The short answer is No, RHEL 9 does not automatically "fix" this conflict, but it provides better tools that make the conflict manageable.

You are still fighting physics: fapolicyd intercepts system calls, and IBM MQ generates millions of them. However, RHEL 9 introduces three specific changes that reduce the "friction" of this stack significantly.

1. The "Rules.d" Feature (Management Fix)

In early RHEL 8, fapolicyd used a single monolithic configuration file (fapolicyd.rules). This made automation (like Chef) dangerous—if Chef made a syntax error in that one file, the whole security daemon crashed or locked everyone out.

 * RHEL 9 Change: fapolicyd now natively supports a /etc/fapolicyd/rules.d/ directory.
 * The Benefit: You can drop a standalone file (e.g., 80-ibm-mq.rules) specifically for MQ without touching the main system rules. If that one file is bad, it’s easier to isolate. This makes your Chef code much cleaner and less prone to "breaking the world."

2. The fanotify Kernel Upgrade (Performance Fix)

fapolicyd relies on a kernel feature called fanotify to intercept files.

 * RHEL 8 (Kernel 4.18): Uses an older version of fanotify that is less efficient at handling high-volume event bursts (like MQ starting up).
 * RHEL 9 (Kernel 5.14): Includes massive optimizations to the fanotify subsystem. It handles event queuing much faster with less CPU overhead.
 * The Result: The "I/O Tax" is lower on RHEL 9, but it is not zero. You still need to tune the queue size (q_size), but the likelihood of the queue overflowing and hanging MQ is lower.

3. SHA-1 Deprecation (New "Gotcha")

RHEL 9 introduces a new problem for this stack: SHA-1 is deprecated.

 * The Conflict: Older versions of IBM MQ (and some legacy Splunk forwarders) still use SHA-1 signatures for their internal Java jars or binaries.
 * The Failure: If you install an older IBM MQ version on RHEL 9, fapolicyd might block it not because it's untrusted, but because it considers the digital signature "weak" and invalid.
 * The Fix: You must ensure you are using IBM MQ 9.3+ and the latest Splunk Universal Forwarder to avoid signature rejection.

Summary: RHEL 8 vs. RHEL 9 for this Stack

Feature	RHEL 8	RHEL 9	Impact on You
Config Style	Monolithic File	rules.d Directory	Chef automation is safer/easier on RHEL 9.
Performance	Higher Overhead	Optimized Kernel	MQ latency is better on RHEL 9, but tuning is still required.
Trust Source	RPMDB Only (mostly)	Improved Trust	Easier to trust third-party installers like Splunk.
Signatures	Allows SHA-1	Blocks SHA-1	Risk: Old MQ versions will fail to start on RHEL 9.

Verdict: RHEL 9 is better, but you still need the Chef recipe I provided above. You cannot just install RHEL 9 and expect fapolicyd to happily ignore IBM MQ's massive I/O load.

This issue catches many administrators off guard because it isn't a "bug" in IBM MQ or Splunk—it is a deliberate security feature in RHEL 9 that breaks older trust models.

Here is the deep dive into the SHA-1 Deprecation Issue and how it specifically wrecks the fapolicyd + IBM MQ stack.

1. The Root Cause: RHEL 9's "Crypto Policy"

RHEL 9 ships with a system-wide cryptographic policy called DEFAULT.

 * The Rule: This policy completely disables SHA-1 for creating or verifying digital signatures.
 * The Logic: SHA-1 is cryptographically broken (collision attacks are possible). RHEL 9 treats any signature created with SHA-1 as if it were unsigned or forged.

2. How It Breaks fapolicyd

The conflict happens in the "Trust Chain."

 * The Install: You install an older IBM MQ version (e.g., 9.1 or early 9.2) via RPM. These RPMs were often signed by IBM using a SHA-1 key.
 * The Check: When fapolicyd starts, it initializes its list of trusted files. If you use trust=rpmdb (the default), fapolicyd asks the RPM database: "Is /opt/mqm/bin/strmqm part of a valid, signed package?"
 * The Failure: The underlying RHEL 9 crypto libraries intervene. They see the IBM SHA-1 key and tell the RPM database: "This signature is invalid."
 * The Result: fapolicyd sees the MQ files as untrusted/unknown, even though they were installed via a legitimate RPM. It blocks execution immediately.

3. How It Breaks Java (IBM MQ's Brain)

Even if you whitelist the binaries in fapolicyd, IBM MQ is heavily Java-based.

 * The JVM in RHEL 9 also respects the system crypto policy.
 * If MQ tries to load internal JAR files that are signed with SHA-1, the JVM will throw a SecurityException and crash the Queue Manager or the web console.

4. The Fixes (Ranked from Best to "Emergency Only")

Option A: The Correct Fix (Upgrade)

You must use IBM MQ 9.3 or higher and Splunk Universal Forwarder 9.0+ on RHEL 9.

 * IBM and Splunk have re-signed their modern packages with SHA-256 keys.
 * This satisfies RHEL 9's default policy, and fapolicyd will trust the RPMs automatically.

Option B: The "Legacy" Policy (The Workaround)

If you are forced to run an older version of MQ on RHEL 9, you can tell RHEL to loosen its standards and accept SHA-1.

The Command:

```
update-crypto-policies --set DEFAULT:SHA1
reboot
```

The Chef Resource:

If you need to automate this exception because you are stuck with old MQ versions:

```
# Enable SHA-1 support for legacy IBM MQ packages
execute 'enable_legacy_sha1' do
  command 'update-crypto-policies --set DEFAULT:SHA1'
  not_if 'grep -q "DEFAULT:SHA1" /etc/crypto-policies/config'
  notifies :reboot_now, 'reboot[system_reboot]', :immediately
end
```

```
reboot 'system_reboot' do
  action :nothing
  reason 'Changing crypto policy to support legacy SHA-1 signatures'
end
```

5. How to Verify if This is Your Problem
If fapolicyd is blocking MQ on RHEL 9, check the fapolicyd debug output or the audit log.
The Smoking Gun Log:
If you see an audit denial where the trust status is 0 (untrusted) for a file you know you installed via RPM, check the RPM signature:
rpm -Kv /opt/mqm/bin/strmqm

 * RHEL 8 Output: Header V3 RSA/SHA1 Signature, key ID ...: OK
 * RHEL 9 Output: Header V3 RSA/SHA1 Signature, key ID ...: BAD (Digest mismatch) or INVALID

If you see BAD or INVALID on the signature check, fapolicyd will never trust it unless you apply Option B (downgrade policy) or Option A (upgrade software).

You can detect this "time bomb" using standard Linux tools available on your current RHEL 8 system. You do not need to start the upgrade to find out if you will fail.
Here are the three specific checks you should run immediately.
1. Check Currently Installed RPMs (The "Am I Safe Now?" Check)
You need to query the RPM database to see what signature algorithm was used to sign your current IBM MQ and Splunk packages.
Run this command on your current RHEL 8 server:
rpm -qa --queryformat "%{NAME}-%{VERSION}: %{SIGALG}\n" | grep -E 'MQSeries|splunk|ibm'

How to read the results:
 * Result: MQSeriesRuntime-9.1.0: RSA/SHA1
   * Verdict: 🛑 DANGER. This package will be rejected by fapolicyd on RHEL 9 (Default Policy). You must upgrade MQ before or during the OS migration.
 * Result: MQSeriesRuntime-9.3.0: RSA/SHA256
   * Verdict: ✅ SAFE. This package uses SHA-256 and is compatible with RHEL 9.

2. Check New/Downloaded Packages (The "Pre-Install" Check)
If you have downloaded the upgrade files for IBM MQ (e.g., the 9.3 tarball) but haven't installed them yet, you should verify their signatures before you schedule the change window.
 * Extract the tarball to see the .rpm files.
 * Run the validation command against one of the core RPMs:
   rpm -Kpv MQSeriesRuntime-*.rpm

Look for the "Digests" output:
 * Bad Output: V3 RSA/SHA1 Signature...
   * This RPM is old. Do not use it for a RHEL 9 migration.

 * Good Output: V3 RSA/SHA256 Signature... or V4 RSA/SHA256 Signature...
   * This RPM is modern and safe.

3. The "Official" Simulation: leapp preupgrade

If you are planning an in-place upgrade (converting the same server from RHEL 8 to 9), Red Hat provides a tool called leapp that analyzes your system and generates a report of blockers.

Why run this?

It specifically looks for SHA-1 signatures in your installed packages and flags them as a "High" risk inhibitor.

 * Install the tool:
   dnf install leapp-upgrade

 * Run the analysis (Does not change the system):
   leapp preupgrade

 * View the Report:
   Open /var/log/leapp/leapp-report.txt. Search for "SHA1" or "deprecated". It will list every single RPM currently installed that will break when you reboot into RHEL 9.

Summary Checklist

Check	Command	Success Criteria
Current MQ	rpm -qa --queryformat "%{SIGALG}\n" ...	Output must verify SHA256
Current Splunk	rpm -qa --queryformat "%{SIGALG}\n" ...	Output must verify SHA256
RHEL 9 Readiness	leapp preupgrade	No "Inhibitor" errors regarding crypto policies

Next Step: Would you like a script that scans your entire server fleet and reports which servers have "SHA-1 Risk" detected?

Sources

Here is a compiled list of official documentation and technical resources to help you research and validate the architecture we discussed.

1. The "Crypto Policy" Conflict (RHEL 9 & SHA-1)

This is the most critical area to research before your upgrade. RHEL 9's default security policy breaks older IBM MQ and Splunk versions.

 * Red Hat Official Guide: Security considerations in adopting RHEL 9 (SHA-1 Deprecation)
   * Key Section: "SHA-1 usage for signatures is restricted in the DEFAULT system-wide cryptographic policy."
 * IBM MQ Deprecations: Deprecated, stabilized, and removed features in IBM MQ 9.3
   * Key Section: "Digital signatures using SHA-1 are deprecated."

2. The "Blocker" (fapolicyd vs. Middleware)

These sources explain why fapolicyd kills IBM MQ and Splunk by default and how to configure the whitelist ("trust") properly.

 * IBM Support: Fapolicyd software framework vs. Db2/MQ

  * Note: While written for Db2, the same mechanism applies to MQ: "fapolicyd controls the execution... and can block the installation... from running."
 * Red Hat Security Hardening: Blocking and allowing applications using fapolicyd
  * Key Concept: This explains how to use the rules.d directory and the fagenrules script we discussed for the Chef recipe.
 * Splunk Knowledge Base: Splunkd stops when Fapolicyd is up/running
  * Validation: Confirms the exact behavior: "Splunkd log could receive an error of operation not permitted... execve: Operation not permitted."

3. Performance Tuning (MQ & Splunk)

These resources back up the recommendations for tuning the q_size (Queue) and handling the "I/O Storm."

 * IBM MQ Tuning: Tuning your IBM MQ network
  * Relevance: Covers network and I/O tuning principles that are impacted when fapolicyd injects latency.
 * Splunk Docs: Include or exclude specific incoming data (Blacklisting)
  * Action: Use this to implement the blacklist = AMQERR*.LOG rule to prevent the "I/O Storm."

4. Upgrade Tools (Pre-Check)

 * Leapp Utility: [suspicious link removed]
  * Tool: This is the official guide for the leapp preupgrade command I recommended running to catch SHA-1 inhibitors.

... RHEL 9 to RHEL 6 SSH or 3rd party services fail due to SHA1 being disabled ...

This video is relevant because it visually demonstrates the exact RHEL 9 SHA-1 deprecation issue and how to use the update-crypto-policies command to fix signature failures.