Elliot Anstey

VULNERABILITY MANAGEMENT · SECURITY TOOLING & SOFTWARE ENGINEERING

ea.infosec@gmail.com
linkedin.com/in/elliotanstey
elliotanstey.com
Orlando, FL · open to remote

Vulnerability management analyst who builds the tooling the role kept asking for. I ran Tenable operations in a HIPAA/ePHI hospital environment — SLA tracking, SOC triage, remediation coordination — then engineered and shipped a Microsoft Store–certified, code-signed Windows product that puts the whole SLA picture on one screen for a fleet of tens of thousands of assets. I take work the whole distance: from the API ingest to the signed, certified release, with the security evidence to back it.

EXPERIENCE

Vulnerability Management Analyst · Orlando Health — Aug 2025 – Jun 2026

- Ran vulnerability-management platform operations across a hospital fleet in a **HIPAA/ePHI environment**, where a missed remediation window is a compliance exposure, not just a ticket.
- Owned **SLA tracking** for scan findings — monitoring time-to-breach across severities and driving the work that kept findings inside their remediation windows.
- Performed **SOC triage** on incoming findings and alerts, separating the genuinely urgent from the noise before it reached engineering queues.
- Coordinated **remediation across owning teams**, matching findings to the people who could actually fix them and following them through to closure.
- Turned the gaps in that daily work into **shipped tooling** — the products below came directly out of problems the platform could not answer.

Licensed Massage Therapist · career change into security — 2014 – 2025

- A decade of licensed, client-facing practice in a regulated healthcare field — the same duty of care, documentation discipline, and confidentiality that HIPAA work demands.
- Studied and built security nights and weekends throughout, then moved into the field full time on the strength of the degree, certifications, and the work in this résumé.

SHIPPED PRODUCTS & ENGINEERING

SLA Analyzer — for Tenable.io · flagship, Microsoft Store certified — Windows desktop

- Answers the question a VM team asks every morning: **which findings are about to breach SLA, who owns the assets, and what was done about it** — configurable SLA policy per severity, with drilldowns from any number on screen to the CVE, the assets, and the owning team.
- **Zero-day exposure in seconds, without waiting for a scan:** matches the CISA KEV, NVD, and ZDI feeds against findings already on disk, so "are we exposed, and where?" is answered from local data.
- **Multi-million-finding tenants on a single desktop** — a disk-backed engine keeps memory flat and the interface responsive; profiled and benchmarked at 2.8 million findings.
- Carried end to end: API ingest → analysis engine → packaging → code signing → **full Microsoft Store certification**.

Remediation toolchain · TicketTimeMachine · SNOW Dispatcher · simulator — Python

- Assembles **idempotent ServiceNow remediation tickets** grouped per owning team, with correlation keys so re-runs never double-file, and a dry-run mode that writes every payload for review before anything is submitted.
- Built a **full mock-ServiceNow environment** with scriptable failure paths, so the entire stack can be developed, tested, and demonstrated with zero live credentials.

Signet · release-signing application — 3 days, idea to verified

- Automates the fragmented last mile of shipping software — the code-signing ceremony — pausing only for the physical hardware-key touch. It signed **its own release build**, and the signature survived a **five-way adversarial verification** including a byte-flip tamper test.

Deadbolt · hardware-key release gate — running in production

- Sensitive data stays encrypted until a person **physically touches a hardware key** — every request logged and attributable. A stolen credential can only ask. The same request → approve → release pattern controls demo distribution on this site.

Secure SDLC — "I attack my own release" · compliance gauntlet — every release

- Built a deterministic security regression suite each release must clear — SAST, dependency audit, secret scanning, license audit, SBOM, and a data-at-rest probe — mapped to **NIST SSDF, OWASP ASVS, and CIS Controls**, producing a dated evidence bundle and a verdict.
- Implemented the tamper-evidence primitives behind Certificate Transparency and modern supply-chain attestation — **RFC 6962 Merkle proofs, RFC 3161 timestamping, DSSE/in-toto provenance** — tested against the specifications' own vectors.

CERTIFICATIONS & COMPETITION

CERTIFIED	**CompTIA Security+ (CE)** — ISO/IEC 17024, DoD 8140 IAT II baseline	CERTIFIED	**EC-Council C|EH** — ANAB / ISO 17024 accredited
NCL TEAM	**19th of 4,199 nationally — 100th percentile**, Spring 2024	NCL SOLO	**432nd of 7,406 — 95th percentile**; top 2% password cracking
TRYHACKME	**Top 1% globally** — 177 hands-on rooms	LABS	SOC Level 1 · Cyber Defense · Red Teaming paths

Every credential is verifiable with its issuer — certificates and score cards are published on the about page.

EDUCATION

B.S., Cybersecurity · ECPI University — May 2024

– Graduated **summa cum laude** from a **NSA/DHS CAE-CD designated** program.

SKILLS

VULN MGMT	Tenable.io API · SLA policy design · KEV/NVD/ZDI intelligence · asset ownership · attack-surface management	LANGUAGES	Python (PyQt6, Textual, SQLite) · PowerShell · Bash · SQL · HTML/CSS/JS
SECURITY ENG	Applied cryptography · tamper-evident logging · Merkle proofs · DSSE/in-toto · threat modelling	SECURE SDLC	Bandit · Semgrep · gitleaks · pip-audit · CycloneDX SBOM · code signing · Store certification
COMPLIANCE	HIPAA · NIST SSDF · NIST 800-53 / 800-171 · OWASP ASVS · CIS Controls	PLATFORMS	Windows · Linux · ServiceNow integration · REST APIs · Git · self-hosted infrastructure

HOW I WORK

- **Evidence over claims.** Every number on my site maps to a specific evidence file, and every release carries a hash manifest anyone can recompute.
- **Honest limits, in writing.** Every project documents what the current version does *not* do — a tool that hides its boundaries is one an operator will eventually trust wrongly.
- **Dry-run first.** Anything that touches an external system of record ships with a mode that writes reviewable payloads to disk before a single live call is made.