TUTORIAL

# How to Configure VMware vSphere Distributed Switch Security Policies

Configure vSphere Distributed Switch security policies with a practical workflow that covers prerequisites, per-port-group settings, validation checks, and safe rollout guidance for production environments.

**Virtualization - August 13, 2026**

## Why these security policies matter

Virtual switch security settings decide whether a virtual machine can send traffic with forged source addresses, spoof MAC addresses, or operate as a promiscuous listener. In a shared virtualization environment, those controls are not optional housekeeping; they are a direct part of tenant isolation, incident containment, and change control.

This tutorial shows how to configure vSphere Distributed Switch security policies on distributed port groups, how to verify that the settings are actually applied, and what to check before you put the configuration into production. By the end, you should be able to decide whether the policy should be applied at the port-group level or inherited from the switch, implement the change safely, and validate the result from both the management plane and the workload side.

## What you are building

The finished state is a distributed port group with explicit security behavior for one or more workloads. In practice, that usually means one of three policy profiles:

- Reject forged transmits, MAC changes, and promiscuous mode for standard VM networks.
- Allow one or more settings only when a workload has a documented need, such as packet capture or a nested virtualization lab.
- Override the switch default at the port-group level so the behavior is unambiguous and easy to audit.

If the environment uses security baselines or cluster hardening standards, align the policy with those standards rather than making ad hoc exceptions. A good companion reference is the VMware vSphere Hardening Guide for Secure Virtualization, especially when you want to compare switch-level controls with broader host and cluster hardening requirements.

## Prerequisites and stop-here checks

Before you change anything, confirm that the distributed switch is already in place and that you have permission to edit networking objects.

## Required access and scope

You need administrative rights that allow you to edit the distributed switch and distributed port group. If role-based controls are used in your environment, verify that your permissions are sufficient before you start. If not, stop here and get the right access rather than testing in a production console session.

## Version and object behavior

Distributed switch behavior can vary with version, licensing, and how the environment is configured. Before making policy decisions, confirm:

- Which distributed switch and port group you are targeting.
- Whether the port group inherits policies from the switch or overrides them.
- Whether the workload depends on promiscuous mode, forged transmits, or MAC changes.

## Stop-here-if warning

Stop here if the port group carries production workloads and you have not identified the traffic characteristics of those workloads. For example, packet capture appliances, nested hypervisors, clustering software, and some security tools may require settings that are unsafe for ordinary VM networks. Enabling those features broadly can weaken isolation and make lateral movement easier.

For environments where permissions need to be tightly scoped, it can help to review How to Secure VMware Virtual Machines with Role-Based Access Control so the networking change does not outpace access governance.

## Decide which policy level should own the setting

## Goal

Choose whether the security policy should be inherited from the distributed switch or overridden at the port group.

## Action

Use the distributed switch as the baseline when you want consistent defaults across many port groups. Use a port-group override when a specific workload needs an exception.

Typical decision rules:

- Switch-level default: use for broad security posture on new port groups.
- Port-group override: use for one-off exceptions or workload-specific requirements.
- Do not mix undocumented exceptions: every override should have a business or technical justification.

## Expected output

You should know exactly which object will own the policy and why.

## Validation

Check the current inheritance state in the networking view before editing. If the port group already overrides a setting, note the current value so you can preserve or deliberately change it.

## Common failure

A common mistake is changing the switch default and assuming all existing port groups immediately behave the same way. If a port group overrides policy, the switch-level change may not apply there.

## Configure the distributed switch security policy

## Goal

Set the baseline security behavior on the distributed switch or on the port group, depending on your design.

## Action

In the networking configuration for the distributed switch, open the security policy for the relevant scope and set the three common controls:

- Promiscuous mode
- MAC address changes
- Forged transmits

For standard VM traffic, the common secure baseline is to reject all three unless a workload has a documented exception.

If the environment is managed through automation, express the same policy in code or configuration management rather than relying only on manual console changes. The exact API or command path depends on your toolchain and version, so verify the supported method for your environment before automating at scale.

## Expected output

A distributed switch or port group policy that clearly states whether each setting is accepted or rejected.

## Validation

Re-open the security settings and confirm that the displayed values match what you intended. If the UI shows inherited settings, confirm the parent object is configured correctly too.

## Common failure

The most common failure is changing the wrong level. Another is assuming the UI saved the change when a later refresh shows the object still inherits the old setting.

## Apply a port-group exception safely

## Goal

Allow a non-default setting only for workloads that genuinely need it.

## Action

If a VM requires a special setting, create or edit a dedicated distributed port group rather than weakening the shared standard network. Then override only the specific policy that is needed.

Good examples of justified exceptions include:

- Packet capture or IDS appliances that require promiscuous mode.
- Nested lab environments that need specific MAC or transmit behavior.
- Legacy appliances that are documented to fail unless a particular setting is enabled.

Keep the exception narrow. Do not enable all three settings unless the workload actually requires them.

## Expected output

A dedicated network segment or port group with an explicit, limited exception and a documented owner.

## Validation

Confirm that only the intended VMs are attached to the exception port group. Verify that no unrelated workloads were migrated there during testing or troubleshooting.

## Common failure

An exception often becomes a convenience network over time. That creates drift, weakens segmentation, and makes later audits harder.

## Validate the policy from the management plane

### Goal

Prove that the configuration is present where you expect it to be.

### Action

Review the distributed switch and the port group after saving the change. Look for three things:

- The policy values are set as intended.
- The inheritance state is correct.
- The affected port group is mapped to the correct workloads.

If you use administrative automation, export or query the object state and compare it against the desired configuration. Keep the validation artifact with the change record.

### Expected output

A verified configuration record showing the final policy values and ownership.

### Validation

A practical validation method is to compare the current state against a documented baseline. If you have a configuration export, verify that the policy flags in the export match the intended profile.

## Common failure

The most frequent issue during validation is checking the switch object while the workload is actually attached to a port group with overridden settings.

## Validate the policy from the workload side

## Goal

Confirm that the configuration affects traffic the way you expect.

## Action

Use a test VM on the target port group and perform a non-destructive connectivity check. For special cases, validate only the specific behavior you changed. For example:

- If promiscuous mode should be blocked, ensure a packet capture tool on an ordinary VM does not see unrelated traffic.
- If forged transmits should be rejected, confirm that a workload attempting to use an unexpected source MAC does not function.
- If MAC changes should be rejected, confirm the guest does not behave as though it can freely adopt a new MAC without authorization.

Keep the test scoped and reversible. Do not use production traffic patterns as your first validation step.

## Expected output

Evidence that the policy is enforcing the desired network behavior for the attached workload.

## Validation

You should be able to explain the result in operational terms: what was blocked, what was allowed, and whether that matches the intended design.

## Common failure

A policy can be correct in the management plane but ineffective for the intended control if the workload is connected to the wrong port group, is using a different network path, or is being validated with the wrong test method.

## Roll out with change control

## Goal

Apply the policy without disrupting unrelated workloads.

## Action

Use a small, controlled rollout:

- Start with a non-production or low-risk port group.

- Validate the policy and workload behavior.
- Move to the production port group only after verification.
- Record the exception rationale if you are allowing any insecure setting.

If the change is tied to a broader hardening effort, document it with the rest of the cluster baseline so future audits can correlate network policy with host and access controls.

## Expected output

A documented configuration change with clear scope, owner, and rollback path.

## Validation

Confirm that only the intended port groups were modified and that the rest of the distributed switch still uses the approved baseline.

## Common failure

A common rollout mistake is changing a shared port group first and discovering later that a critical workload depended on the previous behavior. Another is forgetting to document why an exception exists, which makes later cleanup difficult.

## Operational follow-up

## Goal

Keep the configuration stable and auditable after the initial change.

## Action

Review the security policy periodically and after major platform changes. Add the distributed switch and port-group settings to your configuration review process so deviations are caught early.

When a workload is decommissioned, remove its exception port group or return it to the secure baseline. If you are automating compliance checks, include the three security settings in the drift detection rules.

## Expected output

A maintainable policy state that stays aligned with workload needs and security standards.

## Validation

During audits or maintenance windows, confirm that exception networks still have a current business justification and that ordinary production port groups remain locked down.

## Common failure

The biggest long-term risk is configuration drift: exceptions are created for temporary reasons and then forgotten. That is usually how weak settings survive past their useful life.

## Practical baseline to use in most environments

For most VM production networks, a good starting point is to reject promiscuous mode, MAC address changes, and forged transmits. Then create narrow, documented exceptions only where a workload cannot operate without them. That approach gives you a secure default, an audit trail, and a clean decision process for unusual cases.

If you keep the scope tight, validate inheritance carefully, and test the workload behavior before production use, you will have a distributed switch policy that supports both security and operations without relying on guesswork.

Use this guidance together with EC2 instance recovery and AWS virtual machine migration strategies to connect the workflow with related operational context already available on the site.