ANTARES

The Antares *Decision Model*

A Research Brief on Organizational Cybersecurity Decision-Making

Branden Rowe
FOUNDER & MANAGING DIRECTOR
ANTARES SECURITY LLC

VERSION 1.0
AUGUST 2026

Contents

The Antares Decision Model: A Research Brief on Organizational Cybersecurity Decision-Making
Branden Rowe — Founder & Managing Director, Antares Security LLC · Version 1.0 · August 2026

EXECUTIVE SUMMARY

Security outcomes are a property of decisions.

Most cybersecurity programs are built around the wrong unit of analysis.

Organizations invest in tools, frameworks, and compliance certifications. They build control inventories, conduct annual risk assessments, and publish security policies. They acquire technology that produces dashboards, metrics, and reports.

And then they experience a breach, fail an audit, or lose a client — not because the tools were wrong, but because the decisions behind the tools were fragmented, unowned, or never made in the first place.

The Antares Decision Model (ADM) was developed to address that problem directly. It treats the decision — not the control, the policy, or the tool — as the unit that determines whether a security program actually functions. And it provides a structured approach for making those decisions visible, ownerable, and executable across the functions responsible for acting on them.

This brief applies the ADM to three operational contexts where cybersecurity decision-making most commonly breaks down: vCISO engagement, AI governance, and incident response. In each context, the same pattern emerges. Organizations have governance artifacts. They do not have functioning decision systems.

The brief then documents the failure modes the ADM predicts in each context, and draws implications for organizations trying to build security programs that hold up under pressure rather than on paper.

> *Security outcomes are a property of decisions. Organizations that build structures for making better security decisions will consistently outperform those that focus on accumulating controls and frameworks.*

The central argument is simple: security outcomes are a property of decisions. Organizations that build structures for making better security decisions will consistently outperform those that focus on accumulating controls and frameworks — not because the controls don't matter, but because controls without decision discipline degrade, stall, or fail exactly when they are needed most.

SECTION 01

The Decision Problem

The gap is not in the technology. The gap is in the decisions.

Across industries, the same pattern repeats.

The organization has tools. It has policies. It has a framework it references — NIST CSF, CIS Controls, ISO 27001, SOC 2. It has some form of risk register, some version of an incident response plan, and documentation for most of the controls an auditor would ask about.

It also has a security gap it cannot explain.

The gap is not in the technology. The technology is largely doing what it was configured to do. The gap is in the decisions that determined how the technology was configured, who owns the outcomes it produces, and what happens when the outputs require a leadership response.

When a vendor assessment surfaces a critical finding, who decides whether the vendor relationship continues? When a risk register item has been open for three quarters, who has the authority to accept the risk or require remediation? When the board asks about security posture, who answers — and against what standard?

These are decision questions. They are not answered by acquiring better tools or implementing more controls. They are answered by building organizational structures that make security decisions consistently, with clear ownership, against documented standards, by people with the authority to act.

Most security programs have not been built that way. They have been built around framework compliance, tool acquisition, and audit preparation — all of which matter, none of which substitute for a functioning decision system.

The coordination problem underneath the security problem

The failure most organizations experience is not a failure of technical capability. It is a coordination failure.

Security decisions cross functional lines. A vendor risk decision involves procurement, legal, security, and operations. An incident response decision involves IT, legal, communications, and executive leadership. An AI deployment decision involves technology, compliance, security, and the business unit sponsoring the initiative.

When those functions operate with their own priorities and no shared decision structure, the decision either doesn't get made or gets made differently depending on who is in the room. Accountability becomes difficult to locate because it was never explicitly assigned.

That is the coordination problem the ADM is designed to make visible — not as a philosophical argument but as a practical diagnostic: where in this organization are the decisions that need to be made not being made well, and what is that costing us?

Why this matters for organizations without dedicated security functions

Enterprise organizations have security functions, GRC teams, and governance structures that, however imperfect, create some organizational infrastructure for security decision-making. Organizations without dedicated security leadership typically do not.

In those environments, security decisions are made by whoever is available: an IT Director managing a dozen other priorities, a COO without security context, an outside auditor reviewing point-in-time evidence. The decisions get made, but not well — not consistently, not with clear ownership, not against a documented standard that survives personnel turnover.

The consequence is a security program that looks functional in documentation and is fragile in practice. It passes audits. It fails incidents. It satisfies assessors who review it annually and disappoints clients who ask about it during procurement.

ADM provides a structure for closing that gap — not by adding another framework layer, but by treating the underlying decision problem as the actual problem to be solved.

SECTION 02 · ADM APPLIED

vCISO Engagement

A vCISO engagement exists precisely because an organization has a security decision problem it cannot solve with its current internal resources.

The context in which the ADM's decision structure is most immediately visible is advisory security leadership.

A vCISO engagement exists precisely because an organization has a security decision problem it cannot solve with its current internal resources. The decisions that need to be made — about risk priorities, program direction, board reporting, vendor evaluation, incident posture — require someone with the authority, context, and capacity to make them well. That person doesn't exist inside the organization or exists but is operating without the structural support to be effective.

The ADM provides the operating framework that gives a vCISO engagement its direction. It defines what the advisory function is actually trying to produce: not a security program document, but a functioning security decision system.

What the engagement lifecycle looks like through the ADM

The four-stage decision lifecycle — Understand, Decide, Execute, Improve — maps directly to how a vCISO engagement should be structured and evaluated.

Understand. The first obligation of an advisory engagement is establishing context the client does not yet have in organized form. What risks are actually present in this environment? What business decisions depend on security inputs? What is leadership accountable for — to clients, to regulators, to the board — and how well is the current program positioned to support those accountabilities?

This is not a gap assessment in the traditional sense. It is a decision context inventory. The output is not a list of missing controls. It is a structured picture of the decisions that need to be made and what information each decision requires.

Decide. The decisions themselves — risk prioritization, resource allocation, program direction, risk acceptance — require someone with authority to make them and a clear record of how they were made. The vCISO's role at this stage is not to decide for the organization but to ensure that the decisions the organization needs to make are actually made, by the right people, against the right standard, with the accountability documented.

This is where most advisory engagements lose coherence. Recommendations are made. Reports are delivered. But the decision — who accepts this risk, who prioritizes this initiative, who commits this resource — is never formally taken. The recommendation sits in a document. The security posture doesn't change.

Execute. Decisions have to translate into operational practice. Controls have to be implemented. Vendors have to be managed. Policies have to be operationalized rather than published. The vCISO's execution role is ensuring that the decisions made at the leadership level survive the translation into IT, operations, and vendor management — and that what gets implemented matches what was decided.

Improve. The loop closes when outcomes are measured against the decisions that produced them and what is learned feeds back into context. A risk register that is never re-scoped against what the organization has learned is not improving — it is aging. A board reporting cadence that delivers the same metrics quarter after quarter without adjustment is not governance. The improvement discipline is what keeps the decision system current rather than historically accurate.

CASE STUDY

The practical application of this framework is visible in a recent Antares engagement with a PE-backed manufacturing company. The private equity firm required all portfolio companies to implement a formal security governance program and present quarterly metrics to the board. The target company had no security leadership and no reporting infrastructure.

Through the ADM lens, the engagement was structured as follows:

Understand. The context established at the outset was not primarily technical. It was organizational. What decisions was the board expected to make about security? What risks was the PE firm concerned about across the portfolio? What was the operational environment — what systems, what data, what regulatory exposure? The context inventory shaped every subsequent decision about program direction.

Decide. The NIST CSF alignment was a decision, not a default. It was selected because the framework maps well to board-level reporting and because the PE firm's oversight model was structured around organizational accountability rather than technical controls. The risk register was built around business impact prioritization — not because CVSS scores don't matter, but because the decisions the board needed to make were business decisions, not technical ones.

Execute. The board reporting package was built to function as a decision tool, not a status update. Each presentation contained a specific set of risk acceptance decisions the board was

being asked to make. The quarterly cadence was designed so that board members built literacy over time rather than being reoriented at every meeting.

Improve. The transition to an internal security hire was itself a governance decision — triggered not by a timeline but by program maturity indicators. When the risk register was established and maintained, when board reporting was consistent, when the organization had the infrastructure for a full-time hire to step into, the transition was made. The documentation produced for that transition was an output of the Improve stage, not an afterthought.

> *"The board went from asking basic questions about whether we had antivirus to having substantive conversations about risk tolerance. That shift happened because of how Antares framed the program."*
>
> CFO · PE-BACKED MANUFACTURING COMPANY

That shift is what the ADM is designed to produce. Not a security program. A functioning security decision system.

SECTION 03 · ADM APPLIED

AI Governance

The most concentrated recent illustration of the ADM's core argument.

AI governance has become one of the most discussed topics in enterprise security. It has also become one of the clearest demonstrations of what happens when organizations accumulate governance artifacts without building governance decision systems.

Most organizations deploying AI have policies. They have AI principles documents, acceptable use frameworks, and ethics statements. They have committees designated to oversee AI initiatives. They have compliance checklists oriented toward emerging regulatory requirements.

What most of them do not have is a clear answer to a simple question: when an AI system produces an outcome that causes harm, who is responsible?

That is not a technology question. It is a decision question. And the answer is determined not by the quality of the policy but by the quality of the decision structures that were built — or not built — around the system.

The governance failure pattern

The AI Governance Series developed by Antares over the first half of 2026 documented four structural failure modes that recur across organizations deploying AI. Each maps directly to a stage of the ADM decision lifecycle.

The Coordination Gap (Understand stage failure): Organizations deploy AI faster than they adapt the decision structures responsible for governing it. Cross-functional dependencies exist without shared frameworks. Decisions about risk, ownership, and oversight get made piecemeal — or not at all.

The Ownership Gap (Decide stage failure): AI risk is not one problem. It is four distinct problems — strategic, operational, security, and ethical — each with a different natural owner and a different decision cadence. Organizations that treat it as a single category assign accountability uniformly across functions. Accountability distributed uniformly disappears.

The Runtime Gap (Execute stage failure): Governance designed as a pre-deployment checkpoint cannot govern a post-deployment reality. AI systems change after deployment — prompts evolve, models are updated, integrations expand, new use cases emerge. The system governance approved at launch is not the system running six months later.

The Risk Management Gap (Improve stage failure): Traditional risk frameworks were built for systems that remain stable after approval. AI systems are not stable. Governance teams become bottlenecks when every use case receives the same review regardless of risk profile. The feedback loop that would keep governance current never closes.

The structural answer

The ADM's response to these failure modes is the same in every context: make the decision structure explicit. Define who decides, what they are deciding, with what authority, against what standard, and with what record.

For AI governance, this means establishing ownership at three levels — strategic (CAIO or equivalent), security (CISO), and compliance (CCO) — with a single point of final accountability at the CEO. It means building a governance model that scales with risk rather than applying uniform oversight to every AI initiative. And it means treating governance as a runtime function — something that continues operating after deployment, not something that concludes when approval is granted.

The organizations that will deploy AI effectively are not the ones with the most sophisticated models. They are the ones that can make accountable decisions about AI consistently, at scale, under the conditions that AI deployment actually creates.

SECTION 04 · ADM APPLIED

Incident Response

The context in which decision quality matters most and decision discipline is most frequently absent.

Most organizations have an incident response plan. A significant portion of those plans were written for a compliance requirement, reviewed at implementation, and not tested since. The plan exists. The decision system the plan is supposed to activate does not.

The consequence of that gap becomes visible in the first 48 hours of an actual incident — when the decisions that determine the severity of the fallout are being made simultaneously, under pressure, by people who were not fully prepared to make them and who are operating without a clear decision authority structure.

The incident decision lifecycle

The four-stage ADM lifecycle applies directly to incident response — not as a sequential process but as a structure for understanding where decision discipline breaks down and what the cost of that breakdown is.

Understand. The first obligation in an active incident is establishing what is actually happening. What systems are affected? What data was accessed? When did the attacker enter the environment? What is the scope of the exposure?

This is the stage at which most organizations experience the first significant decision failure. The forensic investigation required to answer these questions takes weeks. Regulators, clients, and insurers are asking for answers in hours. Organizations that have not established a clear process for managing the gap between "we don't yet know" and "we are legally required to notify" find themselves making disclosure decisions with incomplete information and no documented framework for doing so.

Decide. The decisions that define an incident response are consequential and time-pressured simultaneously. Whether to pay a ransom. When to notify regulators. What to tell clients. How to communicate publicly. Whether to bring in outside legal counsel. Which systems to take offline and which business functions that will impact.

These decisions require authority. A COO who has never thought through the ransomware decision framework before is not well positioned to make that call at 2 AM during an active incident. A

communications lead who is receiving conflicting legal guidance from two different advisors cannot draft a consistent client notification.

The ADM's contribution at the Decide stage is the work done before the incident: establishing who makes each category of decision, what information each decision requires, and what escalation paths exist when the primary decision-maker is unavailable.

Execute. Containment, evidence preservation, regulatory notification, client communication, legal engagement, and continued business operations run simultaneously in the execution phase of an incident. Each involves different functions. Each produces outputs that other functions depend on.

The organizations that contain incident damage effectively are the ones that execute these parallel tracks with minimal coordination friction — because the coordination structure was established before the incident, not improvised during it. The functions know their roles. The handoffs are documented. The escalation paths are understood.

Improve. The post-incident review — the step that closes the ADM loop — is the discipline most commonly skipped in incident response.

Organizations that manage incidents without conducting structured post-incident reviews do not improve their decision systems. They may update a technical control, patch a vulnerability, or add a monitoring capability. But the decision failures that determined the severity of the incident — the unclear ownership, the undefined authority, the undocumented notification framework — persist into the next event.

The Improve discipline requires asking not just what technical failure enabled the breach but what decision failures determined how much damage it caused. Those are different questions with different answers and different remedies.

The cascade the ADM is designed to prevent

The fallout of a breach unfolds as a cascade: regulatory, financial, operational, and relational. Each stage of the cascade is driven by decisions — many of which, in organizations without ADM-aligned structures, are being made for the first time under maximum pressure.

The regulatory clock starts at discovery, not containment. Organizations that have not mapped their notification obligations in advance are reconstructing those obligations while simultaneously managing an active incident.

The forensic investigation is expensive, slow, and outside the organization's control. Organizations that have not established what forensic engagement looks like, who authorizes it, and how it

integrates with legal response are adding coordination overhead at exactly the point where capacity is most constrained.

The cyber insurance claim gets complicated when the controls attested to at application cannot be verified in the investigation. Organizations that have not documented their control posture and kept that documentation current are discovering the gap during a claims process rather than before one.

The clients and vendors asking questions deserve consistent, accurate answers. Organizations that have not established a client communication protocol are improvising responses under legal scrutiny — and producing the inconsistency that damages relationships more than the breach itself.

None of this requires prediction. All of it can be prepared for. The ADM's contribution is making the preparation a decision-system discipline rather than an annual plan review.

SECTION 05

The Antares Decision Model — Framework Reference

The terminology, the lifecycle, and the domain structure that constitute the model.

The preceding sections have described the ADM through its application in three operational contexts. This section provides the structural reference — the terminology, the lifecycle, and the domain structure that constitute the model.

The core proposition

The ADM is founded on a single observation: security outcomes are a property of decisions.

A control that was decided well and documented but assigned without a clear owner will degrade. A vendor evaluation that was thorough but lacked the authority to commit will stall. An incident response plan that exists in policy but has never been exercised will fail under the conditions it was written for.

This is not an argument against controls, policies, or frameworks. It is an argument about what determines whether those investments produce the outcomes they were designed to produce. Controls implemented without decision discipline do not hold. Frameworks adopted without ownership structures do not function. Policies published without enforcement authority do not change behavior.

> *The unit of analysis the ADM proposes is the decision — not the framework, the control, or the compliance requirement.*

The question ADM asks of every security program is the same: are the decisions that need to be made actually getting made, by the right people, with the right authority, against the right standard, and with a record that survives the personnel who made them?

The decision lifecycle

Every security decision — at the program level, at the engagement level, and within individual governance choices — is an instance of a four-stage lifecycle.

Stage 1: Understand. Establish the business, technology, and risk context a decision has to be made within. Surface what leadership is accountable for, what the operational environment actually looks

like, and what assumptions are baked into the current posture. The Understand stage produces not a gap list but a decision context: the information base that makes the subsequent decision possible.

Stage 2: Decide. Evaluate priorities, tradeoffs, and ownership. This is the unit of work in the ADM — who decides, against what standard, with what authority, and with what record so the decision survives turnover and scrutiny. The discipline at this stage is ensuring the decision is actually taken, not deferred, not distributed uniformly across functions, but owned by a specific person with the authority to own it.

Stage 3: Execute. Translate the decision into operational practice. Execution lives across engineering, vendors, operations, and compliance — and the decision has to remain coherent as it moves through each of them. The Execute stage fails when translation introduces drift: when what is implemented differs from what was decided, or when the decision's intent is lost as it moves from leadership to operations.

Stage 4: Improve. Measure outcomes against what the decision was supposed to produce. Feed what is learned back into context so the next decision starts from a more accurate picture than the last one did. The Improve stage is what keeps the decision lifecycle functional rather than historical. It is the step most often missing — and its absence is what allows the same security failures to recur across consecutive years.

The lifecycle is not a checklist that completes once. It is a loop. A risk register established but never re-scoped is governance that stopped at Understand. A compliance program that ships controls without measuring what they produce is a cycle that stopped at Execute. The loop closes at Improve — or it doesn't close at all.

The six decision domains

The ADM applies across six domains where security decisions are most critical.

Leadership. Where executive security decisions are made — strategy direction, risk posture, board accountability, and the cadence leadership uses to govern the program. The decision discipline in this domain determines whether security governance functions at the organizational level or exists only on paper.

Risk. How exposure is identified, prioritized, and owned. The structures that translate business consequence into specific decisions leadership can act on. Risk management without decision ownership produces risk documentation. Risk management with decision ownership produces risk posture.

Governance. Decision authority, escalation paths, and the standing cadence that keeps governance functioning between audits, incidents, and changing conditions. This is the domain where the coordination problem most often manifests — where multiple functions each have a partial claim to governance responsibility and no single function owns the decision.

Compliance. How framework obligations are translated into operating controls and evidence — built as a continuous system, not assembled at audit time. Compliance without decision discipline produces point-in-time evidence of past control status. Compliance with decision discipline produces a living system that remains current between audits.

AI. Decision-making where AI systems influence, recommend, automate, or execute actions — accountability, oversight, validation of outputs, and ownership of risk when AI is in the loop. This domain is the fastest-evolving in the ADM and the one where decision structure is most frequently absent.

Operations and Resilience. How the program continues to make defensible decisions through changing conditions, resource constraints, and disruptive events. This is the domain where incident response, business continuity, and vendor resilience decisions live — and where the consequences of decision failures are most immediately visible.

SECTION 06

Failure Mode Analysis

Not theoretical — the recurring patterns Antares observes across security programs.

The ADM predicts specific failure modes at each stage of the decision lifecycle and across the three applied contexts. These failure modes are not theoretical. They are the recurring patterns Antares observes across security programs.

Failure modes at the Understand stage

Context without synthesis. Organizations conduct assessments, vulnerability scans, and risk reviews that produce data without producing decision context. Leadership cannot answer what the organization's most significant risks are, what they are being asked to decide, or what information a specific decision requires. The Understand stage produces artifacts rather than the context that enables decisions.

Assumption preservation. The context established at the beginning of an engagement or program cycle is not updated as conditions change. Risk registers reflect the environment as it was two years ago. Threat assessments assume threat actor behaviors that have evolved. The Understand discipline produces a historical snapshot rather than a current picture.

Failure modes at the Decide stage

Decision diffusion. Accountability is shared uniformly across functions, which is equivalent to assigning it to none of them. The decisions that require a specific person to absorb the risk of being wrong become the decisions nobody makes. This is the most common failure mode Antares observes and the one with the most consistent downstream consequences.

Authority without accountability. A function is designated as responsible for a domain — security, compliance, AI governance — without being given the authority to enforce its decisions. The CISO can recommend. The CCO can flag. The CAIO can assess. None of them can stop a deployment, reject a vendor, or require remediation. Responsibility without authority produces liability rather than governance.

Deferred decisions. Decisions that need to be made before an event are instead made during it. The ransomware payment framework that should have been established in the IR planning process gets improvised at 2 AM during an active incident. The AI deployment governance that should have

been defined before the model went to production gets reconstructed during a regulatory inquiry. The cost of deferred decisions is always higher than the cost of making them in advance.

Failure modes at the Execute stage

Translation drift. The decision made at the leadership level does not survive translation into operational practice. What is implemented differs from what was decided — either because the decision was not specific enough to execute against or because the functions responsible for execution interpreted it differently. The gap between what was decided and what was built is often not discovered until audit, incident, or client review.

Scope expansion without governance. Execution extends beyond the scope of the original decision without triggering a new governance cycle. An AI system approved for one use case gets extended to another. A vendor relationship approved under one set of terms expands in ways that change the risk profile. The Execute stage continues without the Understand or Decide stages resetting.

Failure modes at the Improve stage

Closed loop absence. The most common Improve failure is simply that the stage doesn't exist. Decisions are made, controls are implemented, and the program moves forward without measuring whether the decisions produced the outcomes they were intended to produce. The feedback mechanism that would keep the decision system current is never built.

Post-incident learning failure. Incidents are resolved without structured post-incident review. The technical cause is addressed. The decision failures that determined severity — unclear ownership, undefined authority, undocumented notification framework — persist. The next incident encounters the same decision gaps.

SECTION 07

Implications for Organizations Building Security Programs

The argument this brief has made across six sections can be stated simply.

Most security programs have the artifacts of security governance. They do not have functioning security decision systems. The gap between those two things is where programs fail — not catastrophically, not in ways that are immediately visible, but consistently, in the recurring patterns that make security feel expensive without feeling effective.

The ADM is not a new framework to implement alongside the existing frameworks. It is a diagnostic lens that asks a single question of every governance structure, every compliance program, every risk management process: are the decisions this system is supposed to produce actually getting made well?

What building decision discipline looks like

The organizations that have built functioning security decision systems share a set of observable characteristics.

Ownership is explicit. Every consequential security decision has a named owner — not a function, a committee, or a shared responsibility, but a specific person with the authority to make the call and accountability for the outcome. That ownership is documented and survives personnel changes.

Authority matches accountability. The people designated as responsible for security decisions have the authority to act on them. The CISO who is accountable for security posture can stop a deployment, reject a vendor, or require remediation. The authority is not theoretical — it has been exercised.

Decision records exist. The decisions made — about risk acceptance, vendor relationships, program priorities, governance structures — are documented in a form that outlasts the people who made them. A new hire, a new board member, or a new auditor can reconstruct the decision logic from the record.

The loop closes. The Improve discipline exists and is practiced. Risk registers are re-scoped against what the organization has learned. Board reporting evolves as the program matures. Post-incident reviews produce decision system changes, not just technical remediations.

The starting point

For organizations that recognize the decision discipline gap, the starting point is not another framework assessment or another policy review. It is a decision inventory: an honest accounting of the decisions that shape the organization's security posture and an honest evaluation of how well each of those decisions is currently being made.

Which decisions are being made by default rather than by design? Which are being deferred? Which have unclear ownership? Which lack the authority structure required to act on them?

Those questions — not "what controls are we missing?" but "what decisions are we not making well?" — are the ones the ADM is designed to answer.

The organizations that ask them, and build decision discipline around the answers, will consistently outperform those that don't. Not because they have better tools or more frameworks. Because they have learned to treat security as what it actually is: a decision problem operating at the speed of a changing environment, with real consequences for the quality of every call made along the way.

About Antares Security

Antares Security is an advisory firm helping organizations improve cybersecurity decision-making through executive advisory, governance, risk management, compliance, and incident response.

The Antares Decision Model serves as the firm's operating framework for building security programs that remain effective under real organizational pressure.

EMAIL
info@antaresecurity.com

WEB
antaresecurity.com

PHONE
(312) 725-0296

This paper is intended as the first in an ongoing body of research exploring cybersecurity as an organizational decision system.