**Slide 1 — Title**

“Hello — I’m **[Your Name]**. Today I’ll present my Blue Team Labs Online challenge: **Network Analysis — Web Shell**.  
This exercise focused on investigating a SIEM alert for internal port scanning and finding evidence of a web shell upload.”

**Slide 2 — Project Summary**

“In short: I analyzed a 584 KB PCAP to investigate a ‘Local to Local Port Scanning’ alert.  
Using Wireshark, Tshark, and Tcpdump I confirmed scanning activity, identified suspicious HTTP POSTs, and recovered an uploaded PHP web shell.  
My goal was to determine what happened, who was involved, and produce detection and response recommendations.”

**Slide 3 — Goals & Objectives**

“The objectives were fourfold:

1. Confirm the scanning activity and identify the attacker and target IPs.
2. Reconstruct suspicious network sessions to find payloads.
3. Extract and preserve any malicious artifacts.
4. Produce a clear incident narrative and remediation guidance.”

**Slide 4 — Methodology (Short)**

“Steps I followed:

1. Extracted and unpacked the PCAP.
2. Performed a quick triage with **Tcpdump** and **Tshark** to find top talkers and protocols.
3. Used **Wireshark** filters to detect SYN scans and HTTP POST activity.
4. Exported HTTP objects and inspected files for web shell indicators.
5. Correlated timestamps and IPs with the SIEM alert to build a timeline.”

**Slide 5 — Evidence of Hands-on Activity**

“I carried out the analysis using both GUI and command-line tools:  
• **Tshark/Tcpdump** for fast triage and summary statistics.  
• **Wireshark** to follow TCP streams and export HTTP objects.  
• Exported suspicious files for offline inspection and hashing to preserve evidence.  
These steps gave me the raw facts to support the findings.”

**Slide 6 — Key Findings (bullet delivery)**

“Key findings:

* **Attacker IP:** 10.251.96.4.
* **Port range scanned:** 1–1024.
* **Type of scan:** TCP SYN scan.
* **Recon tools used:** Gobuster 3.0.1 and sqlmap 1.4.7.
* **Exploited file:** Editprofile.php (upload endpoint).
* **Uploaded shell file:** Dbfunctions.php.
* **Command parameter used:** cmd.
* **First command executed:** id.
* **Shell type:** Reverse shell (attacker established a reverse connection).”

**Slide 7 — Evidence Details (short)**

“I located repeated SYN packets from 10.251.96.4 hitting many ports, which matched the SIEM alert.  
Following the relevant TCP streams in Wireshark revealed HTTP POSTs to Editprofile.php carrying a file payload.  
Exporting HTTP objects produced Dbfunctions.php, which contained web-shell-like constructs and used the cmd parameter to execute system commands; the first command observed was id.  
Finally, we saw the attacker spawn a reverse shell connection to a listening port.”

**Slide 8 — Reflection & Learning**

“What I learned:

* Network captures often reveal the earliest indicators of compromise — scans precede exploitation.
* Combining GUI and CLI tools speeds triage (Wireshark for deep dives, Tshark for automation).
* Always preserve exported files and compute hashes for chain-of-custody.  
  How this helps me professionally: it improves my SOC triage skills, speeds investigation, and strengthens my ability to create clear incident reports.”

**Slide 9 — Recommendations**

“Recommended actions:

1. **Containment:** Isolate the compromised host and block 10.251.96.4 at the firewall.
2. **Eradication:** Remove the web shell, patch the vulnerable upload endpoint, and audit web application file upload handling.
3. **Detection:** Add SIEM rules for repeated HTTP POSTs to upload endpoints and anomalous internal scanning.
4. **Recovery & Monitoring:** Restore from clean backups and monitor for re-infection.”

**Slide 10 — Conclusion & Q/A**

“To conclude: this exercise confirmed internal scanning followed by a web shell upload and reverse shell. The analysis produced actionable forensic artifacts and clear remediation steps.  
Thank you — I’m happy to answer any questions