Compliance Detection Dashboard – Iteration 04

Ousmane Toure

November 10th, 2025

1 Dataset Description

This iteration extends the data foundation of the Compliance Detection Dashboard by combining three official datasets: the Electronic Code of Federal Regulations (ECFR), the MITRE Common Vulnerabilities and Exposures (CVE), and the National Vulnerability Database (NVD) from NIST. Each dataset is accessed through its public API and stored locally in structured JSON and CSV formats before ingestion.

The ECFR API exposes hierarchical legal text for federal regulatory titles and sections, while MITRE and NIST provide vulnerability data including CVE identifiers, CVSS scores, descriptions, and Common Platform Enumeration (CPE) references. Together, these sources form the foundation for automated correlation between policy language and technical vulnerabilities.

Dataset Links:

- ECFR API: `https://www.ecfr.gov/developers/documentation/api/v1`
- MITRE CVE API: `https://cveawg.mitre.org/api-docs/`
- NIST NVD API: `https://nvd.nist.gov/developers/vulnerabilities`

These datasets were chosen because they are authoritative, frequently updated, and essential for any real-world compliance or assurance workflow. Their structure supports both automated parsing and incremental updates, aligning directly with the project goal of linking federal policy and cybersecurity intelligence.

2 Tools and Methodologies

Development continues in Python 3 with a focus on automated ETL. The core scripts (`extract_parse.py` and `mitrecve_download.py`) handle the retrieval and transformation of XML and JSON payloads into flattened CSV structures. SQLite is used locally to validate relational joins between CVE, ECFR, and CPE entries before ingestion into Splunk.

Daily ingestion is automated using a Linux `cron.daily` job, ensuring consistent data refresh without manual triggers. Splunk serves as the visualization and correlation platform, where parsed datasets

populate a custom index named `sec_intel`. The Splunk dashboard supports exploratory filtering by vendor, regulation, and vulnerability severity.

GitHub maintains version control across scripts, documentation, and transformation logic. This iteration refined branch management and cron logging to improve reproducibility.

GitHub Repository: `https://github.com/ToureOus/Compliance-Detection-Dashboard`

3 Preliminary Timeline

Week	**Milestone and Deliverables**
Week 9	Refined ECFR and CVE schema mapping; produced validated merged dataset.
Week 10	Completed cross-reference logic between CPE and ECFR citations; documented schema joins.
Week 11	Integrated automated daily ingestion through cron; verified data integrity and log retention.
Week 12	Developed dashboard visuals for compliance–vulnerability correlation.
Week 13	Final documentation, Overleaf report submission, and full GitHub synchronization.

This schedule reflects the remaining stages of the semester and tracks both technical and documentation milestones through final submission.

4 Team Member Contributions

This is an independent project managed and executed solely by the author. All scripting, automation, database testing, and visualization design are developed by Ousmane Toure.

Work this iteration focused on completing ingestion automation, refining schema consistency, and improving traceability through GitHub commits. Each script update is logged, tested, and validated against sample CVE–ECFR intersections before release. Future adjustments will focus on enhancing dashboard interactivity and scaling ingestion to additional ECFR titles.

5 Progress and Next Steps

Progress to Date: The datasets from ECFR, MITRE CVE, and NVD have been collected and reviewed. The focus so far has been on analyzing the structure and unique attributes of each dataset to understand how they can be cross-referenced effectively. Initial exploration compared key identifiers such as CVE

IDs, CPE strings, and regulatory section numbers to identify potential mapping relationships. This phase helped clarify how the data sources differ in structure and terminology, laying the groundwork for the integration and schema design work that follows.

Ongoing Challenges: Some ECFR entries lack consistent formatting, requiring additional parsing logic. The CVE feeds occasionally exceed API rate limits during batch ingestion, so this will be mitigated through staggered calls and caching.

Next Steps:

- Extend dashboard filters for agency-specific views.
- Add validation scripts to track failed cron executions.
- Finalize Overleaf report and GitHub repository for project submission.

The project remains fully aligned with the DS5110 objectives of building reproducible, automated data systems and demonstrates a clear progression toward unifying regulatory and vulnerability datasets within a structured analytical framework.