From Course Project to Production Tool

Building Security Intelligence with Claude Code

A DC 402 Talk

$ whoami

Levi Reuss

Cyber Security Engineer / Manager

The Day Job

Run the security program for a medium-sized beef manufacturing company. Wide variety of security tasks — the title says engineer, the reality is manager.

IT Since 2008

Help desk → Windows admin → full-time cyber security focus for the last ~5 years. Admin by heart — only write code to make my job easier.

CS Degree, Hated Development

Went to college for Computer Science, got the dev background, but hated it. Went the IT route instead. Turns out AI makes coding tolerable.

TLDR

Security guy, not a developer.

I write code only when it makes my life easier.

Claude Code changed the equation.

THE ORIGIN STORY

Where It Started

Wild West Hacking Fest

Deadwood 2025

Course: AI for Cybersecurity Professionals

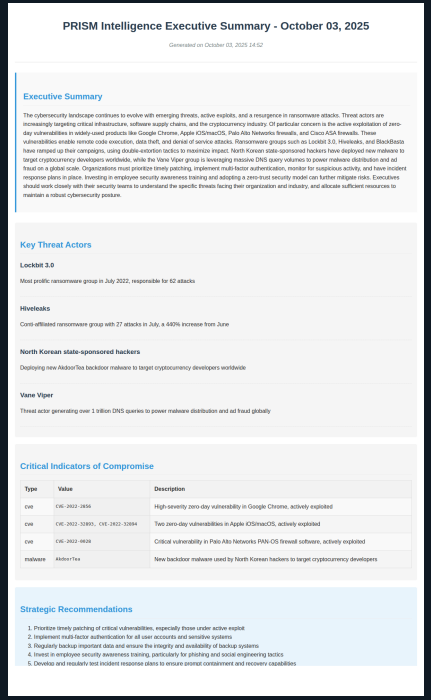
Instructors: Joff Thyer & Derek Banks

The course included a proof-of-concept: a Python script that scrapes security websites and summarizes news using an LLM.

"This course changed how I think about LLMs"

antisyphontraining.com

What the Course POC Produced



PRISM Intelligence Executive Summary - October 03, 2025

Generated on October 03, 2025 14:52

Executive Summary

The cybersecurity landscape continues to evolve with emerging threats, active exploits, and a resurgence in ransomware attacks. Threat actors are increasingly targeting critical infrastructure, software supply chains, and the cryptocurrency industry. Of particular concern is the active exploitation of zero-day vulnerabilities in widely-used products like Google Chrome, Apple iOS/macOS, Palo Alto Networks firewalls, and Cisco ASA firewalls. These vulnerabilities enable remote code execution, data theft, and denial-of-service attacks. Ransomware groups such as Lockbit 3.0, Hiveleaks, and BlackBasta have ramped up their campaigns, using double-extortion tactics to maximize impact. North Korean state-sponsored hackers have deployed new malware to target cryptocurrency developers worldwide, while the Vane Viper group is leveraging massive DNS query volumes to power malware distribution and ad fraud on a global scale. Organizations must prioritize timely patching, implement multi-factor authentication, monitor for suspicious activity, and have incident response plans in place. Investing in employee security awareness training and adopting a zero-trust security model can further mitigate risks. Executives should work closely with their security teams to understand the specific threats facing their organization and industry, and allocate sufficient resources to maintain a robust cybersecurity posture.

Key Threat Actors

Lockbit 3.0

Most prolific ransomware group in July 2022, responsible for 62 attacks

Hiveleaks

Conti-affiliated ransomware group with 27 attacks in July, a 440% increase from June

North Korean state-sponsored hackers

Deploying new AlkdoorTea backdoor malware to target cryptocurrency developers worldwide

Vane Viper

Threat actor generating over 1 trillion DNS queries to power malware distribution and ad fraud globally

Critical Indicators of Compromise

Type	Value	Description
cve	CVE-2022-2856	High-severity zero-day vulnerability in Google Chrome, actively exploited
cve	CVE-2022-32893, CVE-2022-32894	Two zero-day vulnerabilities in Apple iOS/macOS, actively exploited
cve	CVE-2022-0028	Critical vulnerability in Palo Alto Networks PAN-OS firewall software, actively exploited
malware	AlkdoorTea	New backdoor malware used by North Korean hackers to target cryptocurrency developers

Strategic Recommendations

1. Prioritize timely patching of critical vulnerabilities, especially those under active exploit
2. Implement multi-factor authentication for all user accounts and sensitive systems
3. Regularly backup important data and ensure the integrity and availability of backup systems
4. Invest in employee security awareness training, particularly for phishing and social engineering tactics
5. Develop and resiliently test incident response plans to ensure prompt containment and recovery capabilities

"Python script + LLM = this HTML executive summary"

KEY TAKEAWAY

"LLMs are not just chatbots to talk to and get answers — you can use them to generate code to automate IT/Cybersecurity tasks."

If you remember ONE thing from this talk, remember this.

THE ORIGIN STORY

The Problem I Wanted to Solve

The course POC was cool — but it only scraped a few news sites and needed a paid API.

As a security engineer, my real problem is bigger: I can't keep up with everything changing across all my vendors, tools, and threat feeds.

Track Everything That Matters

Microsoft Defender updates • Entra/Intune/ Sentinel changes • Third-party vendor releases • Threat intel from 23+ feeds • AI provider changes

Run It Locally for Free

Frontier API costs add up fast • LM Studio with local models • Free after initial setup • Keep sensitive data on my machine

Automate the Entire Workflow

Scrape → Summarize → Extract IOCs → Generate report • Tiered reporting (1 day to 1 month) • One command, actionable output

"I'm a security engineer, not a developer. I can read code and do basics, but building this myself would take months."

The Discovery: Claude Code

Saw a NetworkChuck YouTube video on Claude Code

Set it up in Windows WSL

"I couldn't believe how good it was at coding"

Key: Claude modifies files on disk AND tests them

RESULT

Months

→

Days

THE JOURNEY

The Timeline

Oct 2025 — Started with Claude GUI

~Nov 2025 — Discovered Claude Code in terminal

Oct-Jan — Free → Pro → Max, Sonnet → Opus

Feb 2026 — Public release on GitHub

Phase 1: Claude GUI

Modified the PRISM POC with Claude Desktop. Tedious — download scripts, run manually, copy-paste. Hit sub limits fast.

Phase 2: Claude Code

NetworkChuck video changed everything. Claude modifies files on disk AND tests them. Massive time savings.

Phase 3: Ship It

Could have released earlier but wanted it perfect and had so many ideas to try. Realistically, years without Claude Code.

"Could have released earlier, but I wanted it perfect and had so many ideas to try"

The Subscription & Model Journey

Subscription Evolution

✗ **Free Plan**

Hit limits almost immediately — frustrating

○ **Claude Pro ($20/mo)**

~1 hour of work, then wait 5 hours. Did this for 2 months.

✔ **Claude Max (5x — ~$100/mo)**

Game changer — never hit caps. Use it every day after work.

Model Evolution

Sonnet (initially)

Had to use Sonnet because Opus maxed out usage almost immediately.

Opus (later)

Extreme improvement: better at coding, better recommendations, actually asks clarifying questions about the project.

"The model matters — Opus is significantly better if you can afford the usage"

"Be prepared to invest in the subscription if you're serious about using it"

Workflow & Tooling

Multi-Machine Sync

Laptop ↔ Desktop: manually copying files to WSL was tedious

Project lives on OneDrive mount — auto-syncs between machines. Two-repo strategy: private (with secrets) + public (for GitHub).

Git & GitHub

No backup if Claude broke something. First public repo ever — Claude taught me the entire git workflow.

Added git for local version history. Claude helped sync to a private repo and sanitize sensitive data for public release.

The Paranoia (Justified)

What if API keys or vendor names I don't want public get synced without my permission?

Disconnected automated GitHub sync entirely. Now manually upload code changes. sync-to-public.sh excludes sensitive files — I control what goes public.

"It's OK to use a simpler workflow if it makes you feel more secure — trust but verify."

Ways to Use Claude

Claude GUI (claude.ai)

Web/desktop chat interface. Use Projects feature at work — solid for documentation, less ideal for scripts (back-and-forth testing slows it down).

Where I started — copy-paste workflow

Claude Code (Terminal)

CLI tool that reads your codebase, edits files on disk, runs commands, and tests code. Full control over file access and permissions.

What I use — admin-friendly, powerful

Claude in IDEs (VS Code, etc.)

Extension for Visual Studio Code and other IDEs. Great for developers, but IDEs aren’t my thing.

For devs — not my workflow

Claude Cowork (Desktop Agent)

Desktop agent for editing files on the fly. Newer option for non-developers — haven't used it personally.

Haven't tried it yet — anyone here have?

"I’m an admin at heart — the terminal is where I live."

GUI vs Terminal

Claude GUI (Phase 1)

Generate code in chat window

Manually download scripts

Manually run and test

Copy-paste back and forth

Hit subscription caps fast

No file system access

Claude Code (Phase 2)

Describe what you want

Claude edits files on disk

Runs and tests automatically

Iterates in project context

Efficient token usage

Control what files Claude sees

"Stupid cool to see it generate code — but the terminal is where the real power is."

Patterns I Didn't Design

Plugin Architecture

BaseTracker abstract class — easy to add new trackers

Factory Pattern

AI providers switchable with one config change

Strategy Pattern

RSS, Web, API, Headless browser — pluggable scrapers

"I described what I wanted. Claude Code suggested these patterns."

Local LLM Support (My Primary Goal)

AIClient Multi-Provider

LM Studio (local)
FREE after setup

Claude API (cloud)
Pay per token

One config change switches providers

```
Recommended: gpt-oss-20b
```

Cost Reality Check

Haiku	Cheap, viable
Sonnet	Too expensive for ongoing ops
Opus	Great quality, cost prohibitive
LM Studio	Free after setup ✔

Hardware: Works on my RTX 4090 (24GB) desktop & RTX 5080 (16GB) laptop. Should run on a 3060/4060 (8GB) too — adjust context window.

LM Studio runs as a local network server. Best on the same machine, but you can run the program on your laptop and point it at your powerful desktop on your home network.

Progressive Build-Out

1	**Defender**	Microsoft Defender news, local AI (LM Studio)
2	**Threat Intel**	23 curated feeds (CISA, Mandiant, CrowdStrike...)
3	**Microsoft Products**	Entra, Intune, Sentinel changes
4	**Third-Party**	Vendor product updates (configurable)
5	**LLM News & YouTube**	AI provider updates + YouTube transcript intel

"I kept building reports for things I had a hard time keeping up with"

THE REALITY

Challenges & Fixes

Parsing Problems

Some sources wouldn't work — added Playwright for JavaScript-heavy sites

Local Model Hallucinations

Paranoid the LLM was making things up — added guardrails and testing

Look and Feel

Spent HOURS refining summaries, parsing, formatting — iterative refinement

Security as an Afterthought

Didn’t lock down WSL initially — now I get why devs skip security to get code working

"Claude helped, but I still had to put in the work"

THE REALITY

What Didn’t Work

Agents Feature

Tried It

Created specialized agents for debugging, research, and design. Didn’t find much benefit from them.

What worked better: Direct prompting with manual review.

Using Multiple LLMs

Skipped It

Watched videos about using multiple LLMs on one project. Considered it, but Claude alone was doing a great job. Didn’t want to over-complicate things.

What worked better: Sticking with one tool I understood well.

"Keep it simple — don’t over-engineer your AI workflow. Not every feature is useful for every workflow."

SECINTEL-AI

Demo

DEMO

What You're About to See

1	**LM Studio (local)**	$0 — gpt-oss-20b runs on my desktop. Free after setup.
2	**Claude Haiku**	$0.77 (full 6-report batch) — Cloud, cheapest tier. ~23 min runtime.
3	**Claude Sonnet**	$2.39 (full 6-report batch) — Cloud, mid-tier. ~39 min runtime.
4	**Claude Opus**	$2.21 partial — Tier-1 rate limit (30K ITPM) blocked the full run. Premium models, premium friction.

"Same scrape. Different brains. You decide which fits."

DEMO

Config: Switch Local vs Cloud

```yaml
# config.yaml — switch ONE line to change provider

ai:
  provider: 'lmstudio'   # local — free, runs on your GPU
  # provider: 'claude'   # cloud — paid, uses Anthropic API

# LM Studio settings (when provider: 'lmstudio')
lmstudio:
  base_url: 'http://localhost:1234/v1'
  model: 'gpt-oss-20b'

# Claude settings (when provider: 'claude')
claude:
  api_key: 'sk-ant-...'
  model: 'claude-haiku-4-5-20251001'
```

Demo Commands

```
# Show available trackers
$ python secintel.py --list

# Test LM Studio connection
$ python secintel.py --test-connection

# Scrape threat intel feeds
$ python secintel.py --tracker threat_intel --scrape

# Run full analysis and generate report
$ python secintel.py --tracker threat_intel --tier 0 --full-run

# View generated HTML report
# reports/threat_intel/report_2026-01-31.html

# Weekly run — all trackers, Tier 1
$ python secintel.py --tracker all --tier 1 --full-run
```

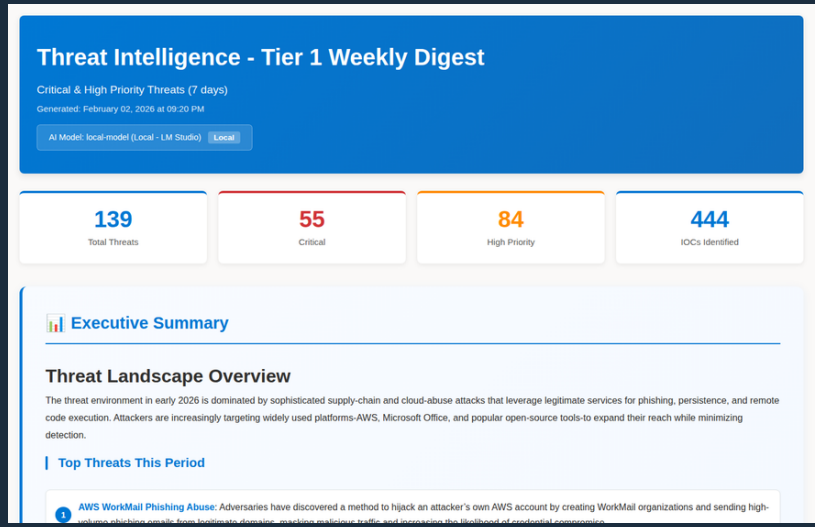
DEMO

Report Output

- Executive summary
- Articles grouped by source
- IOC extraction table
- Action items highlighted

Tiered Reporting System

Tier 0	Scrape & summarize 1 day
Tier 1	Scrape & summarize 1 week
Tier 2	Scrape & summarize 2 weeks
Tier 3	Scrape & summarize 1 month



Threat Intel — Tier 1 Weekly Digest (Feb 2, 2026)

139 threats • 55 critical • 444 IOCs • Summarized by LM Studio (local)

B E Y O N D C O D I N G

Home Lab Documentation

- Hardware inventory (servers, workstations, networking)
- Network architecture (7 VLANs incl. malware isolation)
- VMware configuration (15 VMs)
- Security procedures (malware handling, AI policies)
- Gave Insurance Estimate valuation

"Claude made it stupid easy"

Non-Coding Uses

- Documentation
- Runbooks
- Security SOPs
- Network planning

Beyond SecIntel-AI: Other Tools I've Built

1	**Vulnerability Tracker**	PingCastle + Purple Knight scans → SQLite + local dashboard. Cross-tool dedup. Tracks remediation over time.
2	**IR Tabletop Generator**	AI-generated incident response scenarios — 1-hour tabletops realistic to your environment. Removes prep barrier.
3	**Sec Tool Mapping**	Maps deployed security tools to MITRE ATT&CK. HTML heatmap, Navigator layer, gap recommendations.
4	**Career Tracker**	Career planning, salary data, progression paths. Non-security project — Claude Code is a Swiss Army knife.

"Same workflow. Different domains. Claude Code didn't just build SecIntel-AI — it built a portfolio."

WSL Security Lockdown

"Used Claude to protect myself FROM Claude"

Problem: Claude Code in WSL could access ALL Windows files — browser data, SSH keys, credentials

Layer 1: OS Level (WSL Config)

915-line config

- Disabled WSL automount + NAT mode networking
- Specific mounts only: transfer folder + OneDrive projects
- Claude can only see project folders

Layer 2: Manual Mode

`review everything`

- Manual mode — review every action before it runs
- Monitor connections — what & why
- Global CLAUDE.md: package install rules (anti-typosquat)

`WINDOWS (inaccessible) ← sudo required`

`WSL (Claude workspace) — projects only`

Defense in depth — WSL config + manual mode + SSH passphrase, all set up with Claude's help

Even This Presentation

Claude Code wrote the script to generate this PowerPoint...

But it can’t see .pptx files visually. First time in months I had to leave the terminal.

Switched to Claude GUI	Could have a conversation to fix things — upload the pptx, describe issues, get fixes
Uploaded My Notes	Fed it the docs I’d written (also with Claude) and it reviewed them and made recommendations
Know Your Tool’s Limits	Everything else — coding, debugging, docs, git — Claude Code handled. But visual work needed the GUI.

"Right tool for the right job — terminal for code, GUI for content"

Try Claude Code in Terminal

```
# Install via npm
$ npm install -g @anthropic-ai/claude-code

# Or via Homebrew (macOS)
$ brew install anthropic/tap/claude-code

# Login with your Claude subscription (Pro/Max)
$ claude login  # Don't use an API key — use your sub!

# Start using it
$ cd your-project && claude
```

Works on: macOS, Linux, Windows WSL

Best Practices

1	**Create a CLAUDE.md**	Describes architecture, patterns, and conventions for your project
2	**Be Specific**	The more context you provide, the better the results
3	**Review Everything**	Don't blindly trust — verify all changes before committing
4	**Iterate**	Don't expect perfection on first try: Prompt → Review → Refine → Repeat
5	**Document As You Go**	Open two terminals: one for coding, one for logging. Your future self will thank you.
6	**Reduce Hallucinations**	Tell Claude: don't guess — say if you don't know. Reduces wrong answers.

CALL TO ACTION

Resources

SecIntel-AI

github.com/lreuss07/secintel-ai • MIT Licensed • Local LLM or Cloud API • 23 threat intel feeds included

The Course That Started It

AI for Cybersecurity Professionals • Joff Thyer & Derek Banks • antisyphontraining.com

Claude Code

claude.ai/code • Official Anthropic CLI tool

"LLMs are not just chatbots to talk to and get answers — you can use them to generate code to automate IT/ Cybersecurity tasks."

- Ideas that took months now take days
- You don't need to be a developer
- Start with a problem you want to solve
- Keeping it to yourself doesn't help anyone

"Don't wait for perfect — just put it out there!"

Same lesson from my cybersecurity YouTube channel — waited years to start because I wanted it perfect. Just start.

THANK YOU

Special Thanks

Joff Thyer & Derek Banks

AI for Cybersecurity Professionals • WWHF Deadwood 2025 • The original PRISM POC

NetworkChuck

YouTube video that introduced me to Claude Code in the terminal

Anthropic

For building Claude Code

DC 402

For letting me share this with you

Questions?

LinkedIn	levireuss.com
X / Twitter	@levi_reuss
SecIntel-AI	github.com/lreuss07/secintel-ai
Course Link	antisyphontraining.com/product/ai-for-cybersecurity-professionals-with-joff-thyer-and-derek-banks