Chapter 10: Active Deception & Threat Hunting

Learning Objectives

By the end of this chapter, students will be able to:

1. Explain the theoretical foundations of cyber deception and classify honeypot interaction levels.
2. Design a Q-learning reinforcement learning agent that adapts honeypot personalities to maximize intelligence capture.
3. Implement a SHA-256 hash chain for tamper-evident session recording and verify chain integrity.
4. Map attacker commands to MITRE ATT&CK techniques using regex pattern matching and classify attacker skill levels.
5. Apply Bayesian inference (game-theoretic attacker modeling) to predict attacker behaviour and select optimal honeypot responses.
6. Formulate honeypot placement as a weighted set-cover optimization problem informed by attack graph topology.
7. Construct Merkle tree evidence chains with inclusion proofs and anchor forensic evidence to a tamper-evident ledger.

Agentic Lens

In this chapter, the agent moves from simply observing to actively shaping the environment that attackers experience. Instead of just ranking findings, the agent decides which attacker behaviors need more attention and which can be set aside.

- **The agent man**ages deception and threat-hunting tasks, gathering useful intelligence and making sure production systems stay protected.
- **The agent learns from many** sources, such as live telemetry, key points in attack graphs, decoy interactions, command logs, and past attacker activity.
- **The toolkit offers flexible** honeypot personalities, careful placement, MITRE ATT&CK labeling, and Bayesian profiling. The system keeps track of active decoys, current sessions, and changing views of attacker skill. To verify results, it checks data from different sources, confirms session integrity, and asks analysts to review any major attribution decisions. Decoys must not disrupt real services. Collecting intelligence matters, but not at the cost of system safety. Focusing too much o**n** engagement instead of real threats can cause real intrusions to be missed. Here, deception means careful observation, not random improvisation.
- **The threat model** covers attackers from simple automated scanners to skilled human operators who can probe, log in, move through the network, and adapt to **changes.** Defenders need to balance the risk of missing real intrusions with the risk of misreading decoy activity or making unreliable attributions.
- **D**ecoys are kept separate from production systems and set up to tell apart defender-created artifacts from real **activity.** Session recording, MITRE ATT&CK labeling, and Bayesian profiling help analysts make decisions, but none of these methods alone can fully support attribution claims for legal or organizational purposes.
- **Assumption:** Safety always comes first. High-engagement decoys must not change real syst**ems. T**his chapter makes it clear that deception telemetry is just one part of the process and should be checked against independent network and host evidence. Earlier, the approach was reactive, treating the network as a fixed object. Now, Breakwater takes a more active role by using honeypots that imitate vulnerable devices, carefully recording attacker sessions, and adjusting honeypot behavior. The roots of cyber deception go back to Clifford Stoll's 1986 "Cuckoo's Egg" case, where a small accounting error revealed a KGB-linked hacker. Stoll guided the intruder into controlled spaces to watch their actions. Today's honeypots use this method, offering safe environments to observe and record attackers. Deception works especially well in IoT and OT networks, where devices often can't be

patched or monitored in standard ways. For instance, a honeypot that acts like a Siemens S7-1200 PLC on port 102 can spot attackers moving into OT areas that passive monitoring might miss.

Chapter 10 covers how to deploy honeypots in practice. This includes spreading them across the network, choosing personalities with reinforcement learning, recording sessions with cryptographic hash chains, labeling attacker behavior with MITRE ATT&CK techniques, and anchoring forensic evidence to tamper-proof ledgers. The analyst readout here is meant as a teaching example, not a production standard. Real deployments may look quite different from the examples in this chapter.

10.2.1 Interaction Levels

The depth of interaction classifies the honeypots they provide:

Level	Description	Fidelity	Risk	Example
Low-interaction	Emulates service banners, accepts connections, and records initial probes	Minimal	Very low	Honeyd, Breakwater emulators
Medium-interaction	Emulates command execution, file system navigation, and credential acceptance	Moderate	Low	Cowrie, Kippo
High-interaction	Full operating system with real services, monitored at the hypervisor level	Full	Moderate	Honeynet, T-Pot

Breakwater implements low-interaction honeypots with *personality-driven emulation*. Each honeypot is configured with a personality profile – a dictionary of device characteristics (SSH banner, HTTP server header, open ports, known CVEs, trap credentials) that control how the emulator responds to attacker probes.

```
# From apps/api/app/scanning/deception/honeypot_engine.py
_DEFAULT_PERSONALITIES: dict[str, dict[str, Any]] = {
    "ssh": {"banner": "SSH-2.0-OpenSSH_7.4"},
    "telnet": {"banner": "BusyBox v1.31.1 (2021-04-07 10:15:02 UTC) login: "},
    "http": {
        "server": "lighttpd/1.4.53",
        "title": "Device Management",
        "status": 200,
    },
    "rtsp": {"server": "LIVE555/0.98"},
    "mqtt": {"broker_name": "mosquitto/2.0.15"},
}
```

10.2.2 The HoneypotEngine Lifecycle

The HoneypotEngine class manages the full lifecycle of honeypot deployments: planning, deployment, handling connections, recording sessions, and teardown.

```python
# From apps/api/app/scanning/deception/honeypot_engine.py
class HoneypotEngine:
    def __init__(self, max_honeypots: int = 10, session_timeout: int = 300):
        self.max_honeypots = max_honeypots
        self._deployments: dict[str, DeceptionDeployment] = {}
        self._honeypots: dict[str, HoneypotInstance] = {}
        self._servers: dict[str, asyncio.AbstractServer] = {}
        self._sessions: list[SessionRecord] = []
        self._lock = asyncio.Lock()
```

Each honeypot binds to an asyncio server on the specified IP and port. When an attacker connects, the engine routes the connection through a protocol-specific emulator. The emulator interacts with the attacker, while a SessionRecorder captures every byte with hash-chain integrity.

10.2.3 Deployment Planning

The generate_deployment_plan() method analyses scan results to select unoccupied IPs and observed protocols:

1. **Occupied IP detection:** scan results provide the set of live host IPs.
2. **Unoccupied IP selection** – iterate the subnet, skipping occupied addresses. **Protocol selection:** identify which protocols are observed on the network (SSH, HTTP, MQTT, etc.) and emulate those. those.
3. **Round-robin distribution:** spread honeypots across protocols for coverage diversity.

10.3 Architecture: Deception Deployment Pipeline

```
graph TD
  A[Phase 1-6 Scan Results] --> B[Placement Optimizer]
  B --> C[Honeypot Engine]
  C --> D[Protocol Emulators]
  D --> E[Session Recorder]
  E --> F[Evidence Chain]
  F --> G[Blockchain Anchor]
  D --> H[Chameleon RL Engine]
  H --> D
  E --> I[Attacker Profiler]
  I --> J[Cognitive Honeypot]
  J --> H
  E --> K[MITRE TTP Annotator]
  K --> L[HYDRA Stream D]

  style B fill:#e1f5fe
  style C fill:#e8f5e9
  style D fill:#f3e5f5
  style E fill:#fff3e0
  style F fill:#fce4ec
  style H fill:#e0f2f1
  style I fill:#fff9c4
  style J fill:#ede7f6
  style K fill:#ffebee
```

The pipeline comprises eight interacting engines:

1. **Placement Optimizer** – selects optimal IPs using attack graph topology and weighted set cover.
2. **Honeypot Engine** – manages listener lifecycle, connection routing, and session collection.
3. **Protocol Emulators** – per-protocol handlers (SSH, Telnet, HTTP, RTSP, MQTT) that produce realistic responses.
4. **Chameleon RL Engine** – Q-learning agent that selects which personality to present to each attacker.
5. **Session Recorder** – SHA-256 hash chain over every packet for tamper-evident forensic recording.
6. **Evidence Chain** – Merkle tree over all session evidence, enabling per-item inclusion proofs.
7. **Blockchain Anchor** – anchors Merkle roots to a tamper-evident ledger (local SQLite or Ethereum stub).

Implementation Note: The Ethereum anchor is a stub implementation. It provides the interface for submitting Merkle roots but does not connect to a live blockchain. For production use, replace the stub with a real Ethereum client or a permissioned ledger.

8. **Attacker Profiler** – maps sessions to MITRE ATT&CK techniques and classifies attacker skill level.

10.4 The Chameleon RL Engine

10.4.1 Problem Formulation

The central question for an adaptive honeypot is: *which device personality should we present to this attacker to maximize the intelligence gathered while minimizing the risk of detection?*

This is a sequential decision problem. The attacker's behavior evolves – they may start with reconnaissance, then attempt exploitation, and laterally pivot. The optimal personality to present at step 1 may differ from that at step 15. This naturally maps to a Markov Decision Process (MDP) that is solved via reinforcement learning.

10.4.2 State Space

The attacker session state is encoded as a 4-tuple:

State = (tool_signature, commands_observed_bucket, elapsed_time_bucket, prior_interactions)

Each dimension is discretized into buckets to keep the state space tractable for tabular Q-learning:

```
# From apps/api/app/scanning/deception/rl_chameleon.py
TOOL_SIGNATURES = {
    "masscan": 0,
    "nmap": 1,
    "metasploit": 2,
    "shodan": 3,
    "custom": 4,
    "mirai": 5,
    "unknown": 6,
}
COMMAND_BUCKETS = [0, 3, 10, 30, 9999]    # 5 buckets
TIME_BUCKETS = [10, 60, 300, 9999]         # 4 buckets
```

```
PRIOR_BUCKETS = [0, 1, 5, 9999]            # 4 buckets

N_STATES = 7 * 5 * 4 * 4  # = 560 states
```

The state is encoded as a single integer via mixed-radix encoding:

```
state = tool_idx * (N_CMD * N_TIME * N_PRIOR)
    + cmd_bucket * (N_TIME * N_PRIOR)
    + time_bucket * N_PRIOR
    + prior_bucket
```

10.4.3 Action Space

The action space consists of 8 personality profiles drawn from the PERSONALITY_CATALOG:

Personality	Device Type	Intelligence Value	Detection Risk
ssh-router	Cisco ISR 4321	0.85	0.20
http-camera	Hikvision DS-2CD2185	0.65	0.15
telnet-industrial	Siemens S7-1200 PLC	0.95	0.10
nas-synology	Synology DS920+	0.75	0.25
mqtt-broker	Eclipse Mosquitto	0.70	0.12
http-printer	HP LaserJet Pro	0.50	0.30
ssh-server	Ubuntu Server 22.04	0.80	0.35
voip-phone	Grandstream GXP2170	0.55	0.20

Evidence label: Illustrative. Method note: the “intelligence value” and “detection risk” columns are initial policy priors for the RL environment. Assumptions: the planner needs normalized starting values before any site-specific calibration exists. Boundary: re-estimate these priors from observed session yield, emulation fidelity tests, and decoy-detection rates before operational use.

The telnet-industrial personality starts with the highest intelligence value because ICS-targeting attackers often reveal more about their objectives and follow-on workflow. It starts with a low detection-risk prior because many commodity attackers have less prior exposure to ICS honeypots than to generic SSH or HTTP decoys. Those are plausible priors, not timeless constants.

10.4.4 Reward Function

The reward balances intelligence capture against detection risk:

```
reward = intel_score * intel_value * W_intel
     - detection_risk * W_detect
     + engagement_bonus
```

Where W_intel = 1.0, W_detect = 2.0 (detection penalty is double the intelligence reward), and engagement_bonus = 0.1 per step.

Evidence label: Illustrative. Method note: these reward weights are hand-tuned controller parameters for the chapter controller. Assumptions: the policy should penalize rapid detection more heavily than modest intelligence loss. Boundary: they are not globally optimal values and should be recalibrated for the target environment.

The IntelligenceScorer assigns scores to attacker commands based on pattern matching:

```
# From apps/api/app/scanning/deception/rl_chameleon.py
HIGH_VALUE_COMMANDS: dict[str, float] = {
  "sudo": 0.9, "/etc/shadow": 1.0, "crontab": 0.9,
  "authorized_keys": 0.95, "STOP CPU": 1.0, "START CPU": 1.0,
  "show run": 0.9, "ls": 0.15, "pwd": 0.10, "whoami": 0.30,
}
```

Novel commands (not previously seen in the session) receive a 0.2 novelty bonus. Detection risk is estimated from response timing patterns. Too-uniform timing (low coefficient of variation) or suspiciously fast responses (<1ms) indicate emulation.10.4.5 Q-Learning Update Rule

The agent uses tabular Q-learning with epsilon-greedy exploration:

Q(s, a) <- Q(s, a) + alpha * [r + gamma * max_a' Q(s', a') - Q(s, a)]

```
# From apps/api/app/scanning/deception/rl_chameleon.py
class QLearningChameleon:
    def update(self, state, action, reward, next_state, done):
        q_current = self._qtable[state][action]
        q_next_max = max(self._qtable[next_state]) if not done else 0.0
        target = reward + self.gamma * q_next_max
        td_error = target - q_current
        self._qtable[state][action] = q_current + self.alpha * td_error
        return td_error
```

Default hyperparameters: alpha=0.1, gamma=0.9. Epsilon starts at 0.3 and decays to 0.05 at a rate of 0.995 per episode.

10.4.6 Pre-Training on Synthetic Sessions

The Q-table is bootstrapped with 1,000 synthetic attacker sessions before real deployment. The pretrain_rl_chameleon() function simulates diverse attacker profiles (varying tool signatures, experience levels) interacting with the HoneypotEnvironment:

```
# From apps/api/app/scanning/deception/rl_chameleon.py
def pretrain_rl_chameleon(n_sessions=1000, max_steps_per_session=30, seed=42):
    agent = QLearningChameleon(epsilon=0.5, epsilon_decay=0.997)
    env = HoneypotEnvironment()
    for episode in range(n_sessions):
        state = env.reset(session_id=f"train_{episode}")
        for _step in range(max_steps_per_session):
            action, _ = agent.select_personality(state)
            next_state, reward, done = env.step(action)
            agent.update(state, action, reward, next_state, done)
            state = next_state
            if done:
```

```
                break
        agent.epsilon_decay()
    return agent
```

After pre-training, the agent converges to personality selection policies that match attacker profiles. Metasploit users are shown the Cisco router (high intelligence value from IOS commands), while Mirai-type scanners are shown the Hikvision camera (attracts mass exploitation attempts with low detection risk).

stateDiagram-v2
  [*] --> Observe: New connection
  Observe --> EncodeState: Extract tool signature
  EncodeState --> SelectAction: Q-table lookup
  SelectAction --> PresentPersonality: Epsilon-greedy
  PresentPersonality --> RecordResponse: Attacker command
  RecordResponse --> ScoreIntelligence: Intelligence scorer
  ScoreIntelligence --> ComputeReward: Combine intel + detection
  ComputeReward --> UpdateQ: Bellman update
  UpdateQ --> EncodeState: Next step
  UpdateQ --> [*]: Session ends

10.5 Session Recording with Hash Chains

10.5.1 The Integrity Problem

Forensic evidence from honeypot sessions must satisfy two requirements: (1) every packet is recorded in its original order, and (2) no packet can be modified, deleted, or inserted after recording without detection. The second requirement is important for incident response credibility and later evidentiary review.

10.5.2 SHA-256 Chain Construction

The SessionRecorder maintains a linked hash chain. The genesis hash is the SHA-256 of the session ID. Each subsequent packet hash includes the previous hash as a prefix:

H_0 = SHA-256(session_id)
H_n = SHA-256(H_{n-1} || packet_data)

```
# From apps/api/app/scanning/deception/session_recorder.py
class SessionRecorder:
    def __init__(self, session_id: str):
        self._genesis_hash = hashlib.sha256(session_id.encode()).hexdigest()
        self._previous_hash = self._genesis_hash
        self._packets: list[dict[str, Any]] = []
        self._chain: list[ForensicHash] = []

    def record_packet(self, direction: str, data: bytes, timestamp=None):
        current_hash = self._compute_chain_hash(data, self._previous_hash)
        self._packets.append({
            "index": len(self._packets),
            "direction": direction,
            "data": data.hex(),
```

```
            "sha256": current_hash,
        })
        self._chain.append(ForensicHash(
            packet_index=len(self._packets) - 1,
            sha256=current_hash,
            previous_hash=self._previous_hash,
        ))
        self._previous_hash = current_hash

    @staticmethod
    def _compute_chain_hash(data: bytes, previous_hash: str) -> str:
        h = hashlib.sha256()
        h.update(previous_hash.encode())
        h.update(data)
        return h.hexdigest()
```

10.5.3 Chain Verification

Verification re-computes each hash from the stored packets and compares it against the recorded chain. If any packet has been modified, the computed hash diverges from the stored hash, and all subsequent hashes fail.

```
def verify_chain(self) -> bool:
    prev_hash = self._genesis_hash
    for i, entry in enumerate(self._chain):
        data = bytes.fromhex(self._packets[i]["data"])
        expected = self._compute_chain_hash(data, prev_hash)
        if expected != entry.sha256:
            return False
        prev_hash = expected
    return True
```

The chain also supports PCAP export for analysis in Wireshark and JSON export for API consumption.

10.6 MITRE ATT&CK TTP Annotation

10.6.1 Technique Classification

The AttackerProfiler maps observed commands and events to MITRE ATT&CK technique identifiers using regex pattern matching against 18 technique categories:

```
# From apps/api/app/scanning/deception/attacker_profiler.py
MITRE_TECHNIQUE_PATTERNS: dict[str, list[str]] = {
    "T1110": [r"password", r"auth", r"login"],         # Brute Force
    "T1059.004": [r"cat\s+/etc", r"whoami", r"uname"],  # Unix Shell
    "T1046": [r"nmap", r"scan", r"netstat"],            # Network Service
Discovery
    "T1548": [r"sudo\b", r"su\b", r"chmod\s+[0-7]*s"], # Elevation Control
```

```
    "T1070": [r"rm\s+-rf\s+/var/log", r"history\s+-c"], # Indicator Removal
    "T1496": [r"xmrig", r"minergate", r"cpuminer"],     # Resource Hijacking
    # ... 12 more technique categories
}
```

For each attacker session, the profiler concatenates all commands, credentials, and events into a single text corpus and applies regex matching across all technique patterns. The result is a sorted list of MITRE technique IDs observed in the session.

10.6.2 Tool Detection

Offensive tool signatures are detected in parallel:

```
TOOL_SIGNATURES: dict[str, list[str]] = {
  "nmap": [r"nmap", r"Nmap"],
  "metasploit": [r"msfconsole", r"msfvenom", r"exploit/"],
  "mirai": [r"/bin/busybox", r"MIRAI", r"BOTNET"],
  "hydra": [r"hydra", r"xhydra"],
  # ... 14 more tool signatures
}
```

10.7 Attacker Profiling and Skill Estimation

10.7.1 Multi-Dimensional Skill Classification

Attacker skill is classified into four levels: script_kiddie, intermediate, advanced, and apt. The classification uses a composite scoring system with five dimensions:

Dimension	Weight	Indicators
Tool diversity	1-3 points	Number of distinct offensive tools detected
MITRE technique breadth	1-3 points	Number of distinct ATT&CK techniques observed
Command complexity	0-3.3 points	Pipe chains, variable usage, control structures
Session persistence	0-1.5 points	Duration >10 min indicates patience
Credential variety	0-0.5 points	Beyond simple brute force

The thresholds are: script_kiddie (<2.5), intermediate (2.5-5.0), advanced (5.0-8.0), apt (>=8.0).

10.7.2 Real-Time Skill Estimation

The SkillEstimator provides streaming skill estimation as events arrive. It analyses three dimensions independently:

1. **Command sophistication** (0-10) – checks for pipe chains, variable usage, APT indicators like unset HISTFILE or LD_PRELOAD.
2. **Timing regularity** (0-1) – coefficient of variation of inter-event intervals. Lower dispersion (CV < 0.3) is more consistent with automation. Higher dispersion (CV > 0.8) paired with low error rates can indicate slower human-driven interaction.
3. **Error rate** (0-1) – repeated commands, failed auth attempts, typos. A high error rate indicates lower skill.

```
# From apps/api/app/scanning/deception/skill_estimator.py
_APT_INDICATORS = [
    r"history\s+-c",
    r"unset\s+HISTFILE",
    r"export\s+HISTSIZE=0",
    r"shred\s+",
    r"rm\s+-rf\s+/var/log",
    r"LD_PRELOAD",
    r"/proc/self/exe",
    r"ptrace",
    r"strace\s+",
]
```

```
graph LR
  A[Session Events] --> B[Command Sophistication]
  A --> C[Timing Regularity]
  A --> D[Error Rate]
  B --> E[Composite Score]
  C --> E
  D --> E
  E --> F{Score >= 8.0?}
  F -->|Yes| G[APT]
  F -->|No| H{Score >= 5.0?}
  H -->|Yes| I[Advanced]
  H -->|No| J{Score >= 2.5?}
  J -->|Yes| K[Intermediate]
  J -->|No| L[Script Kiddie]

  style G fill:#ffcdd2
  style I fill:#fff9c4
  style K fill:#e1f5fe
  style L fill:#e8f5e9
```

10.8 Cognitive Game-Theoretic Honeypots

10.8.1 Bayesian Attacker Modeling

The CognitiveHoneypot maintains a belief distribution over attacker types and updates it using Bayesian inference as events arrive. This is a game-theoretic approach: the defender (the honeypot) and the attacker are modeled as players in an information-asymmetry game.

Prior probabilities reflect the real-world distribution of attacker types:

```
# From apps/api/app/scanning/deception/cognitive_honeypot.py
ATTACKER_TYPE_PRIORS: dict[str, float] = {
    "script_kiddie": 0.55,
    "intermediate": 0.25,
    "advanced": 0.13,
    "apt": 0.07,
```

```
}
```

10.8.2 Bayesian Update

When an event is observed, the belief distribution is updated via Bayes' rule:

P(type | observation) proportional to P(observation | type) * P(type)

The observation likelihoods encode the likelihood of each event type for each attacker class. For example, auth_attempt events are 6x more likely from script kiddies (0.60) than from APT actors (0.10), while data_transfer events are 10x more likely from APT actors (0.50) than from script kiddies (0.05).

```
def update_belief(self, event: AttackerEvent) -> CognitiveState:
    likelihoods = _OBSERVATION_LIKELIHOODS.get(event.event_type.value)
    unnormalized = {}
    for attacker_type, prior in self._belief.items():
        likelihood = likelihoods.get(attacker_type, 0.5)
        unnormalized[attacker_type] = prior * likelihood
    # Normalize
    total = sum(unnormalized.values())
    self._belief = {k: v / total for k, v in unnormalized.items()}
```

10.8.3 Action Prediction and Response Selection

The cognitive honeypot predicts the attacker's next action using transition probability matrices:

```
TRANSITION_PROBABILITIES = {
  "apt": {
    "brute_force": 0.02, "scan_ports": 0.05,
    "pivot_lateral": 0.25, "exfiltrate": 0.20,
    "install_backdoor": 0.18, "clean_logs": 0.12, ...
  },
  "script_kiddie": {
    "brute_force": 0.40, "scan_ports": 0.25,
    "pivot_lateral": 0.02, "exfiltrate": 0.01, ...
  },
}
```

The predicted next action determines the honeypot response strategy:

- **APT detected**: deep engagement – realistic breadcrumbed environment, 50ms response delay, high-value fake data, 10-minute session limit.
- **Script kiddie detected**: credential capture – 500ms tarpit delay, no breadcrumbs, 60-second session limit.
- **Brute force predicted**: activate tarpit behavior (1000ms+ delay per attempt).

10.8.4 Information Gain

The system tracks Shannon entropy of the belief distribution. Information gain per observation is the reduction in entropy:

IG = H(belief_prior) - H(belief_posterior)

This quantifies how much each attacker action reveals about their type, guiding the honeypot to elicit actions that maximise information gain.

10.9 Placement Optimization Using Attack Graphs

10.9.1 The Placement Problem

Given a scanned network with N hosts and up to K honeypot slots, where should honeypots be placed to maximise the probability of detecting lateral movement? This is a variant of the weighted maximum coverage problem (NP-hard), which Breakwater approximates with a greedy weighted set-cover heuristic.

10.9.2 Scoring Function

Each candidate's IP receives a composite score based on:

1. **Proximity to high-value hosts:** nearby hosts with CVEs contribute 0.3 points per vulnerability (capped at 5.0). Hosts within 10 IP addresses count as "nearby."
2. **Attack graph centrality** – if the attack graph from Chapter 4 identifies a candidate near a high-centrality node (betweenness centrality > 0.5), a bonus of centrality * 2.0 is **Attack path proximity:** candidates within 5 IPs of an attack path target receive +2.0 points.p**Subnet edge penalty:** IPs at the boundaries of the subnet (last octet <5 or >250) receive a 50% score reduction, as they are less plausible locations for real devices. evices.

```python
# From apps/api/app/scanning/deception/placement_optimizer.py
class PlacementOptimizer:
    def weighted_set_cover(self, candidates, weights, max_select=10):
        remaining = dict(weights)
        selected = []
        while remaining and len(selected) < max_select:
            best_ip = max(remaining, key=remaining.get)
            if remaining[best_ip] <= 0:
                break
            selected.append(best_ip)
            del remaining[best_ip]
            # Penalize same-/24 candidates by 70% to encourage diversity
            best_net = ipaddress.ip_network(f"{best_ip}/24", strict=False)
            for ip in list(remaining.keys()):
                if ipaddress.ip_address(ip) in best_net:
                    remaining[ip] *= 0.3
        return selected
```

10.9.3 Coverage Score

After placement, the optimizer computes a coverage score: the fraction of all network hosts that are within one L3 hop of a honeypot, using the attack graph adjacency list.

graph TD
  A[Scan Results + Attack Graph] --> B[Infer Subnets]
  B --> C[Find Unoccupied IPs]
  C --> D[Score Each Candidate]
  D --> E[Weighted Set Cover]

```
E --> F[Protocol Selection]
F --> G[Deployment Plan]

D --> D1[Proximity Score]
D --> D2[Attack Graph Bonus]
D --> D3[Centrality Bonus]
D --> D4[Edge Penalty]

style E fill:#e1f5fe
style D fill:#fff3e0
```

10.10 Moving Target Defense

10.10.1 The Fingerprinting Problem

Static honeypots are vulnerable to fingerprinting. An attacker who connects to the same IP: port repeatedly can detect the honeypot by observing consistent, stereotyped responses – or by checking databases like Shodan's known honeypot signature library.

10.10.2 Rotation Strategy

The MovingTargetDefense engine periodically rotates honeypot configurations across three dimensions:

1. **IP rotation** – reassign honeypots to new IPs from a pool of unoccupied addresses.
2. **Personality rotation** – swap device personas (e.g., switching from a Hikvision camera to a Siemens PLC).
3. **Port rotation** – shift listening ports to evade port-based fingerprinting. Rotation triggers on two conditions: t**imer-based** configurable interval (default 3600 seconds) and **fingerprinting detected,** which causes immediate rotation when scanning tools (nmap, masscan, Shodan, Censys, Nikto) are detected in session commands. commands.

```python
# From apps/api/app/scanning/deception/moving_target.py
def should_rotate(self, last_rotation, interval, fingerprinting_detected):
    if fingerprinting_detected:
        return True  # Immediate rotation
    return (time.time() - last_rotation) >= interval
```

10.11 Forensic Evidence Chains with Merkle Trees

10.11.1 From Hash Chains to Merkle Trees

The per-session hash chain provides tamper evidence for individual sessions. But an investigation may involve dozens of sessions, hundreds of thousands of packets, and multiple custodians. The EvidenceChain class builds a binary Merkle tree over all evidence items, enabling:

1. **Efficient inclusion proofs** – prove that a specific packet belongs to the evidence set without revealing all other packets.

Note: 'Inclusion proof' is standard Merkle tree terminology. A Merkle inclusion proof cryptographically demonstrates that a specific leaf hash is part of the tree without revealing other leaves. The word 'prove' here is used in its technical cryptographic sense, not as an absolute guarantee.

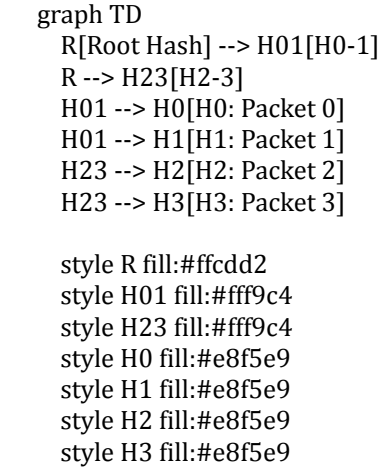
2. **Compact integrity verification** – verify the integrity of the entire evidence set using only the Merkle root (32 bytes).
3. **Chain-of-custody tracking** – record custodians and events in the evidence lifecycle.

10.11.2 Tree Construction

```python
# From apps/api/app/scanning/deception/evidence_chain.py
class EvidenceChain:
    def build_merkle_tree(self) -> str:
        leaves = list(self._leaf_hashes)
        # Pad to power of 2
        while len(leaves) & (len(leaves) - 1) != 0:
            leaves.append(leaves[-1])
        self._tree = [leaves]
        current = leaves
        while len(current) > 1:
            next_level = []
            for i in range(0, len(current), 2):
                left = current[i]
                right = current[i + 1] if i + 1 < len(current) else left
                next_level.append(_hash_pair(left, right))
            self._tree.append(next_level)
            current = next_level
        self._root = current[0]
        return self._root
```

10.11.3 Inclusion Proofs

An inclusion proof for an item at index i consists of the sibling hashes from the leaf level up to the root. The proof length is O(log N), where N is the number of evidence items. A verifier can recompute the root from the item hash and the proof, confirming membership without access to the full data set.



To prove Packet 1 is in the tree, the proof contains [H0, H2-3]. The verifier computes H0-1 = SHA-256(H0 || H1), then Root = SHA-256(H0-1 || H2-3), and checks against the published root.

10.12 Blockchain Evidence Anchoring

10.12.1 Anchoring Architecture

The BlockchainAnchor abstract base class provides a pluggable interface for anchoring Merkle roots to tamper-evident ledgers. Two implementations exist:

LocalLedger (default): An SQLite-backed append-only ledger where each entry contains the Merkle root, a chain hash linking to the previous entry, and metadata. The chain hash is computed as SHA-256(previous_chain_hash || merkle_root), creating a simplified blockchain structure.

```
# From apps/api/app/scanning/deception/blockchain_anchor.py
class LocalLedger(BlockchainAnchor):
    async def anchor(self, merkle_root, metadata=None):
        previous_hash = self._get_last_chain_hash()
        chain_hash = hashlib.sha256(
            (previous_hash + merkle_root).encode()
        ).hexdigest()
        # INSERT into evidence_ledger ...
        return entry_id
```

EthereumAnchor (stub): For high-assurance deployments, a stub implementation provides the interface for submitting Merkle roots as Ethereum transactions. This enables third-party verification without trusting the Breakwater operator's local storage.

10.12.2 Chain Integrity Verification

The verify_chain_integrity() method traverses the entire ledger in chronological order, recomputing each chain hash from the previous entry's hash and the stored Merkle root. Any tampering – modified root, deleted entry, or reordered entries – breaks the chain.

10.D Seminar War Rooms

This section is intentionally demanding. Each war room asks the reader to work like a senior analyst responsible for deception system decisions in live IoT or OT environments. The task is not to recite the chapter. The task is to make a bounded claim under pressure, then defend the evidence boundary when another expert attacks it.

10.D1 Slow Intruder

War Room 1 begins with the slow intruder. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should interrogate the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the slow intruder is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership

sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For a slow intruder, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the slow intruder is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D2 Credential Canary

War Room 2 begins with the credential canary. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should challenge the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the credential canary is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For credential canary, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for Credential Canary is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D3 Timing Probe

War Room 3 begins with the timing probe. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should bound the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the timing probe is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For the timing probe, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and

which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the timing probe is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D4 Fake PLC Physics

War Room 4 begins with the fake PLC physics. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should quarantine the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the fake PLC physics is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For fake PLC physics, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for fake PLC physics is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D5 Privacy-Preserving Lure

War Room 5 begins with the privacy-preserving lure. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should simulate the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the privacy-preserving lure is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For privacy-preserving lure, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for privacy-preserving lure is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D6 Visible Rotation

War Room 6 begins with the visible rotation. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should replay the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the visible rotation is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For visible rotation, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for visible rotation is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D7 Courtroom Evidence Chain

War Room 7 begins with the courtroom evidence chain. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should audit the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the courtroom evidence chain is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For the courtroom evidence chain, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the courtroom evidence chain is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D8 Reward Function Trap

War Room 8 begins with the reward function trap. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should stage the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the reward function trap is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For the reward function trap, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the reward function trap is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D9 Operator Dialogue

War Room 9 begins with the operator dialogue. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should defer the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow-test, collect, rollback, or reject.

The final artifact for the operator dialogue is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For operator dialogue, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for operator dialogue is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D10 Coverage Budget

War Room 10 begins with the coverage budget. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed

artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should escalate the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the coverage budget is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For coverage budget, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the coverage budget is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D11 Attacker Profiling The Defender

War Room 11 begins with the attacker profiling the defender. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should rollback the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the attacker profiling the defender is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For attacker profiling the defender, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for attacker profiling the defender is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove.

10.D12 Honeypot Teardown

War Room 12 begins with the honeypot teardown. The first analyst wants to accept the decoy dialogue because it fits the expected story. The second analyst refuses to move until the containment policy is tied to an observed artifact, an owner, and a clock. The disagreement is productive. It prevents the team from treating a plausible explanation as verified.

The class should instrument the case by changing one fact that a real adversary could influence: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The

answer must state which claim survives the change and which claim fails. A doctoral answer does not hide behind caution. It chooses a precise operational state: admit, hold, shadow- test, collect, rollback, or reject.

The final artifact for the honeypot teardown is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and hunt pivot. It also contains one leadership sentence that is weaker than the analyst wants and stronger than a lawyer would fear. That sentence is the craft. It carries enough truth to guide action without smuggling in facts the system did not prove.

For honeypot teardown, change one adversary-controlled fact: timing, identity, provenance, replay, exception handling, sampling, maintenance state, or operator pressure. The answer must state which claim survives and which claim fails. A doctoral answer chooses a precise operational state rather than hiding behind generic caution.

The final artifact for the honeypot teardown is a compact evidence packet. It contains the input record, transformation, unresolved conflict, decision authority, and recovery or hold gate. It also contains one leadership sentence that guides action without smuggling in facts that the system did not prove. ## 10. E Adversarial Oral Examination

These oral-exam prompts are designed to make expert students uncomfortable in the right way. Each prompt forces a precise claim about deception operations, then changes one operational fact that could occur in a real IoT or OT environment.

10.E1 When Decoy delays match plant physics too well

The examiner gives the student a clean initial narrative, then reveals that the decoy delays match the plant physics too well. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for decoy delays that match plant physics too well has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter.

10.E2 When the Canary credential appears in a real script

The examiner gives the student a clean initial narrative, then reveals that the canary credential appears in a real script. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for the canary credential, as seen in a real script, has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter.

10.E3 When the attacker detects the rotation schedule

The examiner gives the student a clean initial narrative, then reveals that the attacker detects the rotation schedule. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for the attacker detecting the rotation schedule has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter.

10.E4 When the session recorder misses first command

The examiner gives the student a clean initial narrative, then reveals that the session recorder misses the first command. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for the session recorder, when the first command is missing, has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter.

10.E5 When the reward function favors long dwell over evidence

The examiner gives the student a clean initial narrative, then reveals that the reward function favors long dwell over evidence. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for the reward function favors long dwell over evidence has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter.

10.E6 When Fake service exposes defender tooling

The examiner gives the student a clean initial narrative, then reveals that the fake service exposes defender tooling. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for a fake service that exposes defender tooling has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and the next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter.

10.E7 When the Legal reviewer rejects chain metadata

The examiner gives the student a clean initial narrative, then reveals that the legal reviewer rejects the chain metadata. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for a legal reviewer who rejects the chain metadata has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter.

10.E8 When the Hunt pivot threatens production stability

The examiner gives the student a clean initial narrative, then reveals that the hunt pivot threatens production stability. The student must decide whether the original recommendation survives. A passing answer does not retreat into general caution. It names the exact sentence that must be weakened and the exact evidence that would restore it.

The expected response has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary, they have not mastered the chapter.

The expected response for the hunt pivot threatens production stability has four parts: current operational state, evidence still admissible, evidence now contaminated or incomplete, and next collection step. The answer should be short enough to brief during an incident and rigorous enough for another senior analyst to reproduce. If the student cannot name the boundary for this prompt, the student has not mastered the chapter. ## Assumptions and Limitations

Phase 10 makes four assumptions that should stay visible.

First, emulation fidelity is bounded. A low- or medium-interaction honeypot may be convincing during first-pass reconnaissance and early exploitation. However, a patient attacker can still detect the decoy by probing timing, file-system behavior, protocol corner cases, or process-physics inconsistencies.

Second, the RL scores are planning heuristics. Intelligence yield, detection risk, and command value depend on the attacker population, the emulated device family, and the surrounding network story. The values in this chapter are defensible priors, not calibrated scientific constants.

Third, attacker profiling is inference, not attribution. A session may look automated, careful, or operator-driven without proving who the actor is. The model helps with triage and prioritization. It does not identify a person or an organization on its own.

Fourth, Merkle trees and ledger anchors strengthen integrity claims over collected artifacts. They do not, by themselves, establish legal admissibility, prove truthful collection, or remove the need for counsel, chain-of-custody discipline, and jurisdiction-specific review.

> **Caution:** Merkle trees provide cryptographic integrity, not legal admissibility. A court may require chain-of-custody documentation, witness testimony, and forensic methodology review in addition to the technical evidence.

10.B Deep Technical Expansion

10.B1 Decoy safety enA decoy should never become a bridge. It needs egress controls, credential isolation, synthetic data boundaries, teardown rules, and health monitoring. Believability is valuable only within that safety envelope.

The safety review should begin with abuse cases. Can the decoy relay traffic? Store real credentials? Reach production control networks? Can an attacker use it to frame a third party? Can a defender mistake decoy output for plant telemetry? If any answer is uncertain, the decoy is not ready for deployment.

The evidence record should include the egress policy, credential policy, synthetic-data generator, reset interval, containment test, and escalation owner. A convincing decoy without these controls is a liability. A less glamorous decoy with strong containment may produce less drama and better intelligence.

10.B2 Attacker learning model

The attacker is also learning. Each decoy response may update the attacker's belief about the environment. A mature deception system plans for this feedback loop instead of assuming the defender is the only observer.

This makes deception a repeated game. A banner, delay, error message, file path, and credential prompt are all signals the attacker tests. The defender should rotate them with intent, not noise. Randomness that breaks the story helps the attacker. Variation that matches the surrounding network story makes deception harder to discard.

FFor expert students, the hard question is what the attacker is allowed to learn. A decoy may reveal that the defender detects scanning, that a subnet is under surveillance, or that a fake PLC exists. The design should decide which revelation is acceptable because it buys time, attribution evidence, or containment value. 10. A Advanced Practitioner Fieldbook: What Experts Still Miss.

This fieldbook expands Chapter 10 for doctoral students and senior cybersecurity practitioners. It assumes the reader knows the vocabulary. The goal is harder. The reader must reason under pressure, preserve evidence boundaries, and explain why the analysis should be trusted when the environment fights back.

The prose uses short sentences on purpose. They expose weak claims and make the material teachable in a live seminar.

10.A1 The intruder who slowed down

In a manufacturing network, the attacker stops scanning when a decoy answers too perfectly. That pause is evidence. The system learned from hesitation.

Figure 10.18: Attacker Engagement Timeline

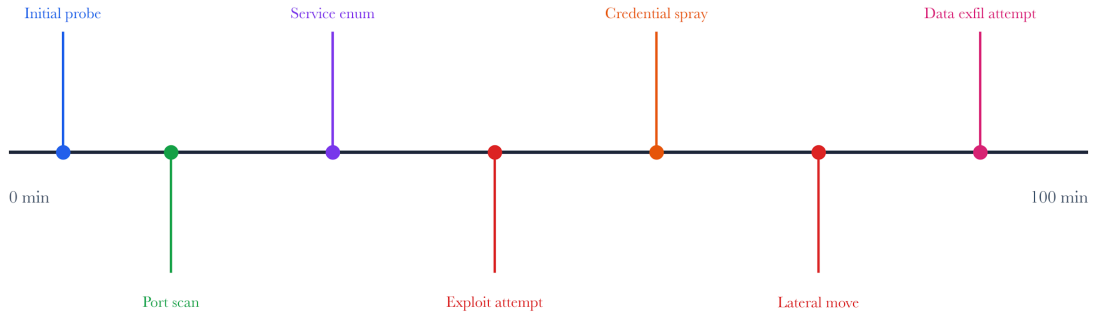


Figure 10.A1: ch10 fig18 engagement timeline.

Figure note: The engagement timeline matches the slowed intruder. It shows how pace changes the defender's interpretation of skill, caution, and objective.

For the intruder who slowed down, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion remains forbidden.

The specific danger in the intruder who slowed down is false closure. containment policy may support one interpretation while the attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for the intruder who slowed down produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. These artifacts should agree on the same boundary. If not, the analyst has confused persuasion with proof.

10.A2 Honeypot fidelity as a measurement contract

A honeypot is a measuring instrument. Low fidelity measures opportunism. High-fidelity measures procedure.

Figure 10.1: Honeypot Interaction Levels

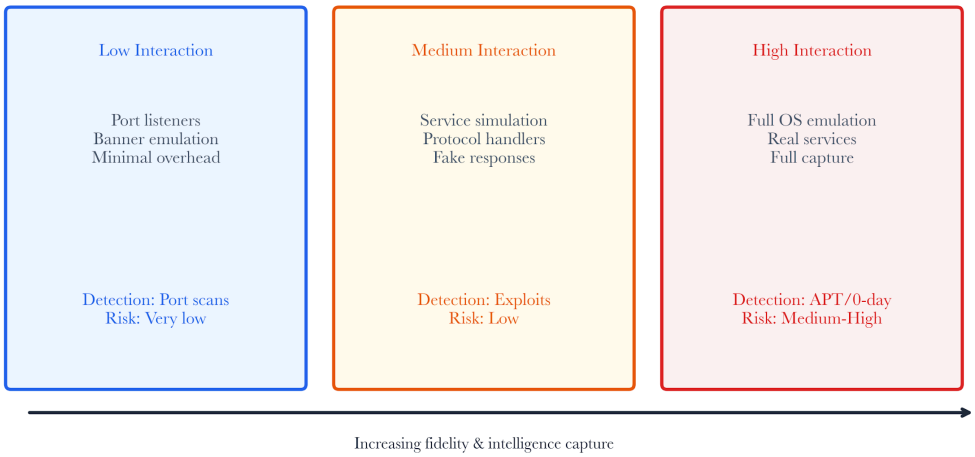


Figure 10.A2: ch10 fig01 honeypot interaction levels.

Figure note: Interaction levels define fidelity boundaries. The figure keeps the subsection honest about what a low-, medium-, or high-interaction decoy can prove.

For honeypot fidelity as a measurement contract, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion remains forbidden.

The specific danger in honeypot fidelity as a measurement contract is the risk of false closure. The containment policy may support one interpretation, while the attacker belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for honeypot fidelity as a measurement contract produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analytic has confused persuasion with proof.

An RL deception policy can learn bad habits. If the reward favors longer sessions, it may create theatrical traps that reveal the defense.

An RL deception policy can learn bad habits. If the reward favors longer sessions, it may create theatrical traps that reveal the defense.

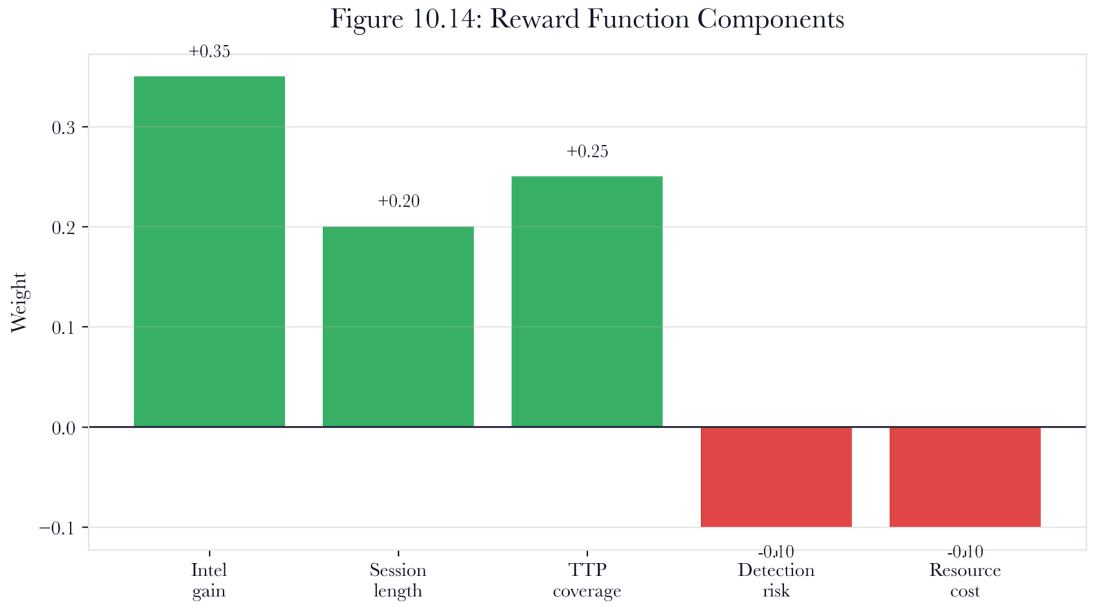


Figure 10.A3: ch10 fig14 reward function components.

Figure note: Reward components belong to deceptive optimization. The figure helps students see when the RL objective rewards the wrong operational behavior.

For reward functions that lie, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion remains forbidden. The specific danger in reward functions that lie is false closure. containment policy may support one interpretation while the attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer to a reward function that lies produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analytic has confused persuasion with proof.

10.A4 Session recording as forensic duty

A deception session without integrity protection is entertainment. A session with hash chains and custody metadata becomes evidence.

Figure 10.4: Session Hash Chain for Evidence Integrity

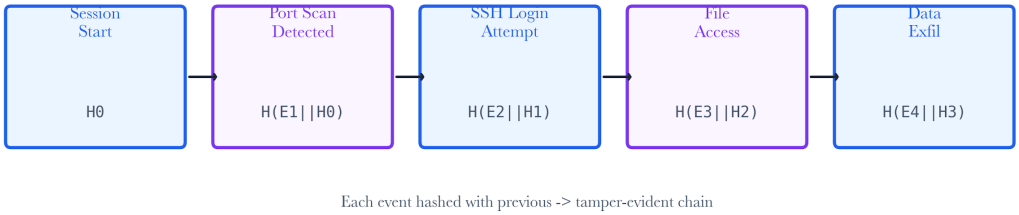


Figure 10.A4: ch10 fig04 session hash chain.

Figure note: The hash chain belongs with session recording. It ties observation, integrity, and later review into one forensic duty.

For session recording as forensic duty, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion remains forbidden.

The specific danger in session recording as a forensic duty is the risk of false closure. The containment policy may support one interpretation, while the attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for session recording as forensic duty produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof.

10.A5 Bayesian attacker modeling

Every command updates a belief. Tool choice, typing cadence, path selection, and failed guesses alter the probability distribution.

Figure 10.6: Bayesian Attacker Skill Model

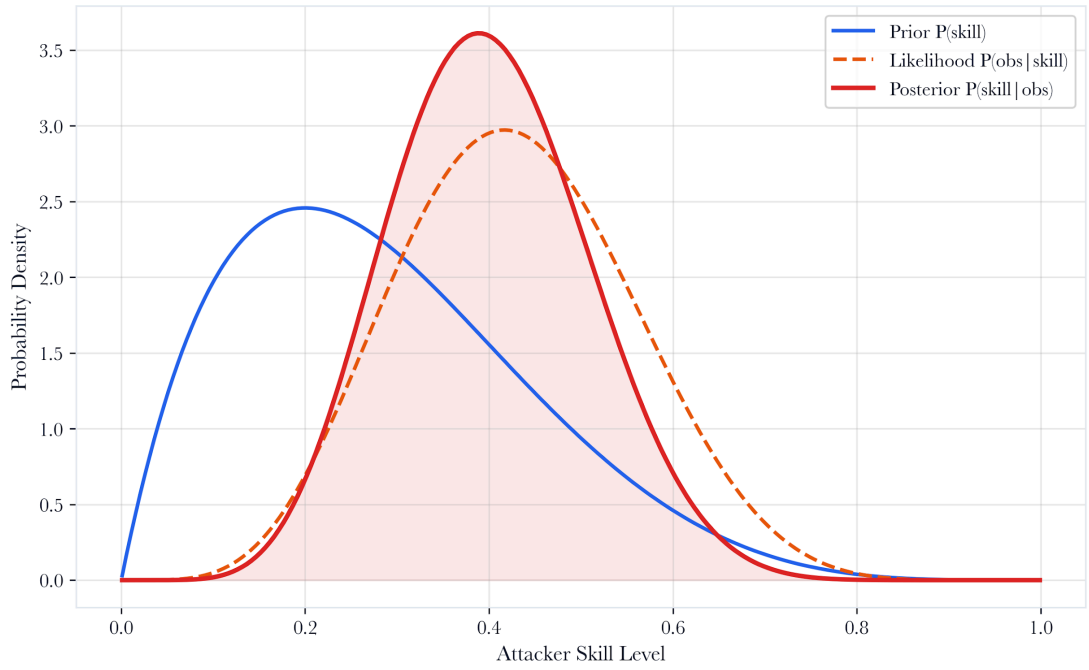


Figure 10.A5: ch10 fig06 bayesian attacker model.

Figure note: The Bayesian model belongs to attacker inference. It expresses uncertainty rather than pretending the session identifies a person or group.

For Bayesian attacker modeling, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion remains forbidden.

The specific danger in Bayesian attacker modeling is the tendency toward false closure. The containment policy may support one interpretation while attacker belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for bayesian attacker modeling produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analytic has confused persuasion with proof.

10.A6 Placement on attack graphs

A decoy behind the wrong choke point is invisible. A decoy on a high-betweenness path turns graph theory into a collection strategy.

Figure 10.7: Honeypot Placement in Network Topology

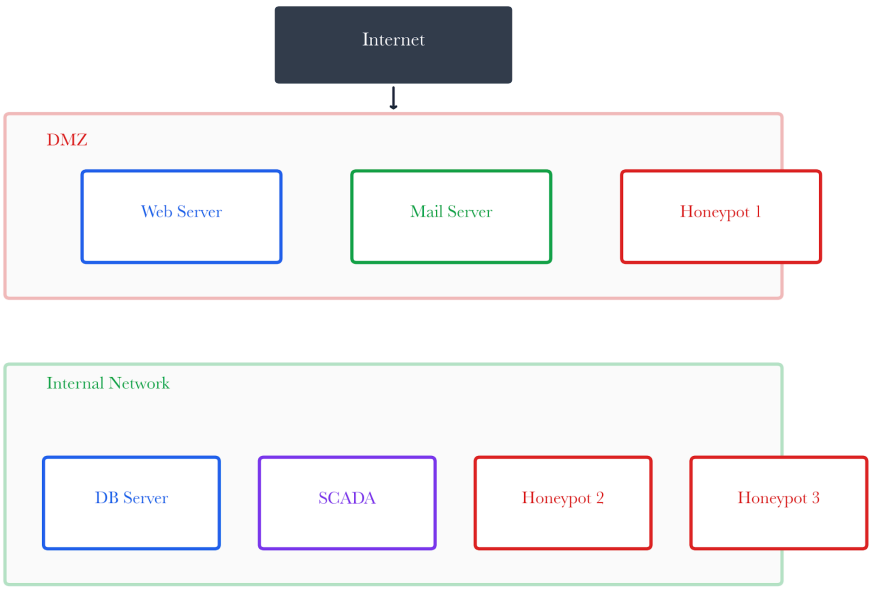


Figure 10.A6: ch10 fig07 honeypot placement topology.

Figure note: Placement topology belongs with attack-graph decisions. It shows where a decoy can observe movement without becoming a bridge.

For placement on attack graphs, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion remains forbidden.The specific danger in placement on attack graphs is false closure. containment policy may support one interpretation while the attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for placement on attack graphs produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof.

10.A7 Moving target defense without chaos

RRotation can help. Randomness can harm. Movement must resist fingerprinting and preserve analyst interpretation.

Figure 10.11: Moving Target Defense with Deception

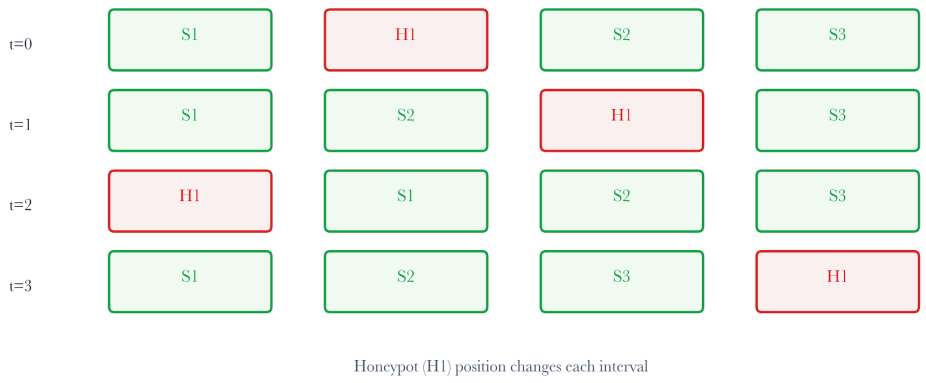


Figure 10.A7: ch10 fig11 moving target defense.

Figure note: Moving target defense belongs with controlled variation. The figure shows the difference between strategic rotation and random noise.

For moving-target defense without chaos, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion remains forbidden. The specific danger in moving target defense without chaos is the risk of false closure. containment policy may support one interpretation, while the attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for moving target defense without chaos produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof.

10.A8 Credential canaries as silent tripwires

AA canary credential is meant to move, not authenticate. Its use creates a high-confidence alert.

Figure 10.10: Deception Response Matrix

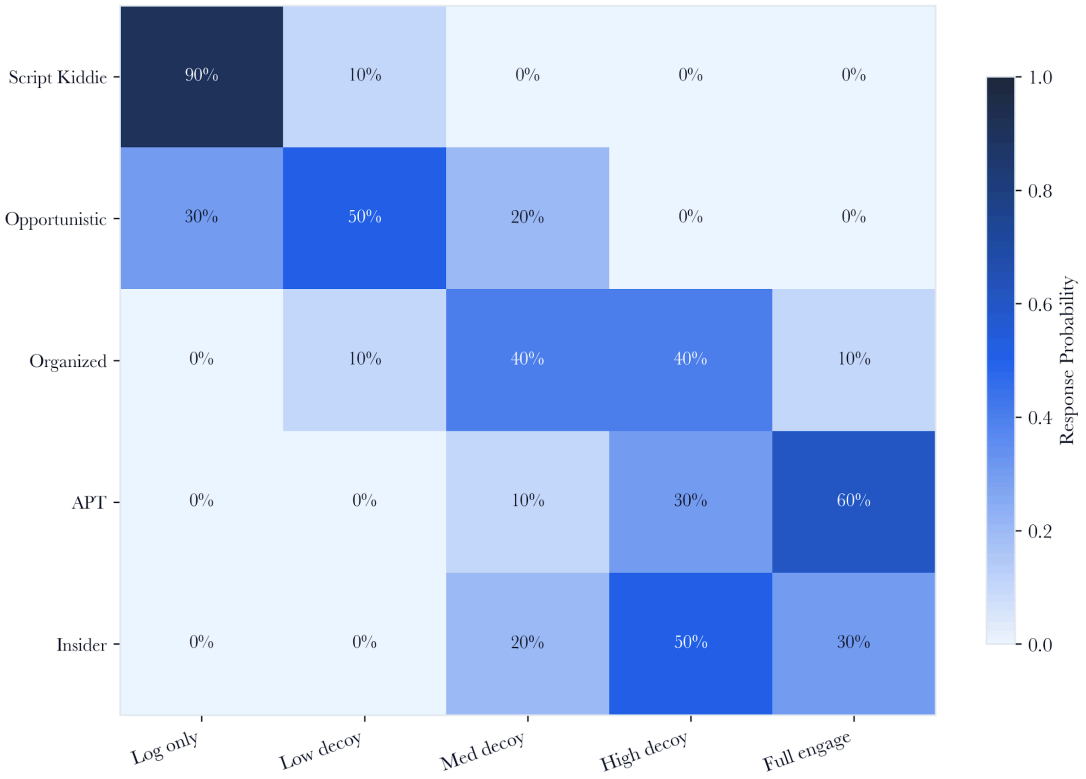


Figure 10.A8: ch10 fig10 deception response matrix.

Figure note: The response matrix belongs with credential canaries. It maps tripwire contact to containment, evidence, and hunt action.

For credential canaries as silent tripwires, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion remains forbidden.

The specific danger in credential canaries as silent tripwires is the false sense of closure. containment policy may support one interpretation while the attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for credential canaries as silent tripwires produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analytic has confused persuasion with proof.

10.A9 TTP annotation under uncertainty

A command can resemble discovery, staging, or error. Mapping to ATT&CK must preserve uncertainty.

For ttp annotation under uncertainty, evidence quality is uneven. The airport operations center may provide a precise decoy session and still hide the condition that matters. That is why chapter-level mastery requires evidence boundaries, not vocabulary recall.

The review question is practical: where can an adversary make the system overconfident? The answer usually lies between the containment rule, the attacker's belief, and the hunt pivot. That gap is where expert students should work.

The section should end with an action: quarantine, shadow-test, narrow the claim, collect one more artifact, or reject the conclusion. A vague recommendation is a failed analytic output.

For ttp annotation under uncertainty, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in the TTP annotation under uncertainty is false closure. The containment policy may support one interpretation, while the attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for ttp annotation under uncertainty produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof.

10.A10 The ethics of believable systems

A decoy should be believable without exposing real secrets or enabling unauthorized movement.

The ethics of believable systems should change how analysts brief on risk. In the police evidence facility, leadership does not need a tour of the algorithm. They need to know which decoy session is admissible, which assumption is fragile, and which decision can be made now.

The system should resist false closure. When containment rule and hunt pivot disagree, the correct state may be unresolved. This is a professional answer, provided the unresolved state has an owner, expiry, and next evidence request.

The highest-grade student will keep model behavior, governance, and mission consequence separate. Merging them into one confidence score destroys the audit trail that makes cyber analytics defensible.

For the ethics of believable systems, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in the ethics of believable systems is the temptation to false closure. containment policy may support one interpretation while the attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for the ethics of believable systems produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof.

10.A11 Merkle roots and courtrooms

Merkle roots provide compact proof that a large evidence collection remains unchanged.

In the factory cell, Merkle roots and courtrooms become a claim-control problem within deception analytics. The analyst starts by naming the observed decoy session, the transformation that touched it, and the smallest defensible conclusion. Anything stronger has to wait for evidence.

The failure path is subtle. The dashboard consolidates containment rule, attacker belief, and hunt pivot into a single status. A senior reviewer should separate them and preserve the disagreement as a first-class record.

The stop rule matters. The subsection should enable the reader to say what blocks the claim, what promotes it, and what keeps it in review. This discipline distinguishes analytics from decorative scoring.

For Merkle roots and courtrooms, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in Merkle roots and courtrooms is the false sense of closure. The containment policy may support one interpretation, while the attack's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for Merkle roots and courtrooms produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof.

10.A12 Deception coverage as a budget problem

The organization cannot decoy everything. It should decoy the attacker into making a choice.

Treat deception coverage as a budget problem as an adversarial hearing, not a feature description. The municipal water site gives the system partial data, delayed data, and political pressure. The answer must still bind the decoy session to a named decision.

A strong implementation keeps the uncomfortable edge visible. If the containment rule says proceed while the hunt pivot says wait, the system should not average the conflict into the confidence. It should record the conflict and assign ownership.

The doctoral move is to ask for the counterfactual. What observation would make the original claim false? If the workflow cannot answer, it has built a belief engine rather than a cyber-analytic control.

For deception coverage as a budget problem, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in deception coverage as a budget problem is the false closure it creates. The containment policy may support one interpretation, while the attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for deception coverage as a budget problem produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof.

10.A13 When the attacker profiles the defender

Skilled attackers test latency, banners, timestamps, and anomalous process trees.

The operational test for when the attacker profiles the defender is whether a second expert can replay the reasoning. In a residential safety platform, this means the input record, transformation, exception handling, and decision authority must survive handoff.

The dangerous shortcut is to trust the most convenient signal. The attacker's belief may look stable, while the hunt pivot carries the real warning. The containment rule may look mathematically clean, while the mission context changes the cost of error.

A useful report should be modest and sharp. It should say what the analyst saw, what it inferred, what it refused to infer, and which collection step would move the case forward.

For when the attacker profiles the defender, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in when the attacker profiles the defender is false closure. A containment policy may support one interpretation, while an attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer to the question of when the attacker profiles the defender produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof.

10.A14 Threat hunting from dialogue

Interactive deception turns hunting into conversation. Each prompt and retry becomes observation.

In threat hunting from dialogue, evidence quality is unevenly distributed. The regional hospital may provide a precise decoy session and still hide the condition that matters. That is why chapter-level mastery requires evidence-based boundaries, not vocabulary recall.

The review question is practical: where can an adversary make the system overconfident? The answer usually sits between the containment rule, the attacker's, and the hunt pivot. That gap is where expert students should work.

The section should end with an action. Quarantine, shadow-test, narrow the claim, collect one more artifact, or reject the conclusion. A vague recommendation is an analytic failure.

For threat hunting from dialogue, the review starts with the decoy session and ends only when the claim has a named owner. The analyst should state what the system observed, what transformation changed the record, and what conclusion is still forbidden.

The specific danger in threat hunting from dialogue is false closure. The containment policy may support one interpretation, while the attacker's belief points toward another. The case should remain open until the hunt pivot explains why the stronger claim is admissible or why it has been refused.

A senior answer for threat hunting from dialogue produces three artifacts: a leadership sentence, a reproducible evidence note, and a hostile objection. The three artifacts should agree on the same boundary. If they do not, the analyst has confused persuasion with proof.

10.A15 Doctoral Mastery Check

A student has mastered Chapter 10 when they can design deception that collects intelligence without becoming unsafe, theatrical, or legally sloppy. The answer must name the decoy story, containment boundary, telemetry value, attacker-learning risk, evidence chain, and hunt pivot.

Perturb the case with five deception shocks: make one decoy too convincing, make one canary credential leak into production, make one rotation visible to the attacker, make one session log incomplete, and make one response increase dwell time without increasing intelligence. The student should revise the policy and explain the safety envelope.