Nevin Shine

Systems Security Researcher

German Citizen | Nürnberg, Germany (May–Aug 2026)
Currently in India (B.Tech Semester 4)
+49 157 54256832 | nevinshine05@outlook.com | github.com/nevinshine

Research Focus

Area: Semantic-to-Execution gap: enforcing correctness across compiler, kernel, and firmware layers
Stack: UEFI SMM, eBPF/LSM, LLVM/inkwell, AMD-V/KVM, Z3 SMT, Rust, C/C++, XDP

Research Projects

Sentinel Stack — Cross-Layer Zero-Trust Architecture 2025–2026 (Active)

- Architected and developed a unified, deterministic 8-pillar defense quadrant enforcing security from Ring -2 (Firmware) to Layer 7 (AI semantics).
- The following components represent the core interconnected subsystems of this monorepo architecture:

Sentinel SMM — Ring -2 Firmware Supervisor 2026

- Operates in System Management Mode (SMM) to sandbox third-party System Management Interrupts (SMI)
- Enforces strict hardware-level "Default Deny" policies for privileged hardware access (MSRs, I/O ports)
- Utilizes an ultra-low latency $\mathcal{O}(1)$ bitmap engine to prevent system jitter and OS starvation
- Neutralizes firmware-level rootkits (e.g., SinkClose) before OS and hypervisor initialization

Sentinel-CC — Compiler-Kernel Execution Integrity 2026

- BFS-based call graph traversal from binary imports reducing syscall attack surface by 81.6%
- Custom LLVM Module Pass extracting syscall provenance and detecting obfuscated patterns
- Cryptographic integrity chain using SHA-256 and Ed25519 verified via Linux kernel keyring
- 25 eBPF hooks enforcing syscall provenance, control-flow integrity, and runtime constraints

Sentinel-KV — Formal Verification for Legacy Drivers 2026

- Pure LLVM C API toolchain for analyzing and formally verifying legacy C driver memory safety
- Employs SMT-backed analysis (Z3) to track stack-spill pointers and restore provenance
- Enforces strict policies against inline assembly and DMA memory manipulation
- Automates sequential analysis of hardware interrupt handlers for concurrency safety

Telos Lang — Kernel-Aware Systems Programming Language 2026

- Dual-target compilation producing both ELF binaries and eBPF-LSM sandbox enforcement
- LLVM-based pipeline using inkwell for simultaneous BPF and x86 code generation
- Integrated Z3 SMT solver proving memory safety, bounds correctness, and invariants
- Compilation aborts on formal verification failure with counterexample generation

Telos Runtime — AI Kernel Containment 2026

- Kernel-level enforcement of AI agent intent against prompt injection and data exfiltration
- Cross-vector taint tracking: sensitive file reads automatically block network access
- Multi-layer domain intelligence pipeline with minimal LLM involvement
- Benchmarked at 100% attack prevention with zero false positives

Hyperion XDP — Wire-Speed Network Enforcement 2025–2026

- XDP-based firewall performing packet drops before kernel `sk_buff` allocation
- Engineered a zero-latency Unix Domain Socket (UDS) IPC bridge mapping Layer 7 AI intelligence directly to Layer 2 enforcement
- Implemented $\mathcal{O}(1)$ LRU Hash Maps to autonomously blacklist IPs at wire-speed upon detecting critical threats
- Zero-copy telemetry and dynamic reconfiguration without network downtime

Sentinel VMI — Hypervisor-Based Introspection 2025–2026

- Hardware-enforced protection using AMD-V Nested Page Tables
- Prevents rootkit modification of `sys_call_table` via #NPF fault trapping
- Engineered the 'Drawbridge' protocol utilizing Unix Domain Sockets for dynamic cryptographic nonce exchange
- Reconstructs process state via direct guest memory introspection (BTF-aware)

Sentinel Runtime — Kernel Intrusion Prevention 2025

- Replaced ptrace-based monitoring with seccomp user notifications
- Reduced overhead from 54× to 1.12×
- Defends against io_uring abuse, stealth processes, and TOCTOU attacks

Technical Skills

Languages:	C, C++, Rust, Go, Python, x86/AArch64 Assembly
Hardware & Firmware:	UEFI DXE, EDK II, SMM Sandboxing, x86 MSRs, AMD-V, ARMv8 EL2
Kernel Systems:	eBPF (LSM/XDP), KVM, namespaces, seccomp
Compiler Systems:	LLVM IR, inkwell, custom passes, dual-target compilation
Formal Methods:	Z3 SMT, Hoare logic, symbolic execution, IFC
Security:	Control-flow integrity, taint tracking, policy enforcement, cryptography
Networking:	TCP/IP, XDP, gRPC, Protobuf
Tools:	GDB, strace, bpftool, perf, QEMU, Git, CI/CD

Education

Bachelor of Technology in Computer Science Expected 2028
Amal Jyothi College of Engineering, India

Languages

German (Native) | English (Fluent) | Malayalam (Native)