



NVIDIA Container Toolkit Security: CDI and CVE Hardening

Plan and build private AI compute with GPU Smith. We connect workload requirements with GPU hardware, networking, deployment and operating decisions so your infrastructure fits the work it must perform.

Published by GPU Smith · 2026-10-29 · [nvidia container toolkit vulnerability](#) · [cve-2025-23266 nvidiascape](#) · [cve-2024-0132 container toolkit](#) · [nvidia container toolkit cdi vs legacy runtime](#) · [nvidia-ctk cdi generate](#) · [gpu container escape kubernetes mitigation](#) · [nvidia container toolkit disable-cuda-compat-lib-hook](#)

Original article: <https://gpusmith.com/articles/en/nvidia-container-toolkit-cdi-security-hardening>



In this article

1. Why the toolkit is a high-value target
 2. How the toolkit touches a container: legacy, CDI and JIT-CDI
 3. The advisory record, 2024 to 2026
 4. CVE-2024-0132 and the legacy-path TOCTOU family
 5. CVE-2025-23266 (NVIDIAScape): a hook bug on the CDI path
 6. What CDI buys you, and what it doesn't
 7. Patch process
 8. Hardening checklist
 9. MIG is not an escape boundary
 10. Detection and auditing
 11. CDI operations
 12. Hardening runbook
-

Why the toolkit is a high-value target

The NVIDIA Container Toolkit runs privileged code on the host every time a GPU container starts. Its job is to put device nodes, driver libraries and symlinks into a filesystem that the container image controls. That code runs as host root while the image may come from a tenant. Every serious toolkit CVE since 2024 exploits this: a crafted image wins a race, plants a symlink, or gets its library loaded into a host-side hook, and gains host filesystem access or root code execution. On a shared node, that means every other tenant's models and data.

Short version for October 2026: run toolkit **1.19.1 or later** and GPU Operator **26.3.2 or later**. Those are the fixed versions for the most recent advisory, CVE-2026-24260 ([NVIDIA bulletin 5850](#)). The current release is v1.20.1, published 2026-09-19 ([GitHub releases](#)), and NVIDIA's 2026 CVE index lists no newer toolkit or GPU Operator bulletin as of 2026-10-28 ([CVE index](#)). Use CDI or JIT-CDI mode, but don't treat CDI as a security fix. The critical 2025 bug, NVIDIAScape, was exploitable *only* in CDI mode on toolkit versions before 1.17.5. Track NVIDIA's bulletins rather than release notes, and keep untrusted images off nodes that host sensitive workloads.

For sharing and isolation performance rather than runtime security, see [MIG](#), [MPS](#) and [time-slicing on Kubernetes](#).

How the toolkit touches a container: legacy, CDI and JIT-CDI

The toolkit can inject GPUs into a container in three ways. Each one runs a different set of host-side code.

Legacy mode. `nvidia-container-runtime` wraps `runc`. It adds the NVIDIA Container Runtime Hook to the incoming OCI spec, and that hook calls `nvidia-container-cli` (`libnvidia-container`) to mount driver files into the container's root filesystem. The library inspects and modifies a rootfs that came from the image, so it is exposed to time-of-check/time-of-use (TOCTOU) and symlink tricks.

CDI mode. The Container Device Interface (CDI) is "an open specification for container runtimes that abstracts what access to a device, such as an NVIDIA GPU, means" ([NVIDIA CDI support docs](#)). A CDI spec on the host lists the device nodes, mounts and hooks for each device, and the runtime applies the spec. NVIDIA's docs say that in CDI mode "the NVIDIA Container Runtime does not inject the NVIDIA Container Runtime Hook" into the OCI spec. CDI does not remove hooks altogether. NVIDIA's generated specs still contain small hooks run by `nvidia-ctk/nvidia-cdi-hook`, such as `update-ldcache`, `create-symlinks` and `enable-cuda-compat` ([v1.17.5](#) and [v1.20.0 release notes](#)).

JIT-CDI mode. Since v1.18.0 (2025-10-21), the runtime defaults to "a just-in-time-generated CDI specification instead of defaulting to the legacy mode" ([v1.18.0 release](#)). With `mode = "auto"`, the runtime picks JIT-CDI on systems that use NVML, and it "does not write a persistent specification" to `/var/run/cdi` or `/etc/cdi` ([CDI docs](#)). NVIDIA's release notes say legacy mode is deprecated but still supported ([toolkit release notes](#)).

A fleet that was installed before October 2025 and never reconfigured may therefore run a different mode on each node. Upgrades alone can change which code path a node uses.

The advisory record, 2024 to 2026

All rows come from NVIDIA's own bulletins, read from NVIDIA's GitHub mirror of its PSIRT bulletins because the `nvidia.custhelp.com` pages returned HTTP 403 on 2026-10-28. Scores are NVIDIA's CVSS v3.1 base scores.

CVE	Bulletin date	Class	NVIDIA score	Toolkit affected	Toolkit fixed	GPU Operator fixed	CDI note
CVE-2024-0132	2024-09-25	TOCTOU (CWE-367)	9.0 Critical	≤ 1.16.1	1.16.2	24.6.2	"does not impact use cases where CDI is used"
CVE-2024-0133	2024-09-25	TOCTOU, empty-file creation	4.1 Medium	≤ 1.16.1	1.16.2	24.6.2	same
CVE-2025-23359	2025-02-11	TOCTOU (CWE-367)	8.3 High	≤ 1.17.3	1.17.4	24.9.2	"does not impact use cases where CDI is used"
CVE-2025-23266	2025-07-15 (rev. 2025-08-12)	Untrusted search path in hooks (CWE-426)	9.0 Critical	≤ 1.17.7; CDI mode only before 1.17.5	1.17.8	25.3.2	CDI was the exposed path on older versions
CVE-2025-23267	2025-07-15	Link following in <code>update-ldcache</code> hook (CWE-59)	8.5 High	≤ 1.17.7; CDI mode only before 1.17.5	1.17.8	25.3.2	same
CVE-2026-24260	2026-06-30	TOCTOU (CWE-367)	8.5 High	≤ 1.19.0	1.19.1	26.3.2	not stated

Sources: [bulletin 5582](#), [bulletin 5616](#), [bulletin 5659](#), [bulletin 5850](#). Smaller fixes in the same period include CVE-2024-0134 (symlink escape, fixed in 1.17.0) and CVE-2024-0135/0136/0137, which the release notes list under 1.17.3 ([toolkit release notes](#)).

Scores can differ by scorer. NVD rescored CVE-2024-0132 at 8.3 High (AV:N/AC:H/PR:N/UI:R/S:C/C:H/I:H/A:H), while NVIDIA scored it 9.0 (NVD [CVE-2024-0132](#)). NVD has published no score of its own for CVE-2025-23266 (status Deferred) or CVE-2026-24260 (Awaiting Analysis) as of 2026-10-28, so only NVIDIA's scores exist for those two. Bulletin 5850 has not been revised and still gives no technical detail. If your SLA keys off scores, record which scorer you use.

CVE-2024-0132 and the legacy-path TOCTOU family

NVIDIA describes CVE-2024-0132 as a TOCTOU flaw "when used with default configuration where a specifically crafted container image may gain access to the host file system" ([bulletin 5582](#)). The toolkit checks a path inside the image's rootfs, then acts on it; a crafted image swaps what the path points to in between. NVIDIA lists code execution, privilege escalation, data tampering and information disclosure as impacts.

CVE-2025-23359, published five months later, has nearly the same description and the same CVSS vector. It affected everything up to 1.17.3 ([bulletin 5616](#)). Patching one legacy-path TOCTOU did not close the class. Both bulletins state that the bug "does not impact use cases where CDI is used." The strong argument for CDI comes from this family: the large libnvidia-container code path that mounts into an image-controlled rootfs is no longer in the loop.

CVE-2025-23266 (NVIDIAScape): a hook bug on the CDI path

Wiz Research disclosed NVIDIAScape on 2025-07-17 ([Wiz](#)). As Wiz describes it, the toolkit registers an OCI createContainer hook that runs `/usr/bin/nvidia-ctk` with `enable-cuda-compat`. That hook "runs as a privileged process on the host". createContainer hooks "inherit environment variables from the container image unless explicitly configured not to", and the hook runs "with its working directory set to the container's root filesystem." Put together, an image can set `LD_PRELOAD` and get its own shared library loaded into the privileged hook. Wiz's proof of concept is three lines:

```
FROM busybox
ENV LD_PRELOAD=/proc/self/cwd/poc.so
ADD poc.so /
```

NVIDIA's bulletin gives the affected range as "All versions up to and including 1.17.7 (CDI mode only for versions prior to 1.17.5)". The fixed versions are toolkit 1.17.8, GPU Operator 25.3.2, Kubernetes Device Plugin 0.17.3 and MIG Manager 0.12.2. The device plugin was affected "only when device-list-strategy is cdi-annotations and/or cdi-cri", and MIG Manager "only when CDI is enabled" ([bulletin 5659](#)). For GPU Operator, the bulletin (revision 2.0.0, 2025-08-12) and NVIDIA's CVE index list versions up to and including 25.3.1 as affected, while the CVE.org record still says 25.3.0; we follow NVIDIA's bulletin.

Read the parenthetical carefully, because it is often read backwards. It does **not** say CDI was safe. It says that before 1.17.5, *only CDI mode* was vulnerable, because CDI specs were the path that ran nvidia-ctk hooks as createContainer hooks. Version 1.17.5 added a dedicated enable-cuda-compat hook for CUDA forward compatibility ([v1.17.5 release](#)). The bulletin's range implies that from 1.17.5 through 1.17.7, legacy-mode nodes were exposed as well. The bulletin also says CVE-2025-23266 "does not affect systems where crun is used as the low-level runtime".

The fix is not described as a security fix in the release notes. The v1.17.8 notes call it a bugfix release ([toolkit release notes](#)). The v1.17.7...v1.17.8 diff includes a commit merged from a private fork, titled "Add NVIDIA_CTK_DEBUG=false to hook envs" ([compare v1.17.7...v1.17.8](#)). Our reading, which NVIDIA has not confirmed: giving the hook an explicit environment stops it from inheriting the image's environment, which is the condition Wiz described. The upstream end-to-end tests now build an nvidiascape-test image with ENV LD_PRELOAD=/proc/self/cwd/poc.so as a regression check ([e2e test](#)).

Interim mitigation if you cannot upgrade yet (per [Wiz](#)). Disable the hook in /etc/nvidia-container-toolkit/config.toml:

```
[features]
disable-cuda-compat-lib-hook = true
```

On GPU Operator, Wiz gives these Helm values:

```
--set "toolkit.env[0].name=NVIDIA_CONTAINER_TOOLKIT_OPT_IN_FEATURES" \
--set "toolkit.env[0].value=disable-cuda-compat-lib-hook"
```

For Operator versions before 25.3.1, Wiz suggests pinning toolkit.version=v1.17.8-ubuntu20.04 (or v1.17.8-ubi8 on RHEL/OpenShift). The flag itself was introduced in v1.17.5 ([v1.17.5 release](#)). Turning it off removes CUDA forward-compatibility library injection, which affects containers that rely on a newer CUDA than the host driver; see the [CUDA compatibility upgrade gate](#) before you flip it fleet-wide.

What CDI buys you, and what it doesn't

Question	Legacy hook	CDI / JIT-CDI
Runs libnvidia-container mount logic against image rootfs	Yes	No
Affected by CVE-2024-0132 / 0133 / 2025-23359	Yes	No, per NVIDIA
Runs privileged nvidia-ctk hooks at container create	From 1.17.5, per the 23266 range	Yes (several hooks)
Affected by CVE-2025-23266 / 23267	1.17.5 to 1.17.7	All versions up to 1.17.7
Device list is declarative and auditable	No	Yes (static spec in /etc/cdi or /var/run/cdi; JIT-CDI keeps it in memory)

Question	Legacy hook	CDI / JIT-CDI
CVE-2026-24260	Not stated by NVIDIA	Not stated by NVIDIA

CDI is a smaller, more auditable attack surface, not the removal of privileged code from container start. Keep patching on bulletin cadence in any mode.

Patch process

1. Inventory per node, not per cluster. Record the toolkit package version, the runtime mode, the low-level runtime (runc or crun) and, if you use it, the GPU Operator chart version. On a host, `nvidia-ctk --version` and the package manager give the toolkit version; `helm list -A` gives the Operator chart. For multi-component upgrades, see the [GPU Operator upgrade gate](#).

2. Subscribe to bulletins, not release notes. Fixed versions have shipped before the bulletins that name them:

Fixed release	Release date	Bulletin date	Gap
1.17.4 (CVE-2025-23359)	2025-01-23	2025-02-11	19 days
1.17.8 (CVE-2025-23266/23267)	2025-05-30	2025-07-15	46 days
1.19.1 (CVE-2026-24260)	2026-05-21	2026-06-30	40 days

Release dates are from [GitHub releases](#); bulletin dates are from bulletins [5616](#), [5659](#) and [5850](#). Worked example for 1.17.8: 1 remaining day of May + 30 days of June + 15 days of July = 46 days. Neither the 1.17.8 nor the 1.19.1 release notes name the CVE they fixed. Teams that upgrade only on CVE announcements sat on vulnerable versions for those weeks; teams tracking releases were already patched. Because the upstream repo lists no GitHub security advisories after the 2024 pair, use [NVIDIA's bulletin repository](#) as the feed.

3. Canary, then roll. Toolkit upgrades can change mode defaults, compat libraries and injected files. Canary one node per GPU and driver combination with a CUDA smoke test and an LD_PRELOAD negative test like the upstream e2e test, then drain and roll.

4. Keep a rollback, but pin forward. Roll back only to a version that is not in a known-affected range. As of 2026-10-28 that means nothing below 1.19.1.

Hardening checklist

- **Patch level:** toolkit $\geq$ 1.19.1 and Operator $\geq$ 26.3.2 ([bulletin 5850](#)).
- **Mode:** CDI or JIT-CDI everywhere. NVIDIA warns CDI "can conflict with" a leftover legacy hook at `/usr/share/containers/oci/hooks.d/oci-nvidia-hook.json`; delete it ([CDI docs](#)).
- **Compat hook:** if you don't use CUDA forward compatibility, set `disable-cuda-compat-lib-hook = true` as defense in depth, even on patched versions.

- **Image environment, not just pod spec:** the NVIDIA Scape vector was an ENV line in the *image*. Admission policies that check only `pod.spec.containers[].env` would not have seen it. Scan the image config for `LD_PRELOAD`, `LD_LIBRARY_PATH` and similar loader variables at the registry or admission layer.
- **Untrusted images:** run tenant-supplied images only on nodes that hold no other tenant's data. Wiz's own conclusion was that "containers are not a strong security barrier and should not be relied upon as the sole means of isolation" (Wiz). For hostile multi-tenancy, use a VM boundary per tenant; see [confidential GPU attestation](#) for the strongest version of that.
- **Capabilities:** deny `CAP_SYS_ADMIN` to tenant pods. Since v1.20.1, a container with `CAP_SYS_ADMIN` that sets `NVIDIA_MIG_CONFIG_DEVICES` or `NVIDIA_MIG_MONITOR_DEVICES` to all gets `/dev/nvidia-caps` MIG management nodes injected ([v1.20.1 release](#)).
- **Pod security:** enforce non-root, no privileged pods, read-only root filesystem where the workload allows. These limit what a compromised container can do, though a hook-level escape runs as host root before the workload process starts.

MIG is not an escape boundary

MIG partitions GPU hardware: "Each GPU slice includes dedicated GPU memory resources", a GPU instance "provides memory QoS", and MIG supports up to 7 partitions per GPU, compared with 48 MPS clients. Within a GPU instance, though, "The CIs share memory and engines" ([MIG concepts](#)). That is isolation *between GPU workloads*; it does nothing about a toolkit hook running on the host CPU. Seven tenants on seven MIG slices share one toolkit, one runc and one kernel, so one NVIDIA Scape image would have compromised all seven.

Detection and auditing

- **Spec drift:** `nvidia-ctk cdi list` shows the CDI devices in use. `nvidia-ctk --debug cdi list` reports errors from loading specs in `/etc/cdi` or `/var/run/cdi` ([CDI docs](#)). Hash static specs and alert when a spec changes outside a maintenance window.
- **Hook process lineage:** alert on `nvidia-ctk` or `nvidia-cdi-hook` processes that open files outside driver and library paths, spawn shells, or make network connections.
- **Image provenance:** require signed images from approved registries for GPU namespaces, and keep the loader-variable scan above in the pipeline.
- **Retrospective hunt:** for any period when nodes ran $\leq 1.17.7$, search registry and pull logs for images whose config sets `LD_PRELOAD`.

CDI operations

Since v1.18.0, the `nvidia-cdi-refresh` service generates `/var/run/cdi/nvidia.yaml` when the toolkit or driver is installed or upgraded and at reboot. To generate the spec by hand:

```
sudo nvidia-ctk cdi generate --output=/var/run/cdi/nvidia.yaml
nvidia-ctk cdi list
```

According to NVIDIA's docs, the spec must be regenerated manually after removing drivers or reconfiguring MIG (CDI docs). Include that step in MIG reprofiling runbooks. A stale spec breaks workloads and is an unreviewed change to what containers can reach.

Hardening runbook

#	Step	Owner	Verify
1	Inventory toolkit, Operator, mode, low-level runtime per node	Platform	Report shows no node below 1.19.1 / 26.3.2
2	Subscribe to NVIDIA PSIRT bulletins; set a 30-day toolkit upgrade SLA	Security	Feed configured; SLA in policy
3	Upgrade the canary node per GPU/driver pair	Platform	CUDA smoke test passes; LD_PRELOAD test image does not execute on host
4	Switch to CDI/JIT-CDI; remove legacy hook file	Platform	nvidia-ctk cdi list populated; no oci-nvidia-hook.json
5	Set disable-cuda-compat-lib-hook where forward compat is unused	Platform	Config diff reviewed
6	Add an image-config loader-variable scan at the registry or admission layer	Security	Test image with ENV LD_PRELOAD rejected
7	Separate untrusted tenants onto their own nodes or VMs	Platform + Security	Scheduling policy test
8	Add MIG reconfiguration → CDI regeneration to runbooks	Platform	Spec hash changes only in change windows

GPU Smith helps teams scope this kind of runtime hardening alongside GPU sizing and deployment.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. <https://github.com/NVIDIA/product-security/blob/main/2026/5850/5850.md>
2. <https://github.com/NVIDIA/nvidia-container-toolkit/releases>
3. https://github.com/NVIDIA/product-security/blob/main/2026/CVE_index.md
4. <https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/cdi-support.html>
5. <https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/release-notes.html>
6. <https://github.com/NVIDIA/product-security/blob/main/2024/5582/5582.md>
7. <https://github.com/NVIDIA/product-security/blob/main/2025/5616/5616.md>
8. <https://github.com/NVIDIA/product-security/blob/main/2025/5659/5659.md>
9. <https://nvd.nist.gov/vuln/detail/CVE-2024-0132>
10. <https://www.wiz.io/blog/nvidia-ai-vulnerability-cve-2025-23266-nvidiascape>

11. <https://github.com/NVIDIA/nvidia-container-toolkit/compare/v1.17.7...v1.17.8>
12. https://github.com/NVIDIA/nvidia-container-toolkit/blob/main/tests/e2e/nvidia-container-toolkit_test.go
13. <https://docs.nvidia.com/datacenter/tesla/mig-user-guide/concepts.html>

THE PUBLISHER

About GPU Smith

GPU Smith provides independent engineering and research for private AI infrastructure. We help technical buyers and operators reason about GPU workload sizing, hardware selection, procurement, deployment and inference operations, from the compute system to the networking, power and cooling requirements around it.

Start with the workload

Useful infrastructure decisions begin with the models, data, throughput, latency and operating constraints a team actually has. GPU Smith helps connect those requirements with system design and deployment choices. The goal is a privately controlled AI environment whose capacity and operational demands are understood before a purchasing decision.

Hardware and supplier research

Our public reference library covers GPUs, complete systems and networking components. The server-vendor directory and manufacturing research help teams investigate suppliers, compare options and follow sources behind technical claims. We distinguish manufacturer specifications from measured benchmarks and advertised capabilities from tested configurations. Reference pages are research resources, not a live inventory listing or a binding equipment quote.

Deployment and operations

GPU Smith's engineering scope includes infrastructure integration, networking, power, cooling and the practical operation of inference workloads. Our published articles explain the tradeoffs behind deployment, procurement and ongoing operations so teams can ask better questions and document decisions.

Work with GPU Smith

Explore the [hardware library](#), [GPU references](#), [networking references](#) and [vendor research](#). [Contact GPU Smith](#) with your workloads and deployment constraints to discuss a project and confirm current scope, pricing and availability.

Inclusion of a manufacturer or product in our research does not imply a partnership, certification or endorsement.

Website: <https://gpusmith.com>

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.