Table of Contents

[VMware NSX Security Framework 2](#_Toc196936653)

[Summary 2](#_Toc196936654)

[Requirements 2](#_Toc196936655)

[Environment Security Matrix 3](#_Toc196936656)

[List of allowed flows. 3](#_Toc196936657)

[Building yaml file - nsx-sf-inventory.yaml 4](#_Toc196936658)

[Tagging Strategy 4](#_Toc196936659)

[Building yaml file - nsx-sf-authorized-flows.yaml 7](#_Toc196936660)

[Tenant Definition: wld09 7](#_Toc196936661)

[Environment Policy 7](#_Toc196936662)

[Application Policy 7](#_Toc196936663)

# VMware NSX Security Framework

## Summary

In this document, we describe the VMware NSX Security Framework and how to implement it using automated Terraform workflows. This framework provides a standardized, scalable, and automated approach to configuring security policies within an NSX environment. This guide is intended for teams responsible for designing, deploying, and maintaining network security within VMware-based infrastructures.

Before we dive deeper, it's important to set expectations: this NSX Security Framework still requires some manual effort before reaching a true Zero Trust state. While tools like VMware Aria Operations for Networks and VMware Aria Operations for Logs will provide visibility into network flows, the goal is not to simply permit every detected flow.  
Zero Trust means **only explicitly authorized traffic is allowed**, based on strict identity verification and least-privilege access principles.

## Requirements

To implement the NSX Security Framework using automated Terraform workflows, the following prerequisites must be in place:

1. Environment Security Matrix.
2. List of Allowed Flows.
3. The FQDN of the NSX Manager.
4. NSX service account with valid permissions.
5. NSX Security Framework Terraform files [link need to be added.]

## 

## Environment Security Matrix

We will adopt the DTAP model for the Environmental Security Matrix.  
In this example, the following Environmental Security Matrix is defined.  
The table below specifies which communication flows are permitted or blocked between environments:

1. Communication between **Development** and **Test** is allowed.
2. Communication between **Test** and **Acceptance** is allowed.
3. Communication between **Acceptance** and **Production** is allowed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Development | Test | Acceptance | Production |
| Development |  |  |  |  |
| Test |  |  |  |  |
| Acceptance |  |  |  |  |
| Production |  |  |  |  |

## List of allowed flows.

How you retrieve a list of flows is up to you. You can use tools like VMware Aria Operations for Logs, VMware Aria Operations for Networks, or refer to the firewall rules and port requirements provided by the software vendors.

Once you have compiled the list, you can proceed to configure the **nsx-sf-authorized-flows.yaml**.

## Building yaml file - nsx-sf-inventory.yaml

To support the consistent and scalable enforcement of micro-segmentation policies within the NSX security framework for example tenant wld09, a structured tagging and grouping strategy is implemented based on the provided YAML structure. This approach ensures that security policies can dynamically adapt to the VM lifecycle and that workloads are logically grouped according to tenant, environment, application, and sub-application.

### Tagging Strategy

The tagging process will be executed in the following order:

#### Tenant Tag and Groups

* A **tenant-level tag** named ten-wld09 will be created based on the tenant key (wld09).
* A corresponding **Tenant Group** named ten-wld09 will be created with **member criteria** based on the tenant tag ten-wld09.
* This tag will be assigned to **all VMs** belonging to the wld09 tenant.
* The tenant group can be used as a top-level grouping in firewall policies to apply controls across the entire tenant environment.

#### Environment Tags and Groups

* For each environment listed under the internal key (e.g., env-wld09-prod, env-wld09-test), an **environment tag** (e.g., env-wld09-prod) will be created.
* Corresponding **Environment Groups** (e.g., env-wld09-prod) will be created with **member criteria** based on the assigned environment tag.
* Every VM defined under a specific environment will be automatically assigned the appropriate environment tag.

#### Application Tags and Groups

* For each application defined under an environment (e.g., app-wld09-prod-3holapp, app-wld09-prod-database), an **application type tag** (e.g., app-wld09-prod-3holapp) will be created.
* **Application Groups** (e.g., app-wld09-prod-3holapp) will be created, with membership determined by the corresponding application type tag.
* All VMs listed under an application will be tagged accordingly.

#### Sub-Application Tags and Groups

* Within application where **sub-application** exist (e.g., app-wld09-prod-3holapp-database, app-wld09-prod-3holapp-application, app-wld09-prod-3holapp-web), additional tags will be created (e.g., app-wld09-prod-3holapp-database).
* **Sub-Application Groups** will also be created, using the specific sub-application tags as membership criteria.
* VMs will be assigned the sub-application tags according to their roles (database, application, web).

#### External Services Groups

* The external key defines **external services** (e.g., DNS, NTP, jumphosts) which are not hosted within the NSX overlay but are critical for the tenant's operation.
* Groups for external services (e.g., ext-wld09-dns) will be created with **membership based on static IP addresses** as listed in the YAML file.
* These groups will be referenced in Distributed Firewall (DFW) policies to allow or control access to or from those external resources.

#### Example Tagging Overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity | Example Tag | Example Group | Group Membership Criteria | YAML values |
| Tenant | ten-wld09 | ten-wld09 | Virtual Machine - Equals - Tag | Tenant Key value will be used to create the tenant tag and group names. All VMs within the tenant will be tagged with this tag. |
| Environment | env-wld09-prod | env-wld09-prod | Virtual Machine - Equals - Tag | Environment Key value will be used to create the environment tag and group names. All VMs within the environment will be tagged with this tag. |
| Application | app-wld09-prod-3holapp | app-wld09-prod-3holapp | Virtual Machine - Equals - Tag | Application Key value will be used to create the application tag and group names. All VMs within the tenant will be tagged with this tag. |
| Sub Application | app-wld09-prod-3holapp-database | app-wld09-prod-3holapp-database | Virtual Machine - Equals - Tag | Sub Application Key value will be used to create the sub application tag and group names. All VMs within the tenant will be tagged with this tag. |
| External Services (DNS) |  | ext-wld09-dns | IP Addresses | External Services Key value will be used to create the external services tag and group names. All IP addresses will be used as member criteria |

#### Implementation Notes

* Tags must be assigned **automatically** based on the YAML file contents during the provisioning phase using automation tool Terraform
* All groups must be **dynamic** where possible (based on tags) to minimize operational overhead.
* Static groups based on IPs should be carefully maintained to reflect external infrastructure changes.
* Naming conventions must be strictly adhered to (env-, app-, ext- prefixes) to ensure traceability and prevent conflicts.
* The tenant-level, environment, application, and sub-application tags should be inherited hierarchically where possible to simplify DFW rule design.

#### Example - nsx-sf-inventory.yaml

---

# Format: Tenant > Internal/External > Environment > Application > Sub Application > Resources

wld09: # Tenant Key

internal:

env-wld09-prod: # Environment Key

app-wld09-prod-3holapp: # Application Key

app-wld09-prod-3holapp-database: # Sub Application Key

- p-db-01a # VM name

app-wld09-prod-3holapp-application: # Sub Application Key

- p-app-01a

app-wld09-prod-3holapp-web: # Sub Application Key

- p-web-01a

- p-web-02a

- p-web-03a

app-wld09-prod-database: # Application Key

- p-db-01a # VM name

app-wld09-prod-application: # Application Key

- p-app-01a

app-wld09-prod-web: # Application Key

- p-web-01a

- p-web-02a

- p-web-03a

env-wldl09-test: # Environment Key

app-wld09-test-database: # Application Key

- t-db-01a # VM name

app-wld09-test-application: # Application Key

- t-app-01a

app-wld09-test-web: # Application Key

- t-web-01a

- t-web-02a

- t-web-03a

external:

ext-wld09-dns: # External Services Key

- 192.168.12.10

ext-wld09-ntp: # External Services Key

- 192.168.12.1

ext-wld09-jumphosts: # External Services Key

- 10.10.89.11

## Building yaml file - nsx-sf-authorized-flows.yaml

### Tenant Definition: wld09

The wld09 tenant defines both **Environment Policies** and **Application Policies** to enforce structured communication controls across different logical zones and applications within the environment. These controls are crucial to ensure **micro-segmentation**, and **environment separation**, core principles of a secure NSX implementation.

### Environment Policy

The **Environment Policy** governs **high-level communication rules** between different *environments* (e.g., env-wld09-prod, env-wld09-test). It enforces an overall trust boundary that must be respected by any lower-level, application-specific policies.

* **Allowed Communications**:
  + **env-wld09-prod ➔ env-wld09-test**:
    - Traffic initiated from the *Production* environment **to** the *Test* environment is **explicitly allowed**.
* **Blocked Communications**:
  + **env-wld09-test ➔ env-wld09-prod**:
    - Traffic initiated from the *Test* environment **to** the *Production* environment is **explicitly blocked**.

#### Implementation Notes

* The application policy will only be applied to the tenant tag (e.g., ten-wld09)
* When a rule has multiple source or destination values, a single rule will be created using multiple groups for the source and/or destination fields, rather than creating multiple individual rules within the policy.

### Application Policy

The **Application Policy** defines **fine-grained communication rules** between specific application tiers or services within the tenant. Each rule specifies allowed flows by **source**, **destination**, **ports**, and **protocols**, enforcing strict service-based segmentation.

#### Implementation Notes

* The application policy will only be applied to the tenant tag (e.g., ten-wld09)
* When a rule has multiple source or destination values, a single rule will be created using multiple groups for the source and/or destination values, rather than creating multiple individual rules within the policy.

#### Example - nsx-sf-authorized-flows.yaml

---

wld09:

environment\_policy:

allowed\_communications:

env-wld09-prod:

- env-wld09-test

blocked\_communications:

env-wld09-test:

- env-wld09-prod

application\_policy:

- source: ext-wld09-jumphosts

destination:

- app-wld09-prod-3holapp-web

- app-wld09-prod-web

- app-wld09-test-web

ports:

- 443

protocol: tcp

- source:

- app-wld09-prod-3holapp-web

- app-wld09-prod-web

- app-wld09-test-web

destination:

- app-wld09-prod-3holapp-application

- app-wld09-prod-application

- app-wld09-test-application

ports:

- 8443

protocol: tcp

- source:

- app-wld09-prod-3holapp-application

- app-wld09-prod-application

- app-wld09-test-application

destination:

- app-wld09-prod-3holapp-database

- app-wld09-prod-database

- app-wld09-test-database

ports:

- 3306

protocol: tcp

- source: ten-wld09

destination: ext-wld09-dns

ports:

- 53

protocol: udp