HOME LAB WRITE-UP

National Cyber League (NCL) Fall Season, Individual Game

Competition	National Cyber League - Fall Individual Game
Categories Covered	Web Exploitation, Digital Forensics, Cryptography, Log Analysis, Network Traffic Analysis (Packet Capture)
Tools Used	Burp Suite, CyberChef, Ghidra, Wireshark, Python
Outcome	Top 25% nationally; full score in the Forensics category

1. Problem / Objective

The National Cyber League (NCL) Fall Individual Game is a scored, capture-the-flag-style competition designed to evaluate and strengthen practical cybersecurity skills across a broad range of technical domains. My first objective going into this event was to demonstrate solid, well-rounded performance across all scored categories, those being web exploitation, digital forensics, cryptography, log analysis, and network traffic analysis. My second objective was to specifically close the gap in two areas identified as personal weak points from prior practice: log analysis and packet capture (PCAP) analysis.

Unlike a single-objective lab, the NCL Individual Game presents dozens of independently scored challenges of varying difficulty spread across the categories above. Each category requires a different mental model and toolset, so part of the underlying problem was building (or refreshing) fluency across all five domains within a compressed practice window, rather than mastering just one.

2. Environment and Tools

Practice and competition work was carried out in a personal home-lab environment I made consisting of an isolated virtual machine used for handling untrusted files (binaries, captures, and downloaded artifacts) and a host system for research and documentation. The table below summarizes which tools I used for which category during both practice and live competition.

Category	Primary Tools
Web Exploitation	Burp Suite (Proxy, Repeater, Intruder) for request interception, parameter tampering, and manual exploitation of injection and auth flaws.

Digital Forensics	Ghidra for static/binary analysis, CyberChef for data recipes and encoding pipelines, plus command-line file and metadata inspection utilities.
Cryptography	CyberChef for cipher identification, encoding/decoding chains, and hash analysis; Python for custom scripted attacks (e.g., brute force, XOR analysis).
Log Analysis	Python (pandas/regex-based parsing) for filtering, correlating, and searching large log sets for indicators of compromise.
Network Traffic Analysis	Wireshark for packet capture inspection, protocol filtering, stream reassembly, and credential/file extraction from PCAPs.

3. Solution Process

3.1 Pre-Competition Preparation

Before the competition window opened, I reviewed prior NCL practice scores and post-game breakdowns to identify weak categories. Log analysis and network traffic analysis (packet capture) stood out as my lowest-scoring areas, so the majority of my dedicated practice time in the two weeks leading up to the event was given to those two categories using NCL's practice environment, the "Gymnasium".

3.2 Web Exploitation

I approached the web challenges by first mapping the target application's functionality manually, then routing traffic through Burp Suite's proxy to observe requests and responses in detail. Burp's Repeater tool was used to iterate on modified requests (parameter tampering, header manipulation, and payload adjustment) while Intruder was used for automated fuzzing of input fields and endpoints where a manual approach would have been too slow. Common vulnerability classes targeted in this category included injection flaws, insecure direct object references, and authentication/session weaknesses.

3.3 Digital Forensics

The forensics challenges spanned file carving, metadata analysis, steganography, and reverse engineering of small, compiled binaries. Ghidra was used to statically analyze binaries, examining decompiled functions and strings to locate embedded flags or logic gating flag disclosure. CyberChef supplemented this work by rapidly chaining decoding operations (Base64, hex, XOR, and compression) on extracted data without needing to write one-off scripts for every transformation. This combination of static analysis plus a flexible decoding tool was the primary driver behind me achieving a full score in this category.

3.4 Cryptography

I tackled the Cryptography challenges by first identifying the likely cipher or encoding scheme (frequency analysis, structure of the ciphertext, and context clues from the challenge prompt), then using CyberChef's Magic and cipher modules to confirm the classification and attempt automated

decoding. Where CyberChef's built-in operations were insufficient (custom or multi-stage encodings, or brute-force key recovery), short Python scripts were written to script the attack directly, giving finer control than a GUI tool alone could give me.

3.5 Log Analysis

This was one of the two categories I specifically targeted for improvement when practicing. The prior approach of manually scrolling through raw log files was replaced with Python scripts using regular expressions and basic pandas filtering to parse logs into structured data, then filter and sort on fields such as timestamp, source IP, HTTP status code, and user agent. Building small, reusable parsing scripts before the competition (rather than writing them from scratch under time pressure) meant that during the live event, most log challenges were reduced to adjusting filter conditions rather than writing new parsing logic.

3.6 Network Traffic Analysis / Packet Capture

This was the second weak area I targeted, and the source of the biggest improvement from this competition. Earlier practice sessions involved opening a .pcap file in Wireshark and scrolling through packets looking for anything unusual, which took a lot of time and proved to be very inconsistent. To fix this, I made a personal checklist and used it for every capture:

- Check the Protocol Hierarchy view first to see which protocols are present and which are unusually large or unexpected for the given context.
- Apply Statistics > Conversations to identify the most active or most unusual IP pairs and ports before looking at individual packets.
- Use Follow TCP/UDP/HTTP Stream on flagged conversations to reconstruct full sessions instead of reading each packet.
- Filter for common exfiltration and credential exposure indicators (plaintext HTTP POSTs, FTP/Telnet traffic, DNS anomalies) before doing a general anomaly sweep.
- Export objects (File > Export Objects) for any HTTP/SMB/FTP transfer to recover transferred files directly rather than reconstructing them by hand.

Following this checklist consistently during practice made the packet capture challenges a lot faster and more repeatable during the live competition, since the first few minutes of every capture was spent gathering context instead of searching blindly.

4. Results

The Fall Individual Game concluded with my final placement in the top 25% of all national competitors. My performance was strongest in the Digital Forensics category, where a full (perfect) score was achieved, reflecting the effectiveness of the Ghidra plus CyberChef workflow developed during practice in the Gymnasium. The two categories that received the most dedicated pre-competition attention, log analysis and network traffic analysis, showed the clearest improvement compared to prior practice performance, validating that the targeted preparation strategy was effective.

Overall Placement	Top 25% nationally, Fall Individual Game
Strongest Category	Digital Forensics
Most Improved Categories	Log Analysis; Network Traffic Analysis (Packet Capture)

5. What I Learned / Key Takeaways

- Structured checklists beat ad-hoc searching. Building a repeatable, ordered checklist for Wireshark analysis turned packet capture from the weakest category into a much faster, more confident process. This same principle is being extended to the log analysis workflow going forward.
- Deliberate practice in weak areas pays off disproportionately. Focusing prep time on the two lowest-scoring categories from initial practice, rather than spreading time evenly, produced the largest measurable improvement.

- GUI tools and scripting are complementary. CyberChef and Burp Suite were fastest for exploration and rapid iteration, while Python scripting was more effective once a repeatable, well-defined task (e.g., log parsing, brute-force key search) had been identified. Knowing when to switch from a GUI tool to a script saved me significant time.
- Static analysis skills transfer across categories. Techniques originally practiced for forensics-oriented binary analysis in Ghidra (string searching, function tracing) also proved useful when investigating suspicious files recovered during packet capture and log challenges.

6. Conclusion

This competition served as an effective, structured way for me to benchmark and improve practical cybersecurity skills against a national group. Placing in the top 25% nationally, with a full score in forensics, confirmed that my existing web exploitation, forensics, and cryptography workflows were solid, while the targeted work on log analysis and packet capture closed a real skill gap. The checklist-driven approach developed for Wireshark analysis in particular is now a permanent addition to my personal home-lab methodology and will be applied to future competitions and real-world investigative work alike.