SEAS-8414 CYBER ANALYTICS

Predictive Attack Path Analytics

Graph Construction, Risk Scoring & Remediation Simulation

Dr. Mallarapu · Breakwater Security Platform

What Phases 1-3 Built

The analytical foundation for attack path computation

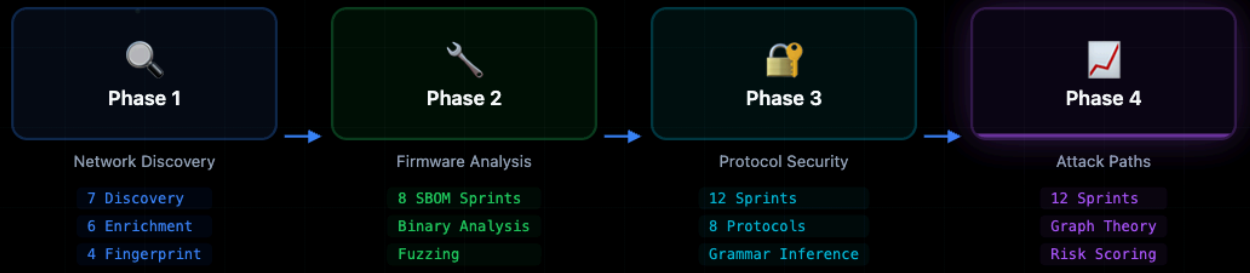
CHAPTER TAKEAWAY

Five arcs today.

ENRICHMENT VALUE

Fifth, what-if remediation simulation.

PROGRESSIVE ANALYTICAL ARCHITECTURE



Phase 1
What is on the network?
573 tests

Phase 2
What is inside each device?
791 tests

Phase 3
How do devices behave on the wire?
2,101 tests

Phase 4
How do risks compound across the network?
2,309 tests

The Phase 3 Limitation

What per-device analysis cannot tell us

CHAPTER TAKEAWAY

Chapter 1 told us what devices exist. Descriptive.

ENRICHMENT VALUE

Today we compose those outputs into a graph.

✔ What We Know

- **Individual CVEs per device**
  47 devices have critical CVEs
- **Default credentials found**
  23 devices with factory passwords
- **Firmware versions analyzed**
  12 with no vendor support
- **Protocol vulnerabilities**
  Per-device fuzzing results

VS

? What We Don't Know

- **How vulnerabilities combine**
  Cross-device attack chains
- **Which device to fix first**
  Topology-aware prioritization
- **Attacker lateral movement**
  Multi-hop path computation
- **Remediation ROI**
  Risk reduction per action

Hospital Network: Why Topology Matters

A medium CVE on a connected device outranks a critical CVE on a segmented one

CHAPTER TAKEAWAY

Keep one scenario in mind all class.

ENRICHMENT VALUE

Treat the scenario numbers as a worked example built to match the chapter model.

Critical CVE on segmented camera

CVSS 9.8 · No route to critical assets

LOW PRIORITY

Medium CVE on flat-network AP

CVSS 5.3 · 4-hop path to BMS

HIGH PRIORITY

Twelve Sprints

From graph construction to deployment-ready analytics

CHAPTER TAKEAWAY

The report answered "what is wrong."

ENRICHMENT VALUE

If an attacker compromises the camera, where do they go?

Graph Risk Intelligence Integration

01 Graph Builder GRAPH	02 BRS Engine RISK	03 Path Engine GRAPH	04 MITRE Mapper INTELLIGENCE
05 What-If Engine RISK	06 Behavioral Baseline INTELLIGENCE	07 Threat Intel INTELLIGENCE	08 Consequence Predictor RISK
09 Risk Timeline RISK	10 STIX Exporter INTEGRATION	11 Dashboard Analytics INTEGRATION	12 Report Generation INTEGRATION

Eight Analytical Paradigms

Each module answers a different question about network risk

CHAPTER TAKEAWAY

Look at the taxonomy table.

ENRICHMENT VALUE

Predictive analytics asks "what attack paths are likely?"

Descriptive

GRAPH

What exists?

Diagnostic

BRS

Why is it risky?

Predictive

PATHS

What could happen?

Prescriptive

WHAT-IF

What should we fix?

Temporal

BASELINE

What changed?

Contextual

THREAT INTEL

Who is attacking?

Causal

CONSEQUENCE

What breaks next?

Prognostic

TIMELINE

When does risk peak?

Phase 4 Architecture

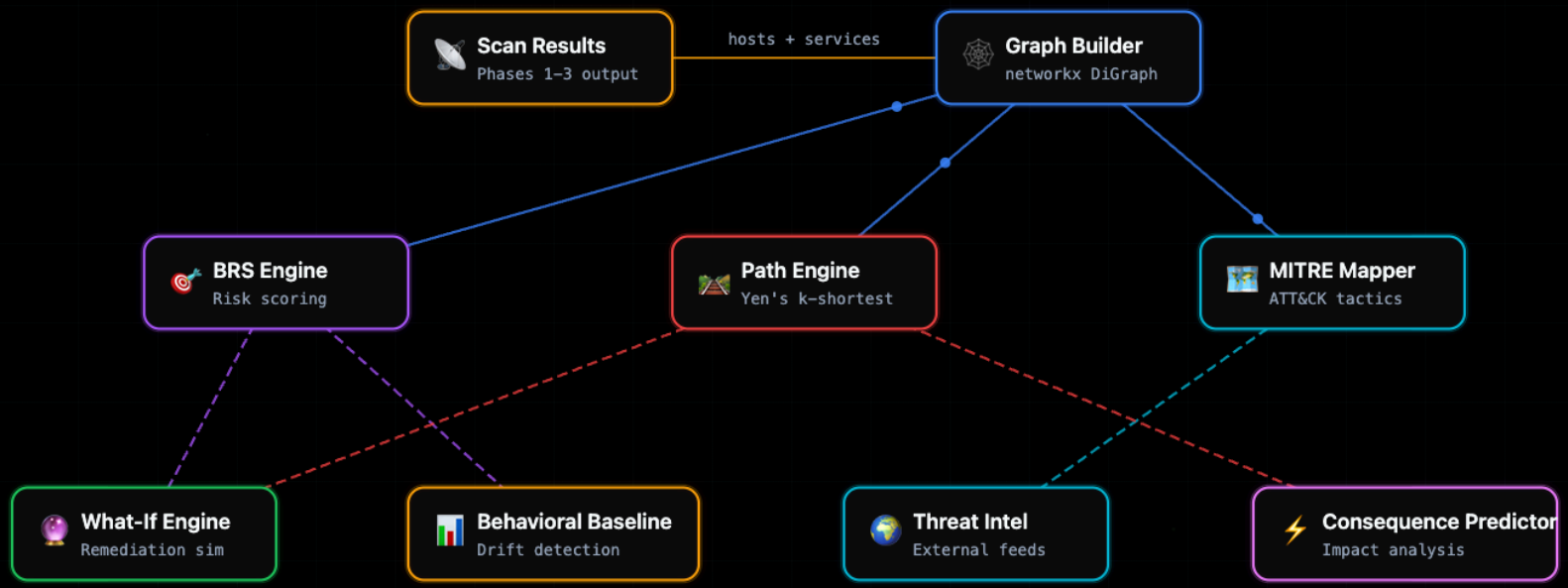
Three-tier analytics pipeline: Ingest, Analyze, Recommend

CHAPTER TAKEAWAY

The attack graph consumes four data categories.

ENRICHMENT VALUE

Quality cascades. A missed device in Phase 1 is a missing node in Phase 4.



Data Flow Layers

Each layer adds analytical depth to the raw scan results

CHAPTER TAKEAWAY

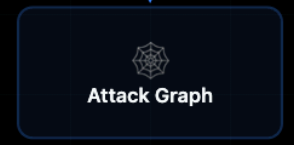
Here is the path that the vulnerability report missed.

ENRICHMENT VALUE

A flat report shows none of this.

IP addresses, open ports, banners, CVE IDs

`47 hosts`
`312 services`
`89 CVEs`



Adds topology, reachability, and edge weights

`47 device nodes`
`312 service nodes`
`1,204 edges`

Adds composite risk, shortest paths, MITRE mappings

`BRS per host`
`k-shortest paths`

SECTION 02

Attack Graph Construction

From Scan Results to networkx DiGraph

Why networkx?

Choosing the right graph library for security analytics

CHAPTER TAKEAWAY

An attack graph is a directed graph. G equals V and E.

ENRICHMENT VALUE

Credential sharing is symmetric. Exploitation is not.

⚖ Alternatives

igraph
10x faster, compiled C backend, great for massive social graphs

graph-tool
C++ backend with OpenMP parallelism, excellent for statistical analysis

Neo4j / ArangoDB
Persistent graph DB, Cypher queries, but adds operational complexity

Custom adjacency lists
Maximum control, zero deps, but re-inventing algorithms

VS

✔ networkx Wins

Yen's k-shortest paths built-in
nx.shortest_simple_paths() -- exact algorithm we need

Arbitrary dict attributes
Nodes/edges carry CVE data, BRS scores, MITRE mappings natively

Zero compiled dependencies
Pure Python -- pip install, no build tools, runs everywhere

Standard ecosystem
matplotlib, pandas, scipy integration out of the box

Five Node Types

Each node type models a different element in the attack surface

CHAPTER TAKEAWAY

Device nodes. One per host. IP, MAC, vendor, device type.

ENRICHMENT VALUE

Physical process nodes. Crane control. Surveillance. HVAC. Terminal.

device

Physical or virtual host on the network

`e.g. NVR at 10.0.5.22`

`ip` `mac` `os` `device_type` `brs_score`

service

Listening port with protocol and version

`e.g. RTSP on :554 (v2.0)`

`port` `protocol` `version` `cpe` `cve_count`

subnet

L3 network segment with routing context

`e.g. 10.0.12.0/24 (OT VLAN)`

`cidr` `vlan_id` `zone` `gateway_ip`

entry_point

Externally reachable interface or service

`e.g. Guest WiFi AP (WAN)`

`interface` `exposure` `auth_type` `acl_rules`

physical_process

Real-world system controlled by a device

`e.g. Building HVAC Loop`

`process_name` `safety_level` `sil_rating` `impact`

Edge Weights = Attacker Effort

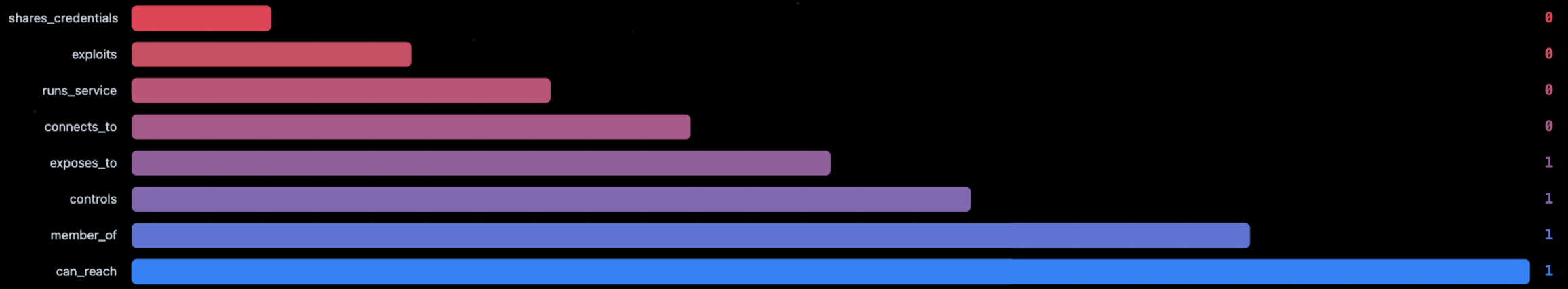
Shortest path algorithm minimizes total weight (sum of traversal costs)

CHAPTER TAKEAWAY

Eight edge types. Each with a default weight.

ENRICHMENT VALUE

can_reach: 1.0. Generic L3 reachability.



Lower weight = easier traversal for attacker

EASIEST
0.1 (shared creds)

HARDEST
1.0 (L3 routing)

Attack Graph Builder: 7-Step Pipeline

Building a weighted directed graph from scan data

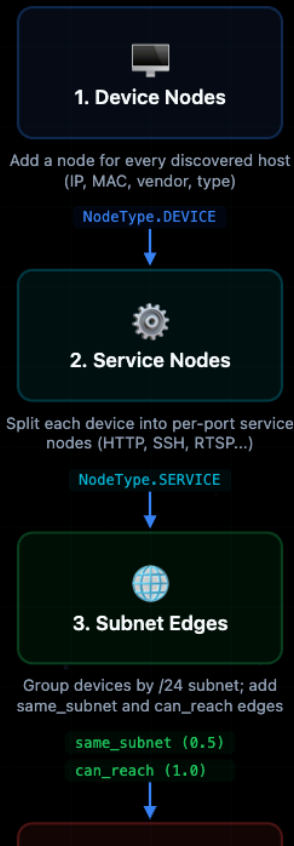
CHAPTER TAKEAWAY

Lower weight equals easier for the attacker.

ENRICHMENT VALUE

These are calibrated against pen test timelines.

`GRAPH_BUILDER.BUILD(SCAN_RESULTS)`



Step 2: Service Nodes

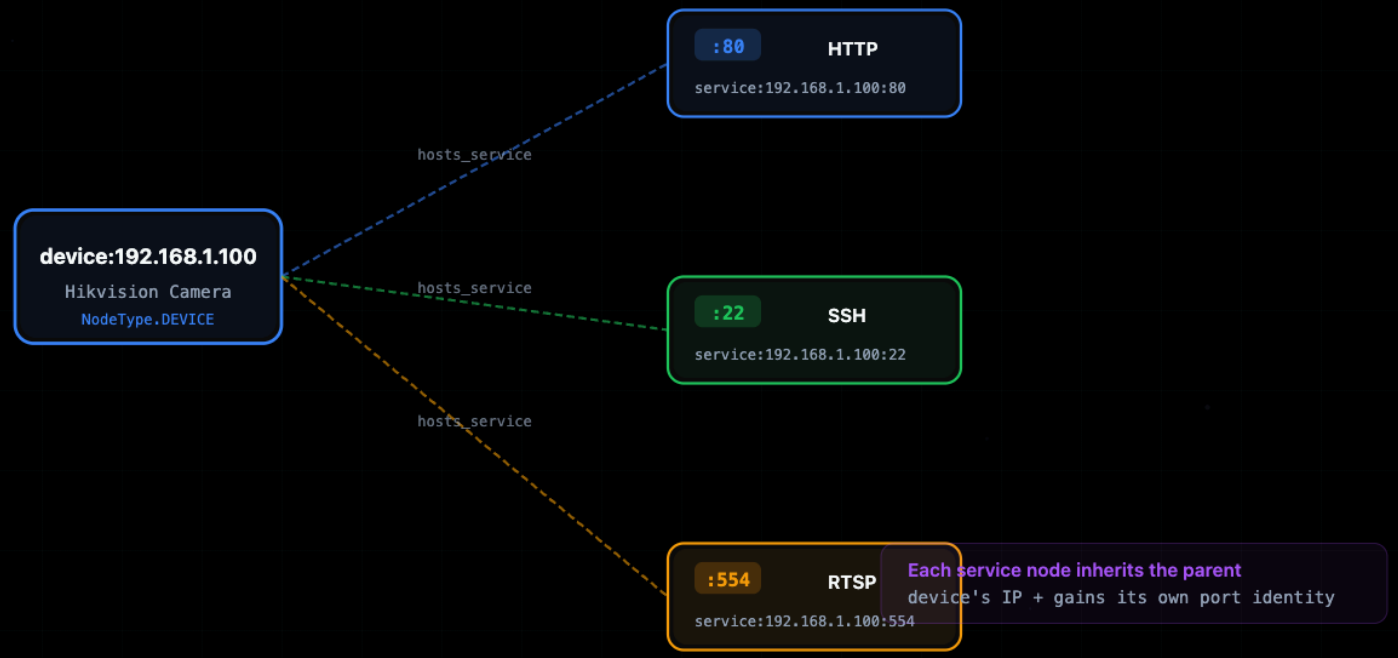
Each open port becomes its own node, linked to the parent device

CHAPTER TAKEAWAY

Step 1: Add device nodes. One host, one node.

ENRICHMENT VALUE

Step 8: Mark entry points. Internet-facing devices.



Step 4: Credential Edges

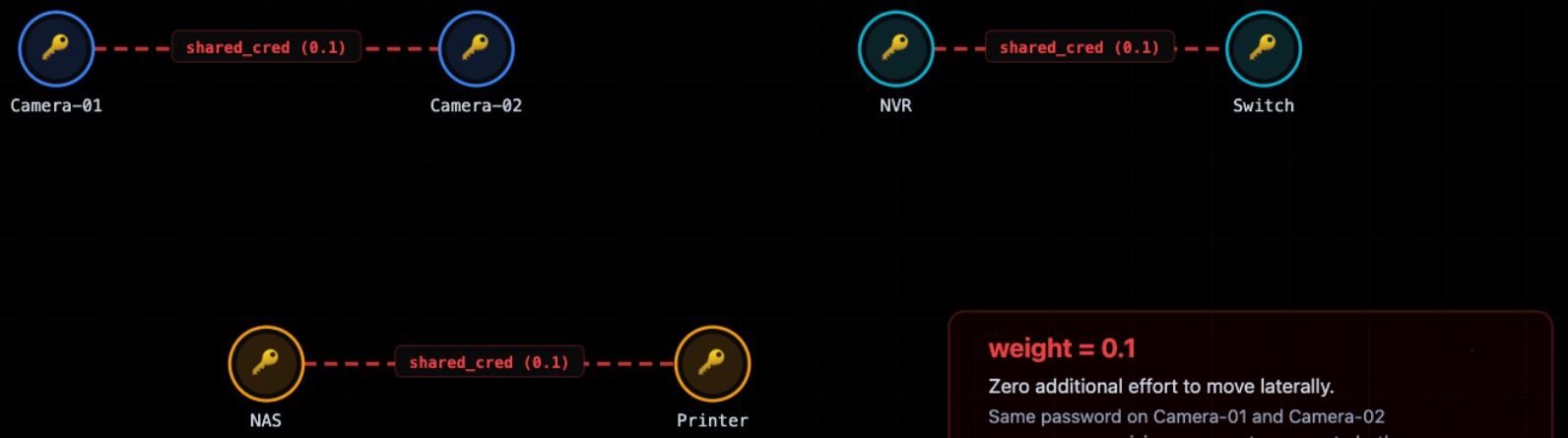
Shared credentials create zero-effort lateral movement paths

CHAPTER TAKEAWAY

214 devices. 47ms to build.

ENRICHMENT VALUE

The graph construction is trivially fast compared to the scan itself.



weight = 0.1

Zero additional effort to move laterally.

Same password on Camera-01 and Camera-02 means compromising one grants access to both.

Graph Statistics

Summary metrics from the constructed attack graph

CHAPTER TAKEAWAY

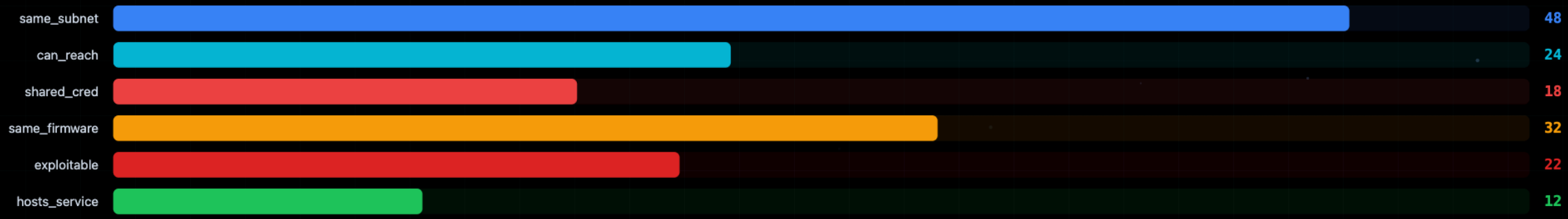
We have a graph. We have paths.

ENRICHMENT VALUE

That number is the Breakwater Risk Score.

Nodes	Edges	Entry Points	Avg Degree
42	156	3	7

EDGE TYPE DISTRIBUTION



SECTION 03

Breakwater Risk Score

Composite Risk Quantification

Why CVSS Alone Is Insufficient

From single-CVE scoring to composite risk assessment

CHAPTER TAKEAWAY

Vulnerability: 0.20.

ENRICHMENT VALUE

These are configurable via environment variable.

✘ CVSS Limitations

Doesn't account for volume
A device with 20 CVEs scores the same as one with 1 CVE of equal severity

No exploitability context
CVSS base score ignores whether exploit code exists or creds are default

No network reachability
An isolated VLAN device scores the same as one directly on the internet

No physical consequences
A compromised PLC controlling a valve scores the same as a printer

VS

✔ BRS Addresses

Volume-weighted severity
V factor scales with CVE count: `min(1.0 + len(cves) * 0.05, 2.0)`

KEV catalog + default creds
E factor: default creds = 10, KEV = 9, critical CVE = 8

Graph-based shortest path
R factor: `10.0 - min_hops * 0.8`, penalizes segmentation

Device-type consequence mapping
P factor: PLC = 10, camera = 6, IoT sensor = 3

BRS Formula

Six weighted factors composing the Breakwater Risk Score

CHAPTER TAKEAWAY

Starts with maximum CVSS score across all CVEs.

ENRICHMENT VALUE

Ten critical CVEs: V is 10.0.

Breakwater Risk Score Derivation

Complete formula

1

```
BRS = w1*V + w2*E + w3*R + w4*P + w5*S - w6*C
```

CVE count x max CVSS

2

```
V = Vulnerability Surface
```

Default creds (10), KEV (9), Critical CVE (8)

3

```
E = Exploitability
```

Shortest weighted path from entry points

4

```
R = Reachability
```

Device-type impact (PLC=10, Camera=6)

5

```
P = Physical Consequence
```

SBOM vulnerable components

6

```
S = Supply Chain
```

V — Vulnerability Surface

Volume-weighted severity factor

CHAPTER TAKEAWAY

Default credentials: E equals 10.0. Full stop.

ENRICHMENT VALUE

Default credentials dominate. They are the cheapest attack.

```
V = max_cvss * min(1.0 + len(cves) * 0.05, 2.0)
```

Volume multiplier caps at 2.0x

Base Score
max_cvss = highest CVSS score among all CVEs on the device (0.0-10.0)

Volume Multiplier
Each additional CVE adds 0.05 to the multiplier, capped at 2.0x (20 CVEs)

V FACTOR BY CVE COUNT (ASSUMING MAX CVSS = 9.8)

E — Exploitability

Priority cascade: highest matching condition wins

CHAPTER TAKEAWAY

Computed from the attack graph.

ENRICHMENT VALUE

Without a graph, falls back to open port count.

1	⚠ Default credentials found	10.0
2	☢ In CISA KEV catalog	9.0
3	☠ Critical CVE (CVSS >= 9.0)	8.0
4	⚡ High CVE (CVSS >= 7.0)	6.0
5	● Medium CVE (CVSS >= 4.0)	4.0
6	✔ No known exploits	0.0

R -- Reachability

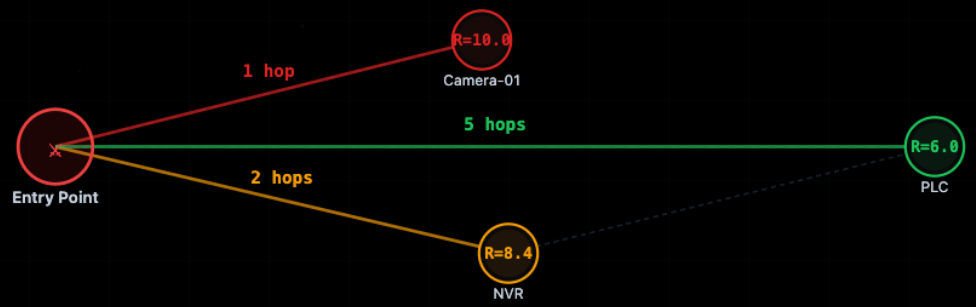
Graph-based distance from entry points

CHAPTER TAKEAWAY

Lookup by device type.

ENRICHMENT VALUE

IT server, printer: P equals 0.0.



$$R = \max(2.0,\ 10.0 - \text{min\_hops} * 0.8)$$

1 hop	2 hops	5 hops	10 hops
10.0	8.4	6.0	2.0

P -- Physical Consequence

Device-type impact severity mapping

CHAPTER TAKEAWAY

Segmentation: plus 3.0.

ENRICHMENT VALUE

Most devices score between 1.0 and 4.0 on compensating controls.

The P factor maps each device type to its potential physical-world impact. A compromised PLC controlling a water pump scores 10.0, while a temperature sensor scores 3.0. This factor is determined during identification (Phase 3) and stored as a device attribute.

CONSEQUENCE SCORE BY DEVICE TYPE

C -- Compensating Controls

Defense factors that reduce the BRS score

CHAPTER TAKEAWAY

9.0 to 10.0: Critical. Immediate action.

ENRICHMENT VALUE

0.0 to 1.9: Info. Acceptable risk.

C is subtracted from the BRS -- good defenses reduce risk

`BRS = ... - w6 * C`

Network Segmentation — -3.0

Device is on a dedicated VLAN with firewall rules restricting lateral movement

No Default Credentials — -1.0

Factory credentials have been changed to strong, unique passwords

Current Firmware — -2.0

Device is running the latest available firmware with known patches applied

Minimal Attack Surface — -1.0

Unnecessary services and debug interfaces are disabled

Maximum C value (all controls active) **-7.0**

Default Weights

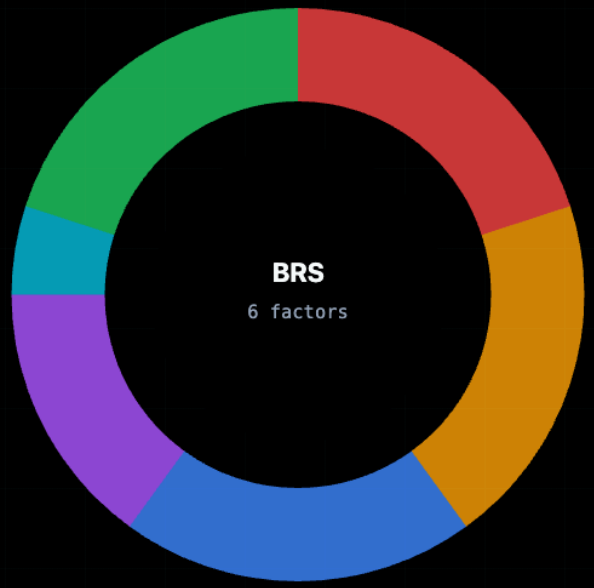
Factor weight distribution in the BRS formula

CHAPTER TAKEAWAY

PLC at 10.0.2.20.

ENRICHMENT VALUE

C equals 1.0. No segmentation, two open ports.



V (Vulnerability)	0.20
E (Exploitability)	0.20
R (Reachability)	0.20
P (Physical)	0.15
S (Supply Chain)	0.05
C (Controls) (subtracted)	0.20

Theoretical Max
8.0

Theoretical Min
0.0

BRS Worked Example

Hikvision Camera -- IP 192.168.86.42

CHAPTER TAKEAWAY

0.20 times 8.5 equals 1.70.

ENRICHMENT VALUE

BRS equals 6.52. Rating: medium.

Camera BRS Calculation

Vulnerability Surface: 20 CVEs, max CVSS 9.8

1. $V = 9.8 * \min(1.0 + 20*0.05, 2.0) = 9.8 * 2.0 = 10.0$

where $V = 10.0$

Exploitability: highest priority -- default creds

2. $E = 10.0$ (default credentials: admin:12345)

where $E = 10.0$

Reachability: 0.5 weighted hops from WiFi AP

3. $R = \max(2.0, 10.0 - 0.5*0.8) = \max(2.0, 9.6) = 9.6$

where $R = 9.6$

Physical Consequence: camera / NVR tier

4. $P = 6.0$ (device type: camera)

where $P = 6.0$

Supply Chain: firmware not analyzed

5. $S = 0.0$ (no SBOM data available)

Scan-Wide BRS Results

Aggregate risk scoring across all discovered devices

CHAPTER TAKEAWAY

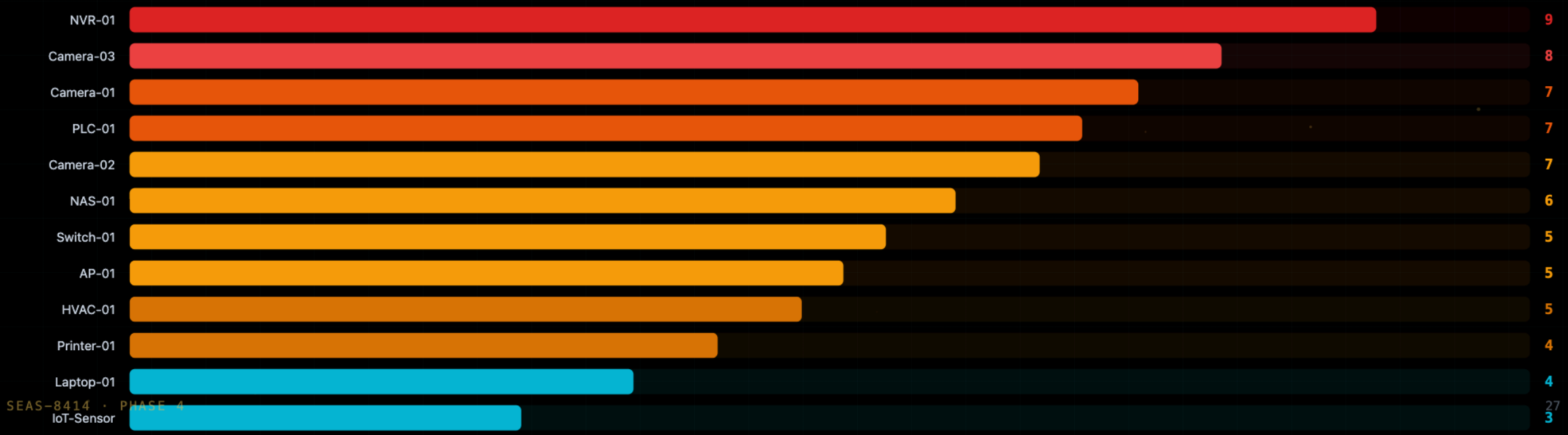
Rotate Modbus credentials.

ENRICHMENT VALUE

No other single action comes close.

6	9	2	5	3	5
Avg BRS	Max BRS	Critical	High	Medium	Low

BRS DISTRIBUTION BY DEVICE



Path Computation

Iterating Yen's algorithm across all entry-target pairs

CHAPTER TAKEAWAY

The path engine starts by finding entry points.

ENRICHMENT VALUE

In the port scenario: firewall, wireless AP, VPN concentrator.

apps/api/app/scanning/attack_graph/path_engine.py — PYTHON

```python
def find_attack_paths(self, graph, k=5):
    paths = []
    for entry in self.entry_points:
        for target in self.high_value_targets:
            for path in nx.shortest_simple_paths(  # ← Yen's algorithm via networkx
                graph, entry, target, weight="weight"  # ← weight-aware shortest paths
            ):
                paths.append(self._score_path(path))  # ← score each path for probability + time
                if len(paths) >= k:  # ← limit to top-K per pair
                    break
    return sorted(paths, key=lambda p: p.cost)
```

Path Cost Calculation

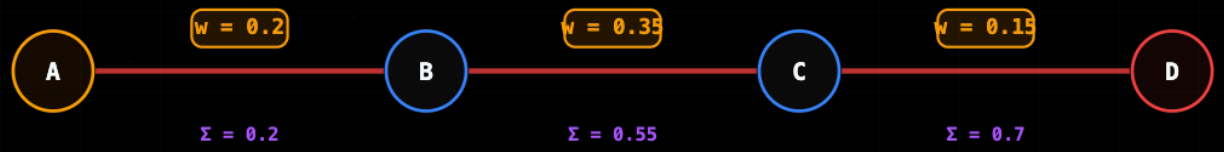
Summing edge weights along the traversal path

CHAPTER TAKEAWAY

The path engine then finds high-value targets.

ENRICHMENT VALUE

Hub devices are high-value because compromising them provides access to many others.



1. Path cost is the sum of all edge weights

$$\text{cost}(path) = \Sigma\, w(e) \text{ for each edge } e \text{ in path}$$

2. Expand for our 3-edge path

$$\text{cost}(A \to D) = w(A \to B) + w(B \to C) + w(C \to D)$$

3. Substitute actual weights

$$\text{cost}(A \to D) = 0.2 + 0.35 + 0.15 = 0.70$$

where lower cost = easier attack path

Time Estimation

Mapping edge weights to real-world compromise timelines

CHAPTER TAKEAWAY

NetworkX provides shortest_simple_paths.

ENRICHMENT VALUE

Maximum hop count: 10.

WEIGHT RANGE	EST. TIME
0.1 – 0.3	Minutes
0.3 – 0.5	Hours
0.5 – 0.8	Days
0.8 – 1.0	Weeks

ESTIMATED COMPROMISE TIME

Hospital Network: Worked Example

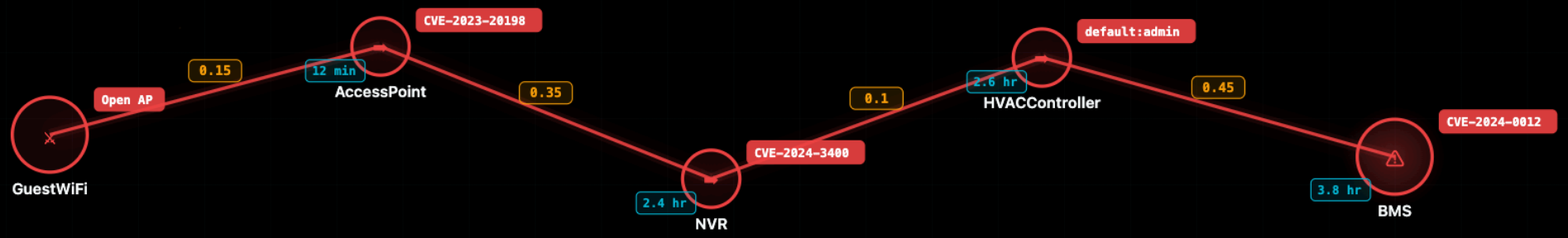
5-hop path from guest WiFi to building management system

CHAPTER TAKEAWAY

Each path is a sequence of attack steps.

ENRICHMENT VALUE

Estimated time in hours from the weight-to-time lookup table.



Total Cost	Est. Time	Hops
1.05	3.8 hr	4

Segmentation Score

Quantifying network isolation effectiveness

CHAPTER TAKEAWAY

Weight 0: probability 1.0. Theoretical maximum.

ENRICHMENT VALUE

Not a true probability. A ranking heuristic.

1. Segmentation score definition

```
seg_score = 1 - (reachable / total)
```

2. Before VLAN segmentation

```
seg_score = 1 - (42 / 50) = 0.16
```

where reachable = 42 of 50 hosts

3. After VLAN segmentation

```
seg_score = 1 - (8 / 50) = 0.84
```

where reachable = 8 of 50 hosts

Before VLANs → After VLANs

Reachable (8) Isolated (42)

Isolation improved by +425%

SECTION 05

MITRE ATT&CK ICS Mapping

Translating Findings to Industry-Standard Tactics

Three Mapping Strategies

How findings translate to MITRE ATT&CK ICS techniques

CHAPTER TAKEAWAY

Measures how well the network is segmented.

ENRICHMENT VALUE

The port scores 6.0. Two subnets but a shared switch.

1

Finding-Type Mapping

`default_credentials`

↓

T0812
Default Credentials

Direct mapping from finding type to ICS technique

2

CWE-Based Mapping

`CWE-798`

↓

T0812
Default Credentials

CVE weakness enum maps to technique via CWE lookup table

3

Device-Type Mapping

`PLC (device_type)`

↓

T0831
Manipulation of Control

Device classification implies likely adversary techniques

MITRE ATT&CK ICS Heatmap

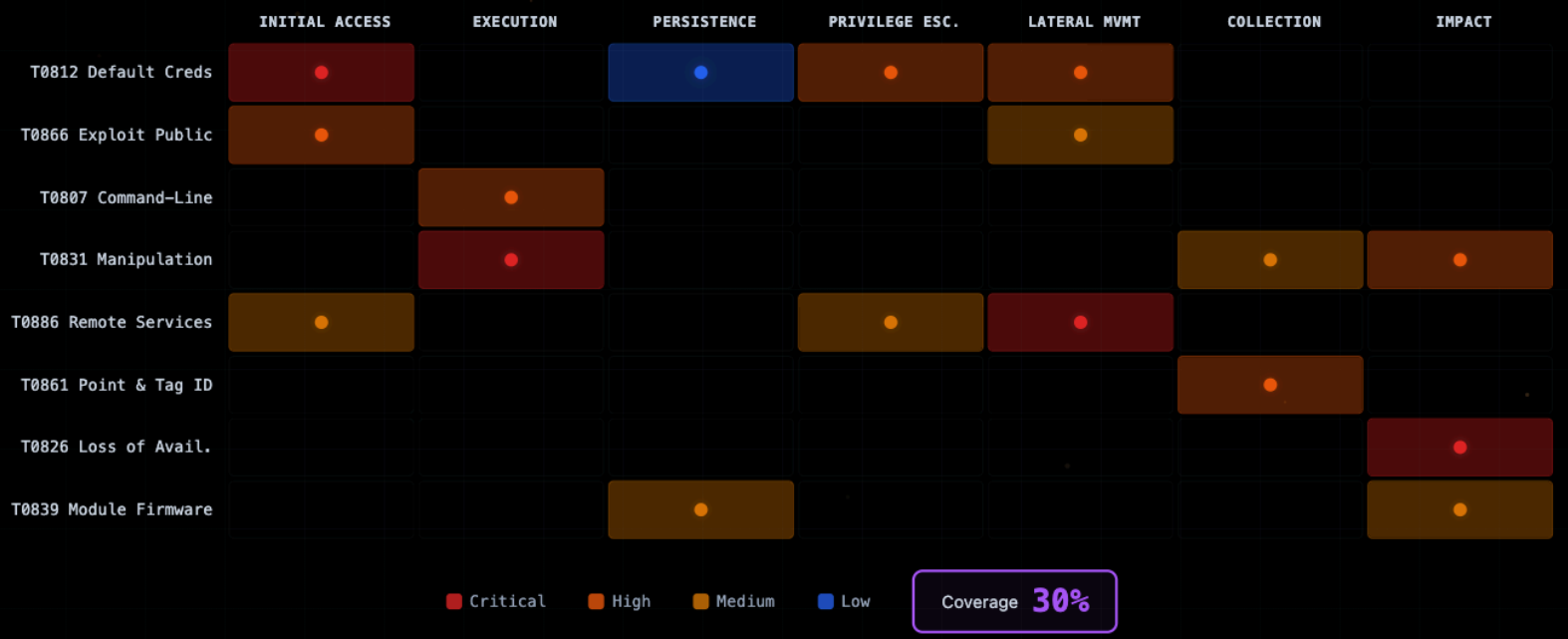
Mapping scan findings to tactic-technique coverage

CHAPTER TAKEAWAY

Three strategies. Device type. CWE. Finding type.

ENRICHMENT VALUE

Deduplicated by technique. Highest confidence wins.



STIX Bundle Structure

Domain objects and relationship objects in the export graph

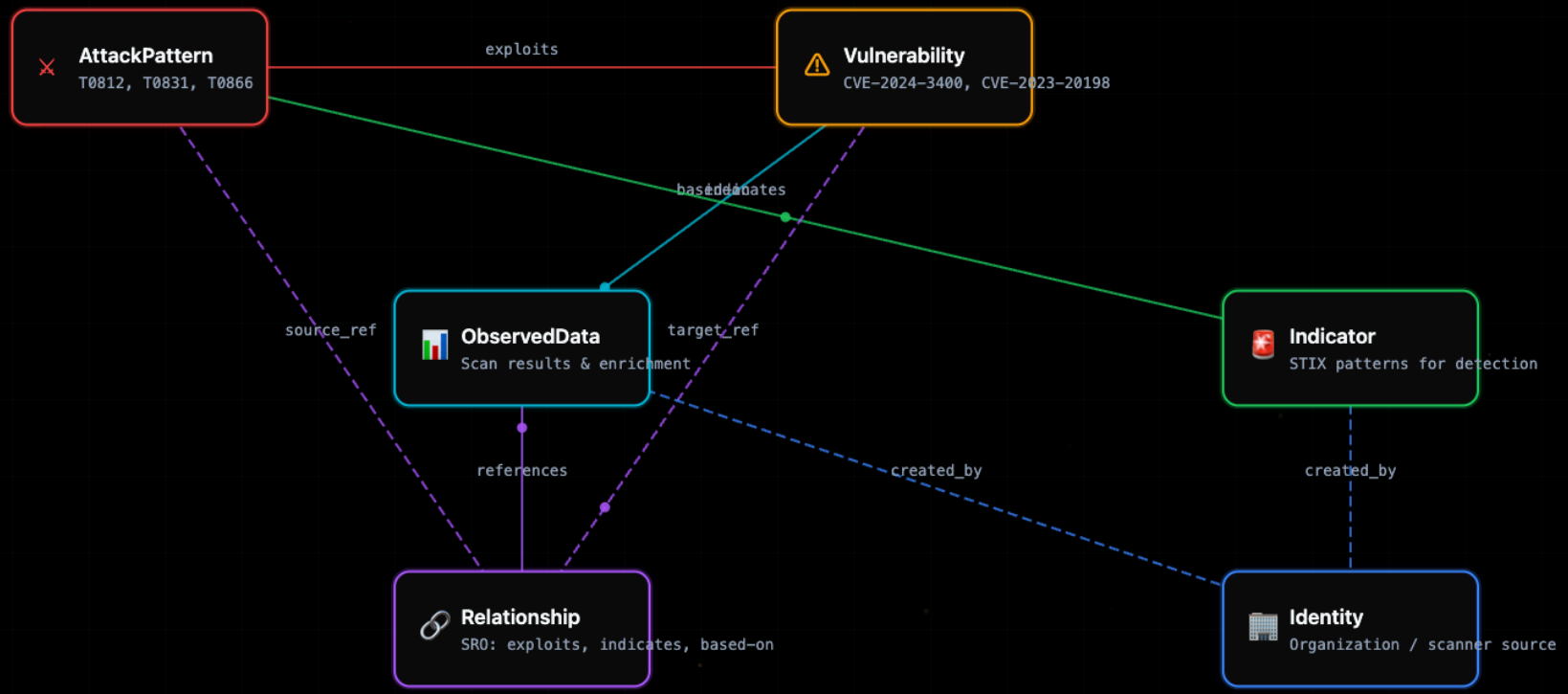
CHAPTER TAKEAWAY

Seven object types in the bundle.

ENRICHMENT VALUE

Available at GET /v1/analytics/mitre/scan_id/stix.

STIX 2.1 OBJECT GRAPH



SECTION 06

What-If Engine

Simulate Before You Remediate

Five Remediation Action Types

Each action modifies the attack graph in a specific, predictable way

CHAPTER TAKEAWAY

Patch: remove all CVEs and exploit edges.

ENRICHMENT VALUE

Firewall rule: remove all inbound edges except subnet.

⚙ Patch

Apply vendor security update to eliminate known vulnerability

`GRAPH: Removes exploit edges`

COST	DOWNTIME
3.0	Service restart

⛓ Segment

Isolate device into a dedicated VLAN or firewall zone

`GRAPH: Removes connectivity edges`

COST	DOWNTIME
5.0	Network reconfiguration

⚿ Rotate Creds

Replace default or shared credentials with unique, strong ones

`GRAPH: Removes sharing edges`

COST	DOWNTIME
1.0	Credential update

⛔ Disable Service

Turn off unnecessary services (telnet, FTP, debug ports)

`GRAPH: Removes service nodes`

COST	DOWNTIME
1.0	Feature loss

⛨ Firewall Rule

Block specific inbound traffic at network or host firewall

`GRAPH: Removes inbound edges`

COST	DOWNTIME
2.0	None

Simulation Pipeline

How the What-If Engine evaluates a candidate remediation plan

CHAPTER TAKEAWAY

Original average BRS vs simulated.

ENRICHMENT VALUE

All in one response object.

Worked Example: Hospital Network

Three targeted actions reduce risk by 45% and eliminate 75% of attack paths

CHAPTER TAKEAWAY

Two actions. Segment the PLC. Rotate all credentials.

ENRICHMENT VALUE

That is a board-ready answer.

⚠	✔	↓	⛓
6	3	−3	9/12
Original Avg BRS	Simulated Avg BRS	Delta ↓	Paths Eliminated

ACTIONS APPLIED

⚿ Rotate NVR credentials	COST	1.0
⚙ Patch IP cameras (CVE-2024-XXXX)	COST	9.0
⛓ Segment PLC into isolated VLAN	COST	5.0
Total Remediation Cost		**15.0**

Greedy Optimization

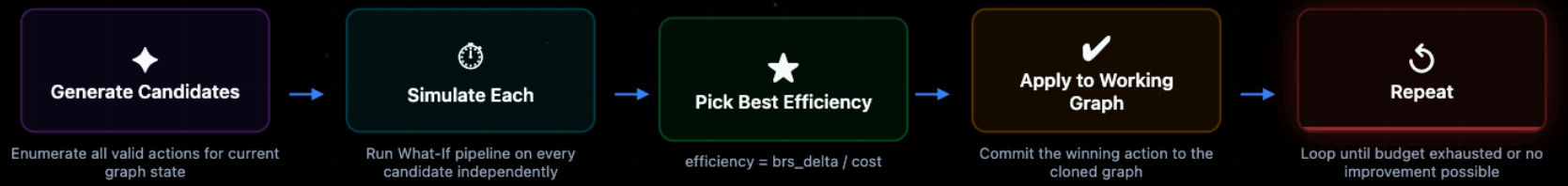
Iteratively select the most efficient action until budget is exhausted

CHAPTER TAKEAWAY

generate_optimal_plan selects actions by efficiency.

ENRICHMENT VALUE

Diminishing returns curve shows the first 3 actions deliver 70% of benefit.



SELECTION CRITERION

efficiency = |avg_brs_delta| / action_cost

At each iteration, the action with the highest efficiency ratio is selected. This greedy approach produces near-optimal plans in O(n * k) simulations.

Remediation Plan Output

Structured JSON output from the What-If Engine optimization

CHAPTER TAKEAWAY

Action 1: rotate credentials. BRS drops 1.2 points.

ENRICHMENT VALUE

Budget accordingly.

whatif_engine.py – RemediationPlan

```
>>> engine.optimize(budget=20.0, max_actions=5)
{
  "actions": [
    {"type": "rotate_creds",   "target": "nvr-01",     "cost": 1.0, "priority": 1},
    {"type": "disable_service","target": "cam-03:23",   "cost": 1.0, "priority": 2},
    {"type": "firewall_rule",  "target": "plc-07:502",  "cost": 2.0, "priority": 3},
    {"type": "patch",          "target": "cam-01..06",  "cost": 9.0, "priority": 4},
    {"type": "segment",        "target": "plc-subnet",  "cost": 5.0, "priority": 5}
  ],
  "total_cost": 18.0,
  "estimated_downtime_minutes": 45,
  "original_avg_brs": 5.8,
  "projected_avg_brs": 2.4,
  "brs_reduction_pct": 58.6,
  "paths_eliminated": 10,
  "paths_remaining": 2,
  "efficiency": 0.189
}
```

Credential Change: The Mirai Signal

Default credentials appearing on a previously secured device is a critical indicator

CHAPTER TAKEAWAY

Rule-based BRS is interpretable and fast.

ENRICHMENT VALUE

Blended: 0.7 GNN plus 0.3 rule-based.

MIRAI BOTNET LIFECYCLE

Detection Signal

When the behavioral baseline detects "credential_change" with default credentials *newly appearing* on a device that previously had strong credentials, this is treated as a **critical anomaly** -- indicating either a factory reset (Step 3) or active compromise. The anomaly type is the only Phase 4 signal rated "critical" severity.

DeviceBaseline Model

8 attributes tracked per device across scan intervals

CHAPTER TAKEAWAY

Two-layer heterogeneous graph attention network.

ENRICHMENT VALUE

Readout MLP: 64 to 32 to 1 times sigmoid times 10.

`class DeviceBaseline(BaseModel)`

open_ports

`Set[int]`

`{22, 80, 443, 8080}`

services

`Dict[int, str]`

`{22: "ssh", 80: "nginx"}`

protocols

`Set[str]`

`{"tcp", "udp", "mdns"}`

credentials

`CredentialState`

`strong | default | none`

firmware_version

`Optional[str]`

`"v3.2.1-build.447"`

response_times

`Dict[str, float]`

`{icmp: 1.2, http: 45.8}`

certificate_info

`CertSnapshot`

`CN=cam.local exp=2026-09`

network_behavior

`BehaviorProfile`

`outbound_hosts, dns_queries`

Worked Example: Baseline Drift Detection

Camera at 10.0.1.44 monitored over 5 scan intervals

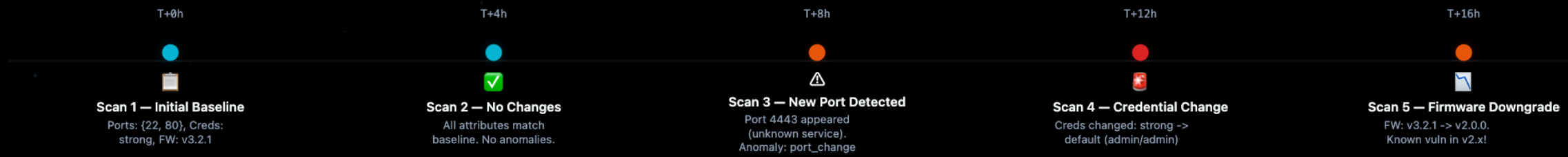
CHAPTER TAKEAWAY

The HYDRA engine extends the six-factor Chapter 4 BRS to 8 components.

ENRICHMENT VALUE

Both have weight 0.05. Small but they accumulate across large fleets.

SCAN INTERVAL TIMELINE



port_change
HIGH

credential_change
CRITICAL

firmware_downgrade
HIGH

Total anomalies: 3 Max severity: CRITICAL Scan intervals: 5

SECTION 07

Threat Intelligence Integration

CISA KEV, External Feeds, and Reputation Scoring

CISA KEV Integration

Matching the Known Exploited Vulnerabilities catalog against scan results

CHAPTER TAKEAWAY

The vulnerability report listed 47 CVEs.

ENRICHMENT VALUE

The eleven-day delay was preventable.

known_exploited_vulnerabilities.json — JSON

```
{
  "title": "CISA Known Exploited Vulnerabilities",
  "catalogVersion": "2026.03.04",
  "count": 1187,
  "vulnerabilities": [
    {
      "cveID": "CVE-2024-21762",    ← Match key
      "vendorProject": "Fortinet",
      "product": "FortiOS",
      "vulnerabilityName": "FortiOS Out-of-Bound Write",
      "dateAdded": "2024-02-09",
      "dueDate": "2024-03-01",
      "knownRansomwareCampaignUse": "Known"
    }
  ]
}
```

threat_intel.py — PYTHON

```python
async def _match_kev_to_scan(self, kev_entries, scan_cves):
    matched = []
    for entry in kev_entries:
        kev_id = entry["cveID"]    ← CVE cross-reference
        if kev_id in scan_cves:                # <-- match!
            matched.append(ThreatMatch(
                cve_id=kev_id,
                source="cisa_kev",
                severity="critical",           # Always critical
                ransomware=entry.get(
                    "knownRansomwareCampaignUse"
                ) == "Known",
            ))
    return matched
```

🎯 KEV Matches Found
out of 249 scan CVEs

7

Reputation Scoring

Cumulative deductions from a perfect 10.0 base score

CHAPTER TAKEAWAY

Attack graphs model how vulnerabilities compose.

ENRICHMENT VALUE

Weights are organizational policy, not technical constants.

```
score = base - sum(deductions)
```

Start with perfect reputation

1. `base_score = 10.0`

Host IP found in threat intelligence feed

2. `base_score - 5.0 (IP match)`

`where ip_deduction = -5.0 per matched IP`

KEV CVE found on this host

3. `5.0 - 3.0 (CVE match)`

`where cve_deduction = -3.0 per KEV CVE`

Suspicious port matches known C2 profile

4. `2.0 - 1.0 (Port match)`

`where port_deduction = -1.0 per flagged port`

Vendor appears in advisory watchlist

5. `1.0 - 0.5 (Vendor match)`

`where vendor_deduction = -0.5 per vendor match`

1

Host Reputation

IP match	-5.0
CVE match	-3.0
Port match	-1.0
Vendor match	-0.5

Worked Example: Threat Intel API

Live query against scan results with matched CVEs and reputation scores

CHAPTER TAKEAWAY

The Phase 4 lab has 10 exercises.

ENRICHMENT VALUE

All against the simulation lab. 22 devices. Real API calls.

breakwater-api -- threat intel query

```
# Query threat intel for scan results
$ curl -s http://localhost:8000/v1/attack-graph/threat-intel/scan_abc123 | jq .
{
  "scan_id": "scan_abc123",
  "kev_matches": [
    {
      "cve_id": "CVE-2024-21762",
      "host": "10.0.1.12",
      "source": "cisa_kev",
      "ransomware": true,
      "severity": "critical"
    }
  ],
  "reputation_scores": {
    "10.0.1.12": 2.0,
    "10.0.1.44": 4.5,
    "10.0.1.3": 8.5
  },
  "total_matches": 7,
  "hosts_flagged": 3
```

SECTION 08

Consequence Prediction

From Cyber Risk to Physical Impact