SEAS-8414 Week 05 Student Lab Guide

SEAS-8414 Week 05 Student Lab Guide: Autonomous Penetration Testing

Start Here: Download the Student ZIP

Before running any lab commands, download the Week 05 student package from the course site:

https://8414.bwater.io/downloads/labs/packages/seas8414-blackboard-week-05-2026.05.0_4e76a52f_aws8414.zip

The recommended path is the package Makefile:

```
unzip seas8414-blackboard-week-05-2026.05.0_4e76a52f_aws8414.zip
cd seas8414-blackboard-week-05-2026.05.0_4e76a52f_aws8414
make week05
```

The Makefile calls `run-week05-lab.sh`, extracts the nested runtime ZIP, starts the lab, runs the Week 05 API workflow, saves evidence under `lab-results/week-05/evidence/`, generates `lab-results/week-05/index.html`, and cleans up containers at exit.

You can also download the runner directly from https://8414.bwater.io/downloads/labs/scripts/run-week05-lab.sh.

Manual extraction uses two ZIP layers. First extract the weekly Blackboard ZIP, then extract the nested runtime ZIP:

```
unzip seas8414-blackboard-week-05-2026.05.0_4e76a52f_aws8414.zip
cd seas8414-blackboard-week-05-2026.05.0_4e76a52f_aws8414
unzip runtime/seas8414-student-lab-2026.05.0+4e76a52f.aws8414.zip
cd seas8414-student-lab-2026.05.0+4e76a52f.aws8414/student-lab
```

Run the instructions in this guide from that `student-lab/` directory. The matching screencast and LLM prompt are published next to the ZIP on the labs page:

- Screencast MP4: https://8414.bwater.io/downloads/labs/screencasts/phase05-lab-screencast.mp4
- LLM Prompt: https://8414.bwater.io/downloads/labs/prompts/phase05-lab-llm-prompt.md
- Run Script: https://8414.bwater.io/downloads/labs/scripts/run-week05-lab.sh

---

Breakwater Phase 5: Autonomous Penetration Testing Lab

Prescriptive Analytics -- Validating Attack Paths Through Controlled Exploitation

Phase 5 introduces the autonomous penetration testing engine of the Breakwater progressive scan pipeline: campaign orchestration, rule-based and PPO reinforcement learning agents, four safety modes, exploit module dispatch, evidence chain integrity, and MITRE ATT&CK coverage analysis. These capabilities transform the theoretical vulnerability data from Phases 1 through 4 into confirmed exploitation evidence.

Within the SEAS-8414 analytics taxonomy (Chapter 5, Section 5.1), this lab is **prescriptive analytics**: *given everything we know, what offensive tests should we run?* This is the first chapter that actively modifies (or simulates modifying) network state:

```
Descriptive   (Ch 1, Lab 1)  "What devices exist on the network?"
Diagnostic    (Ch 2, Lab 2)  "What are these devices doing?"
Detective     (Ch 3, Lab 3)  "What vulnerabilities do they have?"
Predictive    (Ch 4, Lab 4)  "What attack paths are likely?"
Prescriptive  (Ch 5, this lab) "What offensive tests should we run?"
Simulation    (Ch 6, Lab 6)  "What happens if we apply this fix?"
Autonomous    (Ch 7–12)      "Act on it"
```

The analytical tools you will practice include: Markov Decision Process formulation, sequential decision-making under uncertainty, reinforcement learning policy evaluation, safety constraint engineering, cryptographic evidence chain verification, reward shaping analysis, and MITRE ATT&CK kill-chain coverage mapping.

What to Expect

What you will do: Run 10 hands-on exercises that require you to design pentest campaigns, diagnose agent decision-making, compare agent strategies under varying conditions, verify evidence chain integrity, and produce engagement reports. You will launch campaigns via the API, but the intellectual work is in the design, comparison, and analysis of agent behavior.

How long it takes: Approximately 4 to 5 hours, including reading time and written analysis. Individual exercises range from 15 to 30 minutes. Several exercises require written deliverables (campaign strategy memos, CISO briefings, ethics assessments).

What you will learn:

- How safety modes constrain autonomous offensive operations and why each mode exists
- The behavioral differences between rule-based and PPO agents and when each is appropriate
- How evidence chain integrity enables forensic-grade pentest reporting
- How reward shaping influences agent exploration and exploitation strategies
- How to design pentest campaigns that maximise coverage while respecting operational constraints
- How to interpret and communicate pentest results to executive audiences

Prerequisites:

- Docker and Docker Compose installed
- `jq` installed (`brew install jq` on macOS, `apt install jq` on Linux)
- `curl` installed
- Completed Phase 1 through Phase 4 labs
- Basic understanding of penetration testing concepts (exploitation, lateral movement, pivoting)
- **Chapter 5 of the textbook read**, particularly Sections 5.1 through 5.9
- The Breakwater student lab environment running

Lab Environment Setup

```
cd student-lab/
docker compose up -d
sleep 30
docker compose ps
```

Register or authenticate and run prerequisite scan:

> Use `student@example.com`. The API validator rejects `.local` as a special-use domain, so the labs standardize on an RFC 2606 example domain.

```
TOKEN=$(curl -s -X POST http://localhost:8100/v1/auth/register \
  -H "Content-Type: application/json" \
  -d '{"email":"student@example.com","password":"SecurePass!2026","full_name":"Lab
      Student"}' \
  | jq -r '.access_token // empty')
if [ -z "$TOKEN" ]; then
  TOKEN=$(curl -s -X POST http://localhost:8100/v1/auth/login \
    -H "Content-Type: application/json" \
    -d '{"email":"student@example.com","password":"SecurePass!2026"}'
      \
    | jq -r '.access_token')
fi
```

```
# Get existing scan or create new one
SCAN_ID=$(curl -s "http://localhost:8100/v1/scanning/smart-
        scan/history?limit=1" \
  -H "Authorization: Bearer $TOKEN" | jq -r '.scans[0].scan_id')

echo "Token: ${TOKEN:0:20}..."
echo "Scan ID: $SCAN_ID"
```

If no scan exists, create one:

```
SCAN_ID=$(curl -s -X POST http://localhost:8100/v1/scanning/smart-scan
        \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"subnet":"172.30.0.0/24"}' | jq -r '.scan_id')
sleep 120
```

Exercise 1: Design and Compare Safety Mode Strategies

Time: ~25 minutes | **Difficulty:** Intermediate | **Textbook:** Sections 5.2, 5.7

Context

Breakwater supports four safety modes that progressively increase the level of real network interaction:

- **simulation**: No network traffic at all. Results are theoretical projections based on scan data.
- **shadow**: Sends reconnaissance traffic only (port scans, service enumeration). No exploitation.
- **controlled**: Exploitation allowed, but only against an approved target list.
- **autonomous**: Full exploitation with no target restrictions (lab environments only).

Task

Part A: Launch campaigns in three safety modes and compare results:

```
# Simulation mode (no network interaction)
SIM=$(curl -s -X POST http://localhost:8100/v1/pentest/run \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"scan_id\":\"$SCAN_ID\",\"safety_mode\":\"simulation\",\"max_actions\":100,\"agent_type\":\
SIM_ID=$(echo $SIM | jq -r '.data.campaign_id')
echo "Simulation: $(echo $SIM | jq '{findings:
        .data.report.confirmed_exploitable, actions:
        .data.report.timeline.total_actions, hosts:
        .data.report.timeline.unique_hosts_compromised}')"

# Shadow mode (recon only, no exploitation)
SHADOW=$(curl -s -X POST http://localhost:8100/v1/pentest/run \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"scan_id\":\"$SCAN_ID\",\"safety_mode\":\"shadow\",\"max_actions\":100,\"agent_type\":\"rul
echo "Shadow: $(echo $SHADOW | jq '{findings:
        .data.report.confirmed_exploitable, actions:
        .data.report.timeline.total_actions, hosts:
        .data.report.timeline.unique_hosts_compromised}')"

# Controlled mode (exploitation on approved targets only)
CTRL=$(curl -s -X POST http://localhost:8100/v1/pentest/run \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"scan_id\":\"$SCAN_ID\",\"safety_mode\":\"controlled\",\"max_actions\":100,\"agent_type\":\
        [\"172.30.0.10\",\"172.30.0.11\"]}")
echo "Controlled: $(echo $CTRL | jq '{findings:
        .data.report.confirmed_exploitable, actions:
        .data.report.timeline.total_actions, hosts:
        .data.report.timeline.unique_hosts_compromised}')"
```

Part B: Record your results in a comparison table.

Analysis Questions

1. **Quantify the safety-coverage tradeoff.** Simulation mode produces the most findings but zero confirmed exploitations. Shadow mode produces zero findings but provides real reconnaissance data. Controlled mode finds 0-3 confirmed vulnerabilities on 2 approved targets. For a production hospital network with 500 IoT devices, which mode would you recommend for: (a) quarterly compliance assessment, (b) pre-deployment testing of a new network segment, (c) investigating a suspected compromise? Justify each choice.

2. **Controlled mode target selection.** In controlled mode, you chose 172.30.0.10 and 172.30.0.11. If you could only approve 3 targets from the entire network, how would you choose them? Consider: BRS score, blast radius, attack path centrality, device criticality. Design a target selection algorithm and describe it in 4-5 steps.

3. **Legal and ethical analysis.** Shadow mode sends actual network traffic to devices. In what jurisdictions or regulatory frameworks might this require explicit authorization? Draft a three-sentence authorization request that a penetration tester would send to the network owner before running a shadow-mode campaign. Include scope, duration, and liability.

4. **Safety mode escalation protocol.** Design a four-phase escalation protocol for a production network assessment: start with simulation (Phase A), escalate to shadow (Phase B), then controlled (Phase C), and never use autonomous. For each phase transition, specify: what decision criteria trigger escalation, who must approve, and what additional controls are required.

Exercise 2: Analyse Rule-Based Agent Decision-Making

Time: ~25 minutes | **Difficulty:** Intermediate | **Textbook:** Section 5.4

Context

The rule-based agent uses a deterministic priority function to select actions at each step. The scoring function assigns: 100 points for default credential exploitation, 80 points for CVSS >= 7.0 CVE exploitation, 40 points for pivoting, and 20 points for service enumeration. This creates a predictable action sequence.

Task

Part A: Examine the agent's decision timeline:

```
# Get the timeline from the simulation campaign
curl -s http://localhost:8100/v1/pentest/$SIM_ID/timeline \
  -H "Authorization: Bearer $TOKEN" | jq '.data.entries[:15][] |
        {step: .step, action_type, target_ip, success, reward}'
```

Part B: Categorise actions by type and calculate success rates:

```
# Action type distribution
curl -s http://localhost:8100/v1/pentest/$SIM_ID/timeline \
  -H "Authorization: Bearer $TOKEN" | jq '{
    total_actions: (.data.entries | length),
    by_type: ([.data.entries[].action_type] | group_by(.) | map({type:
        .[0], count: length})),
    success_rate: (([.data.entries[] | select(.success == true)] |
        length) / ([.data.entries[]] | length) * 100 | round)
  }'
```

Analysis Questions

1. **Predict the first action.** Before looking at the timeline, predict what action the rule-based agent will execute first. Given that credential exploitation scores 100 and service enumeration scores 20, the agent should always try credentials first -- but only if it knows which devices have default credentials. If Phase 2 identified 12 devices with default credentials, how many credential exploitation actions should appear before the first service enumeration? Verify your prediction against the actual timeline.

2. **Failure analysis.** Identify any failed actions in the timeline (success == false). For each failure, diagnose the cause: did the agent attempt to exploit a device without sufficient information (enumeration first)? Did it try credentials that do not exist? Did

the target refuse connection? What penalty does the agent receive for each failure (-5.0 reward), and how does the cumulative penalty affect the overall campaign efficiency?

3. **Pivot timing analysis.** Find the first pivot action in the timeline. What step number does it appear at? What devices had the agent already compromised before pivoting? The rule-based agent pivots when its score (40 + vulnerability count bonus) exceeds available credential/CVE exploitation scores. Explain why pivoting should NOT occur too early (insufficient foothold) or too late (diminishing returns from current position).

4. **Design an improved scoring function.** The current scoring function is static: credentials always score 100, regardless of the target's BRS score or blast radius. Design a modified scoring function that considers: (a) target BRS score (higher BRS = higher priority), (b) blast radius (devices that reach more other devices), (c) whether the target has already been partially enumerated. Write the formula and explain the weights.

Exercise 3: Compare Rule-Based vs. PPO Agent Strategies

Time: ~30 minutes | **Difficulty:** Advanced | **Textbook:** Sections 5.4, 5.5

Context

The PPO (Proximal Policy Optimization) agent uses a learned policy to select actions, in contrast to the rule-based agent's deterministic priority function. The PPO agent can learn from experience but introduces stochasticity -- running the same campaign twice may produce different results.

Task

Part A: Run both agents with identical parameters and compare:

```
# Rule-based agent
RULE=$(curl -s -X POST http://localhost:8100/v1/pentest/run \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"scan_id\":\"$SCAN_ID\",\"safety_mode\":\"simulation\",\"max_actions\":100,\"agent_type\":\

# PPO agent
PPO=$(curl -s -X POST http://localhost:8100/v1/pentest/run \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"scan_id\":\"$SCAN_ID\",\"safety_mode\":\"simulation\",\"max_actions\":100,\"agent_type\":\

# Compare results
echo "Rule-based:" && echo $RULE | jq '.data.report | {confirmed:
        .confirmed_exploitable, actions: .timeline.total_actions,
        successful: .timeline.successful_actions, hosts:
        .timeline.unique_hosts_compromised}'
echo "PPO:" && echo $PPO | jq '.data.report | {confirmed:
        .confirmed_exploitable, actions: .timeline.total_actions,
        successful: .timeline.successful_actions, hosts:
        .timeline.unique_hosts_compromised}'
```

Part B: Run the PPO agent twice more and check for variance:

```
PPO2=$(curl -s -X POST http://localhost:8100/v1/pentest/run \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"scan_id\":\"$SCAN_ID\",\"safety_mode\":\"simulation\",\"max_actions\":100,\"agent_type\":\

PPO3=$(curl -s -X POST http://localhost:8100/v1/pentest/run \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"scan_id\":\"$SCAN_ID\",\"safety_mode\":\"simulation\",\"max_actions\":100,\"agent_type\":\

echo "PPO Run 1: $(echo $PPO | jq
        '.data.report.confirmed_exploitable')"
```

```
echo "PPO Run 2: $(echo $PPO2 | jq
        '.data.report.confirmed_exploitable')"
echo "PPO Run 3: $(echo $PPO3 | jq
        '.data.report.confirmed_exploitable')"
```

Analysis Questions

1. **Efficiency comparison.** Calculate the efficiency ratio (confirmed_exploitable / total_actions) for both agents. Which agent is more efficient? On a first run (PPO has no training history), the PPO agent uses a random initial policy. How would the efficiency comparison change after 50 training campaigns? Explain the exploration-exploitation tradeoff in RL terms.

2. **Determinism vs. stochasticity.** The rule-based agent produces identical results every time. The PPO agent produces different results each run. For a compliance audit that requires reproducible results, which agent must you use? For a red team engagement that needs to discover novel attack paths, which agent is preferred? Explain the tradeoff.

3. **Variance analysis.** Examine the three PPO runs. Calculate the standard deviation of confirmed_exploitable across runs. If the standard deviation is high, the PPO policy is unstable. If it is low, the policy has converged. What does high variance indicate about the training state of the PPO model? Propose a convergence criterion: "The PPO agent is considered trained when the standard deviation of findings across N runs drops below X."

4. **MDP formulation critique.** The pentest environment is modeled as an MDP with states (device access levels), actions (exploit, pivot, enumerate), and rewards (successful exploitation = +10.0, failure = -5.0). What aspects of real penetration testing does this MDP NOT capture? Consider: time pressure, defender response, tool availability, social engineering. Propose one modification to the MDP formulation that would make it more realistic.

Exercise 4: Explore the Action Budget Tradeoff

Time: ~25 minutes | **Difficulty:** Intermediate | **Textbook:** Section 5.6

Context

The `max_actions` parameter limits the total number of actions the agent can take. A small budget forces the agent to be selective; a large budget allows exhaustive testing. This exercise explores the diminishing returns of increasing the action budget.

Task

Run campaigns at five budget levels and record results:

```
for BUDGET in 20 50 100 200 500; do
  RESULT=$(curl -s -X POST http://localhost:8100/v1/pentest/run \
    -H "Authorization: Bearer $TOKEN" \
    -H "Content-Type: application/json" \
    -d "{\"scan_id\":\"$SCAN_ID\",\"safety_mode\":\"simulation\",\"max_actions\":$BUDGET,\"agent_ty
  echo "Budget $BUDGET: $(echo $RESULT | jq '{confirmed:
        .data.report.confirmed_exploitable, actions:
        .data.report.timeline.total_actions, hosts:
        .data.report.timeline.unique_hosts_compromised, efficiency:
        ((.data.report.confirmed_exploitable // 0) /
        (.data.report.timeline.total_actions // 1) * 100 | round)}')"
done
```

Record results in this table:

Budget	Actions Used	Confirmed	Hosts	Efficiency (%)
20	?	?	?	?
50	?	?	?	?
100	?	?	?	?
200	?	?	?	?
500	?	?	?	?

Analysis Questions

1. **Identify the saturation point.** At what budget level do findings stop increasing? This is the saturation point -- the network has no more exploitable vulnerabilities within the agent's capability. Calculate: what percentage of the total budget is wasted (actions taken but producing no new findings) at budget=500? Draw or describe the diminishing returns curve.

2. **Optimal budget recommendation.** For a production network with 200 devices, extrapolate from the lab data. If the lab's 22-device network saturates at approximately 100 actions, what budget would you recommend for 200 devices? Is the relationship linear (10x devices = 10x actions) or sublinear? Justify your estimate.

3. **Budget allocation strategy.** Instead of a single campaign with 200 actions, you could run 4 campaigns with 50 actions each, each targeting a different subnet or device class. Compare the two approaches: single-campaign provides depth (full lateral movement exploration) while multi-campaign provides breadth (coverage of different network segments). Which approach maximises total confirmed findings?

4. **Time-budget correlation.** Each action takes some amount of wall-clock time (in simulation: milliseconds; in controlled mode: seconds to minutes). For a controlled-mode engagement with a 4-hour window, how many actions can you realistically execute? If each action takes an average of 10 seconds, your budget is 4 * 3600 / 10 = 1,440 actions. Is this enough to saturate a 22-device network?

Exercise 5: Verify Evidence Chain Integrity

Time: ~25 minutes | **Difficulty:** Advanced | **Textbook:** Section 5.9

Context

Every pentest action produces an evidence entry with a cryptographic hash chain. Entry N's hash includes a reference to entry N-1's hash, creating a chain from the genesis entry to the final entry. This ensures that evidence cannot be tampered with after the fact -- any modification to an intermediate entry breaks the chain.

Task

Part A: Retrieve the evidence chain:

```
# Get first 5 evidence entries
curl -s http://localhost:8100/v1/pentest/$SIM_ID/evidence \
  -H "Authorization: Bearer $TOKEN" | jq '.data.evidence[:5][] |
        {step: .step, action_type: .details.method, target:
        .details.target, evidence_hash, previous_hash}'
```

Part B: Manually verify the hash chain for the first entry. The genesis entry has `previous_hash = "0" * 64`. Reconstruct the canonical JSON and compute SHA-256:

```
import hashlib, json

# Replace with actual values from the evidence output
entry_data = {
    "step": 0,
    "action_type": "...",  # from details.method
    "target_ip": "...",
    "target_port": ...,
    "success": True,  # or False
    "details": "...",  # from details
    "previous_hash": "0" * 64
}

canonical = json.dumps(entry_data, sort_keys=True, default=str)
computed_hash = hashlib.sha256(canonical.encode()).hexdigest()
print(f"Computed: {computed_hash}")
print(f"Expected: <evidence_hash from API>")
```

Analysis Questions

1. **Tamper detection scenario.** An adversary gains access to the pentest results database and changes one evidence entry's `success` field from `false` to `true` (making a failed exploitation appear successful). Explain, step by step, how the hash chain detects this tampering. What happens to the hash of the modified entry? What happens to all subsequent entries in the chain?

2. **Chain reconstruction challenge.** If the evidence chain has 50 entries and entry #25 is modified, how many entries would need to be re-hashed to make the chain valid again? Compare this to a system without hash chaining (where each entry is independently hashed). What is the advantage of chaining vs. independent hashing for tamper evidence?

3. **Genesis hash significance.** The first entry's `previous_hash` is `"0" * 64` (64 zero characters). Why not use a random value? What would happen if you used a random genesis hash -- would the chain still be verifiable? How does the fixed genesis ensure that the chain has a known starting point?

4. **Legal admissibility.** In a legal dispute, the evidence chain provides proof that pentest results were not altered after the engagement. What additional metadata would you include in each evidence entry to strengthen its legal admissibility? Consider: timestamps from a trusted time source, the identity of the tester, the scope authorisation reference, and the version of the scanning software.

Exercise 6: Analyse Lateral Movement Patterns

Time: ~25 minutes | **Difficulty:** Advanced | **Textbook:** Section 5.10

Context

Lateral movement is the attacker's progression from an initial foothold to high-value targets. The pentest agent pivots between hosts using credentials, exploits, and network adjacency. Analysing the movement pattern reveals the network's internal defence weaknesses.

Task

Part A: Extract lateral movement actions from the campaign timeline:

```
# Pivot and credential exploitation actions
curl -s http://localhost:8100/v1/pentest/$SIM_ID/timeline \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.data.entries[] | select(.action_type == "pivot" or
        .action_type == "exploit_default_creds" or .action_type ==
        "exploit_cve") | {step, action_type, target_ip, success}]'
```

Part B: Build a movement graph (list of source -> target transitions):

```
# Sequential movement: each successful action's target becomes the
        next action's source
curl -s http://localhost:8100/v1/pentest/$SIM_ID/timeline \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.data.entries[] | select(.success == true) | {step,
        action_type, target_ip}]'
```

Analysis Questions

1. **Draw the movement graph.** Using the successful action data, draw a directed graph showing the agent's path through the network. Label each edge with the technique used (credential exploitation, CVE exploitation, pivot). Identify: (a) the initial foothold, (b) the first pivot, (c) the deepest penetration point (furthest device from the entry point).

2. **Credential reuse detection.** Did the agent reuse credentials from one host to access another? If cameras A and B share the default credential `admin:12345`, the agent exploits A's credentials and then uses the same credentials on B. This is precisely how botnets like Mirai operate. How many unique credential sets did the agent use across all compromised hosts? What does this tell you about the network's credential hygiene?

3. **Chokepoint identification.** In the movement graph, identify any node through which the agent MUST pass to reach certain targets. This is a chokepoint. If you could place a security control (IDS, firewall rule, credential rotation) at one chokepoint, which would you choose to maximise disruption to the agent's lateral movement?

4. **Segmentation effectiveness prediction.** If you implemented the three-VLAN segmentation from Phase 4 Exercise 3, which lateral movement edges in your graph would be eliminated? How many of the agent's successful compromises would have been prevented? Estimate the percentage reduction in confirmed findings.

Exercise 7: Evaluate MITRE ATT&CK Coverage

Time: ~20 minutes | **Difficulty:** Intermediate | **Textbook:** Section 5.13.2

Context

MITRE ATT&CK coverage measures how many adversary techniques the pentest campaign exercised. Higher coverage means more of the attack surface was tested.

Task

```
# MITRE coverage from the campaign
curl -s http://localhost:8100/v1/pentest/mitre-coverage/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" | jq .
```

Analysis Questions

1. **Coverage gap analysis.** List the MITRE ICS techniques that were exercised (e.g., T0812 Default Credentials, T1046 Network Service Scanning). Then list techniques from the ATT&CK ICS matrix that were NOT exercised. For each uncovered technique, explain whether the gap is due to: (a) the agent's limited exploit modules, (b) the network lacking that vulnerability type, or (c) the action budget being exhausted before reaching that technique.

2. **Coverage as a quality metric.** If Campaign A covers 4 techniques and Campaign B covers 6 techniques, is Campaign B objectively better? Not necessarily -- if B's extra 2 techniques are low-severity misconfigurations while A covered 4 critical exploitation techniques. Design a weighted coverage metric that considers both the number and severity of covered techniques.

3. **Coverage improvement strategy.** To increase MITRE coverage from 4 techniques to 8, what changes would you make to: (a) the agent's action budget, (b) the approved target list, (c) the agent type (rule vs. PPO)? Which change has the highest expected impact on coverage?

4. **Red team vs. automated pentest.** A human red team would cover 15+ MITRE techniques because they use social engineering, physical access, and custom tooling. The automated agent covers 4-6 techniques. Does this mean the automated agent is inferior? Discuss the complementarity: what does the automated agent provide that a human red team cannot (scalability, reproducibility, frequency)?

Exercise 8: Design a Reward Shaping Experiment

Time: ~25 minutes | **Difficulty:** Advanced | **Textbook:** Section 5.6

Context

The agent's behavior is shaped by its reward function: +10.0 for successful exploitation, -5.0 for failed actions, +5.0 for discovering new hosts. Changing these rewards changes the agent's strategy -- higher exploitation rewards make it aggressive, higher failure penalties make it cautious.

Task

This exercise is a thought experiment. You cannot modify the reward function via the API, but you can predict and analyse how changes would affect behavior.

Part A: Record the current campaign's reward breakdown:

```
curl -s http://localhost:8100/v1/pentest/$SIM_ID/timeline \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{total_reward: ([.data.entries[].reward] | add | round),
         successful_actions: ([.data.entries[] | select(.success ==
         true)] | length), failed_actions: ([.data.entries[] |
         select(.success == false)] | length), avg_reward:
         ([.data.entries[].reward] | add / length * 100 | round /
         100)}'
```

Analysis Questions

1. **Reward analysis.** Calculate the total positive reward (from successes) and total negative reward (from failures). What is the net reward? If the agent has a net negative reward, it is spending more actions failing than succeeding. What does this indicate about the agent's strategy quality?

2. **Cautious agent design.** If you changed the failure penalty from -5.0 to -20.0 (4x harsher), predict how the agent's behavior would change. Would it attempt fewer risky exploits? Would it spend more time on enumeration (which always succeeds)? Would the total confirmed findings increase or decrease? Justify your prediction.

3. **Aggressive agent design.** If you changed the exploitation reward from +10.0 to +50.0 while keeping failure at -5.0, the agent would be heavily incentivised to attempt exploits. Predict: would it skip enumeration entirely? Would it try low-probability exploits more often? What is the risk of an overly aggressive agent in a controlled-mode engagement (hint: it might trigger security alarms or cause device instability)?

4. **Reward shaping for safety.** Design a reward function for a hospital network pentest where patient safety is the overriding concern. Propose rewards for: successful exploitation of a medical device (-100, because this could endanger patients), successful exploitation of an IT device (+10, standard), failed action on a medical device (-50), network disruption of any kind (-200). Explain how this reward structure would change the agent's target selection strategy.

Exercise 9: Produce a Penetration Test Engagement Report

Time: ~30 minutes | **Difficulty:** Advanced | **Textbook:** Section 5.13

Context

The engagement report is the primary deliverable of a penetration test. It must communicate findings to both technical teams (who will remediate) and executives (who will fund remediation).

Task

Part A: Run a comprehensive campaign:

```
FULL=$(curl -s -X POST http://localhost:8100/v1/pentest/run \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"scan_id\":\"$SCAN_ID\",\"safety_mode\":\"simulation\",\"max_actions\":200,\"agent_type\":\

FULL_ID=$(echo $FULL | jq -r '.data.campaign_id')

# Get the narrative report
curl -s http://localhost:8100/v1/pentest/$FULL_ID/report \
  -H "Authorization: Bearer $TOKEN" | jq '.data.narrative'

# Get the full evidence chain
curl -s http://localhost:8100/v1/pentest/$FULL_ID/evidence \
  -H "Authorization: Bearer $TOKEN" | jq '{total_evidence:
        (.data.evidence | length), by_type:
        ([.data.evidence[].details.method] | group_by(.) | map({type:
        .[0], count: length}))}'
```

Written Deliverable (Required)

Write a **one-page engagement report** with the following sections:

1. **Executive Summary** (3 sentences): What was tested, what was found, what is the most critical risk.

2. **Scope and Methodology** (2-3 sentences): Safety mode used, agent type, action budget, number of targets.

3. **Key Findings** (3-5 bullet points): Each finding should state: the device, the vulnerability exploited, the technique used (MITRE ID), and the business impact.

4. **Lateral Movement Summary** (2-3 sentences): How far did the agent penetrate from its initial foothold? What was the deepest compromise?

5. **Recommendations** (3 numbered items): Priority-ordered remediation actions, each with estimated cost and timeline.

6. **Data Handling Notice** (1 sentence): How the engagement data (evidence chain, credentials discovered) will be stored and when it will be destroyed.

Analysis Questions

1. **Report audience adaptation.** The same findings need to be communicated to three audiences: (a) the IT operations team (needs specific IPs, ports, and remediation steps), (b) the CISO (needs risk scores, business impact, and budget requests), (c) the board of directors (needs one paragraph on risk posture and ROI of remediation). Write the first sentence of the findings section for each audience.

2. **Theoretical vs. confirmed distinction.** The campaign found N theoretical vulnerabilities (from scan data) and M confirmed exploitable (from actual/simulated exploitation). Why is the distinction critical for remediation prioritisation? A CISO who sees "249 vulnerabilities" might panic; one who sees "8 confirmed exploitable paths" can take targeted action. How does the evidence chain support the "confirmed" classification?

3. **Data handling ethics.** The evidence chain contains: IP addresses, confirmed credentials (passwords may be redacted but their existence is documented), exploitation proof data. After the engagement, this data becomes a liability -- if leaked, it is an attacker's playbook. Design a data handling policy: encryption at rest, access control, retention period, and destruction method.

Exercise 10: Design a Complete Pentest Strategy for a Critical Infrastructure Network

Time: ~30 minutes | **Difficulty:** Advanced | **Textbook:** Sections 5.1-5.13

Context

This capstone exercise requires you to synthesise all Phase 5 concepts into a pentest engagement plan for a hypothetical critical infrastructure network.

Scenario

You are the lead penetration tester for a water treatment facility with:

- 50 PLCs (programmable logic controllers) managing chemical dosing
- 30 IP cameras for physical security
- 20 HMIs (human-machine interfaces) for operator workstations
- 10 network devices (switches, routers, firewalls)
- 5 historian servers collecting SCADA data

The facility operates 24/7. Any disruption to PLC operations could affect water quality for 100,000 residents. The last pentest was 3 years ago.

Written Deliverable (Required)

Produce a **pentest engagement plan** that includes:

1. **Safety mode selection** for each device class (PLC, camera, HMI, network, historian). Justify each choice. (Hint: PLCs should NEVER be tested in autonomous or controlled mode.)

2. **Agent selection** (rule-based or PPO) for each campaign phase. Justify based on reproducibility requirements and network sensitivity.

3. **Action budget allocation** across device classes. Total budget: 500 actions. How would you distribute them?

4. **Target prioritisation**: Which 10 devices would you test in the first campaign? Why?

5. **Evidence handling plan**: Who has access to the results? How long are they retained? What happens if the test discovers a credential that could compromise a PLC?

6. **Abort criteria**: Under what conditions would you immediately stop the engagement? Define three specific trigger conditions.

Analysis Questions

1. **Safety-critical pentest paradox.** The most critical devices (PLCs) are the ones you most need to test but the ones you least dare to test. How do you resolve this paradox? Describe a testing strategy for PLCs that provides meaningful security assurance without any risk of disruption. Consider simulation-only assessment combined with firmware analysis.

2. **Regulatory compliance.** The water treatment facility is subject to NIST 800-82 (ICS security) and the American Water Infrastructure Act. These require periodic security assessments but do not mandate penetration testing. Should the facility conduct a pentest anyway? What is the risk of conducting a pentest (potential disruption) vs. the risk of not conducting one (unknown vulnerabilities)?

3. **Agent training dilemma.** The PPO agent improves with training data. But training on the production network risks disruption, and training on a lab network may not transfer (different device types, network topology, vulnerability profiles). Propose a training strategy that provides relevant experience without production risk.

4. **Incident during engagement.** During the controlled-mode campaign, the agent discovers that an HMI has already been compromised by an external attacker (evidence: unknown SSH key, modified configuration files). What do you do? The pentest is now an incident response. Write a 3-step immediate response plan.

Cleanup

```
cd student-lab/ && docker compose down
```

Troubleshooting Reference

Symptom	Cause	Fix
"No hosts found for this scan"	Scan not completed or Redis data expired	Re-run prerequisite scan
Campaign returns 0 findings	Shadow mode or no exploitable vulns	Check safety_mode; ensure scan found vulns
PPO agent produces empty results	No environment data in Redis	Verify scan_id has host data
Evidence hash does not match	Incorrect canonical JSON reconstruction	Check sort_keys, previous_hash, detail stripping
Token is null	User already registered	Use login endpoint

Slide Reference Index

Exercise	Topic	Slides
1	Safety modes and coverage tradeoffs	001-015
2	Rule-based agent decision-making	016-025
3	PPO vs. rule-based comparison	026-035
4	Action budget optimisation	036-042
5	Evidence chain cryptographic integrity	043-050
6	Lateral movement analysis	051-058
7	MITRE ATT&CK coverage	059-065
8	Reward shaping and agent behaviour	066-072
9	Engagement reporting	073-080

Exercise	Topic	Slides
10	Critical infrastructure pentest design	081-095