🛡️ DoD Cyber Sentinel CTF – June 2025 Write-Up

📁 Solved Challenges

1. Secret.txt Society (75 pts) — Web

Summary: Inspected `robots.txt` to discover a disallowed path leading to a hidden page with the flag.
Tools: Browser, manual recon.

---

2. Field Reports Mayhem (150 pts) — Web

Summary: Identified an IDOR by modifying the `id` parameter (`id=1234` → `id=1337`).
Tools: Browser, parameter manipulation.

---

3. Behind the Beat (75 pts) — Forensics

Summary: Used CyberChef to extract metadata from an MP3 file, revealing a hidden flag.
Tools: CyberChef (metadata extractor).

---

4. Hidden in Plain Sight (75 pts) — Forensics

Summary: Extracted hidden EXIF metadata from a social media image to reveal the flag.
Tools: CyberChef, EXIF tools.

---

5. Cafe Confidential (75 pts) — OSINT

Summary: Used Google reverse image search and timestamp matching to triangulate location and flag.
Tools: Google Reverse Image Search, Google Maps.

---

6. Packet Whisperer (75 pts) — Networking

Summary: Analyzed HTTP POST requests in a `.pcap` file with Wireshark to recover credentials.
Tools: Wireshark.

---

7. Problems in North TORbia (150 pts) — OSINT

Summary: Accessed a `.onion` ransom note in Tor Browser and extracted the flag from a hidden HTML field.
Tools: Tor Browser, HTML inspection.

---

8. Hardcoded Lies (75 pts) — Reverse Engineering

Summary: Used CyberChef to extract strings from a binary. Found a base64-encoded configuration string (flag).
Flag: `C1{n0_d3bug_n0_p4yn}`
Tools: CyberChef (strings + base64).

---

9. Encoded Evidence (75 pts) — Forensics

Summary: Analyzed a VBScript that fetched a base64 payload from Pastebin. Decoded to retrieve the flag.
Flag: `C1{n0_d3bug_n0_p4yn}`
Tools: curl, CyberChef.

❌ Unsolved Challenges – Investigative Notes & Methodology

🥖 Hoasted Toasted (150 pts) — Web Enumeration

Challenge Summary:
We were provided with a North Torbian public website:
`https://not-torbian.ethtrader-ai.com/`
Our objective was to discover a hidden, internal-only subdomain that contained a flag.

Steps Taken:

- Accessed the public site and reviewed HTML/JS for clues—no immediate references to internal content.

Launched DNS-based subdomain brute-force using `ffuf` with the `dns-Jhaddix.txt` wordlist:

bash
CopyEdit
```
ffuf -w /usr/share/seclists/Discovery/DNS/dns-Jhaddix.txt -u
https://not-torbian.ethtrader-ai.com -H "Host:
FUZZ.not-torbian.ethtrader-ai.com"
```

-

Tried a few manual `curl` commands using common internal names like:

bash
CopyEdit
```
curl -H "Host: internal.not-torbian.ethtrader-ai.com"
https://not-torbian.ethtrader-ai.com
```

-

What Went Wrong / Roadblocks:

- Long brute-force times (~30+ min per run).
- Likely missed a more concise or context-specific wordlist.
- Could have tried tools like `dnsx`, `crt.sh`, or `tls-scan` for hidden hostnames via certs.
- Did not inspect TLS certificate SANs or leverage deeper passive DNS intel.

🔐 overSSHaring (200 pts) — Network Forensics / Privilege Escalation

Challenge Summary:
A file share was exposed at `https://msoidentity.com/files/` containing multiple artifacts including Python scripts, config files, logs, and two disk images (`backup` and `sys`). The goal was to retrieve credentials for SSH access to `msoidentity.com`.

Steps Taken:

- Downloaded and reviewed all files from the directory listing.
- Inspected text-based files like `opsec.txt`, `log.txt`, and `config.json` for hardcoded secrets or usernames.
- Searched both disk images for keys and passwords:
    - Ran `strings`, `grep`, `binwalk`, and `fdisk`
    - Attempted to mount both with `kpartx` + `mount` after installing missing tools
- Enumerated for private keys, `.ssh`, and sensitive credentials using `strings`, regex, and key signatures.

What Went Wrong / Roadblocks:

- Unable to successfully mount `backup` or `sys` despite identifying sector sizes and using loop devices.
- Extraction tools (e.g. `binwalk -e`) failed—possibly due to unknown or unsupported formats.
- May have missed SSH keys stored with nonstandard file extensions or obfuscated.
- A more aggressive approach (e.g. `photorec`, `foremost`, `sleuthkit`) might have yielded results.

🤖 ChatAPT (150 pts) — AI Prompt Injection / Reverse Engineering

Challenge Summary:
We were instructed to connect via `nc ai.msoidentity.com 31337` to interact with an AI chatbot suspected of hiding a flag in its prompt or instructions.

Steps Taken:

Multiple attempts were made to connect using:

bash
CopyEdit
`nc ai.msoidentity.com 31337`

-
- Experienced difficulty connecting—likely due to high traffic or AI initialization time.

- Explored supporting files (`configuration_deepseek.py`, `config.json`) found in the file share. These pertain to a model named `DeepseekV3ForCausalLM`.

What Went Wrong / Roadblocks:

- Never fully established a working session with the AI over netcat.

- May have misjudged wait time (as the prompt warns connections can take ~30s).

- Didn't identify a reliable trigger to dump the system prompt or exploit via prompt injection.

- Possibly missed required interaction format or magic string to elicit instructions.