

Answers

Reference answers for Lesson 08.

Part 1 — Lab Answers

Exercise 1 — Vulnerable Server Code Review

Expected findings

- A stack-based buffer overflow on fixed-size buffer with unchecked input length.
- Calculated offset from input start to saved return address.
- Protections analysis: canary on or off, NX on, ASLR on or off.
- Verdict: likely RCE if no canary and attacker controls input length.

Exercise 2 — Crash Dump Analysis

Expected verdict

RIP contains attacker-controlled bytes (the signature of a control-flow hijack).
Offset computed from cyclic pattern or buffer boundary.

Exercise 3 — CVE to PoC

Expected PoC

A Python script that connects, sends the characteristic malformed input, and detects vulnerability by response code, connection reset, or timing delta. NO payload delivery.

Exercise 4 — Exploit Template Completion

Expected completion

Offsets filled, safety checks added (target IP allowlist, dry-run default, --check flag). The completed script should run safely in --check mode and refuse to deliver a payload without an explicit flag.

Final Exercise — Disclosure Report

Expected report

Connects source-code line → crash dump register state → CVE description → reproduction via --check PoC. CVSS 3.1 vector present and justified.

Part 2 — Homework Answers

Task 1 — Detection Signature

Expected rule

Suricata rule matching on the malformed packet length field or characteristic byte pattern. Rationale should state FP rate is low because the pattern is anomalous in benign traffic.

Task 2 — Fleet Scanner

Expected shape

~100 lines of Python. Uses `argparse`, reads a host list, enforces rate limits, writes CSV output, logs every probe. Never sends the weaponized payload.

Task 3 — Disclosure Draft

Expected tone

Professional, factual, cooperative. Includes a standard 90-day disclosure timeline, PGP key, and offers to help verify the patch. No hype, no public-disclosure threats in the first contact.