

Answers

Reference answers for Lesson 09.

Part 1 — Lab Answers

Exercise 1 — Target Profile Analysis

Expected training output

Per-person worksheets that name the specific OSINT leak (recent LinkedIn post, conference bio, public project) and the pretext class the person is most vulnerable to (authority for execs, curiosity for engineers, urgency for finance).

Exercise 2 — Phishing Email Forensics

Expected classifications

- One credential harvest (T1566.002).
- One malicious attachment (T1566.001).
- One BEC/invoice fraud (T1566.003 or T1534).
- Distinct pretext triggers per message: authority, urgency, curiosity.

Exercise 3 — Lookalike Domain Detection

Expected verdicts

Several typo-squats and at least one homograph. Recently-registered privacy-protected domains with MX records are the immediate "block" list.

Exercise 4 — Email Header Analysis

Expected findings

- At least one message with SPF/DKIM/DMARC failures that a permissive gateway let through.
- The true origin IP visible in the first external Received: header.
- Reply-to different from From:, a classic BEC tell.

Final Exercise — Campaign Reconstruction

Expected campaign profile

Moderately sophisticated attacker using OSINT for personalization and a small set of typo-squatted domains as infrastructure. Controls that failed: DMARC policy was "none" or "quarantine", gateway did not block newly-registered domains, employees lacked awareness for urgency pretext.

Part 2 — Homework Answers

Task 1 — Awareness Module

Expected tone

Concrete and concise. "Check the sender domain character by character" beats "be vigilant." A good module names one specific red flag per email instead of a

generic checklist.

Task 2 — DMARC Rollout

Expected plan

Phase 1 (30 days): `p=none; rua=mailto:reports@acmecorp.com`. Phase 2: `p=quarantine; pct=25`, increase weekly. Phase 3: `p=reject`. Monitoring: aggregate report volume, third-party SaaS senders that need alignment, legitimate mail flow impact.

Task 3 — Domain Monitor

Expected shape

~80 lines of Python. Generates character swaps, deletions, insertions, homoglyphs (Cyrillic a/o/e), TLD swaps. Optionally uses `socket.gethostbyname` or a DNS library for liveness checks.