL-PCI-APP-VM.
- Static inclusion criteria can state all VMs that are connected to a logical switch "DB-Tier-LS" to be included in Security Group named SG-ALL-DB-Tier-LS.
- A combination inclusion and exclusion criteria could be – Select all VMs included in "DB-Tier" & "APP-Tier" Logical switches that are security tagged as "Production" but exclude all "APP-VMs" in the "PCI Zone" or SG-ALL-PCI-APP-VM.

Security Policy

Security rules are written in a Security Policy. A Security Policy can contain multiple security rules for different security controls. Security Controls are broadly divided into three parts.

- Guest Introspection Services (e.g. AV, Vulnerability Scan, DLP, File Integrity Monitoring)
- Distributed Firewall Rules
- Network Introspection Services (e.g. NGFW, IPS)

Figure 3 shows examples of Security policies and Figure 4 shows the Security Policy screen in the Service Composer.

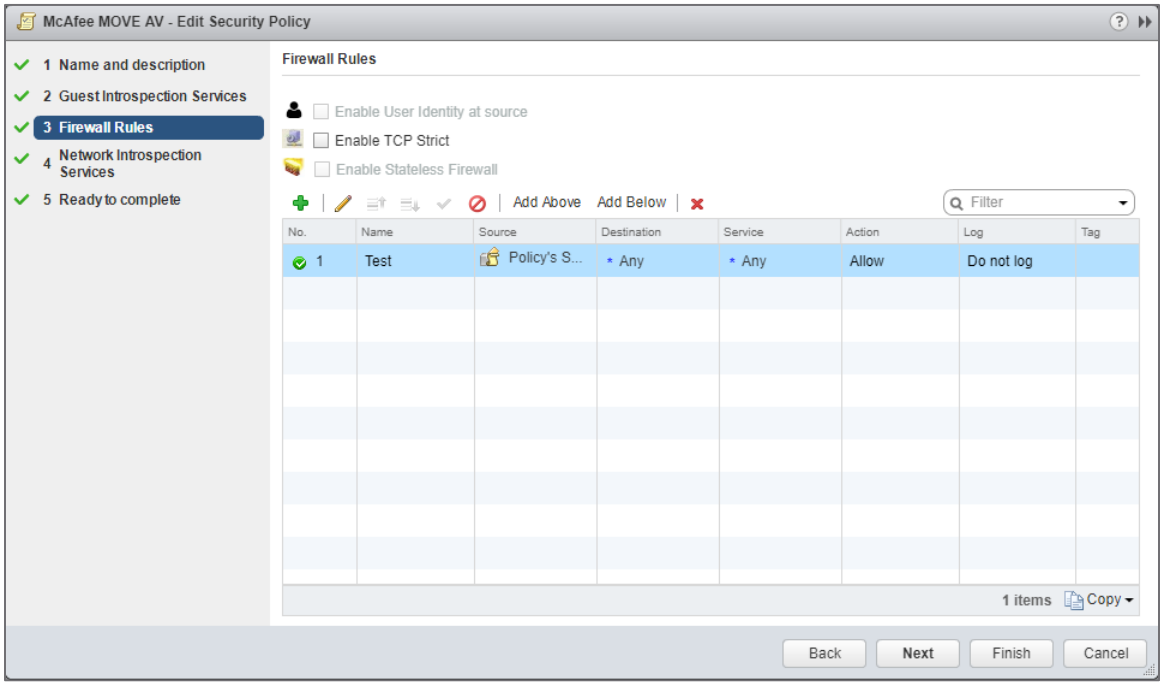


Figure 4 - NSX Security Policy Interface

Inheritance & Precedence in Security Policies

Security Policies leveraged through Service Composer are applied in a weighted fashion. Certain policies may take effect before others. It's also possible to nest policies within policies creating an order of inheritance. Understanding this dynamic is critical when using Service Composer.

Inheritance from a Policy

Inheriting rules from a parent Security Policy can create a child Security Policy.

Inheritance from a Security Group

If a Security Policy is applied to a Security Group, the Security Groups that comprise of the Security Policy will inherit the same Security Policy.

Precedence of a Policy

Policy precedence or weight can be assigned to a Security Policy. If a Security Group contains multiple Security policies applied, the Security Policy of the highest weight will be executed first. If all the policies are of equal weight, re-ordering of policies for a Security Group allows you to decide which policy takes precedence.

Security Rule Model

When building a network least privilege type of security model, the order and classification of security rules is paramount. The following model represents an overview of the different classifications of security rules that can be placed into the NSX DFW either through the Firewall Rule Table or through a Security Policy with Service Composer.

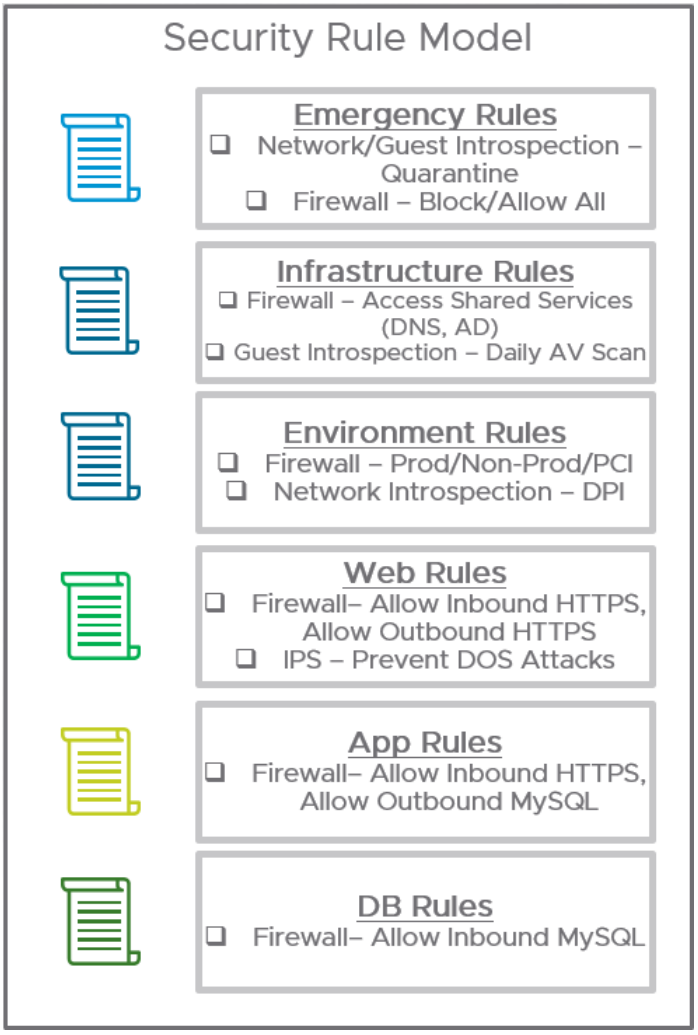


Figure 5 - Security Rule Model

Consumption of DFW

There are two ways to consume DFW functions, the Service Composer and Firewall Rule table. Both of these tools are also configurable via the management plane REST API:

Using Service Composer

Click on Networking & Security -> Service Composer to access Security Policy table:

In this menu, a Security Policy (SP) must be created. Within the SP, DFW policy rules can be defined and then applied to a Security Group that contains your intended VMs

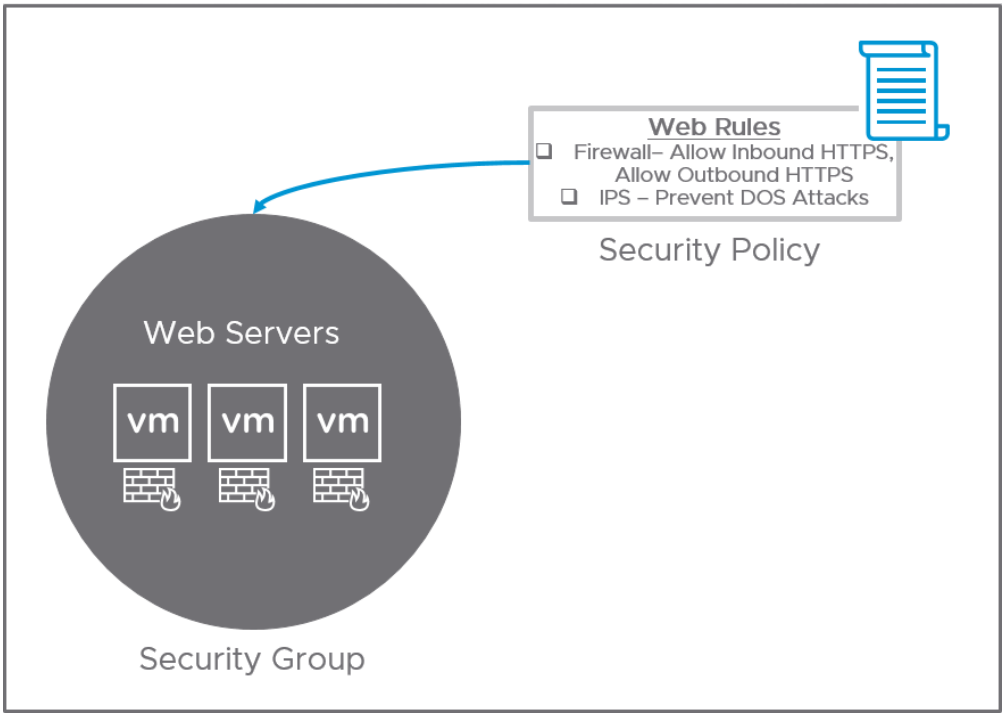


Figure 6 - Web Security Group and Policy Example

Using Firewall Rule Table

Click on Networking & Security -> Firewall to access the Firewall Rule Table.

In this menu, administrator enter Security Policy rule as needed using the standard rule schema: rule name / source / destination / service / action

Table 3 - DFW Rule Table Example

Name	Rule ID	Source	Destination	Service	Action	Applied To
APP Access	1004	Any	SG-APP-ALL	APP-SVG-ALL	Allow	SG-APP-ALL

We will go into detail on defining DFW rules using both the methods in the next section. Let's look at the comparison table between the two methods.

Table 4 - DFW and Service Composer Comparison

	Using Service Composer	Using Firewall Rule Table
Visibility in Firewall Rule Table	●	●
Visibility of Firewall Rules on a VM	●	●
Visibility of blocked flow in Flow Monitoring	●	
Direct Modification of Firewall Rule		●
DFW Rule reusability	●	
Service Insertion	●	
Guest Introspection	●	

DFW Policy Rule Configuration

Using Service Composer

Click on Networking & Security -> Service Composer to access Security Policy table:

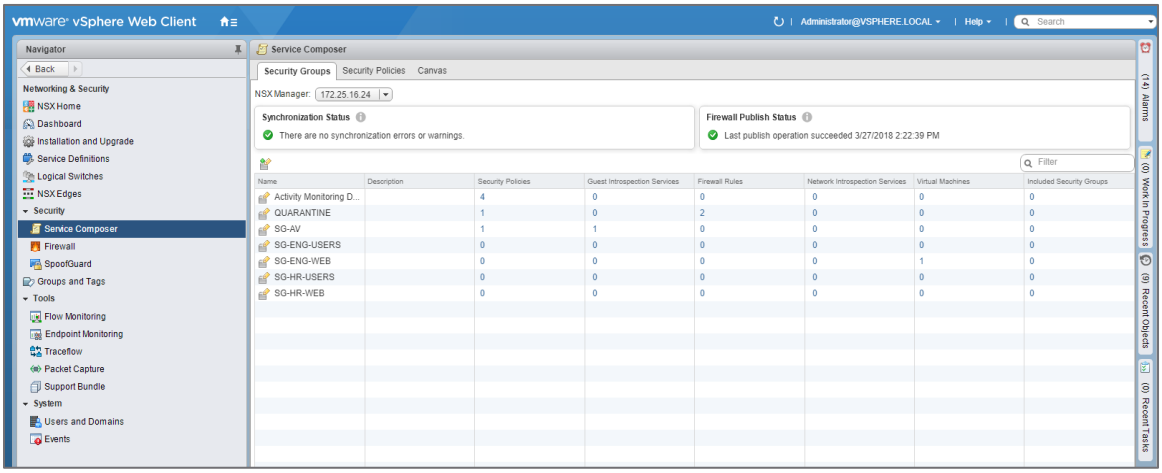


Figure 7 - Service Composer - Security Policy Table

The window displays all Security Policies (SP) defined under NSX. Selecting a particular Security Policy will open a window showing the content of the SP as shown below:

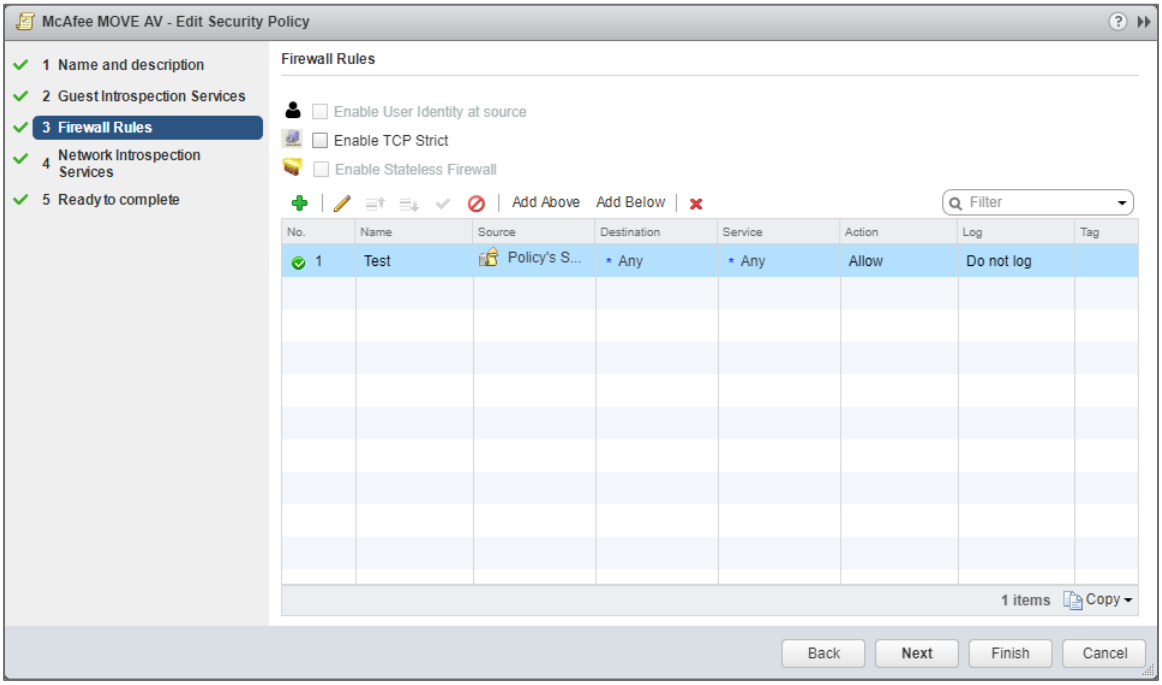


Figure 8 – Service Composer - Security Policy Creation

To create a new policy rule, click on the + button as show in the diagram above. The following window appears:

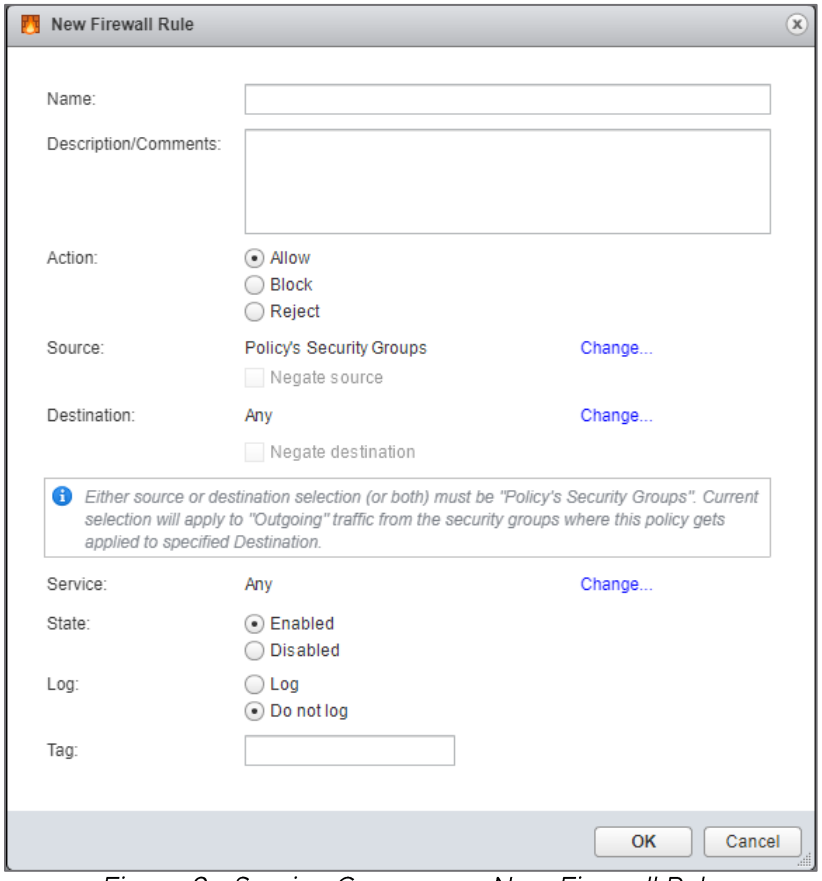


Figure 9 - Service Composer - New Firewall Rule

Admin fills in the following fields:

- **Name**: firewall rule name
- **Description**: firewall rule description
- **Action**: Allow or Block
- **Source / Destination**: can take any of these values

Table 5 - Service Composer - Source/Destination Values

Value	Description
Policy's Security Group	When applying the Security Policy to 1 or more Security Group (SG), this field will be internally replaced with the applied SG(s).
Any	Any.
Select Security Groups	User can select 1 or more Security Groups (NSX will list all available SG in the system for selection).

Following screenshot shows Source/Destination field selection window:

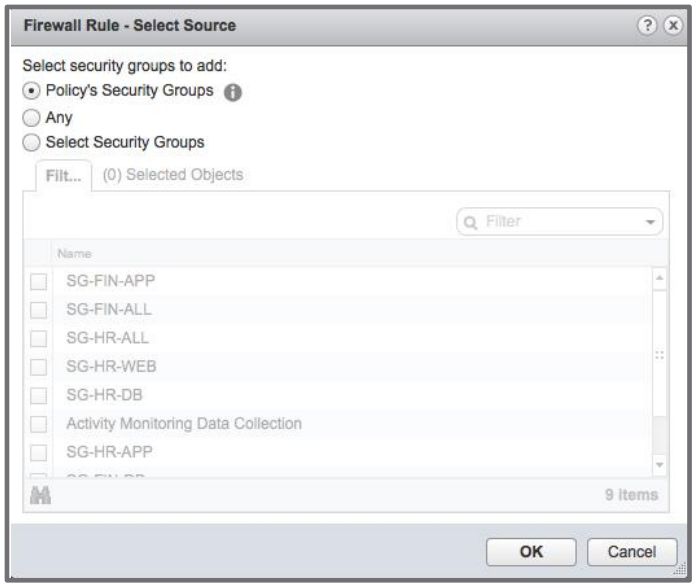


Figure 10 - Service Composer - Select Source

- **Service**: can take any of these values, including Layer 4 and Layer 7 Services.

Table 6 - Service Composer - Service Values

Value	Description
Any	Any Service
Select Services and Service Groups	Pre-defined Services and Services Groups.

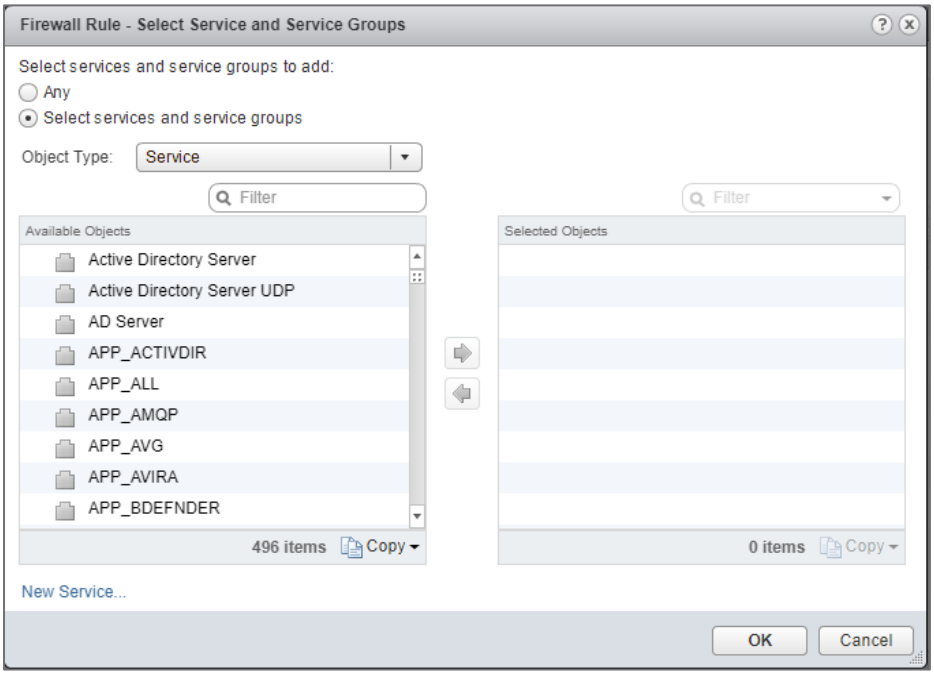


Figure 11 - Service Composer - Select Service and Service Groups

- **State**: Enable or disable this rule.
- **Log**: log or do not log packets matching this rule.

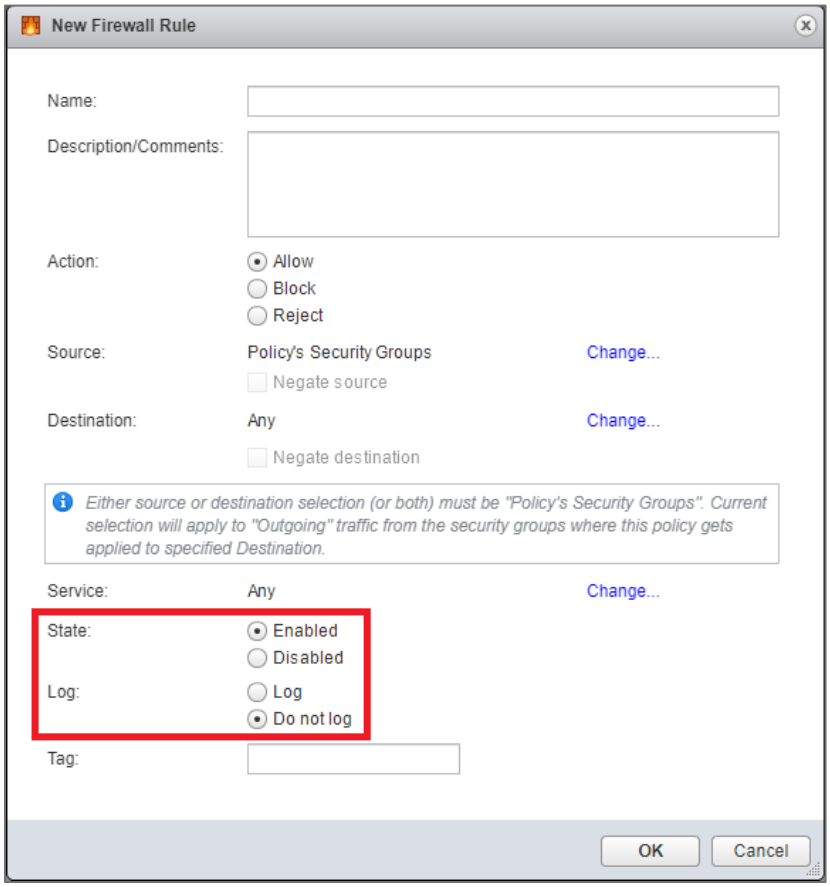
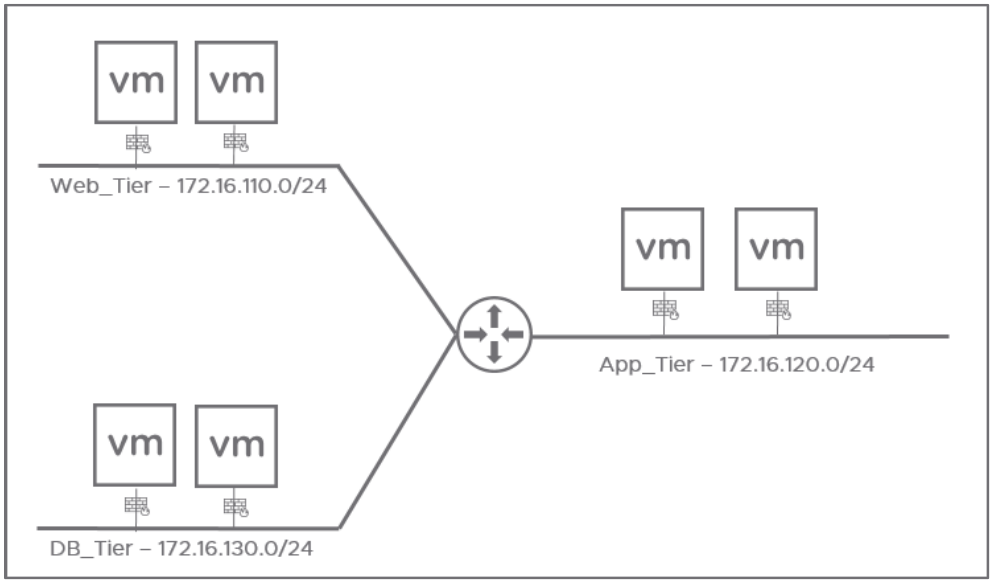


Figure 12 Service Composer Firewall Rule Build

Example I: 3-Tier Application

Let's use the logical network topology and see how to configure firewall (DFW) rules using Service Composer/Security Policy.

Take the simple case of having 3 different set of rules for WEB, APP and DB VMs



First step is to define properly Security Groups (SG).

Table 7 – Service Composer - 3-Tier Application Security Groups

SG name	SG definition
SG-WEB	Static inclusion: Web LS
SG-APP	Static inclusion: App LS
SG-DB	Static inclusion: DB LS

We want to create DFW rules for all applications such that:

- Web VMs can only talk to App VMs using an enterprise bus service.
- Web VMs cannot talk to DB VMs.
- App VMs can only talk to DB VMs via SQL Service.

Security Policy/Firewall rules will then look like this (SP-APP)

Table 8 - Service Composer – 3-Tier Application Security Policy Firewall Rules

Name	Source	Destination	Service	Action
Web to App	SG-WEB	Policy's Security Group	<Enterprise Service Bus>	Allow
App to DB	Policy's Security Group	SG-DB	SQL	Allow
Default	Any	Any	Any	Block

Last step is to apply the Security Policy (SP-APP) to SG-APP to make it operational.

Example II: Production Zone vs DevTest Zone

If some of the VMs are production VMs and some of them are DevTest VMs. Let's say all VMs that end with "01" are production VMs

We want to further apply rules such that – No Production VMs should talk to DevTest VMs

In the above solution, we will create some more SGs and SPs.

Table 9 - Service Composer - Security Group Layout

SG name	SG definition
SG-Prod	Dynamic Inclusion: All VM Names that end with "01"
SG-WEB	Static inclusion: Web LS Static Exclusion: SG-Prod
SG-APP	Static inclusion: App LS Static Exclusion: SG-Prod
SG-DB	Static inclusion: DB LS Static Exclusion: SG-Prod
SG-DevTest	Static Inclusion: SG-WEB, SG-APP, SG-DB

Create a SP (SP-RESTRICT-PROD) to restrict access of DevTest VMs to production. The SP will contain a rule like this:

Table 10 - Service Composer - DevTest/Prod Security Policy Firewall Rules

Name	Source	Destination	Service	Action
Production To DevTest	SG-Prod	Policy's Security Group	Any	Block
DevTest To Production	Policy's Security Group	SG-Prod	Any	Block

Apply this policy to SG-DevTest. Since the SG-DevTest is comprised of SG-APP, SG-WEB, SG-DB, this policy will automatically be applied to these SGs due to inheriting the SG-DevTest policy. Hence none of the VMs in those Security Groups will communicate with the production VMs.

Create the SP for the APP as shown above. Apply it to SG-APP. SG-APP will now contain two policies - SP-RESTRICT-PROD & SP-APP.

You can visualize the second case in the following way –

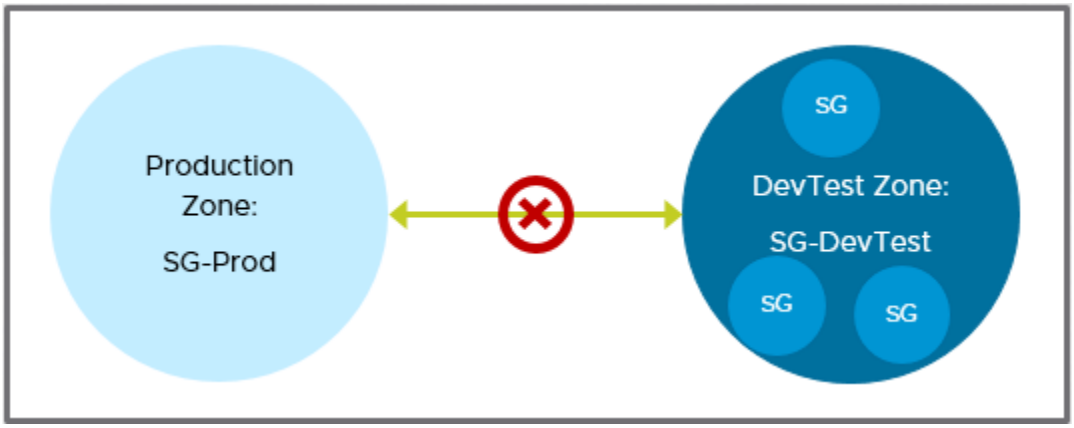


Figure 13 - Prod - Non-Prod Restriction

Using Firewall Rule Table

Click on Networking & Security -> Firewall to access the DFW policy rule table:

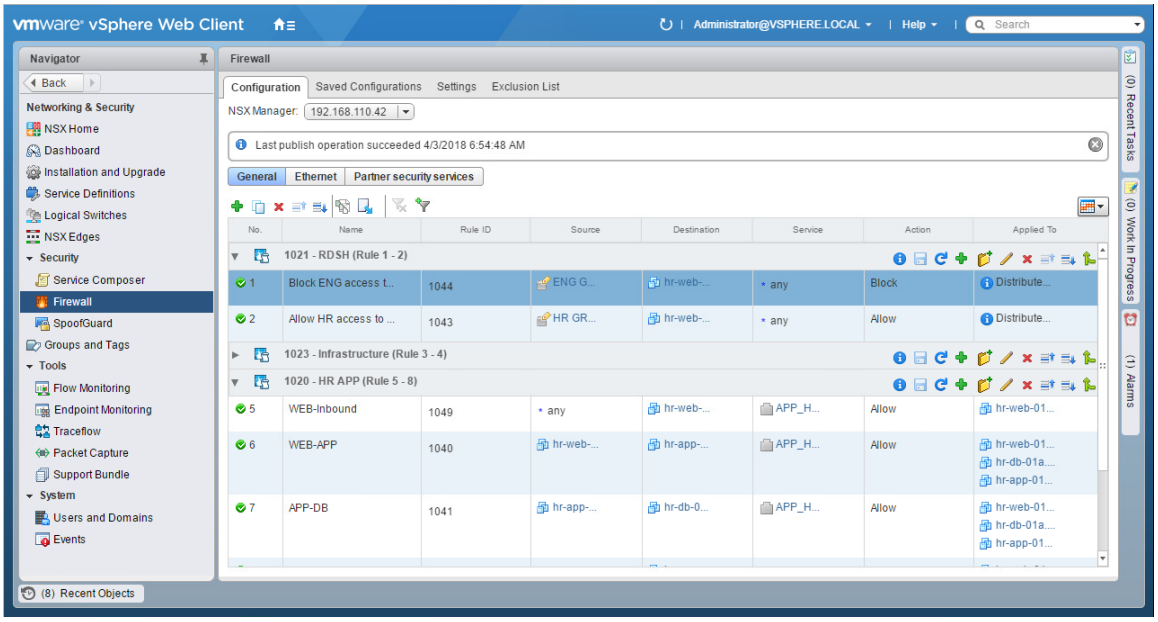


Figure 14 - DFW Policy Rule Table

The window displays all policy rules configured for DFW and packet lookup will be performed from top to bottom. Packet not matching any explicit rule will be enforced by default rule which is always the last one on the table. NSX comes with a DFW default rule set to Allow action. User can change it to Block if desired. VMware recommends using DFW with default rule set to Block and then create explicit rules for allowed traffic.

A policy rule is composed of the following fields:

Name	Rule ID	Source	Destination	Service	Action	Applied To

- **Rule Name**: User field which support up to 255 characters.
- **Rule ID**: Number with 4 digits (e.g. 1013) automatically allocated by DFW.
- **Source and Destination**: Source and Destination fields (respectively) of the packet. Possible entries are:
- IPv4/IPv6 addresses or subnets (e.g. 192.168.200.1, 192.168.200.1/24, 192.168.200.1-192.168.200.24). The following window allows to enter IP address information:

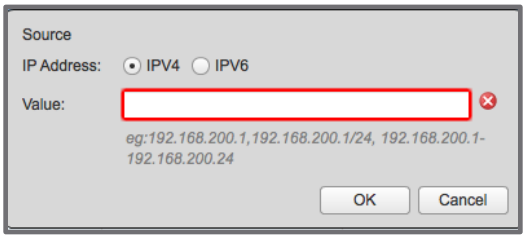


Figure 15 - Source/Destination: IPv4/v6 addresses or Subnets

The window to enter vCenter object for Source/Destination field is displayed below:

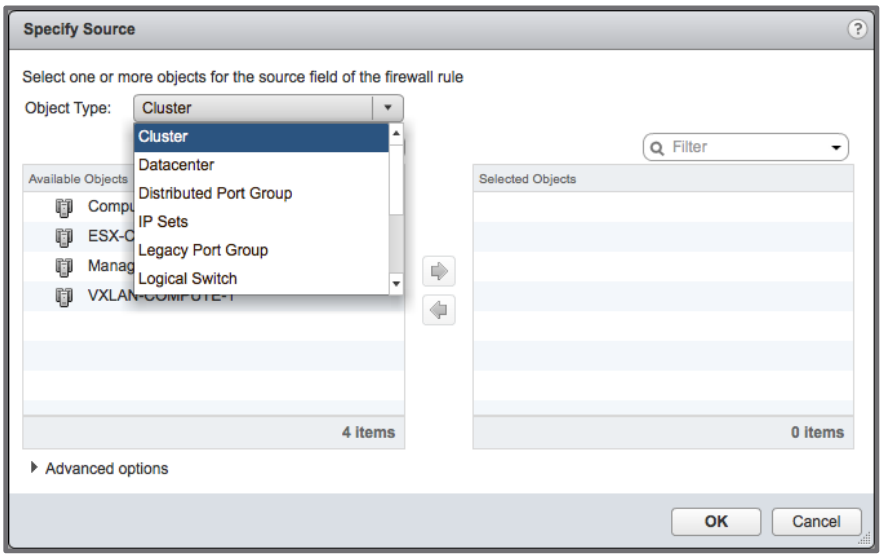


Figure 16 - Source/Destination: vCenter Objects

All permutations are possible for Source/Destination field: IP Address/Subnet and vCenter objects can be used individually or simultaneously.

Important note:
When using vCenter objects in Source or Destination field, it is mandatory to have VMtools installed on guest VM. VMtools enables DFW to retrieve IP address(es) of guest VM in order to enforce properly security rule.
If VMtools cannot be installed on a guest VM, use explicit IP address in Source or Destination field to secure traffic for this VM.

The following table list all possibilities:

Table 11 - Security Group vCenter Resource Values

Object	Description
Cluster	All VM/vNIC within this ESXi cluster will be selected.
Datacenter	All VM/vNIC within this Datacenter cluster will be selected.
Distributed Port Group	All VM/vNIC connected to this DVS port-group will be selected.
IP Sets	Selected IP Sets container will be used. IP Sets contains individual IP address or IP subnet or range or IP addresses.
Legacy port group	All VM/vNIC connected to this VSS port-group will be selected.
Logical Switch	All VM/vNIC connected to this Logical Switch (or VXLAN) segment will be selected.
Resource Pool	All VM/vNIC defined within the Resource Pool will be selected.
Security Group	All VM/vNIC defined within the Security Group will be selected.
vAPP	All VM/vNIC defined within the vAPP will be selected.
Virtual Machine	All VM/vNIC will be selected.
vNIC	This particular vNIC instance will be selected.

- **Service**: Protocols (TCP,UDP,..) or Pre-Defined Services or Pre-Defined Services Group can be selected.

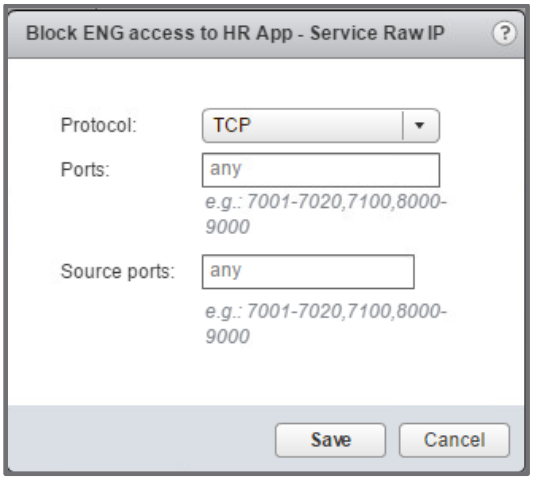


Figure 17 - Source/Destination: Protocols

When selecting Protocols like TCP or UDP, it is possible to define individual ports number (up to a maximum of 15).

User can pick other protocols such as FTP, ICMP, and ORACLE_TNS.

In the advanced options, it is possible to define source ports (up to a maximum of 15).

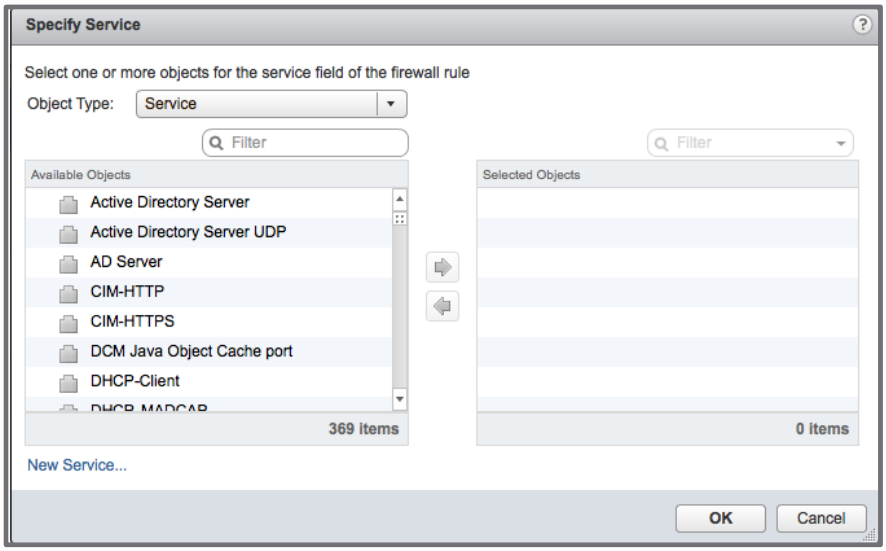


Figure 18 - Source/Destination: Pre-defined Services

User can pick any of the pre-defined Service (from the long list of available objects). This includes both Layer 4 & Layer 7 Services. Layer 7 services names are prefixed with "APP" for clear identification.

It is also possible to define custom Services by clicking on New Service link (in Figure 18). With NSX 6.4, an admin can select Layer 3, Layer4, or Layer 7 for a new custom service.

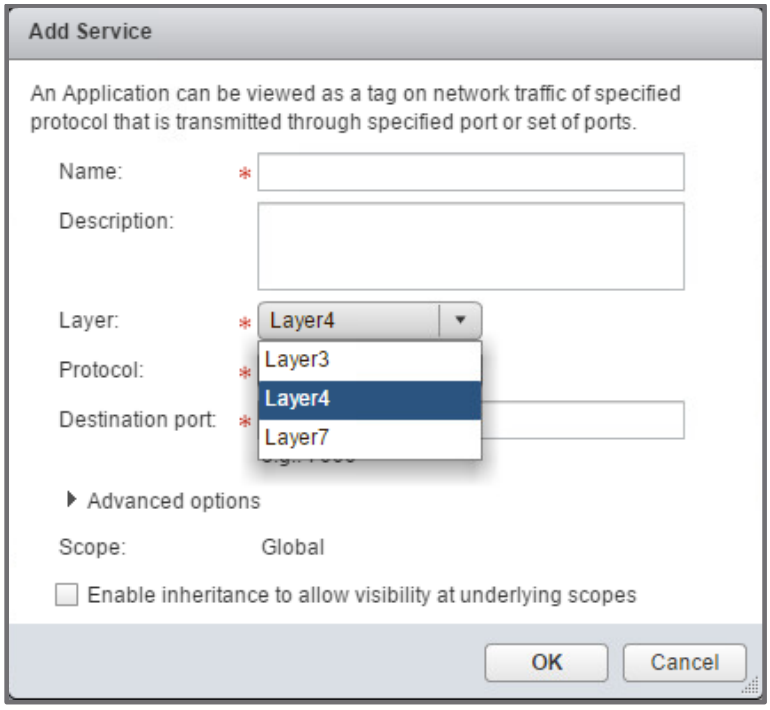


Figure 19 - Source/Destination: Add Custom Services

When selecting protocol like TCP or UDP, it is possible to define individual destination ports or a range of destination ports. This is also true for source ports when expanding advanced options link.

When selecting Layer 7, a list of Application IDs will be necessary to select from. By default, the Protocol field is set to 'Any'. Since Layer 7 context is based on an Application ID, a protocol & port does not need to be defined for the DFW to pick up on the Service. A protocol (Eg: TCP or UDP) and a Destination port can be input to further refine the Service.

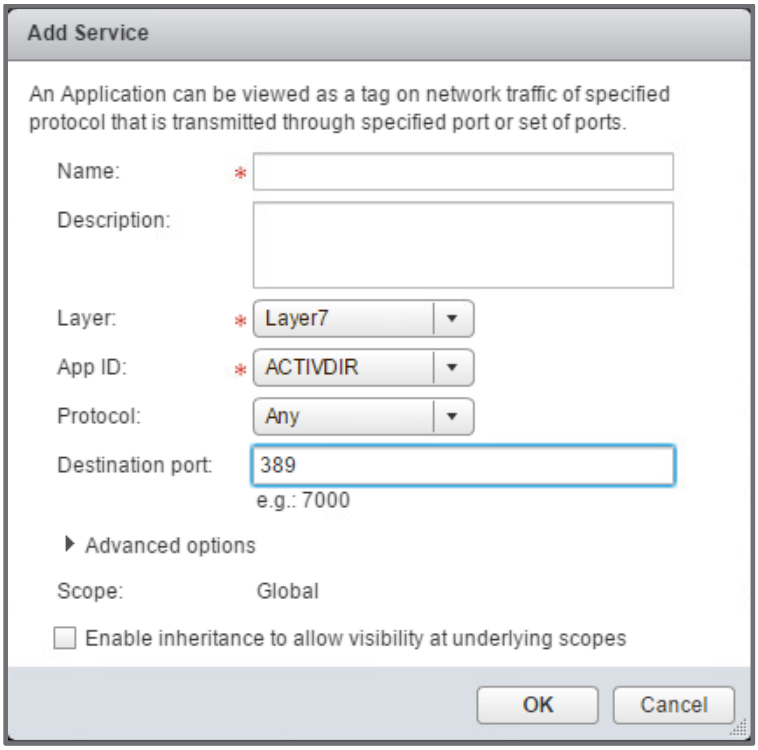


Figure 20 - Layer 7 Service

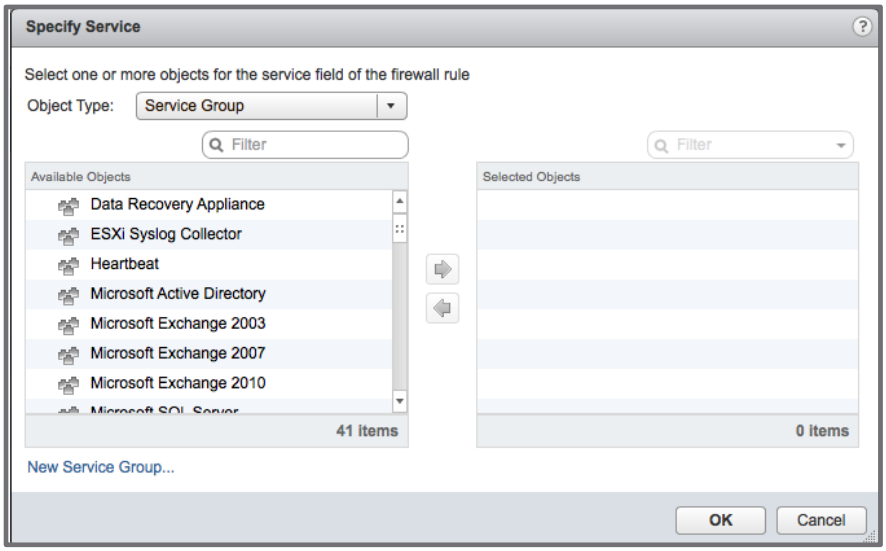


Figure 21 - Source/Destination: Pre-defined Service Groups

User can pick any of the pre-defined Services Group (from the long list of available objects).

It is also possible to define custom Services Group by clicking on New Service Group link (in Figure 21):

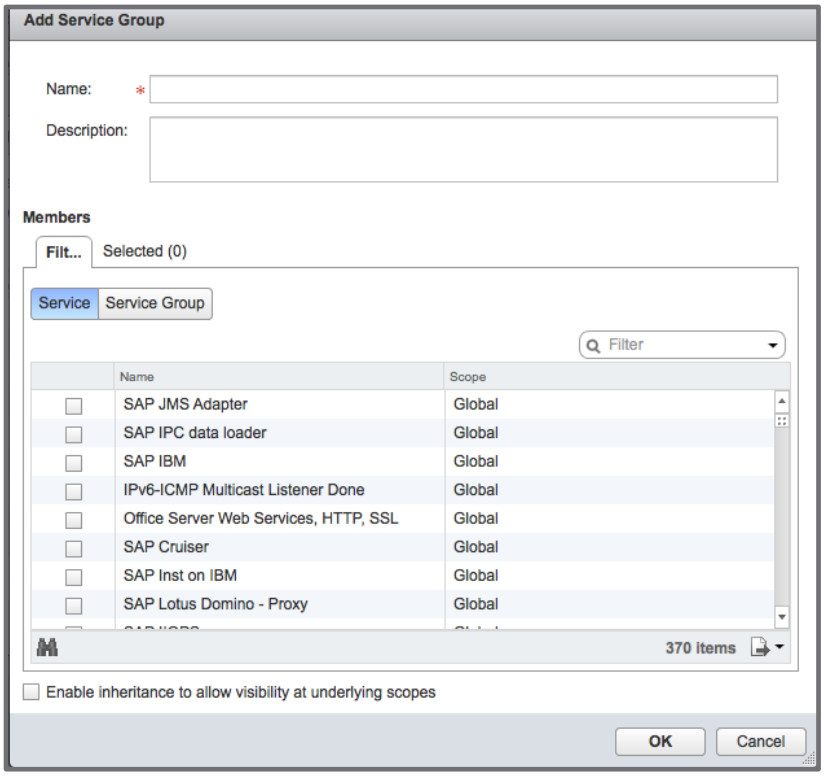


Figure 22 - Source/Destination: Add Custom Service Group

Custom Services Group is a collection of Services or Services Group. These Groups consist of a single, or multiple Services from within NSX.

- **Action**: define enforcement method for this policy rule. Available options are:

Table 12 - Firewall Rule Table - Action Values

Action	Description
Block	Block silently the traffic.
Allow	Allow the traffic.
Reject	Reject action will send back to initiator: • RST packets for TCP connections. • ICMP unreachable with network administratively prohibited code for UDP, ICMP and other IP connections.

Window to define policy rule action is displayed below:

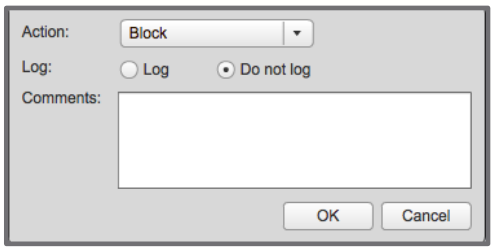


Figure 23 - Action for Policy Rule

On the same window, user can decide to enable packet logging or not. When enabled, the NSX DFW logs from each DFW-enabled vSphere host will send their dfwpktlogs to the configured Syslog server, such as vRealize Log Insight. This information can be used to build alerting and reporting based on the information within the logs, such as blocked or allowed packets.

- **Applied To**: define scope of rule publishing. User can decide to publish policy rule to all clusters where DFW was enabled or restrict publication to a specific object as listed below:

Table 13 - Firewall Rule Table - Applied To Values

Object	Description
Cluster	Selecting Cluster will push the rule down to all VM/vNIC on the ESXi cluster.
Datacenter	Selecting Datacenter will push the rule down to all VM/vNIC on the Datacenter.
Distributed Port Group	Selecting DVS port-group will push the rule down to all VM/vNIC on the Datacenter.
Host	Selecting Host will push the rule down to all VM/vNIC on the ESXi host.
Legacy port group	Selecting Legacy port group will push the rule down to all VM/vNIC on the VSS port-group.
Logical Switch	Selecting Logical Switch will push the rule down to all VM/vNIC connected on this Logical Switch (or VXLAN) segment.
Group	Selecting Security Group will push the rule down to all VM/vNIC defined within the Security Group.
Virtual Machine	Selecting Virtual Machine will push the rule down to all vNIC of this VM.
vNIC	Selecting vNIC will push the rule down to this particular vNIC instance.

Capability to define Security Group in Applied To field.

Note: It is also possible to publish the policy rule to all Edge Service Gateways (ESG) or specific ESG using the appropriate option.

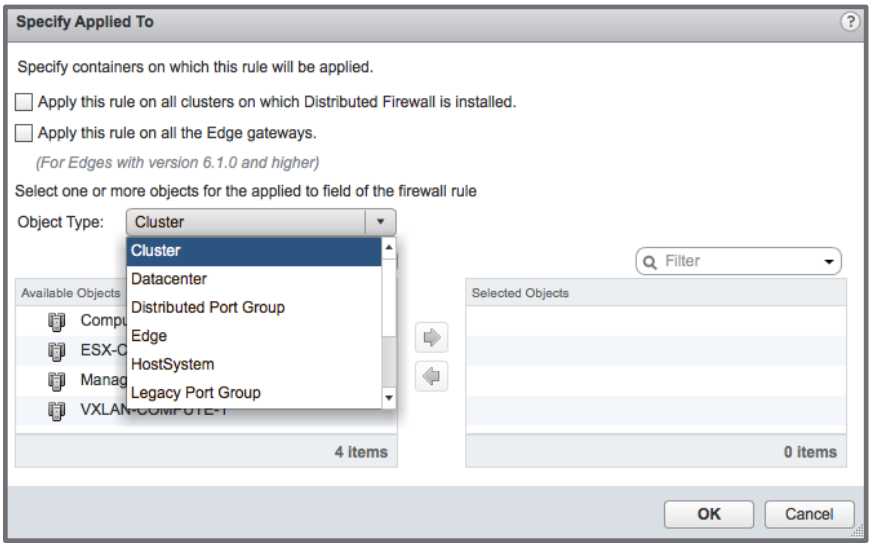


Figure 24 - Applied To scope for the Policy Rule

Let's take different policy rule constructs and let's see how DFW behave for each case.

We will use the same and unique logical network topology for this purpose:

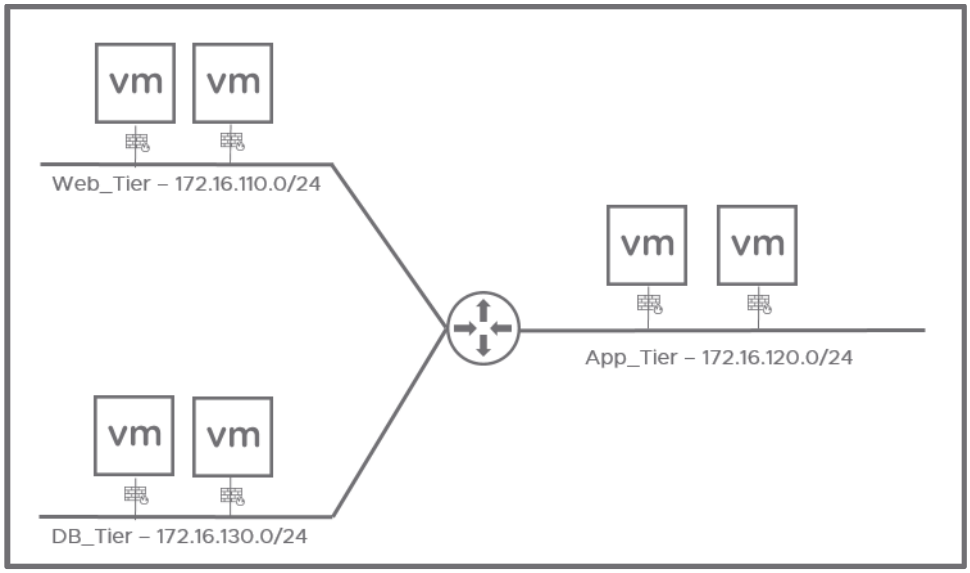


Figure 25 - 3-Tier Application base network topology

This is a standard 3-tier topology with Web/App/DB segmentation. 3 web servers are connected to Web Logical Switch (VXLAN), 2 applications servers are connected to App LS and 2 DB servers connected to DB LS. A Distributed Logical Router is used to interconnect the 3 tiers together by providing inter-

tier routing. DFW has been enabled on the ESXi cluster and as a result, each VM has a dedicated instance of DFW attached to its vNIC.

NSX offers multiple ways to define DFW policy rule configuration. Let's see how some of them look like.

Example 1: Using static IP addresses/subnets in Security Policy rule

Firewall rule policy configuration may look like this:

Table 14 - Firewall Rule Table - Example 1 Layout

Name	Source	Destination	Service	Action	Applied To
Web to App	172.16.10.0/24	172.16.20.0/24	<Enterprise Service Bus>	Allow	All clusters
App to DB	172.16.20.0/24	172.16.30.0/24	SQL	Allow	All clusters
Default	Any	Any	Any	Block	All clusters

When using static IP addresses or subnets in policy rule, there is no need to install VMtools on guest VM. DFW engine is able to enforce network traffic access control based on the provided information.

To use this type of construct, user needs to know the exact IP information and then relay it to policy rule. This construct is quite static and does not fully leverage dynamic capabilities with modern cloud systems.

Example 2: Using Logical Switch object in Security Policy rule

A better way to configure Security Policy rule is by using dynamic objects provided by vCenter/NSX manager (commonly called vCenter objects or containers).

Table 15 - Firewall Rule Table – Example 2 Layout

Name	Source	Destination	Service	Action	Applied To
Web to App	Web LS	App LS	<Enterprise Service Bus>	Allow	All clusters
App to DB	App LS	DB LS	SQL	Allow	All clusters
Default	Any	Any	Any	Block	All clusters

This type of construct necessitates guest VM to have VMTools running.

Reading policy rule table is easier for all teams in the organization ranging from security auditors to architects and operations.

Any new VM connected on any Logical Switch will be automatically enforced with the corresponding security posture (for instance, a new installed Web server will be seamlessly protected by the first policy rule with no human

intervention). On the same way, a VM disconnected from a Logical Switch will have no more Security Policy applied to it.

This type of construct fully leverages the dynamic nature of NSX.

NSX gives capability to check VM connected to a Logical Switch at any point of time. If no more VM are connected to a particular Logical Switch, it then becomes very easy to remove Security Policy rule dealing with this particular Logical Switch.

Example 3: Using Security Group object in Security Policy rule

Before writing DFW policy rule, let's first create the appropriate Security Groups (SG).

Table 16 - Firewall Rule Table - Security Group Example 3

SG name	SG definition
SG-WEB	Static inclusion: Web LS
SG-APP	Static inclusion: App LS
SG-DB	Static inclusion: DB LS

Table 17 - Firewall Rule Table - Example 3 Layout

Name	Source	Destination	Service	Action	Applied To
Web to App	SG-WEB	SG-APP	<Enterprise Service Bus>	Allow	All clusters
App to DB	SG-APP	SG-DB	SQL	Allow	All clusters
Default	Any	Any	Any	Block	All clusters

All statements given for example 2 still prevail here. In fact, using Security Groups provides much more flexibility than anything else.

Using properly dynamic inclusion, static inclusion and static exclusion, user can define in a very granular way what objects to include in this container.

Writing DFW policy rules using Security Groups reduces dramatically number of rules needed in the enterprise and gives the most comprehensible Security Policy configuration.

Using REST API

NSX platform provides a REST API framework for both NSX DFW rule table and Service composer. The REST API is exposed northbound via NSX manager. The REST API is another option to create or consume NSX DFW rules and policies, and automate NSX security workflows. The NSX DFW

management API framework can also be used to build custom third-party application to automate NSX security workflows.

The following NSX DFW Rule Table REST API example shows how to configure a policy in the DFW rule table. This example shows the API path and XML input to create first policy rule, “Web to App”, as defined in the Table 17 into DFW rule table.

More information on NSX DFW API’s are available in the NSX release specific API guide.

Figure 26 DFW RESTful API Example

Automate DFW Rule Creation

Starting with NSX 6.3, the Application Rule Manager (ARM) was implemented into the NSX product line. ARM was built specifically to help administrators with day 2 NSX Micro-segmentation operations. With NSX 6.4, ARM was improved upon by adding in automated recommendations based on the flows it picked up during a monitoring session.

Application Rule Manager

Load

ARM leverages monitoring of the virtual machine vNICs to gather information about how each system is communicating in the session created. Before starting, it helps to understand the virtual machines that make up the application. External dependencies will also show up in the flow table for any systems that talk into the application or the application talks out to even if those systems are NOT included in the Monitor Session.

The length of time that is recommended for monitoring an application is dependent on the application itself. If the application has processes that occur over periods of time or infrequently, monitoring the application for any flows that would occur during those time periods is highly recommended. Here is a sample logical topology of the application that the admin would like to create a Micro-segmentation policy around.

With NSX 6.4, Application Rule Manager introduces a recommendation engine that can quickly provide recommended DFW rules for the flows discovered during the monitoring session. These recommendations can be modified before publishing them to the DFW. With the addition of Layer 7 context awareness, ARM can also chose Layer 7 services picked up during monitoring and recommend those services as well as typical Layer 4 recommendations.

*Note – For ARM to pickup Layer 7 flows during the monitoring session, at least ONE rule in the DFW with a Layer 7 Service must be present before running the monitoring session.

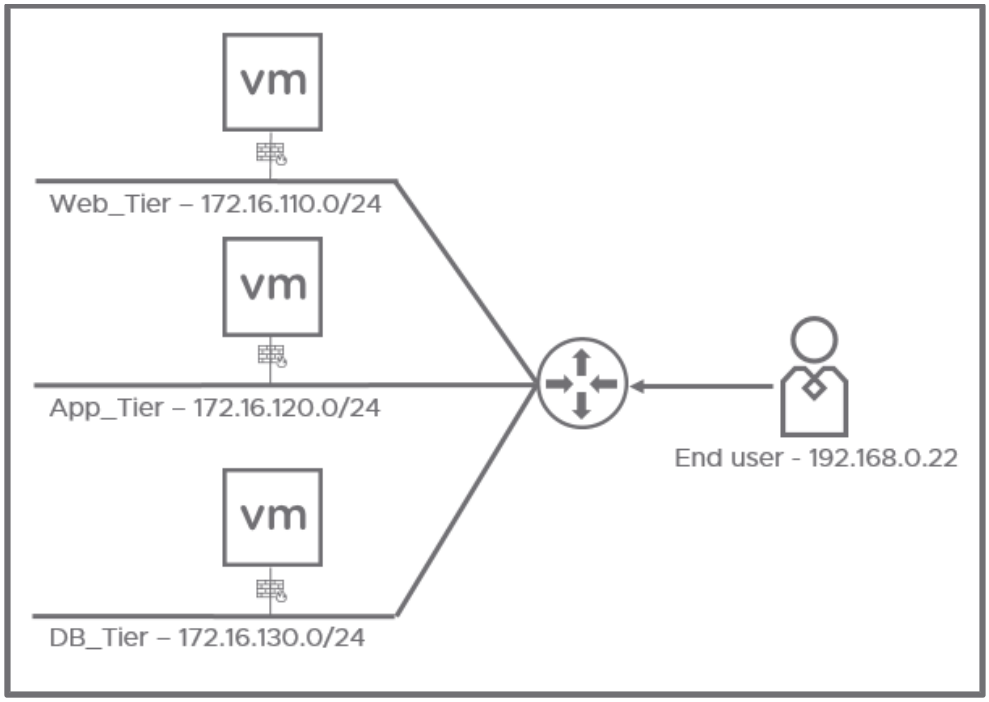


Figure 27 Sample Application Topology

To get started, an admin will log into the vSphere Web Client and navigate to Networking and Security -> Flow Monitoring -> Application Rule Manager. This will bring up the ARM interface.

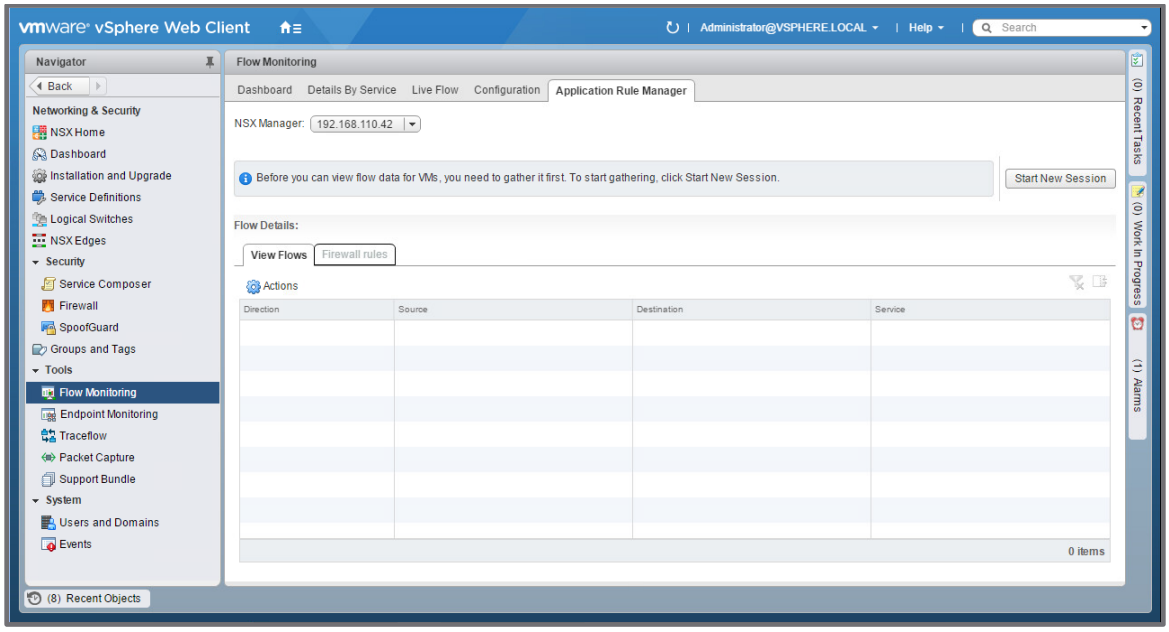


Figure 28 Application Rule Manager interface

From this interface, the admin would 'Start New Session' and select the virtual machines that make up the application that is to be monitored.

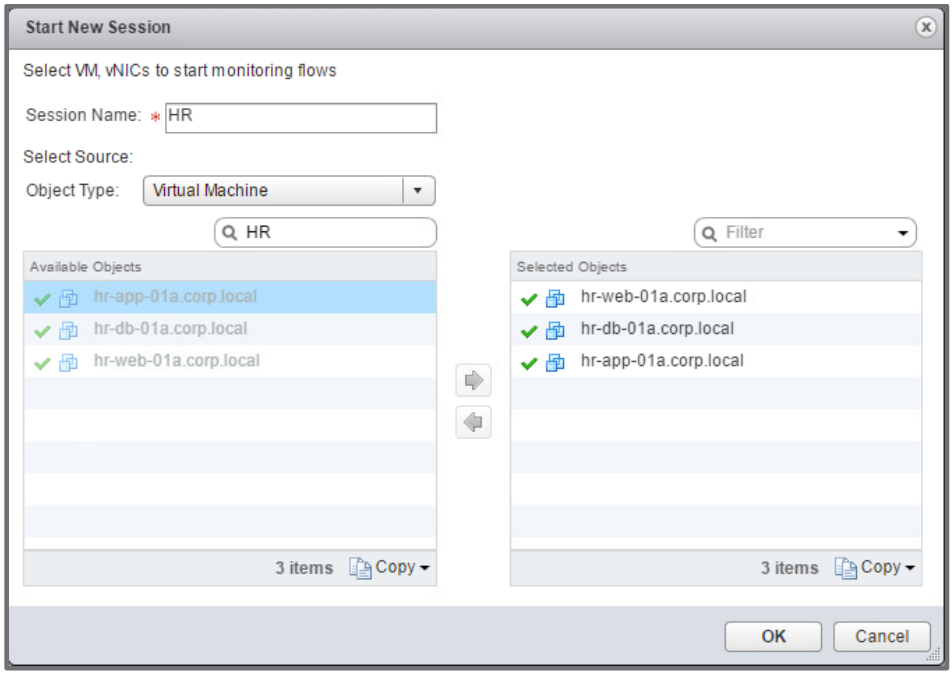


Figure 29 ARM Session Monitor VM Selection

Profile

Once the virtual machines are identified and selected, the monitoring process begins and flows for the application are gathered as the application functions normally. This information is pulled from the vNIC of each virtual machine in the monitored session. This flow information is pulled into the Flow Details table in ARM so the admin can easily view and sort the information.

ARM is capable of monitoring up to 30 virtual machines in each session, with up to 5 sessions simultaneously. ARM has an upper limit of 100,000 flows that can be collected over a 7 day period. If the monitor session sees more than 100,000 flows over the period of time, it will begin dropping older flows from the collection.

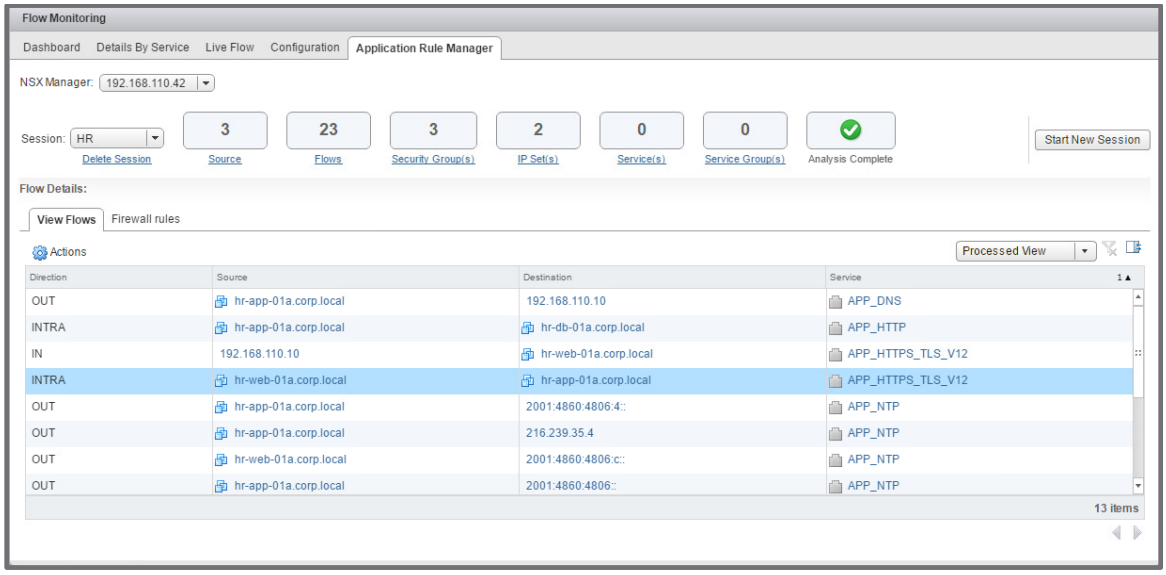


Figure 30 ARM Completed and Analyzed Monitor Session

With the monitor session now complete, the admin will stop the session and prepare to analyze the flows collected.

Analyze

Once the flow information collected, ARM can now analyze the flows and create the Security Groups, IP Sets, and most importantly, the DFW rules from the available information. During the monitoring, ARM may see 100 flows but will deduplicate down to only unique flows for simplier correlation when analyzed. The initial collected flow information is simply IP address-based. ARM will correlate this information with vCenter to determine the virtual machines, if possible, that those IP addresses belong to in the data center.

With the monitor session now analyzed and ARM having correlated the flow information from IP addresses to virtual machine names, ARM will generate recommended rules in the Firewall rules tab.

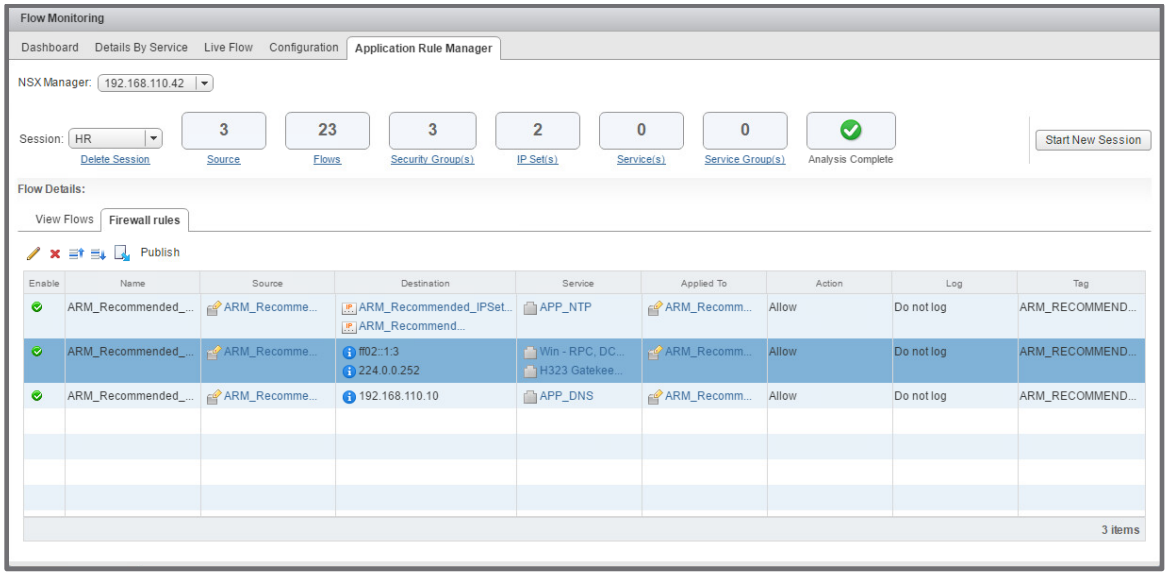


Figure 31 - ARM Recommended Firewall rules

Now that ARM has automatically built the DFW rules, they can be automatically published to the DFW as they are or they can be modified as necessary.

To publish, the admin simply selects the 'Publish' link, inputs a new Section Name and position within the DFW and clicks on 'OK'.

Figure 32 - Firewall Rule Publish

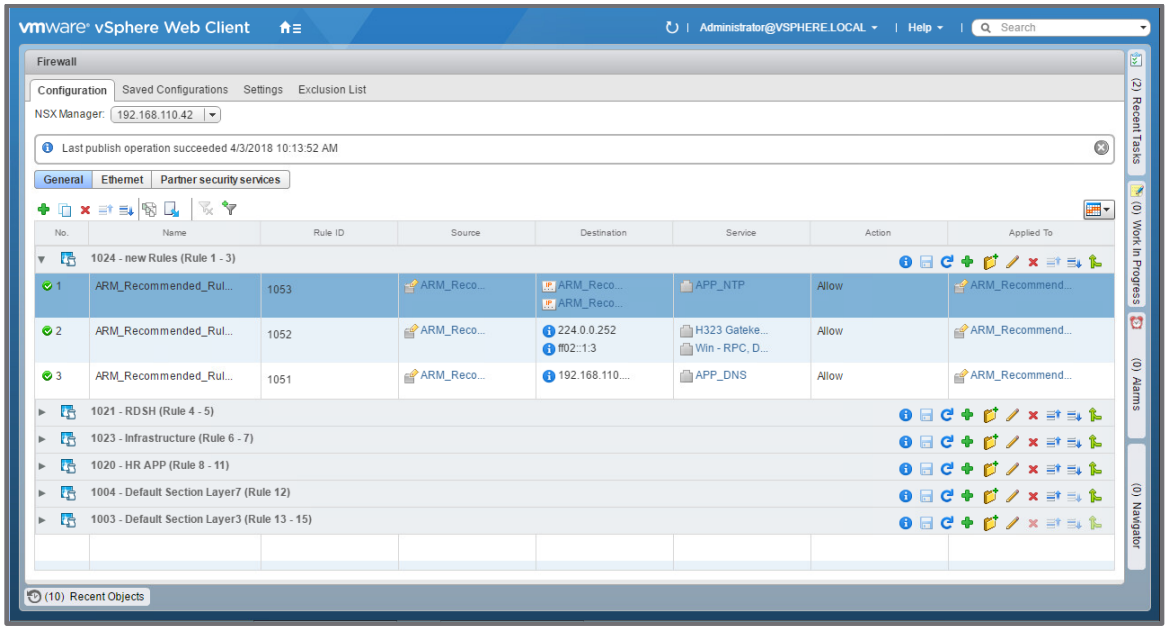


Figure 33 - ARM Recommended Rules Published

Application Rule Manager Details and manual processes

ARM has several details about the flows that it collects and although ARM provides automated recommendations, these recommendations can be modified by the admin to fit their needs before publishing the rules to the DFW. The following section covers the details and the process for modifying any of the recommendations as necessary.

Flow Details

In the View Flows tab, all of the deduplicated flows will show up. Each flow is broken down into the individual details about the flow. These details are used to create a DFW firewall rule based on the information from the observed flow.

- Direction
    - **IN** – This type of flow represents traffic inbound to one of the VMs being monitored. This typically means the Destination VM.
    - **OUT** – This type of flow represents traffic outbound from one of the monitored VMs, typically the Source VM.
    - **INTRA** – This flow type represents traffic going between machines in the monitor session.
- Source/Destination

- **Replace with Any** – Replaces the current entry with the 'Any' entry for any source/destination.
- **Replace with Membership** – Replaces the current entry with an existing NSX Security Group that the virtual machine may already be a part of.
- **Create Security Group and Replace** – Replaces the current entry with a new NSX Security Group that the admin creates and adds the virtual machine to.
- **Add to existing Security Group and Replace** – Replaces the virtual machine by adding the existing entry to a new existing NSX Security Group.
- **Create IPSet and Replace** – Replaces the current IP address entry that can't be resolved to a new IPSet that the admin creates and adds the IP address to.
- **Add to existing IPSet and Replace** – Replaces the current IP address entry that can't be resolved by adding the existing entry to an existing IPSet.

- Service
  - **Resolve Services** – Replaces the current entry with a list of services that meet the criteria for the admin to select from.
  - **Create Services and Replace** – Replaces the current entry with a new service that the admin creates.
  - **Create Services Group and Replace** – Replaces the current entry with a new service group that the admin creates with one or more services nested inside.
  - **Replace Service with any** – Replaces the current entry with the 'Any' entry for any service.
  - **Replace Service with Service Group** – Replaces a service with a service group.
  - **Revery Protocol and Port** – Reverts any changes to the initial entry in the flow details.
- Context*
  - Reserved for future use
- RuleID*
  - Represents the DFW RuleID that the observed flow is being enforced by. Can be clicked on to show the details of the RuleID from the DFW.
- # of Flows*
  - Number of times the flows were seen during the monitor session that were deduplicated down to one flow entry

Note: *Hidden Columns by Default

Once the flows are analyzed and ARM has made it's recommendations, the admin can check those recommendations and make modifications to them as

needed. Each of the groupings at the top of the tables, has each of the DFW objects that was created by ARM during the analysis phase. From these groupings, we can see and modify any of the automated recommendations.

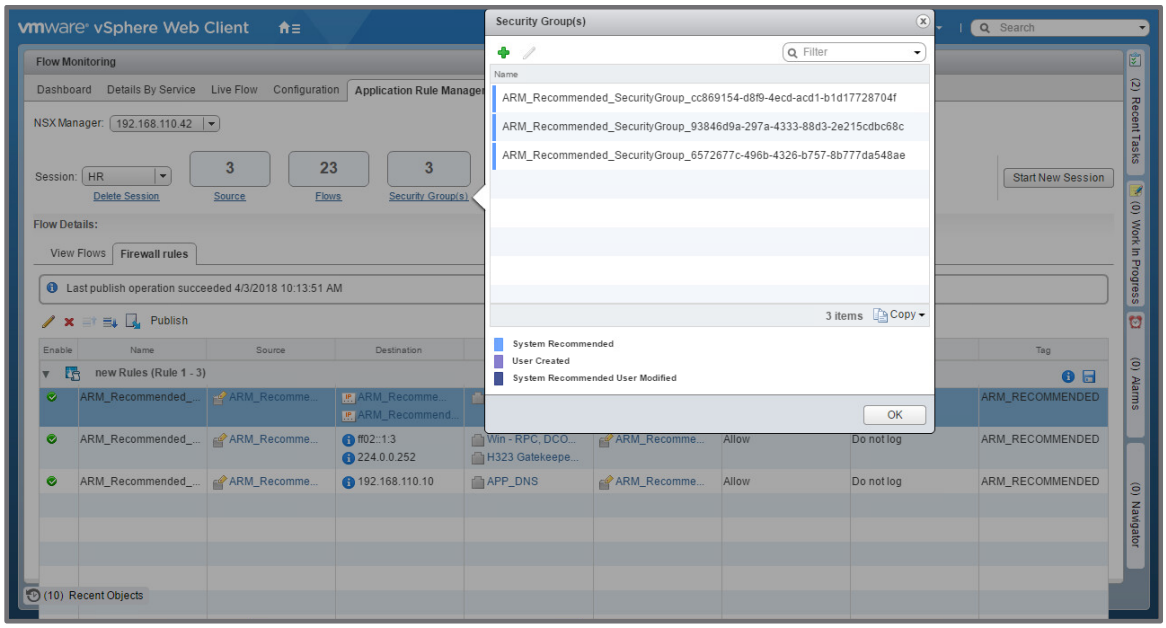


Figure 34 Modifying a Recommended NSX Security Group

If any changes are made, the color legend at the bottom will corresepond to how the change(s) was(were) made. This process can be repeated for any of the other groupings.

Modifications to each rule can be made as well.

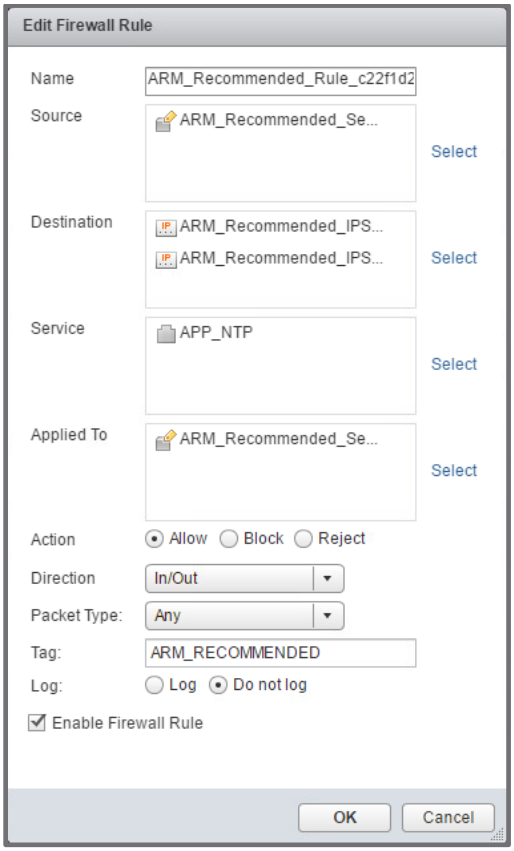


Figure 35 Modifying a Recommended DFW Rule

Verify

ARM is not only for building rulesets in the DFW. It can also be leveraged to ensure that the rulesets built are verified and working as intended. ARM can detect not only allowed flows, but flows that may be dropped due to blocking rules put into the DFW as well. Going back to the new rules for the Application now placed into the DFW Firewall Rule Table, the admin can see the new RuleIDs that were created for each new rule.

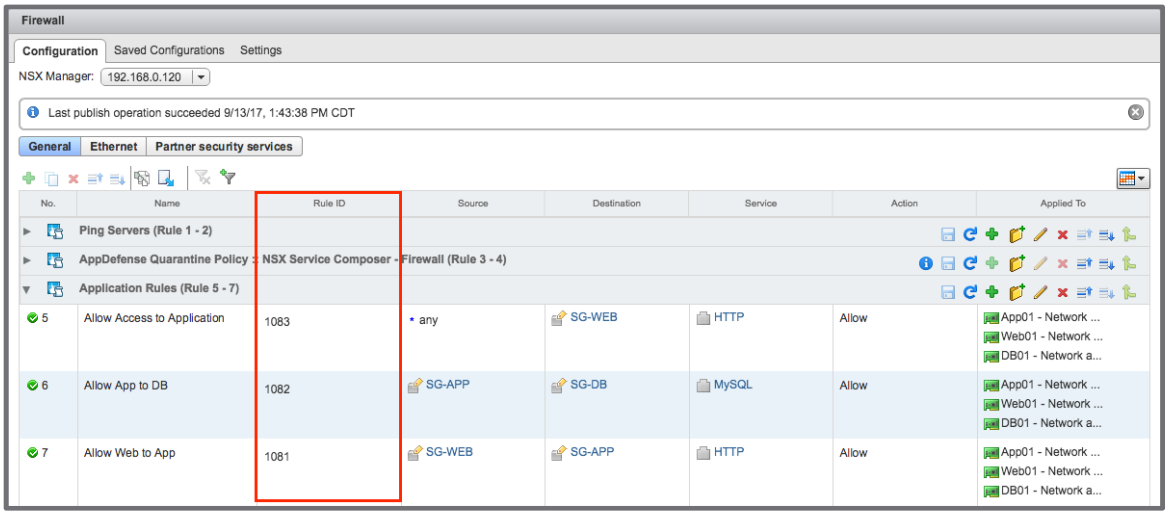


Figure 36 New RuleIDs for New Rules

Just as the admin had created a monitor session for the application previously, the admin can do the same to verify that the new rules created are being observed by the application. The process is the same as setting up a new monitoring session for a new application, the admin just uses the same servers that make up the application as previously defined.

With a new monitor session run against the application and analyzed, the admin can see that the RuleID column numbers have changed to reflect the new RuleIDs from the new rules built in the Build section.

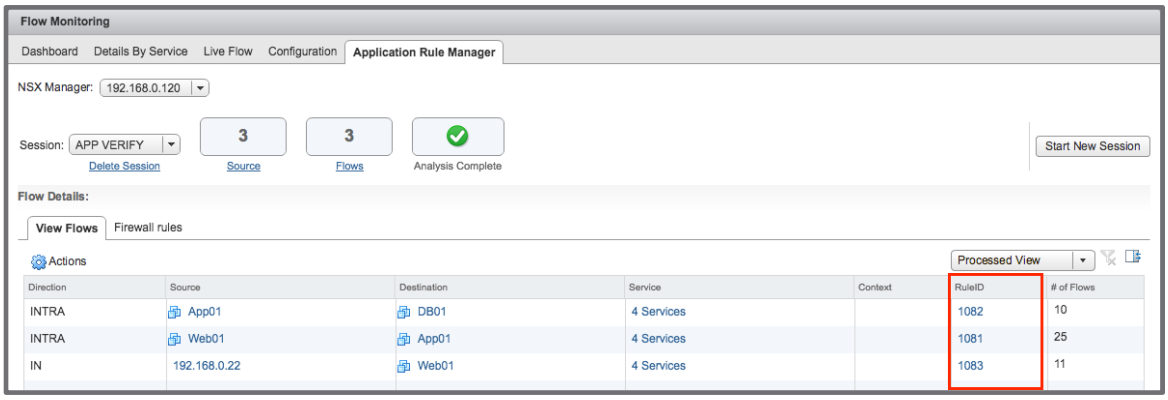


Figure 37 Verify Monitor Session with new RuleID entries

By clicking on one of the links of the RuleIDs listed, the admin can see the DFW rule details.

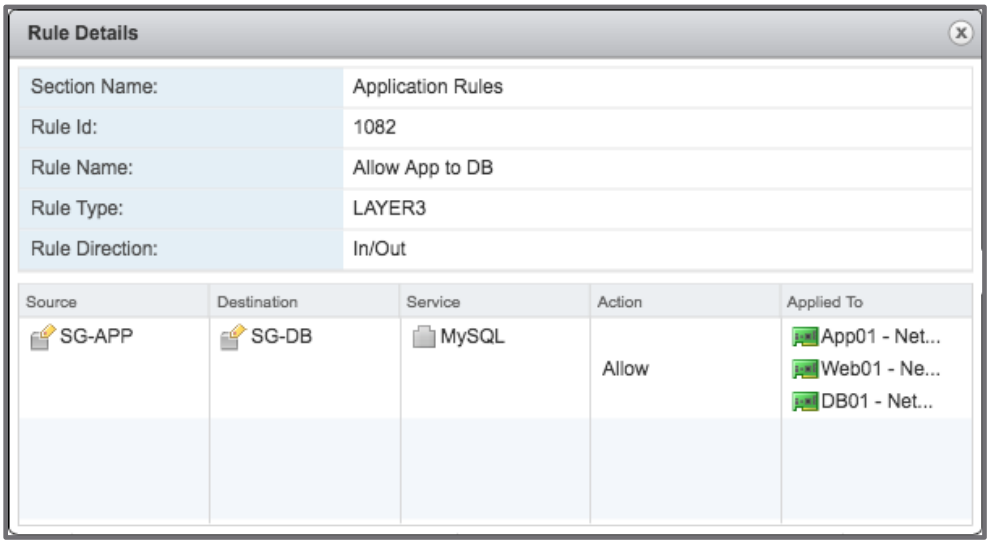


Figure 38 RuleID DFW Details

Help and More Information

NSX-v Documentation

In addition to this document, you can read the following documents for help setting up NSX-v. All are available from https://www.vmware.com/support/pubs/nsx_pubs.html:

- NSX for vSphere Installation and Upgrade Guide
- NSX for vSphere Administration Guide
- NSX for vSphere API Reference Guide
- NSX for vSphere Command Line Interface Reference

NSX Micro-segmentation Guides

The following guides can provide further insights into the process of Micro-segmentation and examples of how and where to start:

- Micro-segmentation Day 1 Guide
- https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/nsx/vmware-nsx-microsegmentation.pdf
- Micro-segmentation Day 2 Guide
- https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/nsx/vmware-micro-segmentation-day-2.pdf

Contacting the NSX Technical Services Team

You can reach the NSX technical services team at http://www.vmware.com/support.html.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com