

Lab Guide

Capstone: plan, execute, and report a simulated engagement using AI at every phase — scope, recon, vuln identification, and timeline reconstruction.

Authorized Use Only

All capstone artifacts are **synthetic**. Treat this exercise as a desk simulation: no packets, no probes, no emails sent outside the lab folder. In a real engagement the scope document is the contract — never exceed it.

Prerequisites

- Completed Lessons 01–09 (this capstone assumes familiarity with every prior tool and workflow)
- Access to an AI assistant (Claude, ChatGPT, or local LLM)
- Downloaded all four files into a folder named `capstone-lab/`
- Your notes / outputs from the previous nine labs — you will reference the techniques, not repeat them

Lab Scenario

You have been engaged for a full red team exercise at **AcmeCorp**. The client has signed a scope document, provided high-level objectives, and given you a 10-day window. A prior team did preliminary recon and left behind a data package. Your mission: read the scope, understand what is in and out, review the recon + vuln data, plan the engagement, execute it on paper, and produce a client-ready after-action report.

Download Lab Files

Save all four files into a folder named `capstone-lab/` before starting the exercises.

↓ [scope-document.txt](#) ~12 KB

↓ [recon-data.txt](#) ~8 KB

↓ [vulnerability-scan.txt](#) ~10 KB

↓ [attack-timeline.txt](#) ~10 KB

Exercise 1 — Scope Document Review

Open `scope-document.txt`. The scope document is the single most important artifact in a red team engagement — it is the contract.

Task 1.1 — Extract rules of engagement

You are a red team lead reviewing a signed statement of work before kickoff. Analyze the scope document and produce:

1. Assets IN SCOPE (IPs, domains, people, applications).
2. Assets explicitly OUT OF SCOPE (never touch).
3. Authorized techniques (recon / exploit / post-exploit / social engineering / physical / DoS).
4. Forbidden techniques (common exclusions: DoS, destructive actions, real executive phishing, production DB writes).
5. Engagement window (start / end / blackout periods).
6. Communication plan (who to contact, how, how often).
7. "Get out of jail" contacts and authorization letters.
8. Reporting format and deadline.
9. Any AMBIGUITY in the scope that should be clarified with the client BEFORE starting (this is the single most important output — ambiguous scopes cause lawsuits).
10. The single biggest risk in this engagement.

Output as a one-page pre-engagement briefing for the team.

[paste scope-document.txt]

✓ **Self-check:** Your briefing should include at least one ambiguity you would raise with the client before touching anything. In real engagements, every scope has at least one gap — finding it is the senior operator's job.

Exercise 2 — Recon Data Review

Open `recon-data.txt`. This is the package left by the prior team: WHOIS, DNS, subdomains, employee profiles, leaked credentials.

Task 2.1 — Validate and extend the recon

Below is recon data collected by a prior team. Your job is to:

1. Validate the data: flag anything out of date, unreliable, or outside scope (reference the scope document).
2. Identify gaps: what recon did they NOT do that you would add?
3. Produce a ranked target list:
 - External attack surface (domains, IPs, services)
 - Internal pivot candidates (VPN, webmail, jump hosts)
 - Human targets (phishing, vishing)
 - Leaked credentials / secrets
4. For each top target, state:
 - The technique from Lessons 01-09 you would apply
 - The expected difficulty
 - The stealth profile (loud / quiet / silent)
5. Map targets to MITRE ATT&CK Initial Access techniques.

Output as a prioritized engagement plan.

[paste recon-data.txt]

✓ **Self-check:** Every target in your plan should reference a specific lesson or technique you already know. If you cannot connect it to a prior lab, you do not actually know how to execute it — that is a training gap.

Exercise 3 — Vulnerability Scan Triage

Open `vulnerability-scan.txt`. This is the raw output from a vulnerability scanner — your job is to separate real findings from noise.

Task 3.1 — Pick the attack paths

Below is vulnerability scanner output covering the in-scope assets. Review as an operator planning exploitation:

1. Filter out: duplicates, false positives, informational findings, vulns on out-of-scope hosts.
2. For remaining findings, bucket by exploitability:
 - Public PoC / Metasploit module exists
 - Documented but no public exploit
 - Theoretical
3. For each HIGH and CRITICAL with a public PoC:
 - Host + port + service version
 - CVE + CVSS
 - Exploitation difficulty
 - Stealth (detectable by EDR/NIDS?)
 - Required post-exploitation (credential capture? persistence? lateral movement?)
4. Build a ranked attack path:
 - #1 primary path (most reliable)
 - #2 fallback (if primary fails or is detected)
 - #3 last-resort noisy option
5. For the #1 path, write the complete OPERATOR CHECKLIST:
 - Pre-checks (scope confirmation, logging evidence)
 - Commands to execute (step by step)
 - Expected outputs at each step
 - Rollback / abort criteria
 - What to do if detected

✓ **Self-check:** Your #1 attack path should be specific enough that another operator on your team could execute it from your notes alone, without asking you questions.

Exercise 4 — Attack Timeline Reconstruction

Open `attack-timeline.txt`. This is a log of actions that were actually taken during the engagement (simulation). Your job is to reconstruct the narrative.

Task 4.1 — Build the narrative

Below is a chronological action log from the red team engagement. Reconstruct the story:

1. Initial access: vector, time, success/failure.
2. Foothold: credentials captured, shell type, user context.
3. Privilege escalation: technique, success, detection risk.
4. Persistence mechanism (if any).
5. Lateral movement: which hosts were touched, how.
6. Objective achieved (domain admin / data exfil / flag file).
7. Detection events: when did blue team notice? What did they do?
8. Did the red team stay within scope at every step? (Flag any actions that pushed boundaries.)

For each step, map to MITRE ATT&CK (tactic + technique ID).

Then build a timeline diagram (text format is fine):

T+00:00 | Phase | Action | Outcome | MITRE ID

Finally: identify THREE moments where a defender could have detected and stopped the attack, and what detection rule would have fired.

[paste attack-timeline.txt]

✓ **Self-check:** Your timeline should read like a case study. Any step that does not map to a MITRE technique is probably mislabeled — go back and categorize it.

Final Exercise — Client-Ready Report

Task 5.1 — The after-action report

Using the outputs of Exercises 1-4, produce a client-ready red team engagement report in the standard format:

1. Executive Summary (3-5 sentences, written for a non-technical executive)
2. Engagement Scope (what was and was not tested)
3. Engagement Timeline (high-level narrative + key milestones)
4. Attack Chain Walkthrough (detailed, with evidence)
5. Findings (ranked by risk, each with: description, impact, evidence, recommendation, CVSS)
6. Detection Gaps (what the blue team missed and why)
7. Strategic Recommendations (3-5 items for the CISO)
8. Tactical Recommendations (10-15 items for IT/SecOps)
9. Appendix: raw artifacts, screenshots, commands

Format as markdown. Max 3000 words for the body; appendix unlimited. Use the findings from Exercise 3 and the timeline from Exercise 4 as the evidence base.

The tone should be: direct, evidence-based, no hyperbole, no "sky is falling" language. Treat the client as a partner.

✓ **Self-check: A good red team report is one the CISO can read in 10 minutes (exec summary + recommendations) and the SecOps team can implement from for 3 months (tactical recommendations). If either reader would feel lost, revise.**

Course Reflection

- Across all 10 lessons, where did AI give you the biggest speedup? Where did it fail you?
- What type of task still requires human judgment that AI cannot replace? Why?
- If you were building an AI-assisted red team workflow at a consulting firm, what would be your top 3 guardrails?

- What is one technique from this course you will use in your next engagement, and one you will NOT use (and why)?
- How has your mental model of "what AI is good at in offensive security" changed since Lesson 01?

Course Complete

You have finished the CTF Red Team course. If you have not yet, run through the companion **CTF Blue Team** course — the same scenarios, viewed from the defender's side. The best red teamers understand the blue team's tools and constraints inside out.