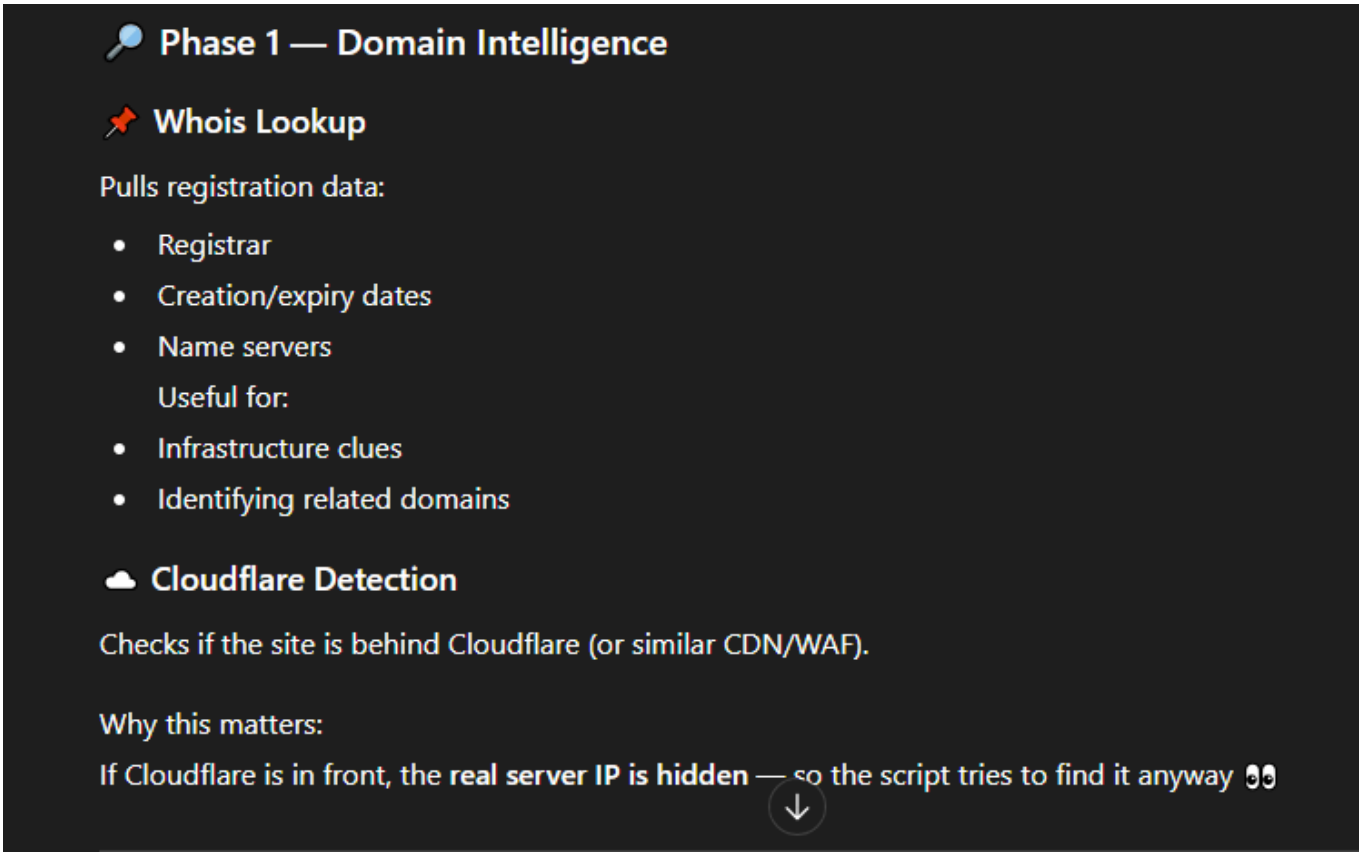
VanguardScan

- So this is a tool we made during this course (TCM) that executes a strong OSINT scan on a domain in an automated way...
- It combines a lot tools executions at once sequencelly...
- We execute this script "./VanguardScan.sh example.com"
- It goes by Phases...

Phase 1



- Here, we do the Whois Lookup thing

Phase 2

🌐 Phase 2 — Origin IP Discovery (If Behind Cloudflare)

Uses the **Censys API** to search internet-wide scan data for:

- Past exposures
- Misconfigured services
- Direct-to-origin IPs

This can reveal the **real server IP** that bypasses Cloudflare protection.

⚠️ This is a common red-team recon technique.

Phase 3

🌍 Phase 3 — Subdomain Enumeration

It runs **multiple tools** because each finds different things:

Tool	Strength
`subfinder`	Fast passive sources
`assetfinder`	Certificate transparency & public data
`amass`	Deep enumeration & graphing

All results are merged into:

Copy code

```
all_subs.txt
```

This builds the **full external attack surface**.

Phase 4

💡 Phase 4 — Find Which Hosts Are Actually Alive

Using:

```bash
httprobe
```

Copy code

It checks which subdomains actually respond over HTTP/HTTPS.

Output:

```
alive.txt
```

Copy code

Now you know which hosts are real web servers vs dead DNS entries.

Phase 5

🕷️ Phase 5 — Crawling & Endpoint Discovery

Tool used: **katana**

It crawls live websites to find:

- Hidden endpoints (`/api/` , `/admin` , etc.)
- JavaScript files
- Deep links not obvious from the homepage

This is how you discover **unlinked or forgotten functionality.**

Phase 6

📜 Phase 6 — Sensitive Data Scraping

From crawled content it extracts:

📧 Emails

Stored in:

```bash
scraped_data/emails.txt
```

Useful for:

- OSINT
- Phishing simulations (in legal engagements)

📜 JavaScript Files

Stored in:

```
javascript files.txt
```

JS files often contain:

- API endpoints
- Keys
- Hidden routes

🧨 Interesting Files

It fuzzes for common sensitive paths like:

- `.env`
- `.git/config`
- `robots.txt`
- Backup files

Saved in:

```
interesting_files.txt
```

These are **classic misconfiguration leaks.**

Phase 7

🚪 Phase 7 — Port Scanning

Runs **Nmap** on live hosts to discover:

- Open ports
- Running services
- Possible entry points beyond web (SSH, FTP, DBs...)

Output:

Copy code

```
nmap_scan.txt
```

Phase 8

🖼️ Phase 8 — Visual Recon

📸 Screenshots with `gowitness`

Takes screenshots of every live web page.

Why this is powerful:

- Quickly spot admin panels
- Identify tech stacks visually
- Detect staging/dev sites

Saved in:

Copy code

```
screenshots/
```

🎯 Favicon Download

Grabs the site favicon.

Favicons can be used to:

- Fingerprint frameworks
- Find other sites using the same icon

Final Output

📊 Final Output = Organized Recon Report

You end up with a structured folder like:

pgsql Copy code

```
VanguardScan_example.com/
│
├── summary.txt              ← High-level overview
├── all_subs.txt             ← Every discovered subdomain
├── alive.txt                ← Reachable web hosts
├── whois.txt
├── nmap_scan.txt
├── potential_origin_ips.txt ← Real IPs (if found)
│
├── scraped_data/
│   ├── emails.txt
│   ├── javascript_files.txt
│   └── interesting_files.txt
│
├── screenshots/
└── logs/                    ← Raw tool outputs
```

Conclusion

So instead of running **10+ tools manually**, you get:

> 🧠 OSINT
> 🌐 Surface mapping
> 🕷️ Endpoint discovery
> 🚪 Service scanning
> 📸 Visual recon
> 📂 Clean report

All in one go.