



## REGULATOR VIEW

# Cyber Response - AI-Driven Containment

**World:** Cyber Response Governance **Outcome:** ALLOW WITH BOUNDS **Evidence Pack ID:** JSP-PUBLIC-20260302-CYBER\_RESPON-0001

## 1. What action happened

SOAR playbook proposes isolating 400 endpoints.

## 2. Who held authority

|                            |   |
|----------------------------|---|
| <b>Primary authority</b>   | Incident Commander  |
| <b>Secondary authority</b> | SOC Lead  |
| <b>Escalation roles</b>    | Incident Commander, SOC Lead                                    |
| <b>Authority principle</b> | No consequential action proceeds without named human authority. |

## 3. What was enforced at runtime

|                        |   |
|------------------------|---|
| <b>Outcome</b>         | ALLOW WITH BOUNDS   |
| <b>Bounds enforced</b> | <ul style="list-style-type: none"><li>- isolate 1 host</li><li>- shorten auth tokens</li><li>- block org-wide isolation</li><li>- increase monitoring</li><li>- time-box escalation</li></ul> |
| <b>Escalation path</b> | If corroboration crosses threshold or IC approves, widen scope in staged steps (10, 50, 200, 400).  |

## 4. What would happen if confidence dropped

**Without governance:** Mass isolation could DDoS internal services and halt operations on a false positive.

**Prevented failure mode:** self-inflicted outage by automated containment

**Illustrative exposure:** hours of downtime + incident costs

Note: exposure figures are illustrative; the structural claim is that the system intercepts and governs at the moment before consequence.

# Evidence & Integrity

## 5. Evidence minimums and signals

- edr\_detection\_ids
- corroboration\_score
- blast\_radius\_estimate
- critical\_service\_dependencies
- incident\_commander
- containment\_playbook\_id

## 6. Portable artefacts included in this pack

|                                  |  |
|----------------------------------|--|
| <b>evidence_pack.json</b>        | Top-level decision record and links.                 |
| <b>authority_contract.json</b>   | Who can authorise what, with what bounds.            |
| <b>policy_trace.ndjson</b>       | Versioned doctrine checks evaluated.                 |
| <b>audit_export.ndjson</b>       | Runtime event log for audit ingestion.               |
| <b>provenance_chain.json</b>     | Illustrative model/prompt/tool-call chain integrity. |
| <b>manifest.json + signature</b> | Tamper-evident integrity for this entire bundle.     |

## 7. Integrity verification

Run **verify\_manifest.py** in the root folder to recompute SHA-256 hashes and verify the signature (when cryptography or OpenSSL is available).

Expected output: **PASS** (no files changed).

This proof object is designed to be inspected without access to the originating systems.

Contact: [founder@judgementspine.com](mailto:founder@judgementspine.com) | <https://judgementspine.com/>