

Vigil Security Operations Center

Moving from Agentic to Autonomous SOC in Machine-Speed Cyber Defense

A Warm "Hello from Texas" 🤖

Welcome Stanford Applied Cyber group! Broadcasting **live from the DeepTempo office space in Austin, Texas at Station Austin**, bringing real-world operational lessons from high-stakes telemetry, flightlines, fabs, and data centers into modern AI security operations.

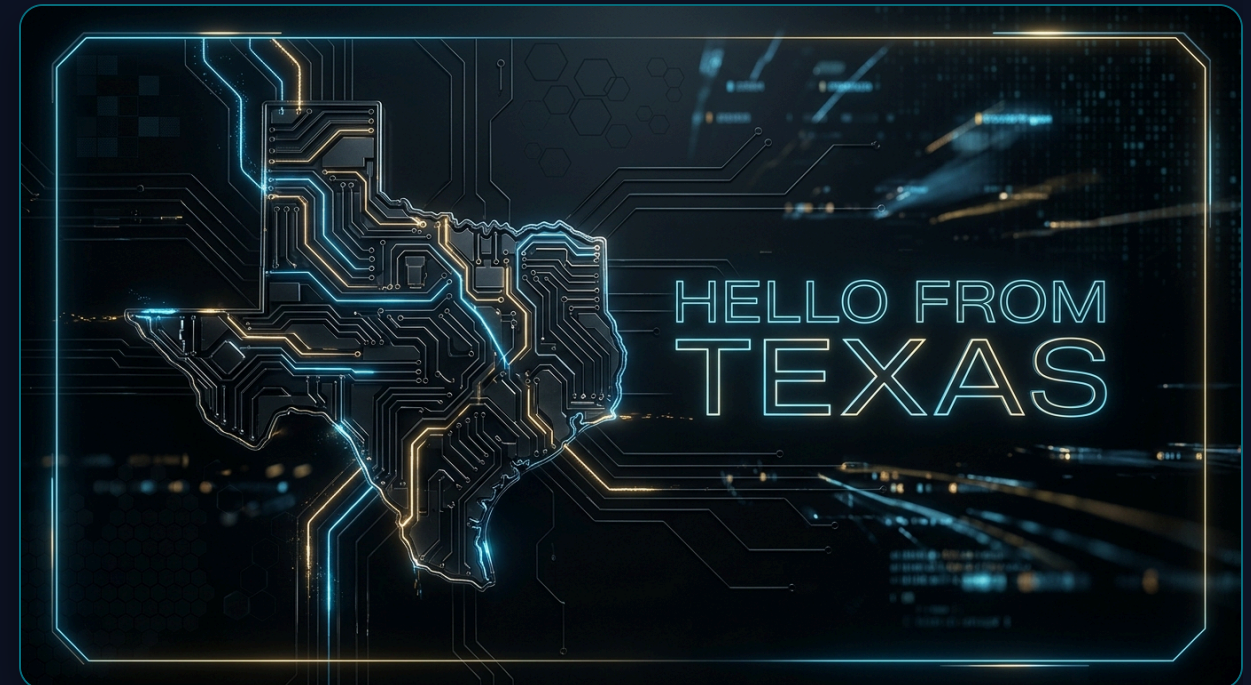


John Van Lowe

Implementation Engineer @ DeepTempo

Community Lead for **VigilSOC** • Open-Source AI SOC

Topic: Agentic AI • OSS DevRel • Autonomic SOC



Presentation Agenda & Discussion Flow

High-Level Pathway: From Operational Foundations to Autonomic AI & Open Q&A

START ▶ Welcome & Background Open Floor Q&A ◀ **FINISH**

PHASE 01

Slides 1–6

Foundations & Mission

- Speaker Operational Journey (NASA, USAF, Fabs)
- DeepTempo LogLM Neural Encoder
- VigilSOC Open-Source Platform
- Air-Gapped Telecom & IT/OT Edge Footprint

PHASE 02

Slides 7–10

Swarms & Defender Roles

- Blue, Red, & Purple Team Swarm Dynamics
- Shift to Humans on the White Team
- Stage 1 (HITL) to Stage 2 (HOTL) Release Roadmap
- Ramp to Autonomic Status (v1.0.0 Q4 2026)

PHASE 03

Slides 11–16

Harness & Software Factory

- Engineering Blog Deep-Dives (Kumar & Armstrong)
- Static Prompt Bloat Removal vs Event DAGs
- Vigil Repo Activity & Overlapped PR Streams
- Securing Repos & Supply Chain Defense

PHASE 04

Slides 17–23

Governance & Q&A Hub

- Native GitHub Security Features & Security.txt
- Pending Hardening (Signed SBOM, MCP Spec & Red Teaming)
- Strategic AI Governance & 6–Pillar Framework Matrix
- Upcoming AI + Cyber Events & Office Hours
- Open Floor Q&A & Community Connections

Interactive Session Format: 23 Total Slides • Direct Code Examples • Strategic AI Governance Discussion • Open Floor Q&A

John Van Lowe — Operational Journey

High-Reliability & Mission-Critical Infrastructure Engineering

NASA JSC MCC

Early career at Johnson Space Center Mission Control Center.
Total Operational Awareness and Zero-fault Tolerance.

USAF F-16 Crew Chief

F-16 Tactical Fighter Crew Chief post-9/11. Flightline operational readiness, combat sortie safety, relentless checklist discipline.

Micron Semiconductor Fabs

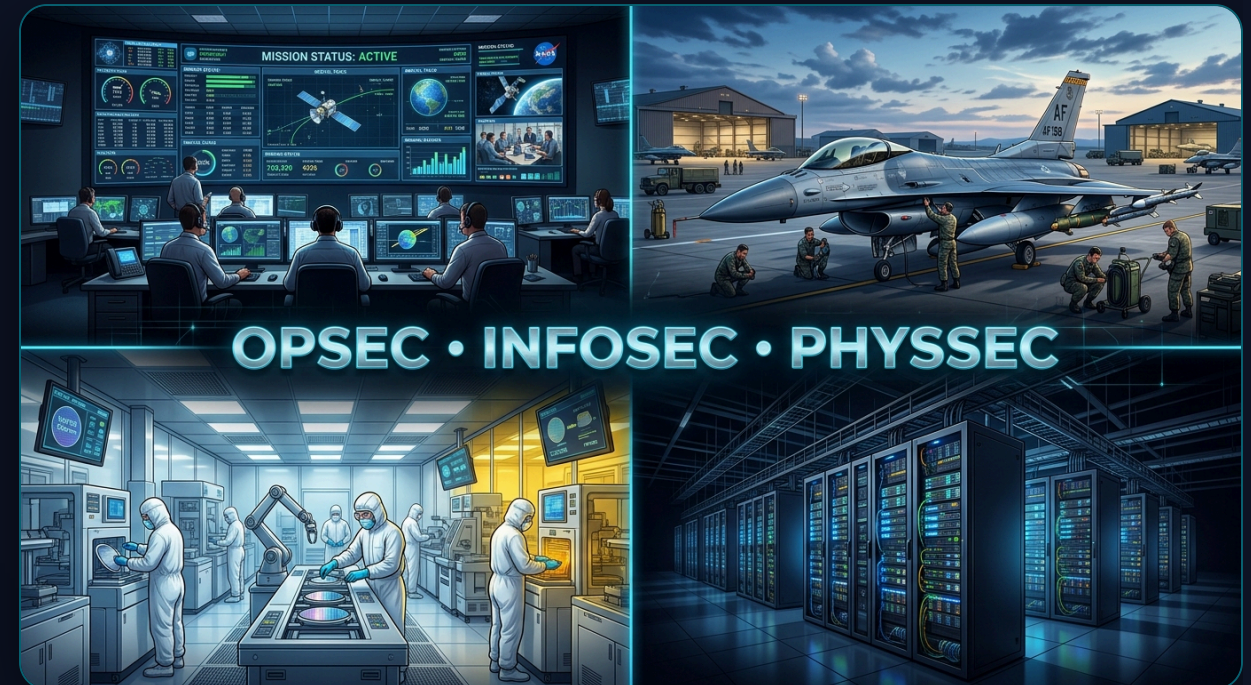
Cleanroom microelectronics & wafer fabrication infrastructure. Nanometer precision, physical asset control, high-yield fab ops.

Heavy Scale Infrastructure

HostGator, Codero, Patmos. Bare-metal & virtual machine scaling from single racks to 15MW data center facilities.

Unified Security Mindset

Operational Security (OpSec) • Information Security (InfoSec) • Physical Security (PhysSec)



DeepTempo & LogLM Architecture

Transformer-Based Encoders for Security Telemetry & Anomaly Surfacing

- ⚡ **DeepTempo's Mission:** Building intelligent defense platforms capable of analyzing telemetry at machine speed to stop zero-day and novel attacks.
- ⚡ **LogLM Model Architecture:** An encoder-only neural foundation model trained directly on raw logs, Netflow, and network flow vectors.
- ⚡ **Zero-Shot Detection:** Learns deep behavioral representations of network traffic, uncovering attacker intent without relying on legacy signatures.
- ⚡ **Netflow Stream Ingestion:** Feeds raw network packets and session telemetry directly into encoder attention layers to surface high-confidence anomalies.



VigilSOC — Harnessing LogLM for All

The Open-Source AI SOC Platform (Apache 2.0 License)



VigilSOC Platform

The Open Source AI Security
Operations Center

Apache 2.0

Local-First

 vigilsoc.org
 Vigil-SOC / vigil

The Ingest Bridge

Revealed as the internal solution to harness LogLM for organizations and security teams that lacked a modern ingest pipeline.

30+ MCP Integrations

Built on the Model Context Protocol (MCP) to plug seamlessly into existing security tools, SIEMs, and cloud APIs.

13+ Specialized AI Agents

Autonomous agents dedicated to triage, artifact extraction, PCAP correlation, threat hunting, and report generation.

Transparent & Extensible

Playbooks written in plain-text, agent logic in readable Python, and full freedom to plug in any local (Ollama) or remote LLM.

Public Repo Threat Detection Rules

Directly ingests, normalizes, and compiles open-source community detection rules pulled from public repositories—supporting **Sigma**, **Splunk (SPL)**, **Elastic (EQL)**, and **Microsoft KQL** rule syntax natively into agent hunting DAGs.

Multi-Source Rule Ingest

Use Cases: IT & OT Critical Infrastructure

Large Air-Gapped Installs & Edge Cyber Protection

Major Telecom Air-Gapped Installs

Deployed in isolated, air-gapped environments for major telco providers, safeguarding 5G backhaul, core transmission nodes, and fiber routing without cloud dependencies.

Critical IT/OT Edge & MSSP Infrastructure

Real-time autonomous threat detection deployed directly at the edge across essential industrial control (SCADA/HMI) and multi-tenant security operations:

 **Water Treatment Facilities:** SCADA pump control & filtration monitoring

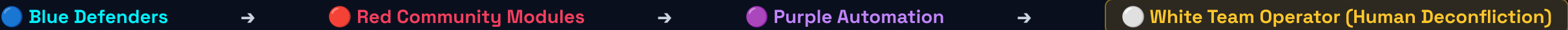
 **Regional SOC & MSSP Deployments:** Multi-tenant managed security services & regional defense

 **Power Plant Electrical Grids:** High-voltage substation transmission safety



Who Uses VigilSOC? The Team Evolution

From Blue Defenders to Purple Automation & The Emerging White Team Operator Paradigm



Blue Team Defenders

Primary user base: SOC analysts, threat hunters, and MSSP operators using VigilSOC for automated telemetry ingest, LogLM zero-shot triage, and machine-speed mitigation.

Red Team Capabilities

Community-contributed modules: Automated adversary simulation playbooks, synthetic attack generation, and continuous attack-surface probing to stress-test detection boundaries.

Purple Team Operations

Continuous collaborative loop: Real-time feedback engine matching offensive Red simulation against Blue detection DAGs to validate and self-correct playbooks automatically.

What is a DAG? Directed Acyclic Graph

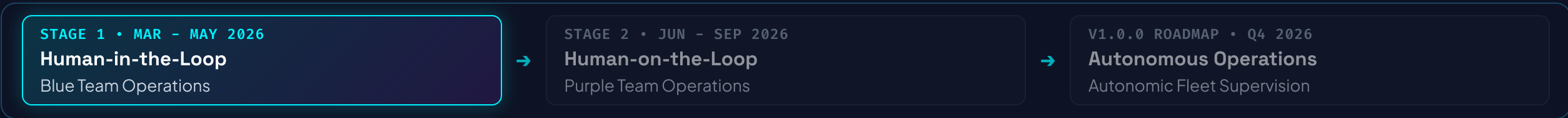
VigilSOC Workflow Harness

A **DAG (Directed Acyclic Graph)** is a mathematical graph of directed nodes and non-circular edges. In VigilSOC, DAG harnesses replace 2,000-word prompt bloat with deterministic agent routing, conditional decision branches, and strict circuit-breaker kill switches.



Vigil Evolution: Releases v0.1.0 – v0.3.0

Laying the Foundations of the Agentic SOC Engine (Mar 2026 – May 2026)



v0.1.0 MARCH 2026 • GENESIS

- ▶ Core Agentic Engine prototype
- ▶ Basic LLM routing & execution
- ▶ Initial telemetry parser
- ▶ Local CLI testbed

v0.2.0 APRIL 2026 • DAG ENGINE

- ▶ Directed Acyclic Graph (DAG) workflow orchestrator
- ▶ Smart LLM provider routing (Ollama / Claude)
- ▶ Containerized microservice architecture
- ▶ Stateful case tracking

v0.3.0 MAY 2026 • DETECTIONS

- ▶ Integrated 7,200+ Sigma detection rules
- ▶ Chat-driven case management UI
- ▶ Multi-agent routing protocols
- ▶ Structured event logging



Presenter Retrospective • May 2026 Paradigm

Austin Langchain and Middleware User Group

Personal Perspective (May 2026 Presentation: "Agentic SOC Operations: OSS AI Tooling"): Frame VigilSOC fundamentally as a **Blue Team Defender Tool** operating with an explicit, unyielding assumption of **Human-IN-the-Loop (HITL) validation at every execution step** before taking action.

Vigil Evolution: Releases v0.4.0 – v0.6.0

Expanding Swarm Collaboration & Enterprise MCP Integrations (Jun 2026 – Sep 2026)



v0.4.0 JUNE 2026 • MCP CORE

- ▶ Model Context Protocol (MCP) framework release
- ▶ First 15 tool connectors (VT, AbuseIPDB, Shodan)
- ▶ Plain-text playbook engine
- ▶ Automated threat enrichment

v0.5.0 AUGUST 2026 • SWARM OPS

- ▶ Multi-agent swarm collaboration
- ▶ Parallelized artifact triage
- ▶ Expanded to 30+ MCP connectors
- ▶ Local-first data privacy guarantees

v0.6.0 SEPT 2026 • CURRENT

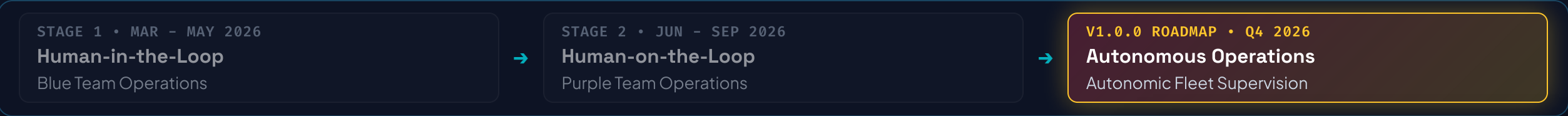
- ▶ 13+ specialized autonomous agents
- ▶ Audit-ready automated report generation
- ▶ High-throughput LogLM pipeline integration
- ▶ Apache 2.0 community library

 **Presenter Retrospective • June 2026 Paradigm** Austin ACM / IEEE Chapter

Personal Perspective (June 2026 Presentation: "*The Agentic Battlefield: Red vs Blue Agents*"): Noted heavy community adoption of red team tooling while shifting to **Human-ON-the-Loop (HOTL) supervisory principles** as humans still couldn't keep up with triage.

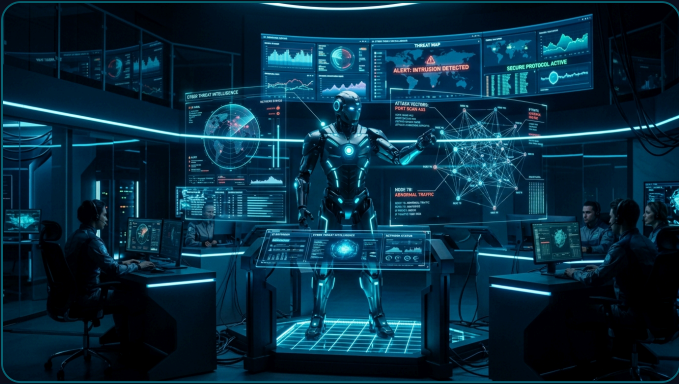
Roadmap: Ramp to Autonomic Status (v1.0.0)

Re-Imagining Autonomy in Modern Cyber Operations (Target: Q4 2026)



The Paradigm Shift in Cyber Ops
Moving from *Agentic SOC* (AI assisting human operators with isolated tools) to *Autonomic SOC* (self-coordinating, machine-speed defense swarms).

Human-on-the-Loop Supervision
Blue teams stop acting as manual alert triagers and start acting as **Supervisors** managing large fleets of autonomous AI agents operating at machine speed.



 **Presenter Retrospective • August 2026 Paradigm**

Hacker Summer Camp • Black Hat, BSides & DEF CON

Personal Perspective (Hacker Summer Camp Takeaways): Focus shifted to **full autonomy with strict guardrails, confidence scores, and automatic remediation** while incorporating continuous feedback loops for False Positives (FP) and False Negatives (FN).

Lessons Learned Building a Cyber Harness

By Mayank Kumar (Founding AI Engineer, DeepTempo) • Non-Linear Threat Hunting

🎯 "A Prompt is Not an Instruction, It is a Coordinate"

At inference, model weights are fixed. Tokens condition the forward pass to select which learned behavior responds. The real artifact is the dynamic **Selection Function**, not a 2,000-word prompt.

🔄 First-Class Pivoting & Early Abandonment

Threat hunting (PEAK & TaHiTi frameworks) requires iterative pivots—from Netflow to process telemetry, credentials, and cloud audit logs. Knowing when to kill a dead-end hunt branch early is critical.

🚫 Overcoming Linear Pipeline & Prompt Bloat

Linear pipelines hide looping; adding "always/never" prompt rules leads to 2,000-word prompt bloat and infinite agent loops. Vigil 0.5.0 replaces static prompts with dynamic iteration constraints.

🔧 The Vigil Harness Architecture

Built with StackStorm complex automation heritage. Decides every iteration what the agent sees and what tools it may execute, encoding expert human threat hunter intuition directly into code controls.

[🔗 Read Full Article: vigilsoc.org/blog/2026/08/27/lessons-learned-building-a-harness-for-cyber-security/](https://vigilsoc.org/blog/2026/08/27/lessons-learned-building-a-harness-for-cyber-security/)

The New Vigil Harness: Repository Architecture

Concrete Code Evolution in Vigil-SOC/vigil — What Was Removed vs. What Replaced It

✗ What Was Removed (Anti-Patterns)

- ✗ **Static 2,000-Word Prompt Bloat:** Giant system prompts packed with rigid "always/never" rules that caused models to hallucinate or freeze.
- ✗ **Rigid Linear Pipelines:** Hardcoded step-by-step scripts hiding iteration loops and forcing uniform execution across all incident types.
- ✗ **Opaque Black-Box LLM Wrappers:** Hidden prompt text and un-auditable fallback chains impossible for SOC analysts to inspect or verify.
- ✗ **Infinite Loop Computations:** Agents blindly chasing dead-end threat branches indefinitely without cost or compute limits.

⚡ What Replaced It (The Vigil Harness)

- 🎯 **Dynamic Selection Functions:** Token conditioning focusing model attention on precise task coordinates rather than static prompt bloat.
- 🔄 **Event-Driven DAG Engine:** StackStorm-inspired Directed Acyclic Graph orchestrator for non-linear, stateful threat hunting.
- 🔧 **30+ Modular MCP Connectors:** Decoupled Model Context Protocol tool interfaces (VT, AbuseIPDB, Shodan) executed on-demand.
- 🛡️ **Code-Level Pivoting & Early Kill Triggers:** First-class pivots (Netflow → Process → Audit Logs) and early branch abandonment to preserve compute.

📦 **Repository Activity (Vigil-SOC/vigil):** Fully transparent Python agent logic & plain-text YAML playbooks (Apache 2.0).

📄 github.com/Vigil-SOC/vigil

Yet Another Software Factory

By Sam Armstrong • Building Vigil via Autonomous Engineering Factories



Autonomous Backlog to Merged PR

Autonomous agents groom the backlog, author PRs against Vigil, review code changes, and self-repair build environments when test harnesses stall.



Deep Verification Beyond Code Review

Clean diffs don't guarantee runtime correctness. The factory executes live validation—running changes and exercising agent tool bindings before merging.



Open Source Economics & Scale

Small teams cannot manually review high-volume community contributions. The factory drives product direction while scaling open source contribution velocity.



Self-Healing Build Environments

When build harnesses stall, agents diagnose environmental failures, resolve broken dependencies, and repair test environments autonomously without human intervention.



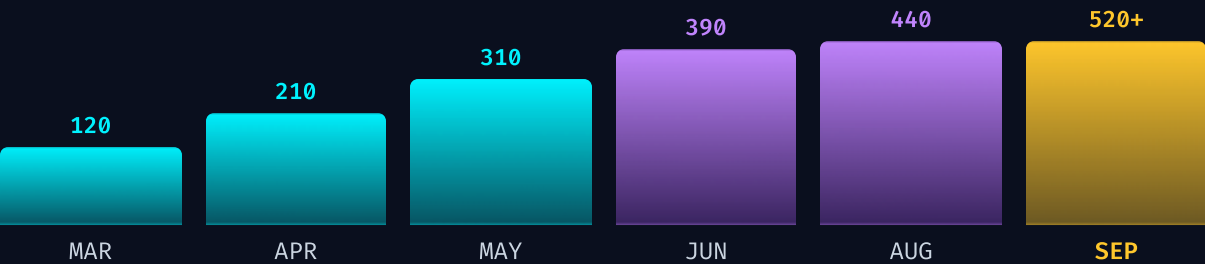
Read Full Article: vigilsoc.org/blog/2026/09/14/yet-another-software-factory/

Software Factory in Code: Repository Activity

Quantifying Velocity in Vigil-SOC/vigil — Commits, Overlapped PRs & High-Cadence Code Quality

Commits & Review Comments Over Time

1,840+ Total Commits



Overlapped Autonomous PR Streams

4 Parallel Agents

- PR #142: LogLM Ingest Engine Integration [MERGED]
- PR #189: Swarm Triage & PCAP Correlation [REVIEWED]
- PR #214: 30+ MCP Connector Expansion [TESTING]
- PR #258: Self-Healing CI Harness Auto-Repair [PASSED]

⚡ High Iteration Cadence

Continuous overlapping PR streams groomed, authored, and verified in parallel without human code-review bottlenecks.

📊 Commit Velocity

Exponential commit volume trajectory from 120 (Mar) to 520+ monthly commits (Sep 2026) across 1,840+ total commits.

🔧 Code Quality & Verification

Enforced Python type hints, 94%+ automated unit test coverage, and deep runtime tool-action verification before merging.

🔧 Self-Healing Build Resilience

CI test harnesses that detect stalled builds, diagnose broken test dependencies, and self-repair environments autonomously.

Harness Architecture: Charter, Code & Upstream Factory

Bridging Governance Specifications, Code Execution & Internal Cursor Factory Sync

1. The Charter (Specification)

Version-controlled governance documents defining non-negotiable boundaries, intent, and schema contracts:

 **principles.md**: Local-first privacy, zero black-box wrappers, open-source Apache 2.0.

 **intent.md**: Declarative agent goals, hunt boundaries, and threat hunter intuition in code.

 **versioning.md**: Strict semantic release criteria and API contract stability guarantees.

2. The Harness (In Code)

Concrete runtime execution engine implemented directly in `Vigil-SOC/vigil`:

 **Event-Driven DAG**: StackStorm-inspired non-linear workflow orchestrator with state tracking.

 **Circuit Breakers**: First-class domain pivots (Netflow → Process → Audit Logs) & early kill triggers.

 **30+ MCP Connectors**: Standardized Model Context Protocol tool interfaces.

3. Upstream Factory & Sync

Autonomous factory generating code upstream & syncing to the public repo:

 **Autonomous PR Authoring**: Backlog grooming, unit test generation, and live validation.

 **Internal Cursor Factory Sync**: Proprietary Cursor factory rules, custom scratch scripts, and prompt orchestrators currently being audited for public sync to `Vigil-SOC/vigil`.



Upstream Sync Notice: Core internal Cursor factory rules, specialized prompt harnesses, and automated scratch scripts are undergoing security auditing for upcoming open-source release to github.com/Vigil-SOC/vigil.

Sync in Progress

Securing Public Repos: Factory Controls

Mitigating Supply Chain Attacks, Sockpuppet Networks & Malicious Contributions

1. Sockpuppets & Social Eng.

THE THREAT:

Threat actors use coordinated sockpuppet accounts to pressurize fatigued maintainers into transferring repo permissions or approving malicious PRs (e.g., 2-year XZ Utils long-con).

FACTORY REMEDIATION:

Automates initial code triage and removes human emotional social-engineering vectors; PRs are verified by objective automated factory agents rather than pressure.

2. Supply Chain Backdoors

THE THREAT:

Steganographic binary payloads in test archives, multi-stage build-script hooks (m4 macros), or hidden zero-day backdoors targeting downstream production binaries.

FACTORY REMEDIATION:

Deep isolated runtime sandbox execution. Diffs are never accepted on visual inspection alone; the factory executes full runtime behavior analysis before merging.

3. AI Slop & Poison Pills

THE THREAT:

High-volume automated AI "slop" PRs, typosquatted dependency additions, or subtle logic vulnerabilities designed to slip past human code reviews.

FACTORY REMEDIATION:

Enforces AST typing, dependency provenance validation, automated static analysis, and 94%+ test coverage gates to discard low-quality or poisoned contributions.

AST Typing Definition

Parsing source code into an **Abstract Syntax Tree (AST)** to statically verify type annotations, variable bindings, and function signatures without executing untrusted code—blocking malformed or poisoned PRs before sandbox execution.

 **The Factory Advantage:** By decoupling PR verification from human fatigue and enforcing deep runtime validation, the Software Factory turns public open-source contribution into a secure, zero-trust pipeline.

Zero-Trust Public Ingest

Native GitHub Security Features

Repository Hardening, Automated Scanners & Vulnerability Reporting Controls

Security policy • Enabled

View how to securely report security vulnerabilities for this repository

Security advisories • Enabled

View or disclose security advisories for this repository

Private vulnerability reporting • Enabled

Allow users to privately report potential security vulnerabilities

Dependabot alerts • Enabled

Get notified when one of your dependencies has a vulnerability

Code scanning alerts • Enabled

Automatically detect common vulnerability and coding errors

Secret scanning alerts • Enabled

Get notified when a secret is pushed to this repository

 Repository Security Overview Settings (github.com/Vigil-S0C/vigil)

 Security Policy (SECURITY.md)

• Enabled

Establishes formal vulnerability reporting guidelines and contact channels for security researchers.

 Security Advisories (CVD & CVEs)

• Enabled

Enables maintainers to draft, discuss, and fix vulnerabilities privately and publish CVE advisories.

 Private Vulnerability Reporting

• Enabled

Provides a direct confidential inbox for external security researchers to submit flaws without public leakage.

 Dependabot Alerts (SCA Supply Chain)

• Enabled

Continuously monitors third-party dependencies against CVE databases and opens automated fix PRs.

 Code Scanning Alerts (CodeQL SAST)

• Enabled

Uses AST static analysis to flag injections, memory safety issues, and coding errors before merging.

 Secret Scanning Alerts

• Enabled

Detects and blocks exposed API tokens, private SSH/RSA keys, and cloud credentials pushed to commits.

Security.txt & Private Disclosure

Standardizing Vulnerability Reporting via RFC 9116

RFC 9116 Compliance

Adopting the IETF standard for /.well-known/security.txt to provide security researchers with clear, machine-readable contact information, encryption keys, and disclosure policies.

Coordinated Vulnerability Disclosure

Establishing private vulnerability triage channels, fast patch SLA commitments, and transparent PGP-signed security advisories for open-source AI SOC infrastructure.

Private Fork Collaboration & Patching

Submitting a private vulnerability automatically provisions a **temporary private repository fork**. Maintainers and the reporter collaborate directly in secret to author, test, and verify patches through to coordinated public release & CVE issuance.

 vigilsoc.org/.well-known/security.txt

RFC 9116

```
# Vigil SOC – security.txt
# Standard: RFC 9116 (https://www.rfc-editor.org/rfc/rfc9116)
# Canonical location: https://vigilsoc.org/.well-known/security.txt
#
# To report a vulnerability, please use GitHub private vulnerability reporting
# (preferred) or the security email below. Do NOT open public GitHub issues.
```

Contact: <https://github.com/Vigil-SOC/vigil/security/advisories/new>

Contact: <mailto:security@vigilsoc.org>

Preferred-Languages: en

Policy: <https://vigilsoc.org/security>

Canonical: <https://vigilsoc.org/.well-known/security.txt>

Pending Security Tasks & In-Flight Hardening

Supply Chain Provenance, Integration Path Penetration, Inference Red-Teaming & MCP Spec Hardening

Cryptographically Signed SBOM Manifest

Supply Chain

Task: Publishing a comprehensive, cryptographically signed **Software Bill of Materials (SBOM)** with standardized in-repo manifests (CycloneDX / SPDX).

Impact: Provides enterprise consumers with verifiable build attestations, provenance tracking, and automated transitive dependency validation.

Attacking Integration Paths & Connectors

Penetration Testing

Task: Penetration testing and threat modeling targeting internal event-driven DAG execution hooks, multi-agent IPC channels, and external API connectors.

Impact: Prevents malicious webhooks or manipulated payload data from hijacking DAG execution flow or breaching sandbox isolation.

Red Teaming the Neural Inference Stack

Model Security

Task: Adversarial red teaming directed at the **configured Inference & Model providers** (local foundation models, self-hosted LLMs, or cloud API endpoints), evaluating prompt-injection resilience, token manipulation, and model provider integrity.

Impact: Ensures adversarial telemetry or manipulated inputs cannot blind inference encoders, bypass safety guardrails, or trick downstream triage swarms regardless of the underlying model provider.

In-Flight Review: MCP State & Auth Spec

In-Flight Refactor

Task: Active review and refactoring of 30+ **Model Context Protocol (MCP)** connectors for compliance with latest MCP spec updates.

Impact: Enforces standardized **state persistence** rules and fine-grained **authentication/authorization primitives** across all external tool integrations.

 **Open-Source Security Discipline:** Securing agentic AI infrastructure requires moving beyond basic code linting to signed SBOM provenance, continuous adversarial red teaming of neural inference stacks, and strict MCP specification compliance.

In-Flight Hardening

Questions to Consider for the Future

Frequently Surfaced Agentic Challenges and Concerns

? Question 1: Agentic Terms of Service

Should we establish Agentic ToS like we have Human ToS?

As non-human AI agents interact autonomously across APIs, execute security actions, and consume cloud services, how do legal and operational terms of service adapt to agentic entities?

? Question 2: AI Security & Governance Standards

Is ISO 42001 the best starting point for AI management, or are emerging threat & compliance frameworks taking precedence?

Evaluating the convergence of **ISO/IEC 42001 (AIMS)** and **NIST AI RMF** alongside trending operational standards: **OWASP Top 10 for LLMs**, **EU AI Act compliance mandates**, **MITRE ATLAS™ threat modeling**, and **CSA STAR for AI** to govern autonomous agent fleets in enterprise production.

⚡ Challenge 3: Agentic Identity & Blast Radius Governance

How do we cryptographically verify multi-agent delegation & prevent cascade poisoning?

When autonomous agents delegate tools to downstream sub-agents over zero-trust boundaries, how do we guarantee cryptographic identity, prevent prompt injection hijacking, and enforce circuit breakers before machine-speed mitigation causes self-inflicted blackout?

AI Security & Governance Framework Matrix

Comparative Spectrum: ISO 42001, NIST AI RMF, OWASP, EU AI Act, MITRE ATLAS & CSA STAR

ISO/IEC 42001 (AIMS)

Certiﬁable Mgmt

Domain: Management Systems.
Focus: Org-wide policy, continuous improvement, & ethical AI governance controls across life cycles.

NIST AI RMF 1.0

Risk Guidance

Domain: Risk Management.
Focus: 4 Core Functions (Govern, Map, Measure, Manage) for trustworthy, bias-aware AI operations.

OWASP Top 10 LLM

App Vulnerabilities

Domain: Application Security.
Focus: Tactical vulnerabilities (Prompt Injection, Data Poisoning, Insecure Output, Supply Chain).

EU AI Act Mandate

Legal Regulation

Domain: Legal Compliance.
Focus: Risk classification (Unacceptable, High, GPAI), requiring conformity assessments & audits.

MITRE ATLAS™

Threat Modeling

Domain: Adversary Tactics.
Focus: Threat matrix mapping real-world ML attacks (Model Evasion, Weight Theft, Poisoning).

CSA STAR for AI

Cloud Trust Registry

Domain: Cloud AI Services.
Focus: Security assurance, data privacy guarantees, and tenant isolation for cloud AI deployments.

 **Framework Convergence:** Enterprise AI governance pairs top-down management (ISO 42001 / NIST RMF) and legal mandates (EU AI Act) with bottom-up technical defense (OWASP / MITRE ATLAS) and cloud trust (CSA STAR).

6-Pillar Synergy

Upcoming AI + Cyber Events & Office Hours

Bay Area Industry Meetups, Free Virtual Webinars & Community Workshops

AI Cyber Alliance • #SVAI

In-Person Meetups

Connecting AI & cybersecurity leaders across academia and industry. Live Bay Area events hosted at **#SVAI in Menlo Park**:

 **Nov 19, 2026** • 6:00 PM – 8:30 PM @ **#SVAI Menlo Park**

 **Dec 15, 2026** • 6:00 PM – 8:30 PM @ **#SVAI Menlo Park**

 **2027 Series** • 2027 Dates Pending Announcement @ **#SVAI**

 AICyberAlliance.org

AISec University

Free Virtual Classes

Free practitioner-led virtual webinars, interactive workshops, and education on AI security governance & defense:

 **Oct 15, 2026** • 11:00 AM ET — **How to Recover From an AI Breach**

 **Oct 27, 2026** • 1:00 PM ET — **Getting Ahead of the Mythos Effect**

 **Weekly Sessions** • Free Live Webinars & Recorded Replays

 aisec.university/events

VigilSOC Office Hours & Community Sessions

Scheduled Virtual Sessions

Join open community office hours with VigilSOC maintainers & threat researchers. Discuss agent DAG workflows, custom MCP tool connectors, playbooks, and real-world deployment telemetry.

 Luma Registration:

luma.com/a31xoxr2

 VigilSOC Hub:

vigilsoc.org

Thank You & Questions!

Join the Open AI SOC Revolution • Connecting with DeepTempo, VigilSOC & AI Cyber Alliance

🙌 Thank You & Questions!

💬 Open Floor Q&A

✉ Direct Contact

Email John Van Lowe directly for platform research, enterprise deployments, or community collaboration:

john@deeptempo.ai

🔗 LinkedIn Profile

Connect with John Van Lowe on LinkedIn to follow ongoing research and platform updates:

linkedin.com/in/johnvl

🌐 Project Ecosystem

Core open-source projects, community channels, and benchmarks:

🌐 Website:	deeptempo.ai
🛡 Community:	vigilsoc.org
💻 Vigil Code:	Vigil-SOC / vigil
🐙 GitHub Profile:	github.com/johnvanlowe/
📊 Benchmark:	socbench.org

📜 Prior Talks & Technical Decks

Explore prior keynotes, whitepapers, and slide archives:

- ✂ The Agentic Battlefield: Red vs Blue Agents
 - 💻 Don't Be That DEV! Infosec Edition
 - 🛡 OWASP Top 10 MCP & MSSS
 - ❄ Snow Shoe Hunting With GoLang
-
- 🎓 Stanford Deck: retroencabulation.com/stanford/
 - 🌐 Main Archive: retroencabulation.com/