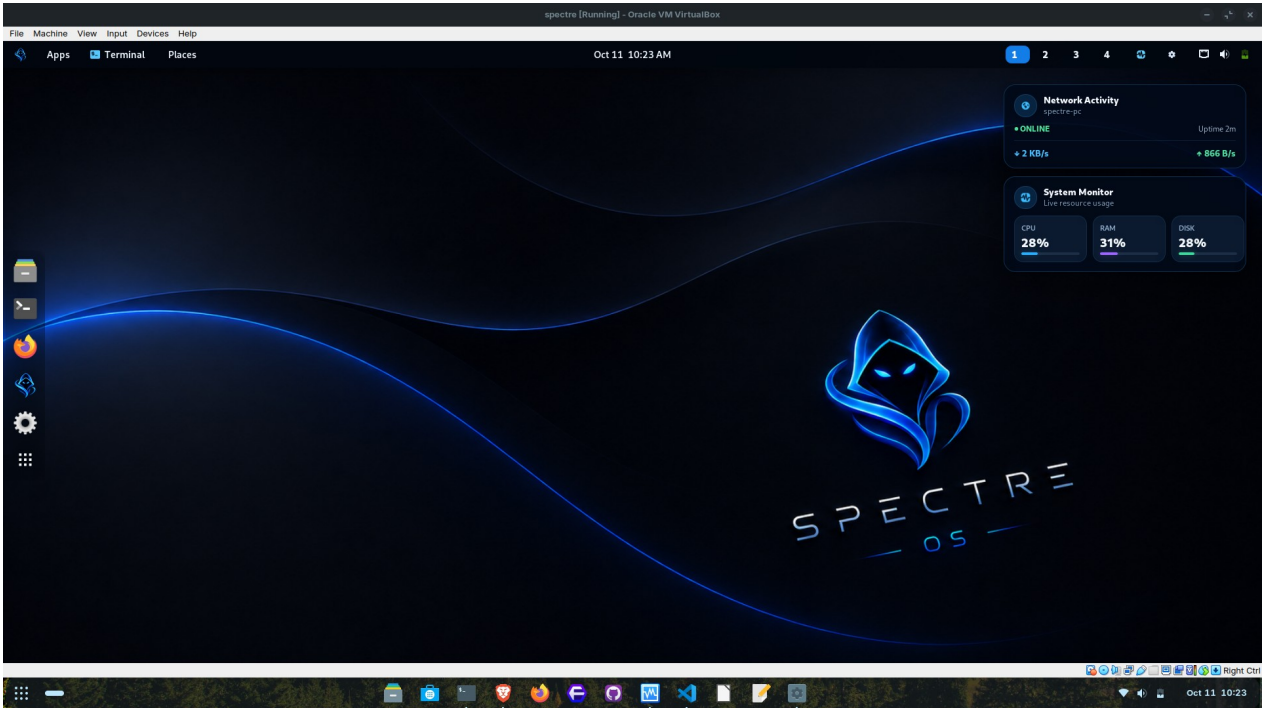


SPECTRE / 1.0

# Linux Spectre 1.0

Technical handbook • architecture, deployment, ToolHub operations, security model, validation and maintenance

|          |                                      |
|----------|--------------------------------------|
| Edition  | Linux Spectre / Spectre OS 1.0       |
| Document | MAINTAINER EDITION • 11 October 2026 |



Actual project screenshot. Display and application availability depend on installed hardware and configuration.

Prepared from the project's source packages, website materials, recorded installation/acceptance logs, and upstream documentation.

## Document control and evidence boundary

| Field                      | Record                                                                                                            |
|----------------------------|-------------------------------------------------------------------------------------------------------------------|
| Product / marketed version | Linux Spectre / Spectre OS 1.0                                                                                    |
| Purpose                    | Daily desktop with optional Kali container tools for authorized security work                                     |
| Maintainer credit          | CodeDev by Edward (project creator)                                                                               |
| Architecture               | x86_64; Debian-family desktop with GNOME; custom 7.1.8-spectre kernel                                             |
| Release artifact           | Spectre-OS-1.0-x86_64.iso (release-named copy of verified RC8.6 image)                                            |
| Verified SHA-256           | a49d7602a772c2db1dfb72411e47574ec908c5337d05131cb10433119f689d76                                                  |
| Verified image size        | Approximately 1.8 GB; exact bytes must be read from release host                                                  |
| Functional test profile    | VirtualBox 7.0.16; 4 GB RAM / 2 vCPUs / 40 GB virtual disk test recommendation                                    |
| Documentation date         | 11 October 2026                                                                                                   |
| Evidence                   | Uploaded patches, site FEATURE_FACTS.md, installer and runtime logs, screenshots and official upstream references |

**Evidence statement.** This handbook documents verified project behaviors and visible source excerpts, not a complete independent audit of every package in the final ISO. Where exact Debian release, package manifest, disk encryption, Secure Boot or hardware support could not be established, this text marks that as unverified.

## Contents

- 01 Product identity and origin
- 02 Scope, audience and deployment assumptions
- 03 System architecture and component boundaries
- 04 Release artifact and provenance
- 05 Installation and first boot
- 06 Desktop and everyday operation
- 07 Spectre ToolHub overview
- 08 ToolHub catalog, APT and package lifecycle
- 09 Rootless Podman and Kali toolbox
- 10 VirtualBox guest display integration

- 11 Security posture and limitations
- 12 Maintenance and updates
- 13 Troubleshooting and recovery
- 14 Release engineering and reproducibility
- 15 Acceptance evidence and remaining gates
- 16 Third-party licensing and attribution
- 17 Roadmap and change management
- 18 Administration quick reference
- 19 Glossary and upstream references

## 01 Product identity and origin

Linux Spectre (also styled Spectre OS) is a personally created x86\_64 Linux desktop project designed to bridge everyday computing and optional security-testing workflows. Its creator wanted a desktop suited to normal use while still making Kali security-tool packages approachable when needed.

The creator was inspired by Kali Linux's penetration-testing ecosystem and by the name associated with a debloated Windows project known as Ghost Spectre. The naming inspiration does not imply a technical derivation from Windows, licensing rights to third-party names, endorsement or affiliation. Linux Spectre is a separate Linux project.

The Debian family was chosen for its established packaging model and stability-oriented ecosystem. Debian is a major ancestor of many Linux distributions, including Kali; it is not the ancestor of every Linux distribution. Whether this ISO tracks Debian Stable repositories specifically must be checked against its `/etc/os-release` and APT sources.

Goal: keep a general-purpose GNOME host separate from an optional Kali Rolling toolbox. This aims to avoid mixing the host's APT configuration with Kali-specific package repositories. It does not turn the host into a certified hardened system.

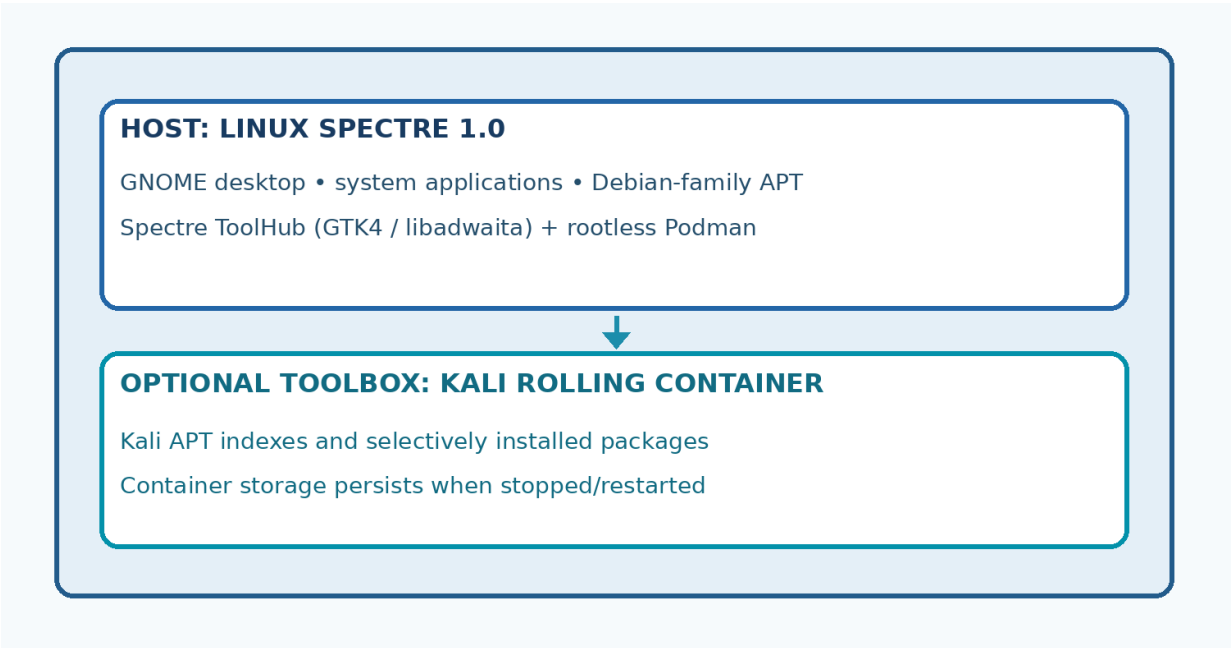
## 02 Scope, audience and deployment assumptions

| Audience              | Primary tasks                                        | Boundary                                                       |
|-----------------------|------------------------------------------------------|----------------------------------------------------------------|
| Desktop user          | Browser, files, terminal, productivity tools         | Application availability must be checked on installed image    |
| Student / learner     | Understand Linux, package management and lab tooling | Work only in labs and authorized targets                       |
| Security professional | Select and update tools through ToolHub              | Hardware and high-privilege tests may need extra configuration |
| Maintainer            | Build images, preserve provenance, test regressions  | Must satisfy third-party license and release gates             |

Known tested platform: a fresh VirtualBox installation with an installed ext4 root filesystem, a 7.1.8-spectre kernel and GNOME desktop. The VM guest displayed full-window desktop output. No claim of broad bare-metal certification or consistent performance on every GPU is made.

### 03 System architecture and component boundaries

Linux Spectre follows a two-environment model: day-to-day desktop services live on the host; Kali package workloads live in an optional container managed by Spectre ToolHub.



Simplified conceptual architecture; tool requests pass from ToolHub through Podman to an on-demand Kali container.

| Layer                   | Implementation / role                                                                    |
|-------------------------|------------------------------------------------------------------------------------------|
| Boot and storage        | BIOS/UEFI-capable installed ISO; ext4 tested in fresh VM                                 |
| Kernel                  | Custom 7.1.8-spectre with driver inventory verified in packaged SquashFS                 |
| Desktop                 | GNOME-based graphical session and branded UI                                             |
| ToolHub front end       | Native GTK4 / libadwaita application; search, categories, package selection and progress |
| Container runtime       | Rootless Podman; named reusable container spectre-kali-toolbox                           |
| Kali image              | docker.io/kalilinux/kali-rolling:latest                                                  |
| Package manager in Kali | APT running inside container, not host Debian APT                                        |
| VirtualBox integration  | VirtualBox-conditional guest userspace service, without building external modules        |

### 04 Release artifact and provenance

The public-facing 1.0 filename is a verified-byte-for-byte copy of the qualified image. Renaming the archive is not a new compile; its SHA-256 does not change if the bytes are identical.

```
ISO="$HOME/Downloads/Spectre-OS-1.0-x86_64.iso"
sha256sum "$ISO"
```

```
printf "%s %s\n" "a49d7602a772c2db1dfb72411e47574ec908c5337d05131cb10433119f689d76" "$ISO" |
sha256sum -c -
```

Source image on the Zorin build host (maintainer-specific):

~/spectre-v3-rc2-source/build/rc8-6-internal/output/Spectre-OS-1.0-RC8.6-VirtualBox-Display.iso. The RC8.6 label is an internal build lineage and is intentionally not part of the public-facing product name.

Provenance: the RC8.6 builder checked a preserved RC1 recovery image, RC8.5 baseline, protected installer/ToolHub/Podman source identities, embedded Guest Additions userspace and boot entries. The output log ended with RUNNER\_EXIT\_CODE=0 and static qualification PASS. Public release approval is a separate event.

## 05 Installation and first boot

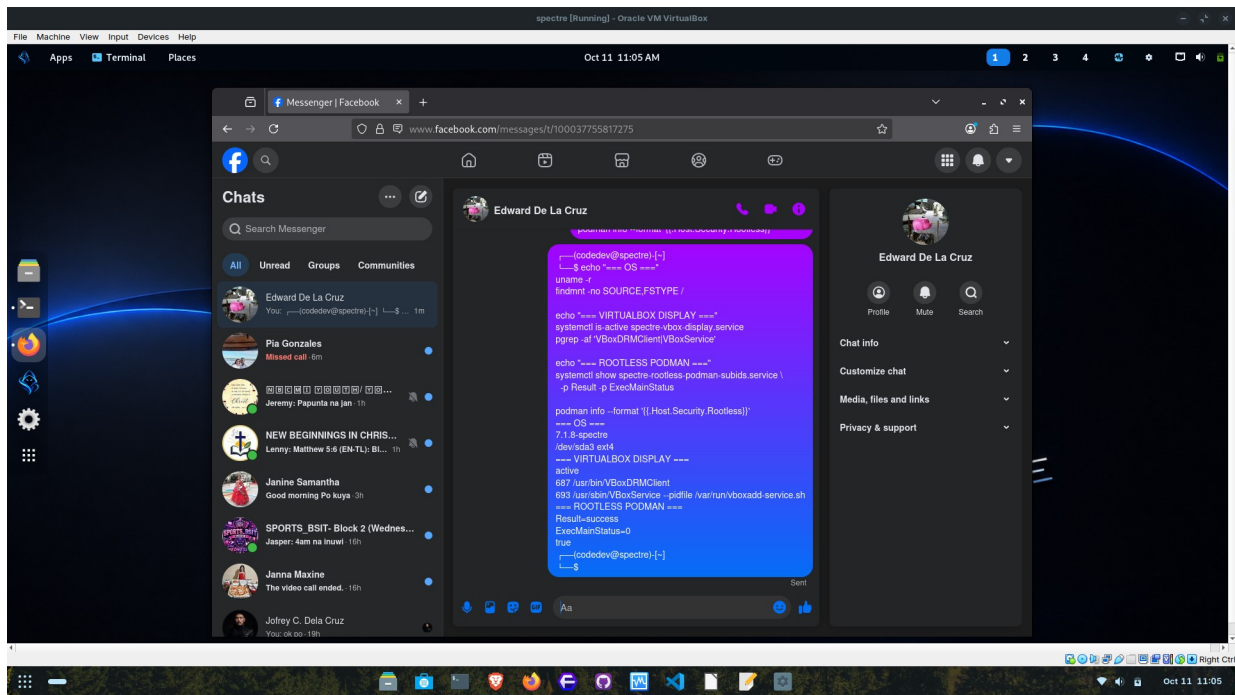
1. Obtain the image only from a source controlled by the project maintainer. Verify its SHA-256 before writing any installation media.
2. Back up all important files. Selecting a physical disk during installation can erase existing data.
3. For a conservative first test, use VirtualBox with Debian (64-bit), at least 4 GB RAM, two vCPUs and a new 40 GB virtual disk; these are tested/illustrative values, not certified minimum hardware requirements.
4. Use VMSVGA graphics and NAT networking in VirtualBox. Attach the ISO, boot and follow the installer's disk and account prompts.
5. Power off, detach the installation ISO, boot from the installed disk, then validate `/etc/os-release` and `uname -r`.
6. For physical installations, verify boot mode, Wi-Fi, graphics, audio, suspend and encryption requirements independently before migrating your daily workstation.

```
uname -r
cat /etc/os-release
findmnt -no SOURCE,FSTYPE /
lsblk -f
```

**Disk safety.** Do not use example device paths such as `/dev/sda` in destructive partitioning commands. Disk names vary by machine, and the installer must be reviewed before confirmation.

## 06 Desktop and everyday operation

GNOME provides the daily-computing interface: application launcher, windows, file management, network settings, display configuration and integrated terminal. Linux Spectre branding and custom UI elements are visible in the recorded desktop screenshots.

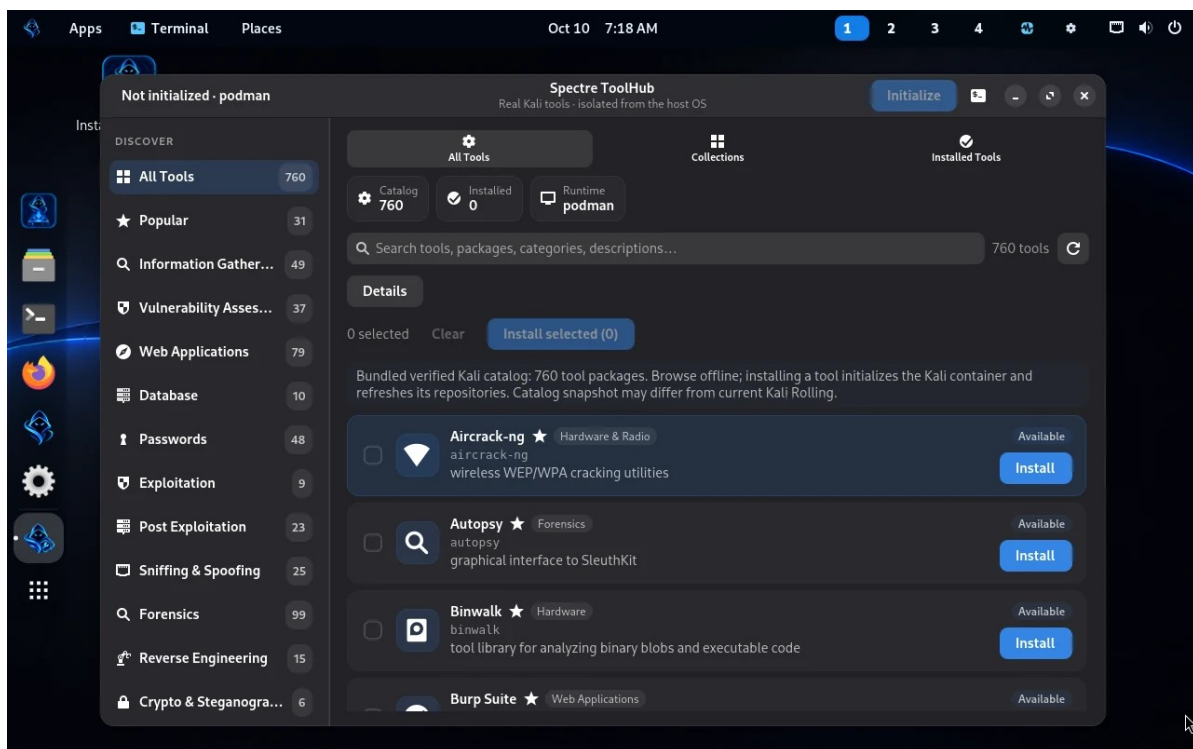


Developer-provided installed Spectre desktop screenshot (VirtualBox).

The distribution is intended for regular work—documents, browsing, development and general administration—as well as security learning. Availability of particular office, communication or development packages is dependent on the installed package manifest and user configuration; no unsupported bundled-app list is asserted here.

## 07 Spectre ToolHub overview

Spectre ToolHub is the native GTK4/libadwaita interface that connects users with a Kali package catalog without requiring Kali to be the host operating system. Its role is discovery and lifecycle management, not automatic installation of every security utility.



ToolHub screenshot from the project website's captured assets.

- Offline catalog: a bundled snapshot containing 760 package entries is searchable before the Kali container exists.
- Selective installation: pick an individual package or a controlled batch; the current source limits each batch to 25 packages to protect storage.
- Real APT progress: the current implementation parses APT status output and avoids inventing percentages for non-determinate steps.
- Collections: Kali metapackage groupings can be discovered from current Kali APT metadata after initialization.
- Lifecycle: initialize, search, install, list installed packages, remove, refresh indexes and open a Kali shell.

**Important terminology.** 760 catalog entries  $\neq$  760 programs already installed. The count is a snapshot inventory, and package availability can change in Kali Rolling.

## 08 ToolHub catalog, APT and package lifecycle

| Operation      | What current source does                                                                 | Caution                                                   |
|----------------|------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| Browse offline | Reads bundled <code>kali_catalog_snapshot.json</code>                                    | Metadata is a snapshot, not a live repository promise     |
| Initialize     | Pulls Kali image if missing; creates and starts container; <code>apt-get update</code>   | Requires internet for first-time pull and package indexes |
| Refresh        | Updates APT index inside Kali then re-discovers tools                                    | Rolling package availability changes                      |
| Install one    | Validates package name; updates APT; runs <code>apt-get install -y</code> inside toolbox | Storage/network consumption varies                        |
| Batch install  | Serialized queue, max 25; verifies each via <code>dpkg-query</code>                      | Failed package reported; others may continue              |
| Remove         | Runs <code>apt-get remove -y</code> inside Kali                                          | Does not necessarily purge configuration/dependencies     |
| Installed list | Uses <code>dpkg-query</code> in Kali                                                     | Container must be accessible                              |
| Kali shell     | Uses interactive container <code>exec</code>                                             | Different filesystem and permissions from host            |

Runtime identifiers recorded in source: image `docker.io/kalilinux/kali-rolling:latest`; container `spectre-kali-toolbox`; Kali hostname `spectre-kali`. The base official Kali image does not include all metapackages by default. Upstream image documentation confirms this design.

```
podman ps -a --format "{{.Names}}" | {{.Status}}"
podman start spectre-kali-toolbox    # if stopped
podman exec spectre-kali-toolbox nmap --version
podman exec -it spectre-kali-toolbox /bin/bash
```

The validated example is Nmap 7.99. It was installed in the container, survived a reboot and ran again after restarting that same container. This verifies the persistence workflow for the tested package, not compatibility with all catalog entries.

## 09 Rootless Podman and Kali toolbox

Spectre initializes the toolbox using a non-root host account and a user namespace. Its create path includes `--security-opt no-new-privileges`, rather than mounting Kali APT sources into the Debian-family host. Package-manager commands inside the container can still execute as root inside the container; that is not the same as host root.

The fresh test reported podman info `Host.Security.Rootless=true`; `spectre-rootless-podman-subids.service` `Result=success` and `ExecMainStatus=0`. The corresponding `/etc/subuid` and `/etc/subgid` ranges were configured automatically. The provisioner was modified to handle account-database lock contention during installation.

```
podman info --format "{{.Host.Security.Rootless}}"
systemctl show spectre-rootless-podman-subids.service -p Result -p ExecMainStatus
grep "^$(id -un):" /etc/subuid /etc/subgid
podman inspect spectre-kali-toolbox --format "Exit={{.State.ExitCode}}
OOMKilled={{.State.OOMKilled}}"
```

Rootless is a useful privilege-reduction mechanism, not a substitute for system hardening, trusted packages, updated software or authorization. Unprivileged default containers usually cannot directly perform wireless monitor mode, USB passthrough and other hardware-specific workflows.

The reboot test showed `Exited (137)` with `OOMKilled=false`. Exit code 137 means SIGKILL; this could occur during shutdown and is not by itself evidence of an out-of-memory event. Restarting the existing container succeeded and Nmap persisted. If repeated exits occur during normal use, review journal and resource limits.

## 10 VirtualBox guest display integration

The tested ISO bundles the Guest Additions 7.0.16 userspace extracted from the developer's already-functioning VM. It does not rerun the Oracle installer or attempt to compile external kernel modules during installation. An enabled `spectre-vbox-display.service` is guarded by `ConditionVirtualization=oracle` and the existence of `/dev/vboxguest` and the expected vendor init script.

```
systemctl is-active spectre-vbox-display.service
pgrep -af "VBoxDRMClient|VBoxService"
systemctl is-enabled vboxadd.service || true
uname -r
```

Fresh-install output showed `spectre-vbox-display.service` active, `VBoxDRMClient` and `VBoxService` running and the 7.1.8-spectre custom kernel loaded. Branded login and desktop were shown in full-window VirtualBox screenshots. The tested environment is not proof of compatibility with every VirtualBox version or every physical GPU.

**Release packaging caveat.** The embedded Oracle binary runtime raises redistribution and corresponding-source obligations. Do not publish the binary until a license inventory and notices/source process is completed.

## 11 Security posture and limitations

| Design decision          | Benefit                                                   | Limits / threats                                         |
|--------------------------|-----------------------------------------------------------|----------------------------------------------------------|
| Host/Kali APT separation | Reduces package and repository mixing                     | Does not stop all host compromise paths                  |
| Rootless Podman          | Avoids a host-root container daemon for standard workflow | Kernel and configuration vulnerabilities remain relevant |

|                                |                                                         |                                                            |
|--------------------------------|---------------------------------------------------------|------------------------------------------------------------|
| no-new-privileges              | Reduces privilege escalation through exec paths         | Not a blanket sandbox guarantee                            |
| Selected installations         | Lower disk footprint than entire Kali bundle            | Installed tools still need patches and safe usage          |
| SHA-256 verification           | Detects changed download bytes against trusted checksum | Checksum must come from an authentic independent source    |
| VirtualBox conditional service | Starts guest userspace only on identified hypervisor    | Non-VirtualBox behavior needs representative test coverage |

No evidence was provided for full-disk encryption default, measured boot, Secure Boot trust chain, mandatory access control profile, SBOM completeness, penetration-test certification, independent vulnerability audit or security update SLA. Documentation therefore does not claim these protections.

Use tools only on systems you own or are expressly authorized to assess. Maintain written scope, comply with local law, retain logs as appropriate, and avoid scanning or exploiting third-party systems without permission.

## 12 Maintenance and updates

### Host maintenance

```
sudo apt update
apt list --upgradable
sudo apt upgrade
```

Run only after checking the actual host repositories and pinning policy. A custom kernel can be sensitive to generic kernel package updates; maintainers should stage and test upgrades before promoting them. Do not add Kali Rolling sources to the host to obtain ToolHub tools.

### Toolbox maintenance

```
podman start spectre-kali-toolbox
podman exec spectre-kali-toolbox apt-get update
podman exec spectre-kali-toolbox apt list --upgradable
```

ToolHub itself performs an APT refresh as part of its documented initialization or install flow. Kali Rolling may change frequently. Snapshot the container or preserve a supported recovery/export method before large updates. Never remove the persistent container while expecting its manually installed packages to remain.

### Storage and recovery practices

- Maintain an immutable copy of the verified release ISO and checksum.
- Keep a tested backup of personal files and a separately restorable copy of project build sources.
- Back up rootless container storage using a supported Podman workflow or a VM snapshot; avoid assuming host file backup includes every layer.
- Stage major package, kernel and bootloader changes in a sacrificial VM before deploying to a daily workstation.
- Document and verify rollback instructions and actual recovery tests.

## 13 Troubleshooting and recovery

| Symptom                          | First inspection                 | Safe response                              |
|----------------------------------|----------------------------------|--------------------------------------------|
| ToolHub has catalog but no tools | podman ps -a; network; free disk | Initialize Kali; first pull needs internet |

|                               |                                              |                                                                        |
|-------------------------------|----------------------------------------------|------------------------------------------------------------------------|
|                               |                                              |                                                                        |
| Nmap unavailable              | podman start; exec nmap --version            | Verify install within same container, do not recreate prematurely      |
| Container Exited (137)        | podman inspect OOMKilled; journalctl         | Determine shutdown versus memory pressure                              |
| Rootless user namespace error | subuid/subgid; provisioner Result            | Do not hand-edit account files without backup                          |
| Window will not resize        | guest service active; clients; hypervisor    | Check virtual display service; do not run module build without headers |
| Unexpected boot failure       | boot mode, disk, ISO hash                    | Restore known ISO/snapshot before invasive edits                       |
| APT package not found         | apt-get update inside Kali, package spelling | Kali Rolling repositories change; do not force host repo mix           |

For a reproducible support ticket collect version, VirtualBox or hardware details, boot logs, ToolHub error output, selected package and exact steps. Redact home paths, usernames and network identifiers where privacy matters.

## 14 Release engineering and reproducibility

The development series used a continuously updated authoritative source approach: apply bounded changes to the master source, retain a source backup, build an isolated cumulative image, run static checks, and then perform fresh-install testing. Incremental patch ZIPs are mechanisms for making changes, not the long-term sole source of truth.

- Preflight: record SHA-256 of trusted source inputs and check free disk and mounted baseline.
- Integrate: permit only reviewed files and build hook changes; preserve prior ISO and installer source.
- Build: assemble SquashFS and BIOS/UEFI installable image; save logs and source manifest.
- Static verify: compare boot entries, critical kernel modules, protected core source hashes and full image SHA-256.
- Runtime UAT: clean installation, ToolHub, Podman, display resize and reboot persistence.
- Release: produce public filename by verified copy, sign off legal notices, publish checksum and actual verified URL only with owner approval.

```
# Maintainer-specific examples, do not run unless relevant build host exists
sha256sum -c "$HOME/spectre-v3-rc2-source/build/rc8-6-internal/output/Spectre-OS-1.0-RC8.6-VirtualBox-Display.iso.sha256"

# Check build qualification result in RC8.6-build.log:
# RC8.6 STATIC QUALIFICATION PASSED ... RUNNER_EXIT_CODE=0
```

A complete machine-independent rebuild guide, full source manifest/SBOM, dependency versions and ISO recipes cannot be guaranteed from the source excerpts provided; maintainers should publish these before promising bit-for-bit reproducibility to external builders.

## 15 Acceptance evidence and remaining gates

|                  |                         |
|------------------|-------------------------|
| Evidence or gate | State as of 11 Oct 2026 |
|------------------|-------------------------|

|                                                   |                                                |
|---------------------------------------------------|------------------------------------------------|
| ISO build completed; SHA-256 a49d...9d76          | PASS: user-supplied build and checksum logs    |
| Fresh VirtualBox install and GNOME login          | PASS: screenshots and installed system reports |
| Custom 7.1.8-spectre kernel; ext4 root            | PASS: user terminal output                     |
| VBoxDRMClient + VBoxService active after boot     | PASS: user terminal output                     |
| Rootless Podman provisioner successful            | PASS: user terminal output                     |
| Nmap 7.99 inside spectre-kali-toolbox             | PASS: user terminal output                     |
| Nmap retained after reboot and container restart  | PASS: user terminal output                     |
| Non-VirtualBox guest / bare-metal regression test | NOT DOCUMENTED                                 |
| Complete source/license/SBOM redistribution audit | NOT DOCUMENTED                                 |
| Owner's public-release signoff / hosted ISO URL   | NOT DOCUMENTED                                 |

Technical acceptance in the tested environment is separate from blanket device compatibility or a formally authorized public binary release.

## 16 Third-party licensing and attribution

Linux Spectre uses upstream free/open-source components and a separately sourced Oracle Guest Additions userspace payload. A redistributor must identify license files per installed component, preserve attributions, provide required corresponding source and applicable notices, and review trademarks and artwork permissions. The Oracle 7.0 base package licensing documentation is a starting point, not a substitute for identifying licenses of the exact embedded Guest Additions files.

- Inventory every shipped binary and source origin; generate a complete software bill of materials (SBOM).
- Collect license texts and copyright notices for the OS, custom changes, GTK/libadwaita ToolHub, Podman/Kali image references and Oracle component.
- Document how recipients can obtain corresponding source where licenses require it.
- Review name/logo similarity to third-party Spectre and Ghost Spectre products; assert no endorsement or affiliation.
- Publish accurate security-contact and vulnerability-reporting channels.
- Require explicit release-owner signoff before posting the ISO publicly.

## 17 Roadmap and change management

Planned improvements must be labeled plans, not current features: broader hardware matrix, signed checksums, SBOM and reproducible-source archive, tested upgrade policy, signed release notes, better shutdown handling for the Kali container, controlled USB/wireless hardware workflows, and support documentation.

Change rule: record issue, affected source, threat and data loss risk; test in staging; preserve source backup; increment version according to release policy; run full regression on a clean VM before replacing the download.

## 18 Administration quick reference

| Question | Command |
|----------|---------|
|----------|---------|

|                       |                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------|
|                       |                                                                                                |
| Kernel / OS           | <code>uname -r ; cat /etc/os-release</code>                                                    |
| Disk                  | <code>findmnt -no SOURCE,FSTYPE / ; df -h</code>                                               |
| Podman rootless       | <code>podman info --format "{{.Host.Security.Rootless}}"</code>                                |
| Rootless provisioning | <code>systemctl show spectre-rootless-podman-subids.service -p Result -p ExecMainStatus</code> |
| Toolbox status        | <code>podman ps -a --format "{{.Names}}   {{.Status}}"</code>                                  |
| Start toolbox         | <code>podman start spectre-kali-toolbox</code>                                                 |
| Installed Nmap        | <code>podman exec spectre-kali-toolbox nmap --version</code>                                   |
| VBox service          | <code>systemctl is-active spectre-vbox-display.service</code>                                  |
| VBox clients          | <code>pgrep -af "VBoxDRMClient   VBoxService"</code>                                           |
| Guest journal         | <code>journalctl -b -u spectre-vbox-display.service --no-pager</code>                          |

## 19 Glossary and upstream references

| Term          | Definition                                                                     |
|---------------|--------------------------------------------------------------------------------|
| Debian        | Linux distribution/project with stable, testing and unstable release tracks    |
| GNOME         | Desktop environment providing the graphical desktop experience                 |
| Kali Rolling  | Frequently updated Kali Linux package branch, accessed through container image |
| Podman        | Container engine; can operate under a non-root account                         |
| Rootless      | Container setup using user namespaces rather than host root                    |
| APT / dpkg    | Debian-family package management tools used on host and separately within Kali |
| ISO           | Installable optical-disc image used for VM boot or installation media          |
| SHA-256       | Cryptographic digest for integrity checking                                    |
| ToolHub       | Spectre's GUI for browsing and managing selected Kali packages                 |
| SubUID/SubGID | Allocated subordinate user/group-ID mapping ranges for                         |

|                 |                                                            |
|-----------------|------------------------------------------------------------|
|                 | user namespaces                                            |
| Guest Additions | VirtualBox guest utilities enabling host/guest integration |

Upstream references (accessed 11 October 2026):

Debian releases: <https://www.debian.org/releases/>

Debian derivatives: <https://www.debian.org/derivatives/>

Debian social contract: [https://www.debian.org/social\\_contract.en.html](https://www.debian.org/social_contract.en.html)

Kali intended audience: <https://www.kali.org/docs/introduction/should-i-use-kali-linux/>

Kali official containers: <https://www.kali.org/docs/containers/official-kalilinux-docker-images/>

Podman rootless: <https://docs.podman.io/en/latest/markdown/podman.1.html>

VirtualBox licensing: <https://docs.oracle.com/en/virtualization/virtualbox/7.0/licensing/licensing-licensing-info.html>

VirtualBox guest additions: <https://docs.oracle.com/en/virtualization/virtualbox/7.0/user/guestadditions.html>

Project website: <https://linux-spectre.myportfoliohub.online/>

Project website and claims are maintained separately from upstream references. Internal packages used in preparation: Spectre ToolHub GUI/runtime, RC8.5 rootless provisioner, RC8.6 VirtualBox overlay, public site FEATURE\_FACTS.md, developer-provided runtime logs and screenshots.