StrongestLayer

Three Axes of Email Security

A Framework for Evaluating Detection Architectures in the AI Era

StrongestLayer Research • December 2025

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Email security has progressed through three architectural generations, each with distinct detection methodologies. The rise of AI-assisted attacks has exposed limitations in earlier approaches more clearly—though these constraints existed from the beginning. They simply mattered less when attacks were more predictable.

This whitepaper proposes a framework for evaluating email security platforms across three essential axes:

- **Completeness**: Can the system detect attacks it has never seen before?
- **Accuracy**: Can the system minimize false positives without creating blind spots?
- **Rapid Response**: When the system is wrong, how quickly can it be corrected—and by whom?

Each generation of email security—rules-based pattern matching, machine learning, and LLM-native reasoning—makes different trade-offs across these axes. Understanding these trade-offs is essential for security leaders evaluating their defensive architecture against AI-enhanced threats.

THE THREE GENERATIONS OF EMAIL SECURITY

Generation 1 / 1.5: Rules-Based Pattern Matching

The first generation of email security relies on human-authored rules to identify threats. Security analysts observe attack patterns, codify them into detection signatures, and deploy rules that match against known indicators. Generation 1.5 platforms extend this model with more sophisticated rule languages, AI-authored rules and customer-authored detection logic.

Core assumption: Attacks can be described in advance. If you know what "bad" looks like, you can write a rule to catch it.

Representative approach: Secure Email Gateways and custom detection rule platforms

Generation 2: Machine Learning

The second generation applies statistical learning to email classification. ML models train on historical attack data to identify patterns that correlate with malicious intent. Rather than explicit rules, the system learns probabilistic relationships between features and outcomes.

Core assumption: Future attacks will resemble past attacks. Statistical patterns in training data generalize to new threats.

Representative approach: Behavioral analytics and ML-based anomaly detection platforms.

THE THREE GENERATIONS OF EMAIL SECURITY

Generation 3: LLM-Native Reasoning

The third generation uses large language models as the central coordinator of detection, not as a feature bolted onto existing architectures. Rather than matching patterns or statistical correlations, these systems reason about intent and other dimensions.

Core assumption: Attacks are defined by what they're trying to accomplish, not how they're constructed. Intent and business context are stable signals that persist regardless of attack novelty.

Representative approach: LLM-as-master, Agentic reasoning and other emerging LLM/agentic first platforms.

A Note on Classification

Most email security platforms incorporate elements from multiple generations. Traditional SEG vendors have added ML-based detection alongside their rules engines. Behavioral analytics platforms combine baseline monitoring with machine learning. Few vendors fit cleanly into a single category.
The relevant question isn't which generation a vendor claims—it's which architecture serves as the primary decision-making foundation when the system encounters a threat it has never seen before. Under pressure, does the system fall back on rules, statistical correlation, or semantic reasoning? That foundational architecture determines the system's ceiling on each axis.

THE FUNDAMENTAL DIVIDE: TECHNIQUE VS. INTENT

The most important distinction between generations is not the technology they use—it's what they're looking for.

Generations 1 and 2 are technique-dependent. They must recognize HOW an attack is constructed to detect it. A rules-based system needs a signature that matches the attack's technical indicators. An ML system needs training data that includes similar attack patterns. When attackers use novel techniques, these systems are structurally blind.

Generation 3 is intent-dependent. It recognizes WHAT an attack is trying to accomplish, regardless of construction. Social engineering still relies on urgency, authority, and fear. Credential theft still requires capturing authentication data. Business email compromise still exploits trust relationships. These intent patterns persist even when attack methods are completely novel.

When AI enables attackers to generate unlimited novel techniques, technique-dependent detection becomes mathematically obsolete. Intent-dependent detection becomes the only viable architecture.

AXIS 1: COMPLETENESS OF DETECTION

Completeness measures whether a system can detect attacks it has never encountered before. In the AI era, this is the defining question—attackers can now generate novel, polymorphic attacks at scale

.

Generation 1/1.5: Low Completeness

Rules-based systems can only detect what they've been programmed to find. Every novel attack requires a new rule. This creates a structural race condition: attackers can generate new techniques faster than defenders can write signatures.

StrongestLayer research on 2,500+ email attacks found that AI-generated threats show only 5-15% Jaccard similarity to historical patterns, compared to 85-95% for traditional template phishing. This means rules written for yesterday's attacks match only fragments of today's threats.

Generation 2: Medium Completeness

ML systems can generalize beyond their training data, but only within the statistical distribution they've learned. When AI generates attacks that fall outside this distribution—novel impersonation patterns, new urgency tactics, unfamiliar business contexts—statistical models lose predictive power.

The fundamental limitation: ML systems learn correlations, not causation. They can identify that certain patterns historically correlated with attacks, but they cannot reason about why something is malicious.

AXIS 1: COMPLETENESS OF DETECTION

Generation 3: High Completeness

LLM-native systems reason about the semantic content of messages—what they're trying to accomplish, whether they match legitimate business patterns, whether the claimed identity is plausible. These signals persist regardless of how the attack is constructed.

When StrongestLayer's TRACE system detected a Microsoft 365 Direct Send exploitation attack that bypassed both Microsoft's native security and leading secure email gateways, it did so not by matching a signature, but by recognizing that the message's claimed purpose didn't match its technical behavior—a reasoning-based detection that required no prior knowledge of the specific technique.

AXIS 2: ACCURACY OF DETECTION

Accuracy measures the system's ability to minimize false positives without creating false negatives—the classic precision/recall trade-off. But the deeper question is architectural: how does the system resolve this tension?

The Prosecutor-Only Problem

Both Generation 1 and Generation 2 systems suffer from what we call "prosecutor-only architecture." They can only hunt for evidence of guilt—suspicious indicators, malicious patterns, anomalous behaviors. They have no mechanism to prove innocence.

This creates an unsolvable tension: make the prosecutor more aggressive, and you convict more innocent emails (false positives). Make it more cautious, and you let more threats escape (false negatives). This trade-off can never be solved within a prosecutor-only architecture.

Generation 1/1.5: Rules Create Blind Spots

When rules generate false positives, organizations create allowlist entries and bypass rules to restore business flow. These exceptions become permanent blind spots—meta-vulnerabilities that attackers can exploit.

Industry research shows that 40-50% of firewall rules become "zombie rules"—obsolete but never removed because removal is risky. The same dynamic applies to email security: every false positive fix creates potential attack surface.

AXIS 2: ACCURACY OF DETECTION

Generation 2: Black Box Trade-offs

ML systems make precision/recall trade-offs inside opaque models. Customers have limited visibility into why the system makes specific decisions, and limited ability to influence those trade-offs. When the model is wrong, the only recourse is to report the error and wait for "dynamic learning" to occur—a process with unclear timeline and uncertain outcome.

This architecture eliminates customer control entirely. When the model is wrong, organizations are left waiting for vendor intervention with no fallback mechanism.

Generation 3: Dual Evidence Reasoning Architecture

LLM-native architecture breaks the prosecutor-only paradigm by simultaneously collecting two types of evidence:

- Prosecutor evidence: Threat signals, authentication failures, suspicious infrastructure, urgency manipulation
- Defender evidence: Business legitimacy patterns, established relationships, documented workflows, communication norms

An impartial LLM judge weighs both evidence streams. Strong legitimacy indicators can outweigh minor threat signals. This resolves the FP/FN tension by giving every email "its day in court" rather than forcing a binary prosecutor verdict.

AXIS 2: ACCURACY OF DETECTION

Real-Time Signal Enrichment

When the dual evidence system identifies gaps in its reasoning chain, Generation 3 architecture fetches missing signals in real-time:

- Stale domain data → Fresh WHOIS/DNS lookup → Domain registered yesterday = elevated risk
- Deferred payload → Re-scan URL at decision time → Payload activated post-delivery = threat identified
- Unverified vendor relationship → Query organizational email history → Zero prior communication = suspicious context

This "what signals are needed for certainty?" approach means the system can reason about its own uncertainty and take action to resolve it—a capability that neither rules nor ML models possess.

AXIS 3: RAPID RESPONSE TO ERRORS

Every detection system makes mistakes. The critical question is: when the system is wrong, how quickly can it be corrected—and who bears the burden of correction?

Generation 1/1.5: Fast but Customer-Owned

Rules-based systems offer immediate response capability: customers can add, modify, or remove rules in minutes. But this speed comes at a cost—the customer owns the correction, and that correction becomes permanent technical debt. Over time, rule libraries accumulate exceptions, allowlists, and workarounds. Each fix addresses an immediate problem while creating long-term vulnerability. The organization becomes responsible for maintaining an ever-growing set of detection logic—logic that requires expert knowledge to audit and prune.

Generation 2: Slow and Vendor-Owned

ML systems remove the customer burden by centralizing model updates at the vendor. But this creates a different problem: when the model is wrong, customers have no agency. They can report errors, but they cannot fix them. Organizations evaluating no-rules platforms have reported that end users frequently reach out to analysts about missing messages, requiring manual release of affected emails. The timeline for model correction remains opaque —"dynamic learning" occurs, but when and how is unclear to the customer.

AXIS 3: RAPID RESPONSE TO ERRORS

Generation 3: Fast and System-Owned

LLM-native architecture introduces a third model: adversarial self-correction with global benefit.

When a false positive is flagged—through user submission, admin review, or system self-doubt at detection time—the platform spins up an adversarial analysis: "Why might this be legitimate?" The system reviews its own reasoning chain, identifies logic gaps or missing signals, and updates its global TTP (tactics, techniques, and procedures) database.

This correction happens in approximately five minutes. More importantly, the correction is global—one customer's false positive report improves detection for all customers. No rules are created. No customer-owned logic accumulates. The system itself becomes more intelligent.

Optional Controls with Architectural Safeguards

Security leaders have been told to "trust the model" before—and been burned when the model was wrong for weeks with no recourse. That experience creates justified skepticism about any platform that eliminates customer control.

Some Generation 3 platforms like StrongestLayer address this directly by providing optional rule-like controls for customers who need an escape valve. But these controls include a critical architectural safeguard: time-to-live (TTL) limits ranging from 7 to 90 days. The system gives you control when you need it—but that control expires rather than accumulates.

AXIS 3: RAPID RESPONSE TO ERRORS

	Gen 1/1.5	Gen 2	Gen 3
Customer Rules	Permanent by default	None (no agency)	TTL-limited (7-90 days)
Zombie Rules	Accumulate forever	N/A	Impossible by design
Default Behavior	Persist unless removed	N/A	Expire unless renewed
Meta-Vulnerability	Grows over time	Unknown	Architecturally eliminated

This flips the default: in Generation 1, rules persist unless actively removed (risky, so they accumulate). In Generation 3, controls expire unless actively renewed (safe, so they fade away as the system proves itself).

SUMMARY: THREE AXES ACROSS THREE GENERATIONS

Axis	Gen 1/1.5 (Rules)	Gen 2 (ML)	Gen 3 (LLM-Native)
Completeness	**Low** Must know technique to write signature	**Medium** Must have seen similar patterns in training	**High** Reasons about intent regardless of technique
Accuracy	**FP storms + blind spots** Allowlists create permanent vulnerabilities	**Black box trade-offs** No visibility into why; no customer control	**Dual evidence resolution** Prosecutor + defender; real-time enrichment
Rapid Response	**Fast, customer-owned** Creates technical debt	**Slow, vendor-owned** Report and wait; no agency	**Fast, system-owned** ~5 min; global benefit; no debt

IMPLICATIONS FOR SECURITY LEADERS

StrongestLayer's AI Advisor embodies these principles in a production-ready platform:

Evaluating Rules-Heavy Platforms
Platforms that emphasize customer-authored rules and detection logic offer rapid response capability, but at significant cost. The detection completeness ceiling is structurally low—you can only catch what you've anticipated. And every rule you write becomes maintenance burden that grows over time.

Key question: How many rules exist in your current system? What percentage have been reviewed in the past year? What percentage could be safely removed?

Evaluating No-Rules Platforms
Platforms that eliminate customer controls entirely solve the technical debt problem, but create a different risk: when the model is wrong, you have no escape hatch. You're entirely dependent on vendor response time and vendor priorities.

Key question: When your platform makes a false positive that blocks critical business email, what is the documented SLA for correction? What can you do in the meantime?

IMPLICATIONS FOR SECURITY LEADERS

Evaluating LLM-Native Platforms

True Generation 3 platforms should demonstrate reasoning about intent (not just pattern matching with AI acceleration), dual evidence architecture (not just AI-enhanced prosecution), and self-correction mechanisms that improve the system without creating customer-owned technical debt.

Key questions: How does the platform handle messages where threat indicators conflict with legitimacy indicators? When a false positive is reported, what happens—and how quickly? Does the correction benefit only your organization, or all customers?

QUESTIONS TO ASK ANY VENDOR

The Three Axes framework provides a structured way to evaluate any email security platform. These questions—applicable to any vendor, including StrongestLayer—will reveal the architectural foundations that determine real-world performance.

Completeness Questions

1. Show me five attacks from the last 90 days that had zero signature matches in threat intelligence databases. How did your system detect them?
2. When an attacker uses a technique you've never seen before, what signals does your system rely on to make a detection decision?
3. What percentage of your detections last quarter were based on novel threat indicators versus known signatures or patterns?

Accuracy Questions

1. Walk me through a recent false positive. What was the reasoning chain? How was it corrected? How long did correction take?
2. When your system sees an email with both suspicious and legitimate indicators, how does it weigh the competing evidence?
3. Can an analyst see exactly why the system made each decision? Show me the explanation for a blocked message.

Rapid Response Questions

1. If I report a false positive at 2 PM on a Friday, what is my realistic expectation for resolution? What can I do in the meantime?
2. When you correct an error for one customer, does that correction benefit all customers automatically?
3. If I need to create a temporary exception, what prevents that exception from becoming permanent technical debt?

CONCLUSION: THE ARCHITECTURE QUESTION

Email security is facing an architectural inflection point. The question is no longer "which vendor has the best rules?" or "which vendor has the most training data?" The question is: which architecture can reason about threats it has never seen, resolve accuracy trade-offs without human intervention, and correct its mistakes without accumulating technical debt?

The Three Axes framework—Completeness, Accuracy, and Rapid Response—provides a structured way to evaluate these trade-offs. Security leaders should use this framework to move beyond feature comparisons and understand the fundamental capabilities and limitations of each architectural approach.

In an era where AI enables attackers to generate unlimited novel threats, technique-dependent detection is no longer viable. Intent-dependent, reasoning-based architecture isn't just an improvement—it's the only path to sustainable defense.