FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

FortiGate Upgrade Dropped LAN-to-LAN Traffic — allow-traffic-redirect and the Implicit Deny Nobody Planned For

A war story about a firewall upgrade that finished cleanly and cut two LANs apart — traffic that had never had a policy, never written a log line, and never been on anyone's test list — and the checks I now run before and after any firmware change.

Daanial Farrukh · 06 November 2026 · https://quabin.com/fortigate-upgrade-lan-to-lan-implicit-deny/

The firmware upgrade was done at night. It finished. The firewall came back. The team sent the usual note asking the customer to verify their services.

The customer found the problem the next morning. Virtual machines on one of their LANs could not reach virtual machines on the other. Packet loss between them was 100 percent, and the system they called their key system was down.

The fix itself was small — one explicit policy, in place by early afternoon. The conversation was harder.

"Whose lapse is this?" their lead asked in the group. "And what measures will follow?"

Two explanations came back from the provider's side. Monitoring could not validate LAN-to-LAN traffic. And the customer should have flagged the problem after the verification note.

Neither one answered the question. The customer rejected both, and they were right to.

I accepted it as our lapse, and I apologised. Then I went looking for the question we should have asked before the change window opened:

"What on this firewall is working without a rule?"

The Trap, Named

That's the trap. A firewall review reads the policy table, and a change plan tests what the policy table says. But a flow that passes with **no policy** does not appear in that table at all. It has no rule to read, no owner to ask, and — as you will see — often no log line to find.

The most dangerous traffic on your firewall is the traffic nobody ever had to allow.

What Actually Happened to the Traffic

The customer had two LANs behind the same firewall interface. There was no LAN-to-LAN policy, because nobody had ever needed one. The old firmware passed the traffic. The new firmware dropped it under the implicit deny.

The path between the two LANs looks like this:

VM on LAN A → firewall interface → route lookup says "out the same interface" → VM on LAN B

Traffic that arrives on an interface and leaves on the same interface is called **hairpin traffic**. On a FortiGate, what happens next is decided by one global setting, allow-traffic-redirect.

- **enable** — same-interface traffic can pass on the routing decision alone, with no matching policy.
- **disable** — it needs a firewall policy whose source and destination interface are the same. Without one, implicit deny drops it.

Here's the part that trips people up. That default changed. Fortinet's FortiOS 7.4.10 release notes list it under "Changes in default behavior": the default moved from enable to disable, for IPv4 and for IPv6. Fortinet's technical tip gives the same change for 7.6.5 and 8.0.0 onward. And the release note says the upgrade sets it to disable **even if it was enabled before**.

So a box can be fully healthy, with every rule intact, and lose a whole class of traffic in the reboot.

I want to be exact about what is proven. The symptom, the old behaviour, the new behaviour and the fix are proven in this case. The specific firmware versions are not in my notes, so I cannot prove this setting was the mechanism on that box. It is the leading explanation, because it matches on every point. The first check in this article is the one that confirms it on yours.

This shape is common. You will find it wherever:

- Two subnets share one interface, usually through a secondary IP address.
- A router or layer 3 switch sits on the same LAN as the firewall, and the firewall is the default gateway that hands traffic back.
- A site grew a second LAN years ago and someone added an IP address instead of an interface.

Why Nobody Sees It in Testing

Three things hide this flow, and they stack.

It has no rule. A policy review cannot find what is not there.

It often has no log. Fortinet's tip notes that traffic logs are generated only when traffic passes a firewall policy. Hairpin traffic passing without a policy leaves no trail in the traffic log, so a "top talkers" report will not show the dependency either.

It is not on the monitor. Most monitoring checks the firewall itself, its uplinks and its published services. All of those were green. The traffic between two groups of a customer's own VMs was never a monitored item, and "firewall is up" says nothing about it.

So the post-upgrade check passed, honestly, on every item it contained. **The list was the failure, not the box.**

How to Actually Fix This

Steps 1 to 3 are read-only. Step 4 changes state, so do it in a window. I have used documentation addresses here: 192.0.2.0/24 for LAN A, 198.51.100.0/24 for LAN B, and port2 for the shared interface.

Step 1: Read the setting and the version.

```
get system status

get system global | grep allow-traffic
```

If you are on 7.4.10, 7.6.5, 8.0.0 or later, expect allow-traffic-redirect: disable. On a box still below those versions, it may say enable — and that box is the one to worry about, because it may depend on this today.

Step 2: Prove the drop with a flow trace. Start a ping from a VM on LAN A to a VM on LAN B, then run this on the firewall. Keep the count small; this is a production box.

```
diagnose debug flow filter addr 192.0.2.10

diagnose debug flow show function-name enable

diagnose debug enable

diagnose debug flow trace start 20
```

When you are done, switch it off:

```
diagnose debug disable
```

The line you are looking for is this:

```
msg="Denied by forward policy check (policy 0)"
```

Policy 0 is the implicit deny. This is not a bug in the firmware. It is the firewall doing exactly what the new default tells it to do.

Step 3: Find the dependency before you upgrade. On a box that has not been upgraded yet, the log will not help you, so look at the wire. Start a fresh ping between the two LANs and run:

```
diagnose sniffer packet port2 'net 192.0.2.0/24 and net 198.51.100.0/24' 4 20
```

The same packet going in on port2 and then out on port2 is hairpin traffic. Now read the policy table:

```
show firewall policy
```

Look for a policy whose srcintf and dstintf are the same. If the sniffer shows the traffic and the table has no such policy, that flow is alive on the old default alone. It will die in the upgrade.

Step 4: Write the policy (this changes state). Create the two address objects, then the policy. Set these up in a window, and take a config backup first.

```
config firewall address
    edit "LAN-A"
        set subnet 192.0.2.0 255.255.255.0
    next
    edit "LAN-B"
        set subnet 198.51.100.0 255.255.255.0
    next
end

config firewall policy
```

```
    edit 0
        set name "LAN-A-to-LAN-B"
        set srcintf "port2"
        set dstintf "port2"
        set srcaddr "LAN-A"
        set dstaddr "LAN-B"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

Repeat it with source and destination swapped if sessions start from both sides. Then narrow service once the logs show you what really flows.

There is a one-line alternative: set allow-traffic-redirect enable under config system global. It restores the old behaviour and it will stop the outage in a second. I use it only as an emergency measure. Fortinet's own article warns that it bypasses the policy check for hairpin flows, which leaves you exactly where you started — a production dependency with no rule, no log and no owner.

Plain takeaway: the setting hides the dependency; the policy documents it.

Step 5: Verify. This step is where most tutorials stop and where mine starts. Run the same ping, then the same trace. You should now see the policy id in the trace instead of policy 0. Then confirm the flow has its own trail: open Log & Report › Forward Traffic and filter on the policy name. You should see entries for LAN A to LAN B.

Now test real traffic, not just ping. If ping works but TCP sessions hang, Fortinet's tip points at anti-replay on the same-interface policy: set anti-replay disable under that policy. Ping alone would have given you a green tick and a broken application.

The Checklist I Run Around Any Firewall Upgrade

1. **Read "Changes in default behavior" in the release notes** for every version on the path, not just the last one. This section exists so that you do not meet it in production.
2. **Save the setting before and after.** get system global | grep allow-traffic goes in the change record, both times.
3. **Sniff for hairpin traffic on every interface that carries more than one subnet.** Any flow without a same-interface policy gets a policy first, before the upgrade.
4. **Ask the customer for east-west flows in writing.** "Which of your servers talk to each other through this firewall?" is a different question from "which services do you publish?"
5. **Give notice, and get a named person on their side.** Ask them to test during the window, not the next morning.
6. **Test it yourself after the reboot.** A provider's "please verify" note is not a verification. Keep one host in each LAN that can ping the other, and watch that path during the window.
7. **Add east-west probes to monitoring.** If the path matters, a monitor should measure the path, not the firewall that carries it.

8 **Check the rest of the fleet.** If one box has this shape, others built by the same people probably do. I have no proof of that for any specific firewall — that is exactly why you run the check.

Absence of errors is not health. A clean upgrade log proves the firmware installed. It does not prove the customer's traffic still flows.

What I Tell Every Customer Now

Before any firewall change, I ask one question:

"Name the traffic between your own servers that would stop the business if it failed — and tell me who tests it, and when, after our change."

If the answer is a list of flows and a name on each side, they're in good shape. If the answer is "the firewall doesn't touch that", we're not done yet.

I write about the questions I ask in real customer rooms — the ones that change architectures and surface the problems no one else noticed. If you found this useful, drop your email below and I'll send the next one when it's ready.

FAQs

Why did LAN-to-LAN traffic stop after a FortiGate upgrade?

A leading reason is the changed default of allow-traffic-redirect. From FortiOS 7.4.10, 7.6.5 and 8.0.0, it is disable, and the upgrade sets it to disable even if it was enabled before. Two subnets on one interface then need an explicit policy with the same source and destination interface, or implicit deny drops the traffic.

What does "Denied by forward policy check (policy 0)" mean?

It means the traffic matched no firewall policy and was dropped by the implicit deny at the end of the policy table. You see it in a debug flow trace. For same-interface traffic after an upgrade, the fix is a policy with the same interface as source and destination.

Should I set allow-traffic-redirect back to enable?

Only as an emergency measure. It brings the traffic back quickly, but that traffic again passes without a policy check. An explicit same-interface policy is better, because it can be logged, inspected and owned by someone.

Do two VLANs on a FortiGate need this policy too?

Two VLAN interfaces are two different interfaces, so that traffic is not hairpin traffic and has always needed a normal policy between them. The setting matters when ingress and egress are the same interface, such as two subnets on one interface with a secondary IP address.

How can I find hairpin traffic before an upgrade if it is not logged?

Use the sniffer on the shared interface with a filter on the two subnets, and start a fresh ping or session. Packets going in and back out on the same interface are hairpin traffic. Then check whether a policy with

matching source and destination interfaces exists.

Is the customer or the provider responsible for testing after a firewall upgrade?

Whoever makes the change owns the proof that it worked. The customer knows which flows matter, so collect that list before the window and agree a named tester. A note asking the customer to verify, sent after the work, moves the risk without removing it.

Read online: https://quabin.com/fortigate-upgrade-lan-to-lan-implicit-deny/