

Phishing Takedown Orchestrator – Week 1, Day 1 Report

Date: Thursday, August 14, 2025

Location: Pune, Maharashtra, India

1. Goals

- Bootstrap the repository in Visual Studio Code
- Create essential metadata files
- Set up Python virtual environment and install dependencies
- Add CLI skeleton using Typer framework
- Configure debugging within VS Code

2. Project Setup in VS Code

Repository Initialization

- Created new project folder: `phish-takedown-orchestrator`
- Initialized Git repository (git init)
- Created core metadata files:
 - `README.md` – overview & project purpose
 - `LICENSE` – MIT placeholder
 - `CODE_OF_CONDUCT.md` – contributor guidelines
 - `CONTRIBUTING.md` – instructions for PRs
 - `SECURITY.md` – security report process
 - `ETHICS.md` – lawful use policy

Git Ignore & Env

- `.gitignore` to exclude sensitive and generated files
- `.env.example` with placeholder environmental variables for SMTP, Google Sheets, etc.

3. Python Environment & Dependencies

- Created `requirements.txt` with libraries:
 - `typer`, `pydantic`, `requests`, `httpx`, `dnspython`, `cryptography`,
`playwright`, `python-rdap`, `ipwhois`, `google-api-python-client`, `oauth2client`, `pytest`
- Created virtual environment and installed dependencies
- Installed Playwright browsers (python -m playwright install)

4. Directory Structure (Initial Scope)

```
src/app/  
src/cli/  
tests/  
docker/  
examples/  
docs/  
...
```

- Stubbed `config.py`, `models.py`, and `main.py`

5. CLI Skeleton Code

```
python
from typing import Optional
import typer

app = typer.Typer(help="Phishing Takedown Orchestrator CLI (Week 1 skeleton)")

def _version_callback(value: bool) -> None:
    if value:
        typer.echo("phish-takedown-orchestrator v0.1.0")
        raise typer.Exit()

@app.callback()
def main(
    version: Optional[bool] = typer.Option(
        None, "--version", "-v",
        help="Show the application's version and exit.",
        callback=_version_callback, is_eager=True
    )
) -> None:
    return

@app.command()
def discover() -> None:
    typer.echo("Discovery step placeholder")

if __name__ == "__main__":
    app()

Test Commands:
bash
python src/cli/main.py --help
python src/cli/main.py --version
python src/cli/main.py discover
```

6. Debugging Configuration in VS Code

Added `launch.json` for interactive debugging:

```
json
{
    "name": "Run CLI (discover)",
    "type": "python",
    "request": "launch",
    "program": "${workspaceFolder}/src/cli/main.py",
    "console": "integratedTerminal",
    "args": ["discover"]
}
```

}

7. Deliverables at End of Day 1

- Clean repo with all essential files and folder structure
- Python virtual environment with dependencies installed
- Working CLI (discover command placeholder)
- Debugger integration

Next Steps (Day 2)

- Implement `config.py` for environment/config loading
- Implement `models.py` for `Finding`, `Evidence`, `Report` dataclasses
- Add initial unit tests to validate models and config loading

Prepared by: [Harshil Buch]

Project: Phishing Takedown Orchestrator

Mentor/Guide: AI-based prompt workflow in VS Code