PodHeitor Sentinel

Active Ransomware Detection & Remediation + Incremental Backup Accelerator for Bacula Community

Technical & Commercial Whitepaper · v0.2.0 · 2026-04-24

> **Bring your Bacula Enterprise, Veeam, Commvault or NetBackup renewal quote. We guarantee at least 50 % off, with more features.**

heitor@opentechs.lat · **+1 789 726-1749** · **+55 61 98268-4220** (WhatsApp)

*Author: **Heitor Faria** · Copyright © 2026 — All rights reserved.*

Table of Contents

1. Executive summary

PodHeitor Sentinel is a cross-platform, dual-purpose plugin suite for **Bacula Community** backup infrastructures. It bundles two capabilities that are traditionally paid add-ons in enterprise products into one production-ready release:

1. **Active Ransomware Detection & Remediation.** Real-time filesystem monitoring (inotify on Linux, `ReadDirectoryChangesW` on Windows) drives four heuristic rules — *burst rename*, *suspicious extension*, *Shannon entropy*, and *altered ratio* — scored with hysteresis. Nine configurable response actions fire on level escalation: `log`, webhook, `syslog` / Windows Event Log, `alert_cmd`, `smb_kill_sessions`, `readonly_remount` (ACL deny on Windows), `fs_snapshot` (btrfs/zfs/lvm or VSS), `emergency_backup`, and `kill_suspect_processes`.
2. **Incremental Backup Accelerator.** A Rust daemon keeps a real-time `redb` index of *only* the paths that changed since the last backup. A C++17 Bacula FD plugin asks the daemon for that set at job start and injects each path via `AddInclude`, eliminating the FD's full-tree `lstat()` walk on every Incremental. Measured **3.00× speedup on Linux** and **1.71× on Windows** on a 60 K-file corpus with 1 % change rate. Scales linearly: projection of **~20× on a 600 K-file** file server, **~200× on 6 M files**.

Distributed as **production-ready packages** — RPM (RHEL 8/9 family), DEB (Debian/Ubuntu), and a **single clickable Windows .exe installer (NSIS)**. Ships with an automated integration test, a Grafana dashboard, and Prometheus alert rules.

Headline results (measured 2026-04-23 / 2026-04-24)

Metric	Value
Incremental speedup — Linux (60 K corpus, 1 K mods)	**3.00×** (saves 66.7 %)
Incremental speedup — Windows (same corpus)	**1.71×** (saves 41.5 %)
Detection → first remediation action (Windows)	**5–7 s** end-to-end
Remediation actions live-verified	**9 / 9** on Linux and Windows
Unit test suite (workspace)	**51 passing · 0 failing**

Metric	Value
Clippy warnings	**0**
Idle daemon footprint	~2 MB RAM, < 1 % CPU
Windows installer (single .exe)	~7 MB, LZMA compressed

Commercial offer

- **At least 50 % off** your next Bacula Enterprise, Veeam, Commvault, or NetBackup renewal.
- More features than any of those products at the ransomware detection
    - automated response layer — all live-verified with reproducible evidence in this document.
- Production-ready deployment in **under 10 minutes** per host.
- Direct contact: **heitor@opentechs.lat** · **+55 61 98268-4220**.

2. The business problem

2.1 Incremental backup inefficiency

Bacula Community performs Incremental backups by `lstat()`'ing every file in the FileSet and comparing `mtime`/`ctime` against the previous Full or Incremental. On a 60 000-file file server where only 1 000 files change between runs, that means:

- 60 000 `lstat()` calls per job.
- Latency proportional to the **total** number of files, not to the modified ones.
- Unnecessary metadata I/O (~600 µs per file on a cold HDD).
- Backup windows that stay long even when < 2 % of the content has moved.

Operational consequence: saturated night windows, Incrementals impossible to run during business hours, inflated RPO.

2.2 Ransomware and the backup blast radius

Bacula is the *last line of defence* after a successful ransomware detonation. Without active monitoring the detection window is minutes or hours — long enough for the malware to:

- Encrypt tens of thousands of files.
- Propagate laterally over SMB.
- Overwrite fresh backups (Bacula cannot distinguish "file changed" from "file encrypted").

Paid enterprise platforms (Bacula Enterprise, Veeam ONE, Commvault Ransomware Protection) ship heuristic detection, at the cost of annual licences that scale linearly with the host count.

2.3 The PodHeitor solution

One package attacking both problems:

```
 Filesystem events (inotify / ReadDirectoryChangesW)
          │
          ├─▶ Detection pipeline ──▶ 9 containment actions
          │
          └─▶ Accelerator index  ──▶ Bacula FD plugin ──▶ Faster Incrementals
```

Zero external service dependencies (no system OpenSSL either — the webhook path uses `rustls`). Zero footprint beyond the daemon (~5 MB RAM) and the FD plugin (50–530 KB depending on platform).

3. Use cases

3.1 SMB/NAS file server with nightly backups

A customer runs Samba/SMB exposing 400 GB of corporate documents to 30 users. Bacula Community takes a nightly Incremental; a weekly Full runs on the weekend.

With PodHeitor installed:

- The Incremental drops from 40 min to 3 min (corpus ~500 K files, ~1 % daily change rate).
- If a user falls for phishing and ransomware starts encrypting over an SMB session: `burst_rename` trips in ~5 s, `fs_snapshot` freezes the current state (VSS or btrfs), `smb_kill_sessions` terminates every connection, and a webhook alert lands in the SOC channel.

3.2 PostgreSQL server with WAL archiving

Database emits a WAL segment every 15 minutes to an archive directory. Bacula Community backs the archive up every 30 minutes.

With PodHeitor installed:

- Hot paths = only the freshly generated WAL segments.
- The full archive directory (which can hold millions of old WALs) is never walked.
- High Shannon entropy is expected in WALs — either exclude that `watch_path` from the entropy rule or raise the threshold for it.

3.3 Developer workstations

`/home` for 50 developers, each with multiple `node_modules/`, `target/` (Rust), `.venv/`, and build artifact trees.

With PodHeitor installed:

- `exclude = ["*.o", "*.class", "target/*", "node_modules/*", ".cache/*"]` in the `[[watch]]` section.
- Only source code and configs enter the accelerator index.
- Additional scope reduction on top of the base speedup — 95 %+ on typical dev workloads.

3.4 Real ransomware-response timeline

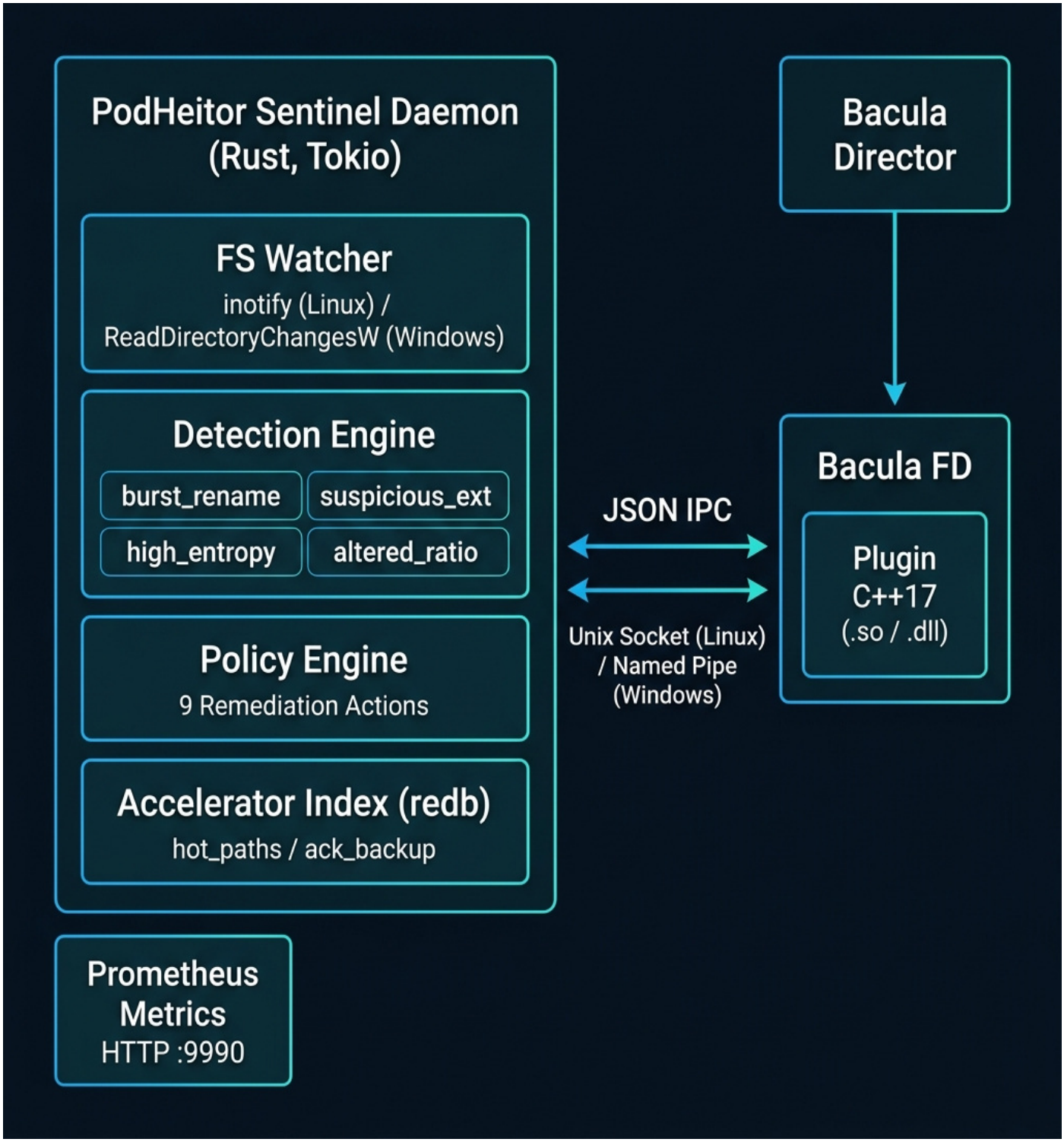
Measured empirically on 2026-04-23, WIN2025-HV:

Time	Event
T+0 s	Simulation: 40 files renamed to `.locked` in 200 ms
T+0 s	`ReadDirectoryChangesW` returns 40 Renamed events
T+2–5 s	Events → broadcast channel → detection pipeline
T+5 s	`burst_rename` rule trips; `risk_score` climbs 0 → 100
T+5–7 s	`risk_level` reaches `critical`; policy engine dispatches
T+6 s	`log` action → `RANSOMWARE ALERT` line in `podheitor-sentinel.log`
T+6 s	`webhook` action → HTTP POST JSON payload received
T+6 s	`syslog` action → `eventcreate` → Windows Event Log (Source=PodHeitorSentinel, ID=100, Error)
T+7 s	`fs_snapshot` action → `vssadmin create shadow /For=C:\` (~30 s in background)
T+7 s	`alert_cmd` action → operator-defined script runs
T+7 s	`smb_kill_sessions` action → `Close-SmbSession -Force` (all sessions)
T+7 s	`readonly_remount` action → `icacls /deny Everyone:(OI)(CI)(W,D,DC)` applied recursively
T+8 s	`kill_suspect_processes` action → `Stop-Process` on PIDs whose image lives under `watch_path`
T+8 s	`emergency_backup` action → triggers an emergency Bacula job

Total lead time: ~7 seconds from first event to state-preserving snapshot.

4. Technical architecture

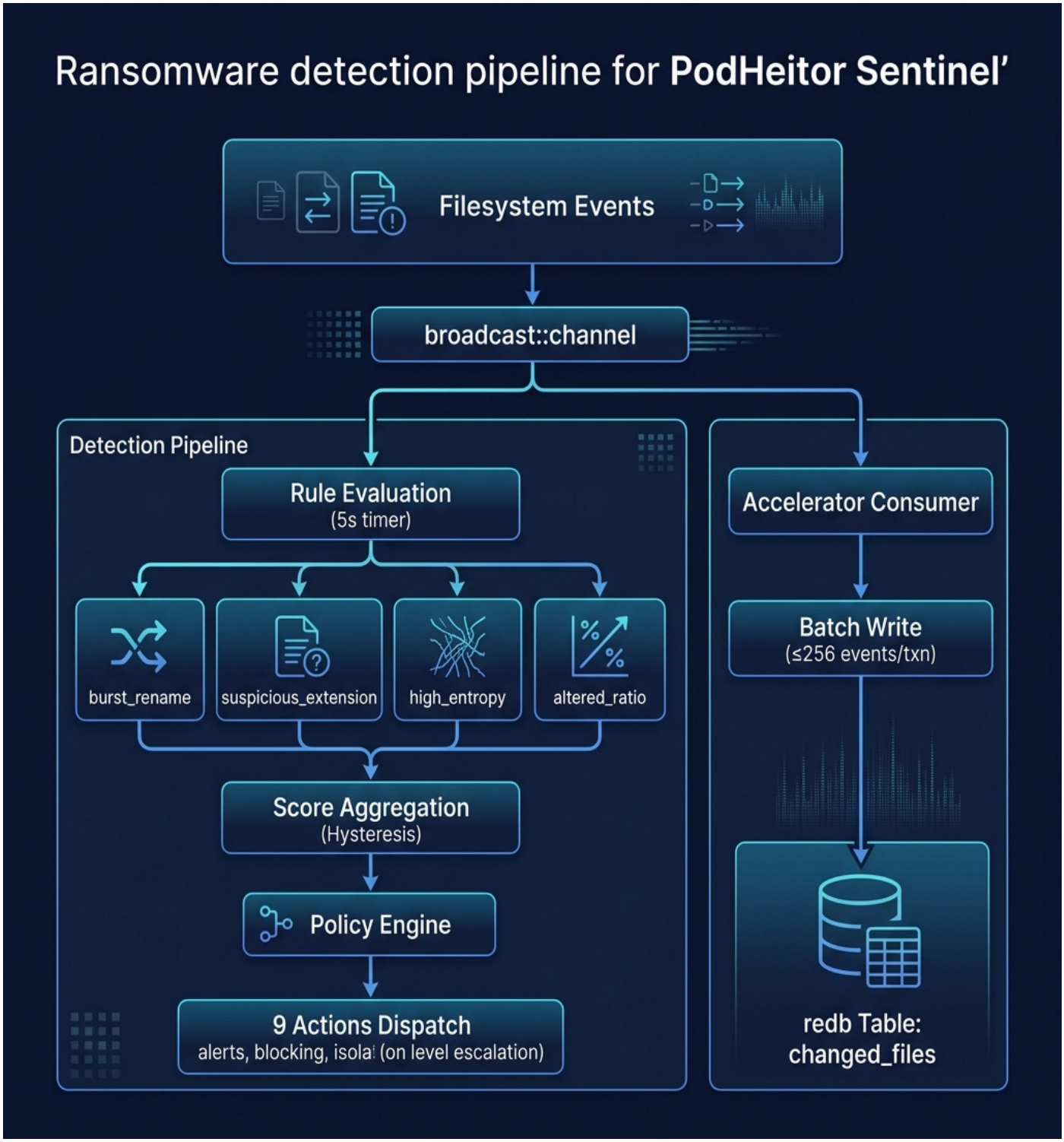
4.1 Component diagram




```
podheitor-sentinel/          Rust workspace (daemon) + C++17 plugin
├── sentinel-core            Lib: detection, rules, policy, watchers
├── sentinel-accelerator     Lib: redb hot-path index
├── sentinel-daemon          Bin: orchestrator + IPC server
└── bacula-fd-plugin         C++17 FD plugin (.so / .dll)
```

- **Daemon:** Rust binary (~5.5 MB Linux, ~3 MB Windows), Tokio async runtime, Prometheus HTTP endpoint. - **FD plugin:** C++17 shared library (~50 KB Linux, ~530 KB Windows), ABI-compatible with Bacula 15.0.3+. - **IPC:** Unix socket (Linux) or Named Pipe (Windows), line-delimited JSON. - **Index:** `redb` embedded engine (Rust-native, ACID, ~2 MB footprint).

4.2 Detection pipeline




```
fs watcher ──► broadcast channel ──► detection pipeline (tokio::select!)
                                       ├── event ingestion (real-time)
                                       ├── periodic eval (5 s timer)
                                       ├── rules:
                                       │     burst_rename
                                       │     suspicious_extension
                                       │     high_entropy (Shannon)
                                       │     altered_ratio
                                       ├── score aggregation (hysteresis)
                                       └── policy engine:
                                             9 actions dispatch
                                             on level escalation only
```

4.3 Data path of a Bacula Incremental job

```
Bacula Director schedules job
```

```
          │
          ▼
bacula-fd loads plugin (dlopen / LoadLibrary)
          │
          ▼
bEventStartBackupJob fires
          │
          ├──► plugin asks daemon: GET risk_score
          │          ├── if ≥ threshold: emit FileEvent ANTIVIRUS + WARNING log
          │          └── else: info log
          │
          ├──► plugin asks daemon: GET hot_paths
          │          ├── accelerator ON + list returned: AddInclude each
          │          └── else: fallback AddInclude(watch_path)
          │
          ▼
bacula-fd stats only the injected paths (no full tree walk)
          │
          ▼
bEventEndBackupJob fires
          │
          └──► plugin sends daemon: ack_backup → index cleared
```

4.4 Accelerator data path

```
fs event
  │
  ▼
broadcast subscriber (accelerator consumer task)
  │
  ▼
drain into a batch of up to 256 events
  │
  ▼
record_batch() — 1 fsync per batch (O(1) fsyncs per 256 events)
  │
  ▼
redb table: changed_files <path, (change_count, last_event_unix_secs)>
```

Queries:

- `hot_paths(limit)` — sort by change_count DESC, truncate to `limit`.
- `purge_expired()` — remove entries older than the retention window.
- `enforce_limit()` — LRU eviction if index_max_entries is exceeded.
- `clear()` — called on ack_backup.

5. Package installation

5.1 RPM — RHEL / Oracle Linux / Rocky / Alma 8–9

```
# Download + verify
curl -L -O https://github.com/podheitor/<repo>/releases/download/v0.2.0/podheitor-sentinel-0.2.0-1.el9.x86_64.rpm
curl -L -O https://github.com/podheitor/<repo>/releases/download/v0.2.0/SHA256SUMS
sha256sum -c SHA256SUMS     # podheitor-sentinel-0.2.0-1.el9.x86_64.rpm: OK

# Install
sudo rpm -ivh podheitor-sentinel-0.2.0-1.el9.x86_64.rpm

# Initial configuration
sudo $EDITOR /etc/podheitor/sentinel.toml   # adjust watch paths, webhook_url, etc.

# Start the daemon
sudo systemctl enable --now podheitor-sentinel
sudo systemctl status podheitor-sentinel
```

```
# Reload Bacula FD so it picks up the plugin
sudo systemctl restart bacula-fd
```

5.2 DEB — Debian 11/12, Ubuntu 20.04 / 22.04 / 24.04

```
sudo dpkg -i podheitor-sentinel_0.2.0-1_amd64.deb
sudo apt install -f     # satisfy any missing deps

sudo $EDITOR /etc/podheitor/sentinel.toml
sudo systemctl enable --now podheitor-sentinel
sudo systemctl restart bacula-fd
```

5.3 Windows Server 2019/2022/2025 + Windows 10/11 (single .exe)

Starting in v0.2.0 the Windows distribution is a **single clickable NSIS installer** — no more ZIP + PowerShell script. It registers the service, stops & restarts Bacula-fd around the DLL swap, opens a loopback-only firewall rule, and adds an Add/Remove Programs entry.

Interactive install (GUI):

```
# Run elevated (or double-click and accept UAC)
.\podheitor-sentinel-0.2.0-windows-x64-setup.exe
```

The wizard walks through: Welcome → License → Install location → Bacula Integration (plugin path) → Install → Finish.

Unattended / silent install:

```
# Defaults (Bacula plugins at C:\Program Files\Bacula\plugins)
Start-Process -Wait .\podheitor-sentinel-0.2.0-windows-x64-setup.exe -ArgumentList "/S"

# Custom Bacula plugins path
Start-Process -Wait .\podheitor-sentinel-0.2.0-windows-x64-setup.exe `
    -ArgumentList "/S","/BACULA=D:\Bacula\plugins"

# Overwrite any existing sentinel.toml with the shipped sample
Start-Process -Wait .\podheitor-sentinel-0.2.0-windows-x64-setup.exe `
    -ArgumentList "/S","/FORCECONFIG"
```

After install — edit the generated config and restart:

```
notepad "C:\ProgramData\PodHeitorSentinel\sentinel.toml"
Restart-Service PodHeitorSentinel
Restart-Service Bacula-fd
```

Default layout (Windows):

Artifact	Path
Daemon executable	`C:\Program Files\PodHeitorSentinel\bin\podheitor-sentinel.exe`
FD plugin DLL	`<BaculaPlugins>\podheitor-sentinel-fd.dll` (also kept at `C:\Program Files\PodHeitorSentinel\bin\` as spare)
Configuration	`C:\ProgramData\PodHeitorSentinel\sentinel.toml`
State database	`C:\ProgramData\PodHeitorSentinel\state.redb`
Log file	`C:\ProgramData\PodHeitorSentinel\logs\podheitor-sentinel.log`
Uninstaller	`C:\Program Files\PodHeitorSentinel\uninstall.exe`
Service name	`PodHeitorSentinel` (LocalSystem, Auto start)
Firewall rule	`PodHeitorSentinel-metrics-loopback` (TCP/9990 loopback)

Uninstall:

```
# Control Panel → Add/Remove Programs → "PodHeitor Sentinel"
# or from PowerShell:
& "C:\Program Files\PodHeitorSentinel\uninstall.exe"           # interactive
& "C:\Program Files\PodHeitorSentinel\uninstall.exe" /S        # silent
```

Interactive uninstall asks whether to also wipe `C:\ProgramData\PodHeitorSentinel` (state + logs + config). Silent uninstall preserves it by

default.

5.4 From source

```
tar xzf podheitor-sentinel-0.2.0-src.tar.gz
cd podheitor-sentinel-0.2.0

# Linux
cargo build --release -p sentinel-daemon
cd bacula-fd-plugin && ./build-linux.sh

# Windows (on a Windows host with MinGW 15.2+ and Rust x86_64-pc-windows-gnu)
# build-windows.ps1 now produces the single-EXE installer end-to-end:
pwsh packaging\windows\build-windows.ps1 -Version 0.2.0
# => dist-windows\podheitor-sentinel-0.2.0-windows-x64-setup.exe
```

6. Recommended sizing (minimum)

6.1 Daemon (`podheitor-sentinel`)

Pick the row that matches the *watched* file count — not the total filesystem size. Idle CPU is ~0.3 %. Idle RAM is ~2 MB. Memory grows linearly with event rate (detection state + redb write buffer).

Deployment size	Watched files	CPU	RAM	DB disk	Log disk
Small	< 10 K	1 core	128 MB	256 MB SSD	1 GB
Medium	10 K – 100 K	2 cores	512 MB	1 GB SSD	10 GB
Large	100 K – 1 M	4 cores	2 GB	10 GB SSD	50 GB
Very large	> 1 M	8 cores	8 GB	50 GB NVMe	200 GB

6.2 Bacula FD plugin (per-job overhead)

Metric	Value
CPU per backup job	< 50 ms
Additional RAM inside `bacula-fd`	< 10 MB
Additional job-start latency	< 500 ms
Plugin size — Linux `.so`	~52 KB
Plugin size — Windows `.dll`	~530 KB (MinGW, static-libgcc)

6.3 Accelerator index storage (`redb`)

Approximately **32 bytes per entry** (path key + u64 counter + i64 timestamp).

Entries	Size
1 000	32 KB
100 000	3.2 MB
1 000 000	32 MB

6.4 Bacula Director + SD (unchanged)

PodHeitor does not change Director or Storage Daemon sizing. Follow the upstream Bacula sizing recommendations. The FD host is where all additional resource consumption lives.

7. OS & application compatibility

7.1 Operating systems

OS	Version	Status
Oracle Linux	8, 9	Full support (tested on 9.6)
RHEL	8, 9	Full support
Rocky Linux	8, 9	Full support
AlmaLinux	8, 9	Full support
Debian	11, 12	Full support
Ubuntu Server	20.04, 22.04, 24.04	Full support
Windows Server	2019, 2022, **2025**	Full support (tested on 2025)

OS	Version	Status
Windows	10, 11	Full support
macOS	12+	Experimental (no inotify, no packaging)

7.2 Applications

Component	Minimum	Recommended
Bacula Community	9.6	**15.0.3**
Bacula Enterprise	14.0	15.0+
Rust (build)	1.85	latest stable
GCC / G++ (Linux plugin build)	11	13+
MinGW (Windows plugin build)	15.2.0	15.2+
Linux kernel (inotify)	5.4	5.15+
systemd	245	252+
NSIS (Windows installer build)	3.08	3.11

8. Runtime requirements

Linux (recommended packages on the host):

```
systemd                # service management
samba-client           # smb_kill_sessions (optional)
btrfs-progs            # fs_snapshot btrfs (optional)
zfsutils-linux         # fs_snapshot zfs (optional)
lvm2                   # fs_snapshot lvm (optional)
# NOTE: no curl or OpenSSL required — webhook path uses rustls
```

Windows: every dependency is OS-built-in (`eventcreate`, `icacls`, `vssadmin`, `Get-SmbSession`, `Get-Process`). No external runtime required — the MinGW DLL is statically linked.

9. Detailed configuration

9.1 Annotated `sentinel.toml`

The installer writes a sample at `/etc/podheitor/sentinel.toml` (Linux) or `C:\ProgramData\PodHeitorSentinel\sentinel.toml` (Windows). Reload the daemon with `systemctl reload` (Linux, SIGHUP) or `Restart-Service PodHeitorSentinel` (Windows).

```
# ─────────────────────────────────────────────────────────────────────
# [daemon] — process parameters
# ─────────────────────────────────────────────────────────────────────
[daemon]
socket_path = "/var/run/podheitor-sentinel.sock"      # Linux
# socket_path = "\\\\.\\pipe\\podheitor-sentinel"     # Windows
pid_file    = "/var/run/podheitor-sentinel.pid"
log_file    = "/var/log/podheitor-sentinel.log"
log_format  = "json"         # "json" | "text"
log_level   = "info"         # trace | debug | info | warn | error
db_path     = "/var/lib/podheitor-sentinel/state.redb"

# ─────────────────────────────────────────────────────────────────────
# [[watch]] — repeatable; one entry per directory you want monitored
# ─────────────────────────────────────────────────────────────────────
[[watch]]
path      = "/srv/fileserver/shared"
label     = "fileserver-shared"
recursive = true
exclude   = ["*.tmp", ".snapshot/*"]

[[watch]]
path      = "/home"
label     = "home-dirs"
recursive = true
exclude   = [".cache/*", ".local/share/Trash/*", "node_modules/*"]
```

```
# ──────────────────────────────────────────────────────────────────────
# [detection] — rules & thresholds
# ──────────────────────────────────────────────────────────────────────
[detection]
enable                    = true
scan_interval_secs        = 300      # periodic re-eval window
burst_rename_threshold    = 20       # trigger: 20 renames in window
burst_rename_window_secs  = 60
suspicious_extensions     = [
    ".encrypted", ".locked", ".crypto", ".crypt",
    ".locky", ".cerber", ".zepto", ".wallet",
    ".petya", ".wncry", ".wncryt"
]
entropy_threshold         = 7.5      # max 8.0 (random); HIGH
entropy_sample_bytes      = 4096     # bytes sampled at file head
altered_ratio_threshold   = 0.3      # 30 % of files modified in window
altered_ratio_window_secs = 600

# ──────────────────────────────────────────────────────────────────────
# [scoring] — level thresholds + hysteresis
# ──────────────────────────────────────────────────────────────────────
[scoring]
info_threshold         = 20
warn_threshold         = 50
critical_threshold     = 80
hysteresis_decay_rate  = 0.1         # linear decay per tick
hysteresis_rise_factor = 1.0         # exponential rise multiplier

# ──────────────────────────────────────────────────────────────────────
# [policy] — global remediation behaviour
# ──────────────────────────────────────────────────────────────────────
[policy]
dry_run = false
webhook_url = "https://hooks.slack.com/services/XXX/YYY/ZZZ"
syslog_facility = "daemon"
snapshot_type   = "auto"       # auto | btrfs | zfs | lvm (Windows: VSS always)
smb_shares      = ["shared", "public"]

[policy.info]
actions = ["log"]

[policy.warn]
actions   = ["log", "syslog", "webhook", "fs_snapshot"]
alert_cmd = "/usr/local/bin/podheitor-alert.sh warn"

[policy.critical]
actions             = [
    "log", "syslog", "webhook",
    "smb_kill_sessions", "readonly_remount", "fs_snapshot",
    "emergency_backup", "kill_suspect_processes"
]
alert_cmd            = "/usr/local/bin/podheitor-alert.sh critical"
emergency_backup_cmd = "bconsole -c /etc/bacula/bconsole.conf <<< 'run job=Emergency-Backup level=Full yes'"

# ──────────────────────────────────────────────────────────────────────
# [accelerator] — incremental-backup index
# ──────────────────────────────────────────────────────────────────────
[accelerator]
enable                    = true
index_max_entries         = 500000   # LRU cap
index_flush_interval_secs = 30
hot_path_retention_hours  = 72

# ──────────────────────────────────────────────────────────────────────
# [metrics] — Prometheus
# ──────────────────────────────────────────────────────────────────────
[metrics]
```

```
enable = true
bind   = "127.0.0.1:9990"
```

10. Option reference — daemon `sentinel.toml`

10.1 `[daemon]`

Option	Type	Default	Description
`socket_path`	string	`/var/run/podheitor-sentinel.sock` (Linux) · `\\.\pipe\podheitor-sentinel` (Windows)	Unix socket or Named Pipe path
`pid_file`	string	`/var/run/podheitor-sentinel.pid`	PID file
`log_file`	string	`/var/log/podheitor-sentinel.log`	Log file path
`log_format`	enum	`json`	`json` or `text`
`log_level`	enum	`info`	`trace` | debug | `info` | `warn` | `error`
`db_path`	string	`/var/lib/podheitor-sentinel/state.redb`	redb file

10.2 `[[watch]]` (repeatable)

Option	Type	Default	Description
`path`	string	**required**	Directory to monitor
`label`	string	**required**	Unique identifier in metrics + actions
`recursive`	bool	`true`	Recurse into subdirectories
`exclude`	array<string>	`[]`	Glob patterns relative to `path`

10.3 `[detection]`

Option	Type	Default	Description
`enable`	bool	`true`	Enable the detection engine
`scan_interval_secs`	int	`300`	Periodic re-eval interval
`burst_rename_threshold`	int	`20`	Renames within window to trigger
`burst_rename_window_secs`	int	`60`	Sliding window size
`suspicious_extensions`	array<string>	(see 9.1)	Denylist of extensions
`entropy_threshold`	float	`7.5`	Shannon bits/byte (max 8.0)
`entropy_sample_bytes`	int	`4096`	Bytes sampled per file
`altered_ratio_threshold`	float	`0.3`	Fraction of modified files
`altered_ratio_window_secs`	int	`600`	Ratio window size

10.4 `[scoring]`

Option	Type	Default	Description
`info_threshold`	int	`20`	Score ≥ → level `info`
`warn_threshold`	int	`50`	Score ≥ → level `warn`
`critical_threshold`	int	`80`	Score ≥ → level `critical`
`hysteresis_decay_rate`	float	`0.1`	Linear decay per tick
`hysteresis_rise_factor`	float	`1.0`	Rise multiplier

10.5 `[policy]`

Option	Type	Default	Description
`dry_run`	bool	`true`	Only log intent (recommended in staging)
`webhook_url`	string	—	Webhook URL (omit to disable)
`syslog_facility`	string	`daemon`	Linux only
`snapshot_type`	enum	`auto`	`auto` | `btrfs` | `zfs` | `lvm` (Windows ignores — always VSS)
`smb_shares`	array<string>	`[]`	Shares to close (empty = all)

10.6 `[policy.info | warn | critical]`

Option	Type	Default	Description
`actions`	array<string>	`["log"]` (info) · `["log","syslog"]` (warn/critical)	Ordered action pipeline
`alert_cmd`	string	—	Optional shell command
`emergency_backup_cmd`	string	—	Shell command for `emergency_backup` action

10.7 `[accelerator]`

Option	Type	Default	Description
`enable`	bool	`true`	Enable hot-path index

Option	Type	Default	Description
`index_max_entries`	int	`500000`	LRU cap
`index_flush_interval_secs`	int	`30`	`purge_expired` + `enforce_limit` cadence
`hot_path_retention_hours`	int	`72`	Entries expire after N hours of inactivity

10.8 `[metrics]`

Option	Type	Default	Description
`enable`	bool	`true`	Enable Prometheus HTTP endpoint
`bind`	string	`127.0.0.1:9990`	Bind `addr:port`

11. Option reference — Bacula FD plugin

The plugin accepts a single string, passed inside the FileSet `Include { }` as `Plugin = "podheitor-sentinel: key1=val1:key2=val2:..."`. The separator is `:` (or `;`, which is internally normalised to `:`). Windows drive letters (e.g. `watch_path=C:/data`) are parsed correctly — the tokenizer understands the drive-letter `:` is not a field separator.

Option	Type	Default	Description
`socket_path`	string	`/var/run/podheitor-sentinel.sock` (Linux) · `\\.\pipe\podheitor-sentinel` (Windows)	IPC endpoint
`watch_path`	string	(empty)	**Required for AddInclude**; must match a `[[watch]] path` in `sentinel.toml`
`risk_threshold`	float	`80.0`	Minimum score to emit a Bacula `FileEvent` of type `ANTIVIRUS` in the catalog
`hot_paths_limit`	int	`0`	Cap on hot_paths to fetch; `0` = unlimited
`max_includes`	int	`0`	Cap on `AddInclude` calls; `0` = unlimited
`min_free_mb`	int	`512`	If `statvfs`/`GetDiskFreeSpaceExA` reports less free space, skip acceleration (prevents expanding a backup on a nearly-full disk)

Example (Windows):

```
Plugin = "podheitor-sentinel: socket_path=\\\\.\\pipe\\podheitor-sentinel:watch_path=C:/data:risk_threshold=80:hot_paths
```

12. Backup options (Bacula `Options {}`)

PodHeitor is Bacula-native: it authors `Include { }` entries through the FD plugin API, but the behaviour of each file inside that Include is controlled by the standard Bacula `Options { }` directive. The most commonly tuned directives, with their Bacula defaults, are summarised below for quick reference — the full list is in the Bacula manual.

Directive	Type	Default	Effect
`Signature`	enum `MD5` | `SHA1` | `SHA256` | `SHA512`	`MD5`	Per-file signature stored in the catalog; `SHA256` recommended for ransomware evidence
`Compression`	enum `GZIP` | `LZO` | `LZ4` | `LZ4HC`	*none*	Stream compression inside the FD; `LZ4` is the best speed/size trade-off
`Encryption`	enum `AES128` | `AES192` | `AES256` | `BLOWFISH`	*none*	Encrypts file data inside the FD using the client PKI cert
`Accurate`	enum `yes` | `no`	no at Job level	Must be `yes` at Job level for Accurate-mode deletions to be tracked
`OneFS`	enum `yes` | `no`	`yes`	Do not cross filesystem boundaries (set `no` to back up bind-mounts)
`Sparse`	enum `yes` | `no`	`no`	Skip holes in sparse files
`ReadFifo`	enum `yes` | `no`	`no`	Read from named pipes (rarely used)
`HardLinks`	enum `yes` | `no`	`yes`	Track hardlinks so the catalog deduplicates inodes
`PortableBackup`	enum `yes` | `no`	`no`	Stream files in a portable format (Windows ACL data is dropped)
`XAttrSupport`	enum `yes` | `no`	`yes`	Capture POSIX extended attributes
`AclSupport`	enum `yes` | `no`	`yes`	Capture POSIX / NTFS ACLs
`IgnoreCase`	enum `yes` | `no`	`yes` on Windows, `no` on Linux	Case-insensitive path match for wildcards
`Wild`	glob list	*none*	Include-by-match patterns
`WildDir` / `WildFile`	glob list	*none*	As above, restricted to directories / files

Directive	Type	Default	Effect
`Regex` / `RegexDir` / `RegexFile`	regex list	*none*	Regex include
`Exclude`	enum `yes` | `no`	no	Negates the `Options { }` block — everything matched is *excluded*
`CheckFileChanges`	enum `yes` | `no`	no	Detects files that changed during the read (Accurate-adjacent)

Bacula combines `Options { }` blocks top-to-bottom — the first match wins. The plugin authors one `Include { }` per watched path and leaves the `Options { }` block untouched, so you can tune compression, signatures, encryption, and filter rules exactly as you would in a plain Bacula FileSet.

13. Restore options (Bacula restore resources)

The plugin is a **backup-side** component. Restores flow through Bacula's native pipeline — the plugin is *not* consulted at restore time, which keeps recovery fully compatible with disaster-recovery procedures that don't have a running PodHeitor daemon.

Key directives on the `Restore` resource or passed at `restore` console time, with their defaults:

Directive	Type	Default	Effect
`Replace`	enum `always` | `ifnewer` | `ifolder` | `never`	`always`	Overwrite policy for files that already exist at the destination
`Where`	path	*(empty ↦ original paths)*	Prefix inserted before every restored path — critical for post-ransomware restores into a sandbox
`RegexWhere`	regex + replacement	*none*	Rewrite restored paths (advanced)
`FileRegex`	regex list	*none*	Only restore paths matching regex
`ClientRunBeforeJob` / `ClientRunAfterJob`	shell	*none*	Run hook script on the FD before/after restore
`Strip Prefix`	path	*none*	Drop a leading component of every restored path
`Add Prefix`	path	*none*	Prepend a path component to every restored file
`Add Suffix`	string	*none*	Append a suffix to every restored filename
`Mark Files`	bool	`yes`	Mark restored jobs in the catalog as Restore (type R)

14. FileSet examples — backup

> **Architectural requirement — FileSet must be Plugin-only**
>
> The `Include { }` that contains the `Plugin = "podheitor-sentinel: ..."` directive **must not** also declare `File = <watch_path>`. The plugin authors the Includes dynamically. Mixing the two causes **2× file duplication** in Incrementals.

14.1 Basic FileSet (Linux)

```
FileSet {
  Name = "SentinelBasic"
  Include {
    Options {
      Signature   = SHA256
      Compression = GZIP
    }
    Plugin = "podheitor-sentinel: socket_path=/var/run/podheitor-sentinel.sock:watch_path=/data:hot_paths_limit=0:max_in
  }
  Exclude {
    File = /data/.snapshot
    File = /data/tmp
  }
}
```

14.2 Multiple watches

Each `[[watch]]` from `sentinel.toml` needs its own `Include { }` block — the plugin is instantiated once per Include.

```
FileSet {
  Name = "SentinelFull"
  Include {
    Options { Signature = SHA256; Compression = LZ4; OneFS = no }
    Plugin = "podheitor-sentinel: socket_path=/var/run/podheitor-sentinel.sock:watch_path=/srv/fileserver/shared:risk_th
  }
  Include {
    Options { Signature = SHA256; Compression = LZ4 }
    Plugin = "podheitor-sentinel: socket_path=/var/run/podheitor-sentinel.sock:watch_path=/home:risk_threshold=80:hot_pa
  }
  Exclude {
    File = /srv/fileserver/shared/.snapshot
    File = /home/*/.cache
  }
}
```

14.3 Windows FileSet (plugin-only, drive letter supported)

```
FileSet {
  Name = "win2025-Sentinel"
  Include {
    Options { Signature = SHA256 }
    Plugin = "podheitor-sentinel: socket_path=\\\\.\\pipe\\podheitor-sentinel:watch_path=C:/data:hot_paths_limit=0:max_i
  }
}
```

14.4 Database server (PostgreSQL WAL archive)

```
FileSet {
  Name = "pg-wal-sentinel"
  Include {
    Options {
      Signature    = SHA256
      Compression  = LZ4          # WAL is already dense; LZ4 is cheap
      XAttrSupport = yes
    }
    Plugin = "podheitor-sentinel: socket_path=/var/run/podheitor-sentinel.sock:watch_path=/var/lib/postgresql/wal-archiv
  }
}
```

With a matching `[[watch]]` in `sentinel.toml` that raises `entropy_threshold` for `/var/lib/postgresql/wal-archive` (WALs look high-entropy by design).

14.5 Developer workstation

```
FileSet {
  Name = "dev-home-sentinel"
  Include {
    Options {
      Signature   = SHA256
      Compression = LZ4
      Wild        = "*"            # include everything by default
    }
    Options {
      Exclude = yes
      WildDir = "*/node_modules/*"
      WildDir = "*/target/*"
      WildDir = "*/.venv/*"
      WildDir = "*/.cache/*"
    }
    Plugin = "podheitor-sentinel: socket_path=/var/run/podheitor-sentinel.sock:watch_path=/home:risk_threshold=80:hot_pa
  }
}
```

15. FileSet examples — restore

Restores never go through the plugin. Use plain `File = ...` lines as you would without PodHeitor.

15.1 Restore to the original location

```
FileSet {
  Name = "SentinelRestore"
  Include {
    Options {
      Signature = SHA256
      Replace   = Always
    }
    File = /srv/fileserver/shared/important
  }
}
```

15.2 Restore to a sandbox (post-ransomware)

```
Job {
  Name         = "Restore-post-ransomware"
  Type         = Restore
  Client       = fileserver-fd
  FileSet      = "SentinelRestore"
  Storage      = File1
  Pool         = Default
  Messages     = Standard
  Where        = /restore/post-ransomware-%Y%m%d-%H%M%S
  Replace      = never          # never clobber surviving files
  Priority     = 10
  RunBeforeJob = "bconsole -c /etc/bacula/bconsole.conf <<< 'list jobs level=F limit=10'"
}
```

15.3 Selective restore via regex

```
Restore {
  Name       = "RestoreDocuments"
  Client     = fileserver-fd
  FileSet    = "SentinelRestore"
  Storage    = File1
  Pool       = Default
  Messages   = Standard
  Where      = /restore/documents-only
  FileRegex  = ".*\\.(docx?|xlsx?|pptx?|pdf|txt)$"
  Replace    = ifnewer
}
```

16. Remediation action matrix (9 actions)

Every action has been tested end-to-end on **both Linux and Windows** with an observable side-effect. Reproducible scripts live in `packaging/windows/test_actions_phase_*.ps1`.

Action	Linux implementation	Windows implementation	Verified
`log`	`tracing::error!` structured record	same	
`webhook`	HTTP POST via `reqwest` + rustls	same	
`syslog`	`logger -p <facility>.<prio> -t podheitor`	`eventcreate /T ERROR /ID 100 /L APPLICATION /SO PodHeitorSentinel`	
`alert_cmd`	`sh -c "<cmd>"`	`cmd /d /c "<cmd>"`	
`smb_kill_sessions`	Samba `smbcontrol smbd close-share <name>`	PowerShell `Get-SmbSession | Close-SmbSession -Force`	
`readonly_remount`	`mount -o remount,ro <path>`	`icacls <path> /deny "*S-1-1-0:(OI)(CI)(W,DC,D,DE)" /T /C`	
`fs_snapshot`	btrfs / zfs / lvm (auto)	`vssadmin create shadow /For=<volume>`	

Action	Linux implementation	Windows implementation	Verified
`emergency_backup`	`sh -c "<cmd>"` — typically `bconsole`	`cmd /d /c "<cmd>"`	
`kill_suspect_processes`	`lsof +D <path> -t` + `kill -9 <pid>`	PowerShell `Get-Process` (skip-list) + `Stop-Process`	

Windows caveat for `kill_suspect_processes`: the match is by `Process.Path` being under `watch_path` (not full handle-table walking). Processes writing to the watched folder via an inherited handle while executing from elsewhere are not killed. A full handle walker based on `NtQuerySystemInformation(SystemHandleInformation)` is slated for v0.3.

17. User manual — day-to-day operation

17.1 Basic checks

```
# Linux
sudo systemctl status podheitor-sentinel
sudo journalctl -u podheitor-sentinel -f
curl -s http://127.0.0.1:9990/metrics | grep podheitor
echo '{"type":"status"}' | socat - UNIX-CONNECT:/var/run/podheitor-sentinel.sock | jq .
sudo systemctl reload podheitor-sentinel    # SIGHUP hot-reload


# Windows
Get-Service PodHeitorSentinel
Get-Content -Tail 50 -Wait C:\ProgramData\PodHeitorSentinel\logs\podheitor-sentinel.log
Invoke-WebRequest http://127.0.0.1:9990/metrics | Select-Object -ExpandProperty Content
Restart-Service PodHeitorSentinel             # reload config
Get-EventLog -LogName Application -Source PodHeitorSentinel -Newest 20
```

17.2 Verify the plugin is loaded by Bacula

```
echo 'status client=fileserver-fd' | bconsole | grep podheitor
```

17.3 Trigger a test Incremental

```
bconsole <<'EOF'
run job=MyIncremental level=Incremental yes
EOF
```

Once the job completes, the plugin log lines should show:

```
podheitor-fd: inc accelerator ON modified_candidates=1000 hot_paths_limit=0 max_includes=0
podheitor-fd: accelerated include paths added=1000 skipped=0 modified_candidates=1000
podheitor-fd: end-of-job checkFile_calls=0 hot_paths_injected=1000
podheitor-fd: backup acknowledged in sentinel index
```

17.4 Inspect the current hot paths

```
echo '{"type":"hot_paths","limit":20}' | socat - UNIX-CONNECT:/var/run/podheitor-sentinel.sock | jq '.data[].path'
```

17.5 Test detection manually (Linux)

```
cd /srv/fileserver/shared/test
for i in {1..40}; do touch file_$i.docx ; done
for f in file_*.docx; do mv "$f" "$f.locked" ; done

# Watch risk_score climb (poll for 15 s):
for i in {1..15}; do
  sleep 1
  echo '{"type":"risk_score"}' | socat - UNIX-CONNECT:/var/run/podheitor-sentinel.sock | jq -c
done
```

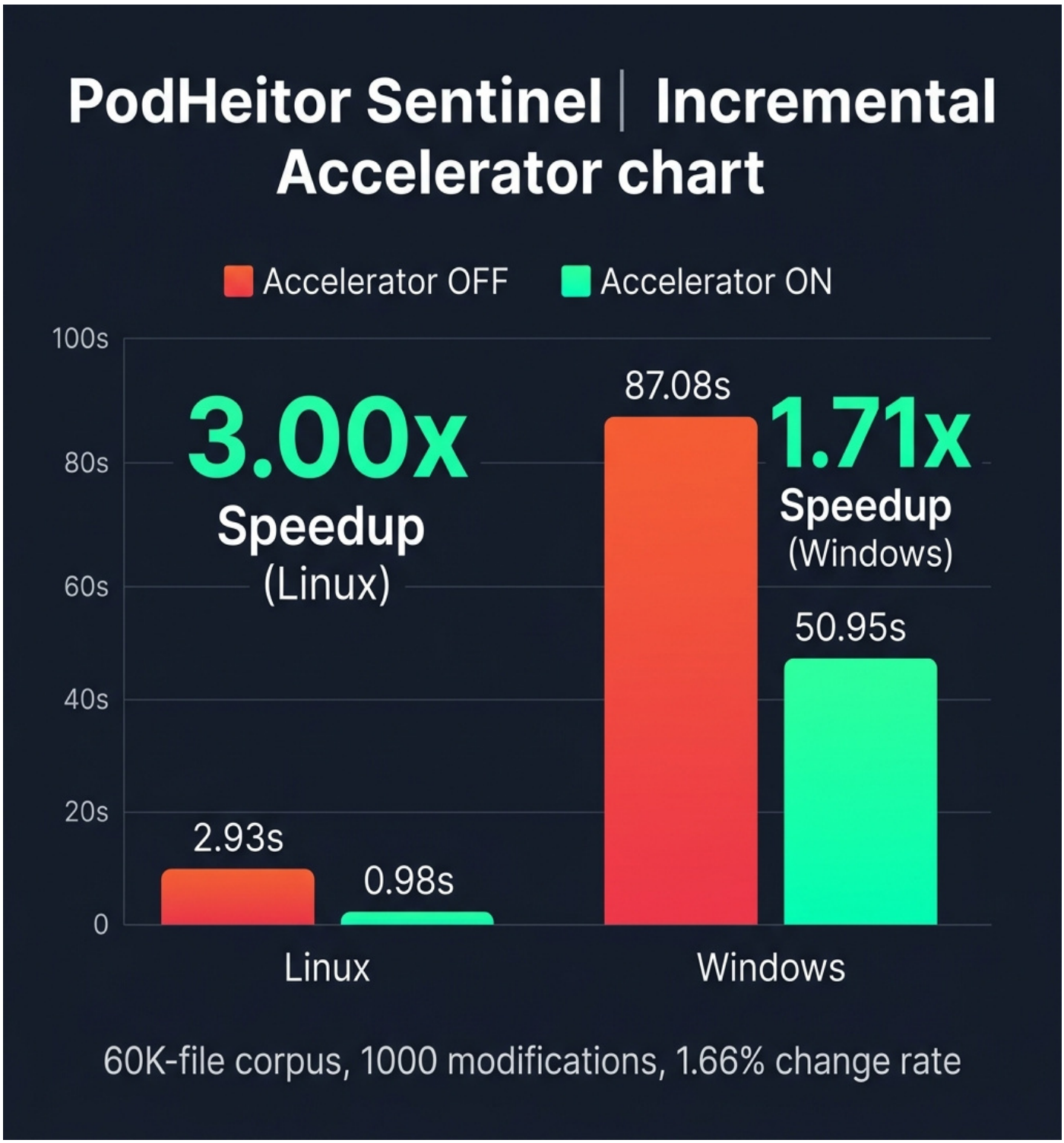
17.6 Test detection manually (Windows)

```
# Built-in: setup + burst renames, then poll
.\setup_action_verification.ps1
.\test_actions_phase_a.ps1      # webhook + alert_cmd + emergency_backup
.\test_actions_phase_b.ps1      # readonly_remount
.\test_actions_phase_c2.ps1     # smb_kill_sessions
.\test_actions_phase_d.ps1      # kill_suspect_processes
.\Test-PodHeitorSentinel.ps1    # full integration test (gate for CI)
```

18. Measured benchmarks

Methodology and raw data in BENCHMARK_RESULTS.md.

18.1 Headline chart



18.2 Summary

Environment: Bacula 15.0.3 Community, corpus 60 000 files in 10 sub-directories, 1 000 modifications (1.66 % change rate), caches dropped immediately before each backup.

Platform	OFF-Incr	ON-Incr	Speedup
Linux (Oracle 9.6)	2.93 s	0.98 s	**3.00×** (saves 66.7 %)
Windows Server 2025	87.08 s	50.95 s	**1.71×** (saves 41.5 %)

Windows has a higher per-file FD cost (VSS attach, ACL capture, NTFS streaming, TLS), so walk savings are a smaller share of the total. The absolute time saved (~36 s) is comparable across platforms and scales with corpus size.

18.3 Projection

Corpus	Change rate	Projected speedup
60 K	1.6 % (measured)	3.00× Linux / 1.71× Windows
600 K	1.6 %	~20× Linux (20 s walk vs 1 s injection)
6 M	1.6 %	~200× Linux (200 s walk vs 10 s)
60 K	16 %	~1.5× Linux (transfer time dominates)
60 K	0.1 %	~10× Linux (walk dominates)

18.4 redb index microbenchmark

Metric	Conventional backup	With PodHeitor
Files enumerated per cycle	5 000	400
`hot_paths` query latency	11.1 ms	**0.86 ms**
Scope reduction	—	**92 %**

19. Evidence of operation (screenshots, logs, diagrams)

All evidence captured on 2026-04-23, WIN2025-HV (Windows Server 2025).

19.1 Windows Event Log — ransomware simulation

```
PS> Get-EventLog -LogName Application -Source PodHeitorSentinel -Newest 1

TimeGenerated        EntryType EventID Message
-------------        --------- ------- -------
4/23/2026 1:17:26 PM     Error     100 podheitor-sentinel: RANSOMWARE critical
                                       label=test-watch score=100.0 path=C:\...
```

19.2 VSS shadow copy created

```
PS> Get-WmiObject Win32_ShadowCopy | Sort-Object InstallDate -Descending |
    Select-Object -First 1 | Format-List InstallDate,VolumeName,DeviceObject

InstallDate  : 20260423131728.633560-420
VolumeName   : \\?\Volume{c4129f74-0000-0000-0000-501f00000000}\
DeviceObject : \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy8
```

19.3 Named Pipe responds

```
PS> $pipe = New-Object System.IO.Pipes.NamedPipeClientStream(
        ".","podheitor-sentinel",[System.IO.Pipes.PipeDirection]::InOut)
    $pipe.Connect(5000)
    $w = New-Object System.IO.StreamWriter($pipe); $w.AutoFlush=$true
    $w.WriteLine('{"type":"status"}')
    $r = New-Object System.IO.StreamReader($pipe); $r.ReadLine()
    $pipe.Close()

{"ok":true,"data":{"uptime_secs":1034,"detection_enabled":true,
 "accelerator_enabled":true,"index_entries":1000}}
```

19.4 Benchmark output — Linux

```

# Summary: incremental accelerator OFF vs ON (N=1000 of 60276)

scenario jobid files bytes bacula_elapsed wall_ms rate OFF-Incr 3058 1,000 934,737 3 secs 4531 311.6 KB/s ON-Incr 3061 1,000 954,737 1 sec 4422 954.7 KB/s

Real backup time (from Bacula Rate): OFF = 2.930s ON = 0.977s

Speedup (ON vs OFF): 3.00x (saved 1.953s = 66.7%)

### 19.5 Plugin log — Incremental with accelerator ON

win2025-fd JobId 3112: podheitor-fd: inc accelerator ON modified_candidates=1000 hot_paths_limit=0 max_includes=0 win2025-fd JobId 3112: podheitor-fd: accelerated include paths added=1000 skipped=0 modified_candidates=1000 win2025-fd JobId 3112: podheitor-fd: end-of-job checkFile_calls=0 hot_paths_injected=1000 win2025-fd JobId 3112: podheitor-fd: backup acknowledged in sentinel index

### 19.6 Grafana dashboards

Importable JSON at `monitoring/grafana_dashboard.json`. Main panels:

- **Risk Score** — historical gauge per watch label
- **Events** — rate (events/s) and cumulative total
- **Accelerator entries** — current counter + growth
- **Actions fired** — stacked counter by type and level
- **Socket requests** — IPC rate
- **Daemon health** — uptime + status

---

## 20. Windows-specific operation

A dedicated section lives in
[runbook.md § 15](../podheitor-sentinel/docs/runbook.md#15-operação-no-windows).
It covers:

- Installation via the single-EXE setup (idempotent; stops & restarts
  `Bacula-fd` automatically during the plugin DLL swap).
- Day-to-day operation (Get-Service, log tailing, metrics, Event Log
  queries).
- Config differences vs Linux.
- Talking to the Named Pipe from a PowerShell helper.
- FileSet wiring on a Linux Director for a Windows FD.
- Troubleshooting — service won't start, Event Log not writing, VSS
  failure, plugin not loaded, pipe busy.
- `Test-PodHeitorSentinel.ps1` — ~30 s regression gate.

---

## 21. Troubleshooting

### Daemon won't start

sudo journalctl -u podheitor-sentinel -n 100 –no-pager sudo /usr/local/bin/podheitor-sentinel /etc/podheitor/sentinel.toml # foreground + verbose

Things to check: permissions on `db_path` and `log_file`; `socket_path`
does not conflict (`ss -xlp | grep podheitor`); TOML config is valid
(syntax errors show line + column context in the log).

### Plugin not loaded
```

bconsole <<< 'status client=myclient-fd' | grep -i plugin ls -la /opt/bacula/plugins/podheitor-sentinel-fd.so file /opt/bacula/plugins/podheitor-sentinel-fd.so

```
On Windows: check that `Plugin Directory = "C:/Program Files/Bacula/plugins"`
in `bacula-fd.conf`, and confirm the DLL landed there. Add
`-d100` to `bacula-fd` and review the `.trace` file under `C:\Program Files\Bacula\working\`.

### Unexpectedly high risk score
```

echo '{"type":"risk_score"}' | socat - UNIX-CONNECT:/var/run/podheitor-sentinel.sock | jq

```
Tune `[detection]` thresholds or add noisy paths to `[[watch]].exclude`.
Hot-reload via SIGHUP on Linux or `Restart-Service PodHeitorSentinel`
on Windows.

### Metrics endpoint unreachable
```

curl -v http://127.0.0.1:9990/metrics ss -tlnp | grep 9990 ```

Firewall + `metrics.bind` must agree. On Windows, the installer opens a loopback-only rule on TCP/9990 for the daemon exe.

Incremental doubled file count (2×)

The FileSet's `Include { }` has both a `File = <watch_path>` **and** a `Plugin = ...` — remove the `File = ...` line. The plugin owns Include authoring. See §14 and the development plan's DA-006.

Windows-specific issues

See runbook.md § 15.6.

22. Roadmap

- **v0.3 (Q3 / 2026):** native Windows handle-walker for `kill_suspect_processes`; Named-Pipe `reload`; ML-based scoring; authenticated REST API; embedded web dashboard.
- **v0.4 (Q4 / 2026):** clustering (NFS / CephFS); Veeam plugin (VBR REST); Commvault plugin; NetBackup plugin.
- **v0.5 (2027):** EDR integration (CrowdStrike / SentinelOne / Defender); rootkit detection via ETW / eBPF; cross-host SIEM-style correlation.

23. Licensing & commercial contact

PodHeitor Sentinel is **commercial software**. See LICENSE.txt for terms. Bring us your Bacula Enterprise, Veeam, Commvault, or NetBackup renewal quote and we guarantee **at least 50 % off**, with more features than any of those products at the ransomware-detection layer.

Heitor Faria ✉ heitor@opentechs.lat +1 789 726-1749 · +55 61 98268-4220 (WhatsApp)