SEAS-8414 Week 04 Student Lab Guide

SEAS-8414 Week 04 Student Lab Guide: Predictive Attack Path Analytics

Start Here: Download the Student ZIP

Before running any lab commands, download the Week 04 student package from the course site:

https://8414.bwater.io/downloads/labs/packages/seas8414-blackboard-week-04-2026.05.0_4e76a52f_aws8414.zip

The recommended path is the package Makefile:

```
unzip seas8414-blackboard-week-04-2026.05.0_4e76a52f_aws8414.zip
cd seas8414-blackboard-week-04-2026.05.0_4e76a52f_aws8414
make week04
```

The Makefile calls `run-week04-lab.sh`, extracts the nested runtime ZIP, starts the lab, runs the Week 04 API workflow, saves evidence under `lab-results/week-04/evidence/`, generates `lab-results/week-04/index.html`, and cleans up containers at exit.

You can also download the runner directly from https://8414.bwater.io/downloads/labs/scripts/run-week04-lab.sh.

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

```
unzip seas8414-blackboard-week-04-2026.05.0_4e76a52f_aws8414.zip
cd seas8414-blackboard-week-04-2026.05.0_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/phase04-lab-screencast.mp4
- LLM Prompt: https://8414.bwater.io/downloads/labs/prompts/phase04-lab-llm-prompt.md
- Run Script: https://8414.bwater.io/downloads/labs/scripts/run-week04-lab.sh

---

Breakwater Phase 4: Predictive Attack Path Analytics Lab

Predictive Analytics -- Modelling, Scoring, and Simulating Attack Paths

Phase 4 introduces automated attack graph construction, composite risk scoring (BRS), attack path computation, MITRE ATT&CK ICS mapping, STIX 2.1 threat intelligence export, what-if remediation simulation, behavioral anomaly detection, threat intelligence correlation, and risk timeline projection. These capabilities move Breakwater from reactive vulnerability reporting to predictive risk analytics.

Within the SEAS-8414 analytics taxonomy (Chapter 4, Section 4.1), this lab is **predictive analytics**: *given the vulnerabilities, what attack paths are likely?* This is the first chapter where the analysis generates new intelligence beyond what any single scan finding provides:

```
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, this lab) "What attack paths are likely?"
Prescriptive  (Ch 5, Lab 5)  "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: graph theory (attack graph construction, shortest-path algorithms, blast radius via reachability), composite risk scoring (multi-factor weighted aggregation), constrained optimisation (greedy remediation planning under budget constraints), information exchange formats (STIX 2.1), temporal risk modelling (logarithmic growth projection), and behavioral baseline analysis (anomaly detection across scans).

What to Expect

What you will do: Run 10 hands-on exercises that require you to interpret attack graphs, diagnose why devices receive specific risk scores, compare remediation strategies using what-if simulation, design optimal remediation plans under budget constraints, and build executive-level risk briefings. You will use the Breakwater API to generate data, but the intellectual work is in the analysis, comparison, and design.

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 (risk assessments, remediation plans, CISO briefings).

What you will learn:

- How attack graphs transform isolated vulnerability findings into connected risk intelligence
- Why composite risk scores (BRS) provide better prioritisation than CVSS scores alone

- How attack path analysis reveals lateral movement opportunities invisible in per-device assessments
- How what-if simulation enables evidence-based remediation planning
- How to interpret and export STIX 2.1 threat intelligence for cross-organisation sharing
- How temporal risk modelling creates urgency arguments for remediation investment

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 3 labs
- Familiarity with graph theory concepts (nodes, edges, directed graphs, shortest paths)
- Basic understanding of the MITRE ATT&CK framework
- **Chapter 4 of the textbook read**, particularly Sections 4.3 (Graph Construction), 4.4 (BRS), 4.5 (Attack Paths), and 4.8 (What-If)
- 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 get scan ID:

> 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

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"
```

> **Troubleshooting:** If `Scan ID: null`, run `python scan_report.py 172.30.0.0/24` to create a scan.

Phase 4 API Cheatsheet

Endpoint	Method	Description
`/v1/analytics/graph/{scan_id}`	GET	Attack graph (nodes + edges)
`/v1/analytics/attack-paths/{scan_id}`	GET	All computed attack paths
`/v1/analytics/attack-paths/{scan_id}/{ip}`	GET	Attack paths targeting a host
`/v1/analytics/brs/{scan_id}`	GET	BRS scores for all devices
`/v1/analytics/brs/{scan_id}/{ip}`	GET	BRS score breakdown for one device
`/v1/analytics/mitre/{scan_id}`	GET	MITRE ATT&CK ICS mappings
`/v1/analytics/mitre/{scan_id}/stix`	GET	STIX 2.1 bundle export
`/v1/analytics/what-if/{scan_id}`	POST	Simulate remediation actions
`/v1/analytics/what-if/{scan_id}/optimal-plan`	GET	Auto-generated optimal plan
`/v1/analytics/behavioral/{scan_id}`	GET	Behavioral baselines and anomalies
`/v1/analytics/threat-intel/{scan_id}`	GET	Threat intelligence matches
`/v1/analytics/segmentation/{scan_id}`	GET	Network segmentation score
`/v1/analytics/blast-radius/{scan_id}/{ip}`	GET	Blast radius for a device
`/v1/analytics/timeline/{scan_id}/{ip}`	GET	Risk timeline projection

Exercise 1: Interpret the Attack Graph and Identify Structural Vulnerabilities

Time: ~25 minutes | **Difficulty:** Intermediate | **Textbook:** Sections 4.3, 4.5

Context

An attack graph is a directed graph where nodes represent network entities (devices, services, subnets, entry points) and edges represent relationships an attacker could traverse. The graph is constructed automatically from scan data. Unlike a vulnerability list, the graph reveals *connectivity* -- how compromise of one device enables attacks on others.

Task

Part A: Retrieve the attack graph and characterise its structure:

```
# Graph overview
curl -s http://localhost:8100/v1/analytics/graph/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{node_count: .data.node_count, edge_count: .data.edge_count}'

# Node type distribution
curl -s http://localhost:8100/v1/analytics/graph/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.data.nodes[].node_type] | group_by(.) | map({type: .[0],
        count: length})'

# Edge type distribution
curl -s http://localhost:8100/v1/analytics/graph/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.data.edges[].edge_type] | group_by(.) | map({type: .[0],
        count: length})'
```

Part B: Identify the entry points and credential-sharing edges:

```
# Entry points (attacker's initial footholds)
curl -s http://localhost:8100/v1/analytics/graph/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.data.nodes[] | select(.node_type == "entry_point") | {id,
        label, ip}]'

# Credential-sharing edges (lateral movement via shared passwords)
curl -s http://localhost:8100/v1/analytics/graph/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.data.edges[] | select(.edge_type == "shares_credentials") |
        {source, target, weight}]'
```

Part C: Calculate the graph density and identify hub nodes:

```
# Graph density = edges / (nodes * (nodes - 1))
curl -s http://localhost:8100/v1/analytics/graph/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{nodes: .data.node_count, edges: .data.edge_count, density:
        (.data.edge_count / (.data.node_count * (.data.node_count -
        1)))}'
```

Analysis Questions

1. **Node multiplier effect.** The graph has more nodes than the number of hosts discovered in the scan. Explain why. How does the service-per-port expansion (each open port becomes a service node) affect the graph's utility for attack path analysis? Would a simpler graph with only device nodes miss important attack vectors?

2. **Edge weight semantics.** Examine the edge weight table:

Edge Type	Weight	Meaning
`shares_credentials`	0.1	Devices share default credentials
`exploitable_via`	0.3	Known CVE enables exploitation
`same_firmware`	0.4	Devices run same firmware version
`same_subnet`	0.5	Devices on same L2 subnet
`runs_service`	0.7	Device exposes a network service
`can_reach`	1.0	Entry point can reach a device

Lower weight means easier traversal. Why is `shares_credentials` (0.1) the easiest edge to traverse? In terms of attacker effort, compare: reusing a known password (0.1) vs. exploiting a CVE (0.3) vs. network reachability (1.0). Does this weight ordering match your intuition about real-world attack difficulty?

3. **Credential-sharing topology.** Examine the credential-sharing edges. If cameras A and B share credentials, an attacker who compromises A can immediately access B. Draw the credential-sharing subgraph (just the devices connected by `shares_credentials` edges). Identify any "credential clusters" -- groups of devices all sharing the same credentials. What single remediation action would break the largest credential cluster?

4. **Entry point classification.** What criteria does Breakwater use to classify a node as an entry point? Examine the entry points and check their device types and services. Propose a modification to the classification logic: should a device with default credentials on RTSP (video streaming) be an entry point even if it does not have a web interface? Justify your answer.

Exercise 2: Diagnose BRS Scores and Identify Scoring Anomalies

Time: ~25 minutes | **Difficulty:** Advanced | **Textbook:** Sections 4.4, 4.11

Context

The Breakwater Risk Score (BRS) combines six factors into a 0-10 composite score:

```
BRS = w_V*V + w_E*E + w_R*R + w_P*P + w_S*S - w_C*C
```

Where: V = Vulnerability (0.20), E = Exploitability (0.20), R = Reachability (0.20), P = Physical consequence (0.15), S = Supply chain (0.05), C = Compensating controls (0.20).

Task

Part A: Retrieve BRS scores and factor breakdowns for three different device types:

```
# Fleet-wide BRS summary
curl -s http://localhost:8100/v1/analytics/brs/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{avg_score: .data.avg_score, max_score: .data.max_score,
         count: .data.count}'
```

```
# Compare camera, NAS, and router
for IP in 172.30.0.10 172.30.0.30 172.30.0.32; do
  echo "--- $IP ---"
  curl -s http://localhost:8100/v1/analytics/brs/$SCAN_ID/$IP \
    -H "Authorization: Bearer $TOKEN" \
    | jq '{ip: .data.ip, score: .data.score, rating: .data.rating,
           V: .data.factors.vulnerability, E:
        .data.factors.exploitability,
           R: .data.factors.reachability, P: .data.factors.physical,
           S: .data.factors.supply_chain, C:
        .data.factors.compensating,
           top_contributors: .data.top_contributors}'
done
```

Part B: Find the device where BRS and CVSS rankings disagree:

```
# All devices sorted by BRS
curl -s http://localhost:8100/v1/analytics/brs/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.data.scores[] | {ip, brs: .score}] | sort_by(-.brs) | .[:5]'

# All devices sorted by max CVSS (from scan results)
curl -s http://localhost:8100/v1/scanning/smart-scan/$SCAN_ID/results
       \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.hosts[] | {ip, max_cvss: ((.cves // []) | map(.cvss_score //
       0) | max // 0), total_cves: (.cve_summary.total // 0)}] |
       sort_by(-.max_cvss) | .[:5]'
```

Part C: Check blast radius for the top BRS device vs. the top CVSS device:

```
# Blast radius comparison
for IP in 172.30.0.10 172.30.0.32; do
  echo "--- $IP ---"
  curl -s http://localhost:8100/v1/analytics/blast-radius/$SCAN_ID/$IP
       \
    -H "Authorization: Bearer $TOKEN" \
    | jq '{ip: .data.ip, reachable_count: .data.reachable_count}'
done
```

Analysis Questions

1. **Factor decomposition.** The camera has P=6.0 (physical consequence) because it controls a surveillance process. The NAS has P=0.0. The router has R=10.0 (maximum reachability). Manually recompute the BRS for each device using the formula and weights. Does your calculation match the API output? If there is a small difference, what rounding or normalisation step might explain it?

2. **BRS vs. CVSS ranking disagreement.** Identify a device that ranks higher by BRS than by CVSS (or vice versa). Explain why. The most common case is a router with moderate CVEs but maximum reachability -- CVSS ranks it lower because individual CVEs are less severe, but BRS ranks it higher because compromising the router gives access to everything.

3. **Weight sensitivity analysis.** The default weights assign equal importance to V, E, R, and C (0.20 each). If you changed the weights to prioritise reachability (R=0.40, V=0.15, E=0.15, P=0.10, S=0.05, C=0.15), which devices would move up in the ranking? Which would move down? For an OT/ICS environment where physical consequence matters most, propose a weight configuration and justify it.

4. **Compensating controls diagnostic.** The `C` (compensating controls) factor reduces BRS. Examine a device with non-zero `C`. What controls is the BRS engine detecting? If you applied network segmentation to the highest-BRS device (isolating it on its own VLAN), how would you expect the C factor to change? What other factors (R, E) would also be affected?

Exercise 3: Trace Attack Paths and Evaluate Segmentation

Time: ~25 minutes | **Difficulty:** Intermediate | **Textbook:** Sections 4.5, 4.5.5

Context

The attack path engine uses Yen's k-shortest-paths algorithm to find the easiest routes from entry points to high-value targets. Each path has a total weight (lower = easier), probability (`1/(1+weight)`), and estimated time.

Task

Part A: Retrieve attack paths and identify the easiest route:

```
# Top 5 easiest paths
curl -s http://localhost:8100/v1/analytics/attack-paths/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.data.paths[] | {path_id, description, total_weight,
        probability, estimated_time_hours, hop_count}] |
        sort_by(.total_weight) | .[:5]'

# Examine the easiest path step by step
curl -s http://localhost:8100/v1/analytics/attack-paths/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '.data.paths | sort_by(.total_weight) | .[0] | {path_id,
        description, total_weight, probability, steps: [.steps[] |
        {node_id, node_type, technique, weight}]}'
```

Part B: Compare paths to two different targets:

```
# Paths to the NAS vs. paths to a camera
for IP in 172.30.0.30 172.30.0.11; do
  echo "--- Paths to $IP ---"
  curl -s http://localhost:8100/v1/analytics/attack-paths/$SCAN_ID/$IP
       \
    -H "Authorization: Bearer $TOKEN" \
```

```
    | jq '{target: .data.target_ip, path_count: .data.count,
        easiest_weight: (if .data.count > 0 then
        [.data.paths[].total_weight] | min else null end)}'
done
```

Part C: Get the segmentation score:

```
curl -s http://localhost:8100/v1/analytics/segmentation/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" | jq .
```

Analysis Questions

1. **Path narration.** Take the easiest attack path and narrate it as a story: "An attacker first... then... then..." For each step, identify: the attacker's technique, the time estimate, and what the attacker gains (access to a device, lateral movement capability, data). This narrative format is how you would explain the risk to a non-technical executive.

2. **Path count as a risk metric.** Device A has 12 attack paths and device B has 4. Does this mean A is 3x more at risk? Not necessarily -- if all 12 paths to A go through the same chokepoint, blocking that chokepoint eliminates all 12 paths. Examine the paths to the device with the most paths. Do they share any common edges or nodes? Identify the chokepoint if one exists.

3. **Segmentation impact prediction.** The segmentation score is 0.0 (single flat subnet). If you split the network into three VLANs (cameras, NAS/storage, infrastructure), which attack paths would be eliminated? Estimate the new segmentation score and the reduction in attack path count. What inter-VLAN traffic would need to be explicitly allowed?

4. **Time estimation critique.** The path engine estimates time-to-compromise in hours. Examine the time estimates for credential-sharing edges (0.5 hours) vs. CVE exploitation edges (8 hours) vs. can_reach edges (72 hours). Are these estimates realistic? An automated tool like Mirai can try default credentials in seconds, not 30 minutes. Propose revised time estimates for automated vs. manual attackers and explain how this changes the path analysis.

Exercise 4: Map Findings to MITRE ATT&CK and Design Detection Rules

Time: ~25 minutes | **Difficulty:** Intermediate | **Textbook:** Section 4.6

Context

MITRE ATT&CK for ICS provides a standardised taxonomy of adversary behaviours. Breakwater maps scan findings to ATT&CK techniques using three strategies with different confidence levels: direct findings (1.0), CWE-to-ATT&CK inference (0.7), and device-type inference (0.5).

Task

Part A: Retrieve MITRE mappings and analyse tactic coverage:

```
# Summary
curl -s http://localhost:8100/v1/analytics/mitre/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{total_mappings: .data.total_mappings, unique_techniques:
        .data.unique_techniques}'

# Tactic distribution
curl -s http://localhost:8100/v1/analytics/mitre/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '.data.tactic_counts | to_entries | sort_by(-.value) |
        map({tactic: .key, count: .value})'

# Top techniques
curl -s http://localhost:8100/v1/analytics/mitre/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '.data.technique_counts | to_entries | sort_by(-.value) | .[:8]
        | map({technique: .key, count: .value})'
```

Part B: Compare technique mappings for a camera vs. a NAS:

```
for IP in 172.30.0.10 172.30.0.30; do
  echo "--- $IP ---"
  curl -s http://localhost:8100/v1/analytics/mitre/$SCAN_ID \
    -H "Authorization: Bearer $TOKEN" \
    | jq --arg ip "$IP" '.data.by_host[$ip] // [] | {ip: $ip,
        technique_count: length, tactics: [.[].tactic_name] | unique,
        techniques: [.[].technique_name] | unique}'
done
```

Analysis Questions

1. **Tactic gap analysis.** The MITRE ATT&CK ICS kill chain includes: Initial Access, Execution, Persistence, Privilege Escalation, Evasion, Discovery, Lateral Movement, Collection, Command and Control, Inhibit Response Function, Impair Process Control, Impact. Which tactics appear in your scan results? Which are ABSENT? For each absent tactic, explain whether the gap is because (a) the scan cannot detect that tactic, (b) the network does not have that vulnerability, or (c) the mapping logic does not cover it.

2. **Confidence calibration.** T0812 (Default Credentials) appears with confidence 1.0 because it was directly confirmed. A CWE-287-based mapping appears with confidence 0.7. A device-type inference appears with confidence 0.5. Design a SOC alert rule that uses these confidence levels: at what confidence threshold should a finding trigger an automatic incident ticket? A human review? A log entry only?

3. **Detection rule design.** For the most prevalent technique (likely T0812 -- Default Credentials), design a network-level detection rule that would alert when this technique is being actively exploited. Specify: what network traffic pattern would you look for, what log sources would you correlate, and how would you distinguish legitimate admin access from an attacker using default credentials?

4. **ATT&CK heatmap interpretation.** If lateral_movement is the most populated tactic, what does this tell you about the network's defensive posture? Compare this to a network where initial_access dominates. What would each pattern suggest about the attacker's likely entry strategy and the defender's priorities?

Exercise 5: Export and Evaluate a STIX 2.1 Bundle

Time: ~20 minutes | **Difficulty:** Intermediate | **Textbook:** Section 4.7

Context

STIX (Structured Threat Information Expression) 2.1 is the standard format for sharing cyber threat intelligence. Breakwater exports its findings as a STIX bundle that can be imported into threat intelligence platforms.

Task

Part A: Export the STIX bundle and analyse its structure:

```
# Export and save
curl -s http://localhost:8100/v1/analytics/mitre/$SCAN_ID/stix \
  -H "Authorization: Bearer $TOKEN" > /tmp/breakwater-stix-bundle.json

echo "Bundle size: $(wc -c < /tmp/breakwater-stix-bundle.json) bytes"

# Object type distribution
curl -s http://localhost:8100/v1/analytics/mitre/$SCAN_ID/stix \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.objects[].type] | group_by(.) | map({type: .[0], count:
        length})'

# Attack patterns
curl -s http://localhost:8100/v1/analytics/mitre/$SCAN_ID/stix \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.objects[] | select(.type == "attack-pattern") | {name,
        external_id: .external_references[0].external_id}]'
```

Analysis Questions

1. **Interoperability assessment.** The STIX bundle contains `ipv4-addr` objects with internal IP addresses (172.30.0.x). If you shared this bundle with a partner organisation via a TAXII server, what privacy and operational security concerns arise? Design a data sanitisation process that preserves threat intelligence value while removing organisation-specific identifiers.

2. **Completeness evaluation.** Compare the number of `vulnerability` objects in the STIX bundle to the number of unique CVEs in the scan results. Are they equal? If not, what CVEs were included or excluded? What criteria should determine whether a scan finding becomes a shared threat intelligence indicator?

3. **Platform integration scenario.** Your SOC uses OpenCTI as its threat intelligence platform. Describe the workflow: how would you import this STIX bundle, what automated correlations would OpenCTI perform, and what new detections would be enabled? If a peer organisation shared a STIX bundle indicating that CVE-2021-36260 was being actively exploited against their cameras, how would your platform correlate that with your scan data?

4. **Standards critique.** STIX 2.1 represents attacks as `attack-pattern` objects and vulnerabilities as `vulnerability` objects. But it does not natively represent "this device at this IP has this vulnerability with this confidence." What STIX relationship types would you use to encode device-specific vulnerability assignments? Is the current STIX schema sufficient for IoT vulnerability sharing?

Exercise 6: Compare Remediation Strategies Using What-If Simulation

Time: ~30 minutes | **Difficulty:** Advanced | **Textbook:** Sections 4.8, 4.8.4

Context

The what-if engine clones the attack graph, applies hypothetical remediation actions, and recomputes BRS scores and attack paths on the modified graph. This exercise requires you to design, execute, and compare multiple remediation strategies.

Task

Part A: Simulate three different single-action strategies and compare their impact:

```
# Strategy 1: Rotate credentials on the highest-BRS device
curl -s -X POST http://localhost:8100/v1/analytics/what-if/$SCAN_ID \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $TOKEN" \
  -d '{"actions": [{"action_type": "rotate_creds", "target_ip":
        "172.30.0.10", "cost": 1.0, "downtime_hours": 0.5}]}' \
  | jq '{strategy: "rotate_creds_camera", delta: .data.delta,
        paths_eliminated: .data.paths_eliminated, cost: 1.0}'

# Strategy 2: Patch the QNAP NAS
curl -s -X POST http://localhost:8100/v1/analytics/what-if/$SCAN_ID \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $TOKEN" \
  -d '{"actions": [{"action_type": "patch", "target_ip":
        "172.30.0.31", "cost": 3.0, "downtime_hours": 2.0}]}' \
  | jq '{strategy: "patch_nas", delta: .data.delta, paths_eliminated:
        .data.paths_eliminated, cost: 3.0}'

# Strategy 3: Segment the router
curl -s -X POST http://localhost:8100/v1/analytics/what-if/$SCAN_ID \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $TOKEN" \
```

```
  -d '{"actions": [{"action_type": "segment", "target_ip":
        "172.30.0.32", "cost": 5.0, "downtime_hours": 4.0}]}' \
  | jq '{strategy: "segment_router", delta: .data.delta,
        paths_eliminated: .data.paths_eliminated, cost: 5.0}'
```

Part B: Simulate a multi-action combined strategy:

```
# Combined: rotate creds + patch + segment
curl -s -X POST http://localhost:8100/v1/analytics/what-if/$SCAN_ID \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $TOKEN" \
  -d '{
    "actions": [
      {"action_type": "rotate_creds", "target_ip": "172.30.0.10",
        "cost": 1.0, "downtime_hours": 0.5},
      {"action_type": "rotate_creds", "target_ip": "172.30.0.30",
        "cost": 1.0, "downtime_hours": 0.5},
      {"action_type": "patch", "target_ip": "172.30.0.31", "cost":
        3.0, "downtime_hours": 2.0},
      {"action_type": "segment", "target_ip": "172.30.0.30", "cost":
        5.0, "downtime_hours": 4.0}
    ]
  }' | jq '{strategy: "combined", original: .data.original_score,
        simulated: .data.simulated_score, delta: .data.delta,
        paths_eliminated: .data.paths_eliminated}'
```

Part C: Compare the greedy optimal plan at different budget levels:

```
# Budget-constrained: 3 actions
curl -s "http://localhost:8100/v1/analytics/what-if/$SCAN_ID/optimal-
        plan?max_actions=3" \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{actions: (.data.actions | length), reduction:
        .data.score_reduction, cost: .data.total_cost, efficiency:
        .data.efficiency}'

# Moderate budget: 7 actions
curl -s "http://localhost:8100/v1/analytics/what-if/$SCAN_ID/optimal-
        plan?max_actions=7" \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{actions: (.data.actions | length), reduction:
        .data.score_reduction, cost: .data.total_cost, efficiency:
        .data.efficiency}'

# Full budget: 15 actions
curl -s "http://localhost:8100/v1/analytics/what-if/$SCAN_ID/optimal-
        plan?max_actions=15" \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{actions: (.data.actions | length), reduction:
        .data.score_reduction, cost: .data.total_cost, efficiency:
        .data.efficiency}'
```

Analysis Questions

1. **Efficiency ranking.** Compute the efficiency (delta/cost) for each single-action strategy. Which strategy provides the most BRS reduction per dollar? Which provides the most paths eliminated per dollar? Do these two metrics agree? If they

disagree, which metric is more important for security decision-making?

2. **Superadditivity test.** Compare the combined strategy's delta to the sum of the three individual strategies' deltas. Is the combined delta greater than, equal to, or less than the sum? Explain why (hint: credential rotation on device A eliminates credential-sharing edges to device B, which also benefits from the segmentation action -- they interact synergistically).

3. **Diminishing returns curve.** Plot (or describe) the relationship between action count and BRS reduction using the three budget levels. At what point does each additional action produce less than 0.1 BRS improvement? Use this to recommend an "optimal budget" for this network.

4. **Greedy vs. optimal.** The greedy algorithm selects the locally best action at each step. Describe a scenario where the greedy approach misses a better global plan. Hint: action A alone produces delta 0.5, action B alone produces delta 0.4, but actions A+B together produce delta 1.5 (synergy). The greedy algorithm would select A first, then re-evaluate -- but what if the order matters?

Exercise 7: Detect and Investigate Behavioral Anomalies

Time: ~20 minutes | **Difficulty:** Intermediate | **Textbook:** Sections 4.9, 4.10

Context

The behavioral baseline engine tracks device state across scans and detects deviations: new ports, missing devices, firmware changes, credential changes. Anomalies indicate either legitimate changes or security incidents.

Task

Part A: Examine the current baselines:

```
# Baseline overview
curl -s http://localhost:8100/v1/analytics/behavioral/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{baseline_count: .data.baseline_count}'

# Examine a specific device baseline
curl -s http://localhost:8100/v1/analytics/behavioral/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '.data.baselines["172.30.0.10"] | {ip, observation_count,
        ports, device_type, firmware_version, has_default_creds}'
```

Part B: Review the anomaly types and their severities:

Anomaly Type	Severity	Implication
`new_port`	medium	Unexpected service exposed
`missing_port`	low	Service disabled (could be legitimate)

Anomaly Type	Severity	Implication
`firmware_change`	high	Firmware updated (or tampered)
`credential_change`	critical	Default creds newly detected (factory reset?)
`new_device`	medium	Unauthorized device on network
`missing_device`	medium	Device offline or removed

Analysis Questions

1. **Anomaly scenario design.** For each of the six anomaly types, write a one-sentence scenario where the anomaly is (a) benign and (b) malicious. Example for `new_port`: (a) "Admin enabled SSH for maintenance" (b) "Attacker opened a backdoor listener." This exercise tests whether you can distinguish operational changes from security incidents.

2. **Baseline observation threshold.** The engine requires a minimum number of observations before flagging anomalies (default: 2). Why? What would happen if the threshold were 1 (flag anomalies on the first scan)? What would happen if it were 10 (require 10 scans before flagging)? Propose an optimal threshold for a weekly-scanned network and justify it.

3. **Factory reset detection.** A `credential_change` anomaly (critical severity) occurs when default credentials are newly detected on a device that previously did not have them. This strongly suggests a factory reset. Design an investigation playbook: what should the SOC analyst do in the first 15 minutes after this alert fires? List 5 specific actions in order.

4. **Shadow IT detection.** A `new_device` anomaly is critical in OT/ICS environments because unauthorized devices can create unmonitored attack surfaces. In a hospital, an unauthorized Raspberry Pi connected by a nurse to monitor patient metrics could bypass all security controls. Design a `new_device` response policy that balances security (quarantine the device immediately) with patient safety (the device might be monitoring a critical patient).

Exercise 8: Evaluate Threat Intelligence Enrichment

Time: ~20 minutes | **Difficulty:** Intermediate | **Textbook:** Sections 4.12, 4.13

Context

The threat intelligence engine matches scan findings against CISA's Known Exploited Vulnerabilities (KEV) catalog and other feeds. A KEV match means a CVE has confirmed in-the-wild exploitation, which dramatically changes its priority.

Task

Part A: Retrieve threat intelligence matches:

```
# Summary
curl -s http://localhost:8100/v1/analytics/threat-intel/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
```

```
  | jq '{indicators_checked: .data.indicators_checked, match_count:
        (.data.matches | length), kev_matches: .data.kev_matches}'

# KEV matches with details
curl -s http://localhost:8100/v1/analytics/threat-intel/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.data.matches[] | select(.indicator.source == "cisa_kev") |
        {ip, cve: .indicator.value, required_action:
        .indicator.metadata.required_action, due_date:
        .indicator.metadata.due_date}]'
```

> **Note:** If running in an air-gapped environment, KEV data may not be available. In that case, proceed with the analysis questions using hypothetical data.

Part B: Cross-reference KEV matches with BRS scores:

```
KEV_IPS=$(curl -s http://localhost:8100/v1/analytics/threat-
        intel/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq -r '[.data.matches[] | select(.indicator.source == "cisa_kev")
        | .ip] | unique | .[]' 2>/dev/null)

for IP in $KEV_IPS; do
  echo "--- $IP ---"
  curl -s http://localhost:8100/v1/analytics/brs/$SCAN_ID/$IP \
    -H "Authorization: Bearer $TOKEN" \
    | jq '{ip: .data.ip, brs: .data.score, exploitability:
        .data.factors.exploitability}'
done
```

Analysis Questions

1. **KEV vs. CVSS prioritisation.** A CVE with CVSS 6.5 appears in the CISA KEV catalog (confirmed exploited in the wild). Another CVE has CVSS 9.8 but is NOT in KEV (no known exploitation). Which should you patch first? CISA mandates federal agencies remediate KEV entries within the due date regardless of CVSS score. Should private organisations follow the same policy? Argue for and against.

2. **Threat feed strategy.** Design a threat intelligence feed integration for a SOC monitoring three threat landscapes: (a) ICS/SCADA threats (feed: CISA ICS-CERT advisories), (b) IoT botnet indicators (feed: BadPackets/GreyNoise), (c) ransomware indicators (feed: Abuse.ch). For each, describe what indicator types it provides (IPs, domains, CVEs, file hashes) and how you would correlate them with your scan data.

3. **Confidence calibration.** The threat intel engine assigns different confidence levels: CVE match = 1.0, port match = 0.5, IP match = 0.8. Why is a port-based match (e.g., "port 23 is commonly targeted by botnets") lower confidence than a CVE match? Design a scenario where a port-based match is more actionable than a CVE match.

4. **KEV aging analysis.** Examine the `due_date` fields on KEV matches. Some due dates may have passed (they were set years ago). What does a passed due date mean for a non-federal organisation? Should past-due KEV entries be treated as higher or lower priority than current KEV entries?

Exercise 9: Project Risk Forward and Build an Investment Case

Time: ~25 minutes | **Difficulty:** Advanced | **Textbook:** Sections 4.14, 4.15

Context

The risk timeline engine projects BRS scores forward using logarithmic growth: `projected = current + (10 - current) * (1 - e^(-rate * months))`. This models the increasing probability of exploitation over time as new exploits are published and attacker awareness grows.

Task

Part A: Project risk for three devices with different current scores:

```
for IP in 172.30.0.10 172.30.0.30 172.30.0.22; do
  echo "--- $IP ---"
  curl -s http://localhost:8100/v1/analytics/timeline/$SCAN_ID/$IP \
    -H "Authorization: Bearer $TOKEN" \
    | jq '{ip: .data.ip, current: .data.current_score,
           at_3mo: .data.projections[1].projected_score,
           at_12mo: .data.projections[3].projected_score,
           at_24mo: .data.projections[4].projected_score}'
done
```

Part B: Find when each device crosses the critical threshold:

```
DEVICE_IP="172.30.0.10"
curl -s http://localhost:8100/v1/analytics/timeline/$SCAN_ID/$DEVICE_IP
        \
  -H "Authorization: Bearer $TOKEN" \
  | jq '[.data.projections[] | select(.rating == "critical") |
        {months, projected_score, rating}] | .[0]'
```

Written Deliverable (Required)

Draft a **two-paragraph investment case** for the CISO, using the timeline projections:

Paragraph 1: "If no remediation action is taken on [device], its risk score will increase from [current] to [projected] within [months], crossing the critical threshold. The growth model predicts [X% increase] in the first 6 months, driven by [increasing exploit availability / attacker awareness / firmware aging]."

Paragraph 2: "The estimated cost of remediation today is [cost from Exercise 6]. The estimated cost of incident response if the device is compromised is [estimate -- use $50,000 for a camera breach, $200,000 for a NAS data breach, $500,000 for a router compromise]. At a [Y%] annual probability of compromise, the expected loss exceeds the remediation cost within [Z] months."

Analysis Questions

1. **Growth curve analysis.** The logarithmic model means risk grows faster initially and plateaus near 10.0. A device at BRS 8.0 grows slowly (little room left), while a device at BRS 3.0 grows quickly (lots of room). Which device is the better remediation investment: the one at 8.0 that will cross critical in 12 months, or the one at 3.0 that will reach 5.0 in 12 months? Justify your answer considering both the probability of compromise and the blast radius.

2. **Model limitations.** The 2% monthly aging rate is a fixed parameter. What events could make the actual rate much higher (e.g., a major zero-day affecting the device's vendor, a new botnet targeting the device type)? What events could make it lower (vendor releases a patch, device is moved to an isolated segment)? Propose a model modification that adjusts the rate based on threat intelligence feed activity.

3. **Comparative urgency.** Create a table comparing three devices: current BRS, months to critical, remediation cost, and expected loss if compromised. Use this to rank them by ROI of remediation. Does the ranking differ from the BRS-only ranking?

Exercise 10: Produce a Comprehensive Risk Assessment

Time: ~30 minutes | **Difficulty:** Advanced | **Textbook:** Sections 4.15, 4.16

Context

This capstone exercise requires you to synthesise all Phase 4 outputs into a single-page risk assessment suitable for executive decision-making.

Task

Gather all necessary data:

```
# Fleet BRS summary
curl -s http://localhost:8100/v1/analytics/brs/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{avg_brs: .data.avg_score, max_brs: .data.max_score,
         device_count: .data.count}'

# Attack path summary
curl -s http://localhost:8100/v1/analytics/attack-paths/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{total_paths: .data.count}'

# Segmentation score
curl -s http://localhost:8100/v1/analytics/segmentation/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" | jq '.data.segmentation_score'

# MITRE technique count
curl -s http://localhost:8100/v1/analytics/mitre/$SCAN_ID \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{techniques: .data.unique_techniques}'
```

```
# Optimal remediation plan (5 actions)
curl -s "http://localhost:8100/v1/analytics/what-if/$SCAN_ID/optimal-
        plan?max_actions=5" \
  -H "Authorization: Bearer $TOKEN" \
  | jq '{actions: (.data.actions | length), cost: .data.total_cost,
        reduction: .data.score_reduction, efficiency:
        .data.efficiency}'
```

Written Deliverable (Required)

Produce a **one-page risk assessment** with the following sections:

1. **Network Risk Posture** (3 sentences): Current average BRS, number of high/critical devices, segmentation score interpretation.

2. **Top Three Attack Paths** (3 bullet points): For each, state the entry point, the target, the technique chain, and the estimated time.

3. **MITRE ATT&CK Coverage** (2 sentences): Which tactics are represented, which are absent, and what this tells you about the network's exposure profile.

4. **Recommended Remediation Plan** (table): Top 5 actions from the optimal plan with cost, downtime, and expected BRS reduction.

5. **Risk Timeline** (2 sentences): If no action is taken, when does the average BRS cross the critical threshold? What is the projected cost exposure?

Analysis Questions

1. **Single metric selection.** If you could present only ONE metric to a board of directors to convey the network's security posture, which would you choose: average BRS, attack path count, segmentation score, or MITRE technique count? Defend your choice and explain why the other three are less suitable for this audience.

2. **Pre/post comparison.** Using the what-if results, describe the network's posture AFTER implementing the optimal remediation plan. How many attack paths remain? What is the new average BRS? Which devices are still at high risk and why?

3. **Phase 4 vs. Phase 3 insight.** What insights does Phase 4 (attack graphs, BRS, paths) provide that Phase 3 (per-device CVE lists) cannot? Specifically, identify one finding from Phase 4 that would be invisible in a Phase 3-only assessment (e.g., a device with low CVE count but high reachability, or a credential-sharing path that spans multiple low-risk devices).

4. **Continuous monitoring proposal.** Design a monitoring cadence for Phase 4 analytics: how often should the attack graph be rebuilt, BRS recalculated, and what-if simulations re-run? Consider the tradeoffs between computational cost, data freshness, and the speed at which the network changes.

Cleanup

```
rm -f /tmp/breakwater-stix-bundle.json
docker compose down -v    # If finished with the lab environment
```

Troubleshooting Reference

Symptom	Cause	Fix
`"Attack graph not found"` (404)	Scan did not complete all phases	Run a new scan and wait for completion
What-if returns 404	Redis was restarted, clearing host cache	Run a new scan
Empty threat intel matches	No internet access to CISA KEV	Expected in air-gapped labs; use hypothetical data
Behavioral baselines show 0 anomalies	First scan (no prior baseline)	Run a second scan to generate baselines
`400 Bad Request` on what-if	Invalid `action_type`	Use: `patch`, `segment`, `rotate_creds`, `disable_service`, `firewall_rule`
Optimal plan returns 0 actions	Fleet BRS already low	Verify scan found vulnerable devices

---

Slide Reference Index

Exercise	Topic	Slides
1	Attack graph structure and interpretation	010-025
2	BRS scoring and factor decomposition	026-038
3	Attack path analysis and segmentation	039-045
4	MITRE ATT&CK mapping and detection rules	046-048
5	STIX 2.1 export and interoperability	049-050
6	What-if remediation simulation	051-057
7	Behavioral anomaly detection	058-064
8	Threat intelligence enrichment	065-069
9	Risk timeline projection	101-105
10	Comprehensive risk assessment	106-111