Phishing Takedown Orchestrator – Week 1, Day 2 Report
Date: Friday, August 15, 2025
Location: Pune, Maharashtra, India

1) Objectives
- Implement centralized configuration with safe defaults.
- Define validated data models for pipeline records using Pydantic.
- Add a CLI command to inspect effective configuration.
- Write quick sanity tests to validate defaults and model behavior.

2) Files Added/Updated
- src/app/config.py — configuration loader and schema
- src/app/models.py — Pydantic models for pipeline data
- src/cli/main.py — CLI updated with show_config command
- examples/sample_config.yaml — example, overridable configuration
- tests/test_config_and_models.py — sanity tests

3) Configuration (src/app/config.py)
Overview:
- Loads from environment variables and optional YAML (examples/sample_config.yaml if present).
- Creates standard directories (artifacts, .outbox, .runs) automatically.
- Safe defaults: reporting disabled, headless browser with JavaScript off, sane timeouts.
- Single accessor get_config() returns an immutable Config object.

Key structures:
- BrowserSettings: headless, javascript_enabled, user_agent, nav_timeout_ms, max_redirects.
- Paths: root, artifacts_dir, outbox_dir, runs_dir (created on demand).
- Reporting: enable_reporting, SMTP settings, rate_limit_per_min.
- Sheets: Google Sheets identifiers (optional).
- Heuristics: suspicious_tlds, brand_keywords, url_length_warn.
- Config: top-level container for all settings, includes to_json() to print effective configuration.

Usage:
- from src.app.config import get_config
- cfg = get_config() # returns validated, cached configuration

4) Example Config (examples/sample_config.yaml)
Purpose:
- Documents configurable fields and mirrors defaults for quick edits without changing env.
- Keeps reporting disabled.
- Seeds heuristics (suspicious TLDs, common “brand-like” keywords) for future discovery logic.

Sample content includes:
- env: dev
- browser: headless true, JS disabled, 10s timeout, 5 redirects
- heuristics: suspicious TLDs and keywords
- reporting: disabled, 15 messages/min rate limit
- sheets: optional IDs and credentials path

5) Data Models (src/app/models.py)
Validated records for pipeline stages:

- Finding
  - url (validated), discovered_at (UTC now), source (default local_csv), risk_score optional
  - Trims source; falls back to “unknown” if empty.

- Evidence
  - url, final_url, redirects, screenshot_path, html_hash, html_size
  - Output of headless capture stage (to be added next).

- NetworkMeta
  - dns_records, ip, asn, tls_issuer
  - Populated during enrichment (DNS/TLS/ASN).

- Parties
  - registrar, registrar_abuse, hoster, hoster_abuse, brand_contact
  - Used by routing/reporting stages.

- Report
  - url, recipients, message_id, sent_at, attachments, status (“draft” default)
  - Tracks sent reports and status.

- Outcome
  - url, status (pending/reported/taken_down/false_positive/unknown), last_seen_http_status, observed_takedown_at, sla_days
  - Used by monitoring/recheck lifecycle.

Why Pydantic:
- Enforces correctness early (types/required fields).
- Easy JSON serialization for JSONL pipelines.

6) CLI Update (src/cli/main.py)
Added:
- show_config command to print effective configuration as JSON for inspection.
- discover remains a placeholder.

Example:
- python src/cli/main.py show_config
- python src/cli/main.py discover

Value:
- Verify configuration/paths without network actions.
- Confirms env and YAML are merged as expected.

7) Tests (tests/test_config_and_models.py)

Coverage:
- Ensures enable_reporting is False (safety default) and headless is True.
- Verifies artifacts_dir creation by config loader.
- Validates model creation for Finding, Evidence, Report using simple example.com URL.

Command:
- pytest -q

Outcome:
- Quick feedback loop that config and models are stable before adding I/O/network code.

8) Day 2 Deliverables

- Centralized configuration with safe defaults and auto-created directories.
- Strongly typed, validated data models for the pipeline.
- CLI command to print and inspect effective configuration.
- Sanity tests that pass locally.

9) Notes and Guardrails

- Reporting remains disabled globally; requires explicit opt-in.
- No network access yet; Day 2 remains safe/local.
- Artifacts and outbox paths are ignored by version control.
- Config is cached for consistent behavior within a process.

10) Next Steps (Preview of Day 3)

- Discovery MVP:
  - Read benign local CSV of URLs.
  - Normalize, deduplicate, apply simple heuristic filters.
  - Output JSONL list of Finding records, validated by Pydantic.

Prepared by: HARSHIL AMIT BUCH
Project: Phishing Takedown Orchestrator
Mentor/Guide: AI-assisted workflow in VS Code