WESTGATE DATA SCIENCE / TECHNICAL WHITEPAPER

Churchill

Fail-Closed Runtime Integrity for Production Workloads

SECTION 01

The Problem

Modern attackers no longer stop at the perimeter. They reach inside the running process. They debug it, read its memory, redirect its calls, and substitute its components, while the application appears, from a distance, to be operating normally. The on-disk binary is unchanged. Antivirus and file-integrity monitors see nothing unusual. The running process simply isn't the program that was deployed.

For workloads handling financial settlement, regulated data, mission-critical control, or sovereign infrastructure, that quiet substitution is the worst-case outcome. The boundary that needs defending isn't the disk or the network. It is the running process itself. That is where Churchill operates.

SECTION 02

The Outcome

Churchill provides continuous, fail-closed runtime integrity for production applications. While the application runs, Churchill validates that its code, its data, and its execution environment match what was deployed. Any tamper attempt is detected, contained, and answered automatically. The protected workload either runs untouched or stops cleanly. There is no quiet middle state.

This is not theoretical. Across an independent third-party adversarial evaluation, Churchill held the boundary against active attack, including periods in which the adversary was granted full administrator privilege on the host. Operators saw every event in real time, with a cryptographically chained evidence record available for audit and incident review.

INDEPENDENT ADVERSARIAL EVALUATION RESULTS

100%	Detection rate of adversarial tamper attempts
0	Protected assets modified; zero data exfiltration events
Under 1 second	Auto-recovery from tamper, no operator involvement

11.3 hours / 29 sessions	Of live enforcement, including 1.5 hours under root-level adversary

SECTION 03

How It Works

Churchill operates in three cooperating trust domains. Each watches the others, with no single one trusted alone.

INSIDE THE APPLICATION	**Integrity sentinel** Continuous self-attestation. Verifies code, data layout, privilege context, and operating environment while the application runs.
ON THE HOST, OUTSIDE THE PROCESS	**Privileged sentinel** Watches the protected application from a separate trust domain. Validates binary execution before it begins. Executes immediate, irrevocable termination on violation.
OFF-HOST, CONTROL PLANE	**Remote control plane** Receives mutually authenticated heartbeats and integrity telemetry. Locks the first baseline and refuses subsequent drift. Operator dashboard for visibility, evidence review, and administration.

Three trust domains: an integrity sentinel inside the protected application, an external sentinel on the same host, and an off-host control plane.

Inside the protected application. A library embedded in the application performs continuous self-attestation: its code, data layout, privilege context, and operating environment are verified continuously while it runs. The application and the integrity sentinel exchange ongoing liveness proofs, so neither can be silently disabled while the other continues.

On the host, outside the protected process. A privileged sentinel watches the protected application from a separate trust domain. It validates binary execution before it begins, monitors process lifecycle in real time, and executes immediate, irrevocable termination through the operating system the moment a violation is detected. Termination persists across reboots, so a tampered application cannot be silently restarted.

Off-host, in the control plane. A remote service receives mutually authenticated heartbeats and integrity telemetry from every protected host. It records the first integrity baseline it sees and refuses any drift afterward, making code rewriting in memory detectable from outside the host the attacker controls. An operator dashboard provides real-time visibility, evidence review, and administrative control.

After bootstrap, Churchill's runtime artifacts no longer exist on disk. Payloads are cryptographically signed, delivered through authenticated channels, and unpacked off-disk for the life of the boot. On host platforms that support modern operating-system security extensions, Churchill installs additional enforcement at that layer; on others, the same conditions are caught upstream and answered the same way.

SECTION 04

What This Means For You

Churchill is built to fit into a CISO's existing operational model.

— **Deployment is one binary and one token per host.** Operators don't manage a fleet of agents, daemons, or shared secrets. They issue an enrollment token and Churchill takes it from there.

— **Tamper events flow into existing incident response.** A pluggable enforcement chain feeds Churchill's signals into your alerting, isolation, and automation stacks alongside its own immediate termination. The dashboard tells your team what happened; your runbooks tell them what to do next.

— **Evidence is audit-ready.** Every event is cryptographically chained and archived. Auditors and incident responders can verify exactly which attempts were detected, which were contained, and how long any compromise window remained open.

— **Platform fit is broad.** Churchill runs on all Linux platforms, with a Windows edition in development.

SECTION 05

Where Churchill Fits

Churchill defends the running process against userspace adversaries who reach the host. It complements rather than replaces the rest of your defense in depth. Pair Churchill with Secure Boot, kernel lockdown, and hardware roots of trust to address kernel-rootkit threats; with IOMMU configuration, encrypted memory, and tamper-evident hardware to address physical-access threats; and with build-pipeline integrity controls to address supply-chain compromise of the toolchain that produces Churchill's runtime payloads. Each layer covers what the others cannot.

SECTION 06

Conclusion

For workloads where compromise has consequences, the question isn't whether to detect tampering. It is whether the answer arrives in time. Churchill's three-domain architecture, off-disk runtime, and locked baselines give you a containment boundary that holds even when the adversary already controls the machine. The platform's verified record speaks to the rest.

To request a technical briefing or schedule an evaluation, contact the WGDS engineering team at **info@wgds.tech**.

ABOUT WESTGATE DATA SCIENCE

*WestGate Data Science is a security and data-engineering firm building infrastructure for high-assurance workloads. Churchill is part of WGDS's runtime-integrity portfolio, alongside **Digital Detectives**, the company's intelligence layer built on the same doctrine and validated on IBM LinuxONE. Each Detective is designed for a different use case; the supply-chain Detective is the first to launch. WGDS supports customers with regulated, mission-critical, and sovereign workloads on all Linux platforms.*