

FALSE-POSITIVE SCREENING, REARCHITECTED

Let the Cheap Model Look

Nullify replaced a one-shot frontier-model false-positive screen with a small open-weight agent that can read the repository. It catches more false positives, and it costs a third as much per call.

99.1%
of eventual false positives caught before the full classifier ran

⅓
the per-call cost of the one-shot frontier screen, at $0.0103

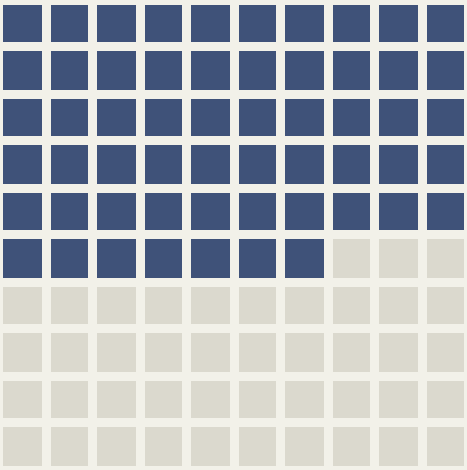
5.6×
lower miss rate than the one-shot screen on control tenants

0
real vulnerabilities dismissed across 46 hand-audited dismissals

Based on “Let the Cheap Model Look” by J. Lam, Nullify · v0.1.0 · 18 August 2026 · First nine days in production across six tenants

Most Scanner Findings Are Noise

Application-security triage is as much an economics problem as an accuracy problem. Scanners are cheap to run and prolific in what they report, so investigating every finding deeply is an expensive way to confirm that most of them do not matter.



57%

of **14,847** SAST findings were ultimately labelled false positive.

Five representative production tenants, 90-day window. Each square is 1% of findings.

$0.40–$1.60

to give one finding the full triage treatment at frontier-model rates

10,000s

of findings per quarter, per tenant, at production scale

The Usual Answer Is a Staged Pipeline

STAGE 1

Scanner findings

Cheap, prolific, and mostly noise.

→

STAGE 2 · THIS PAPER

False-positive screen

Removes obvious false positives before deeper analysis begins.

→

STAGE 3

Full classifier

Context gathering, exploitability analysis, severity drafting and prioritisation.

The harder design questions are what the screen should be built from, and how much authority it should have.

Where $6,356 of Triage Spend Went

Five tenants, 90 days



83% · Legitimate deep analysis of real findings. Intentionally left unchanged.

8.4% · Avoidable deferral: false positives that paid for a full classifier run. This is the slice the work targets.

8.6% · Remaining spend.

Without a screen, most triage spend would go toward carefully confirming that noise is, in fact, noise.

Feb 2025

when Nullify first put a false-positive screen in front of its full classifier

A Sensible Design for 2025 Outlived Its Constraints

The original screen made one call to the best available frontier model, with a forty-line window around the finding and no tools. Every one of those choices was reasonable when it was made.

LINEAGE OF THE FALSE-POSITIVE SCREEN

Feb 2025
One-shot classifier ships on Claude 3.5 Sonnet v2 via Bedrock.

Jul 2025
Full triage classifiers become multi-step. The screen stays one-shot.

Jul 2026
90-day production study. Agent screen built and piloted in dev.

Aug 2026
Agent screen goes live on production tenants.

What Changed in Eighteen Months

CONSTRAINT	EARLY 2025	MID 2026
Reliable judgment	Frontier models were the only serious option for a security decision with asymmetric failure costs.	For a bounded question answered by concrete evidence, access to the evidence matters more than the remaining capability gap.
Open-weight models	Trailed badly on instruction adherence, calibrated refusal and structured output, and were hard to justify in design review.	Labs such as Qwen and DeepSeek are credible sources of production-quality models.
Tool use	Multi-turn tool loops were dependable on frontier models and flaky everywhere else.	Open-weight models run reliable multi-turn tool loops.
Token prices	At $3 / $15 per million tokens, keeping the screen cheap meant keeping its context small: a ±20-line window.	Pay-as-you-go hosted inference at roughly one-tenth of frontier per-token prices.

THE CLAIM

For this task, the ability to look is worth more than the ability to think.

The claim is not that the cheaper model is intrinsically better. It is that repository access can now be bought at open-weight prices.

STILL RUNNING IN MID 2026

- **One call** per finding
- **Forty lines** of context
- **No tools** and no follow-up turn

A cost-control decision that hardened into an architectural one.

The One-Shot Screen Works Until the Evidence Lives Elsewhere

Across 14,847 triaged findings, the old screen proved genuinely useful. Its limit is architectural: when the fact that clears a finding sits outside the forty-line window, the screen has to defer.

~88%
of eventual false positives caught, at roughly $0.05 each

~$5,600
of downstream classifier work avoided over 90 days

12%
of false positives it structurally could not resolve: **1,025** findings

One-Shot Catch Rate by Tenant

Share of eventual false positives dismissed before the classifier ran · five tenants, 90 days



Company C keeps the relevant evidence outside the flagged file more often, so 37% of its false positives needed context the screen could not see.

What a Deferral Costs

CAUGHT BY THE SCREEN
~$0.05

DEFERRED, THEN RULED FALSE POSITIVE
$0.38–$0.92

Same conclusion, a full classifier run later. Across five tenants that came to roughly **$536 per 90 days**, almost all of it INFO-severity noise.

Why a Stronger Model Does Not Help

- A hash function looks like a broken security control until its **caller** shows it only computes a cache key.
- A raw SQL fragment looks injectable until its operand is traced to a **hardcoded alias**.

A stronger model inside the same window still lacks the decisive fact. The missing piece is access, not intelligence.

A single grep outside the window can provide the fact that a stronger model inside it still lacks.

30–100×
more spent by the full classifier to reach the conclusion the screen could not

Give a Cheaper Model Access to the Repository

The replacement keeps the pipeline shape and changes only the screen. It starts from exactly the same seed context, then investigates with three read-only tools under a real turn and dollar budget.

BEFORE · 2025–2026

One-shot screen

- Claude Sonnet, **one call**
- ±20-line window around the finding
- No tools

Confident false positive → dismissed at ~$0.05

Unsure (12% of false positives) → full classifier at $0.40–$1.60

NOW · BOUNDED INVESTIGATION

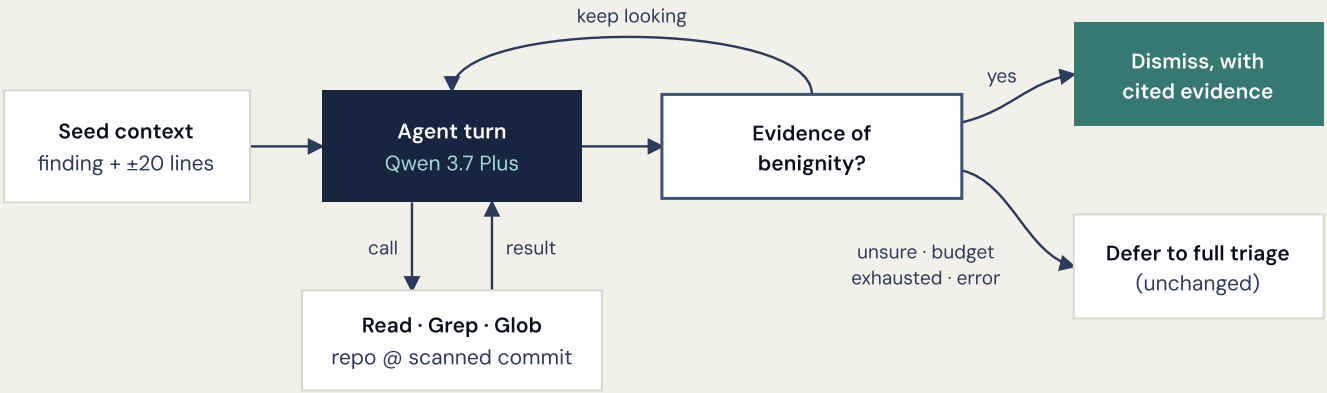
Agent screen

- Qwen 3.7 Plus, **multi-turn**
- Read, Grep and Glob over the repo at the scanned commit
- Turn and dollar budget

Evidence of benignity, cited → dismissed at ~$0.01

No evidence found → full classifier, unchanged

The Bounded Investigation Loop



MODEL

Qwen 3.7 Plus on Fireworks

$0.40 in, **$1.60** out and **$0.08** cached per million tokens. The design depends on the price class, not on this specific model.

STOPPING

3.9 turns on average

The agent is asked whether one specific security claim survives inspection, which gives the investigation a natural stopping point.

SERVING

Pay-as-you-go inference

$180–330 per month for the measured workload. A dedicated GPU at $1.6–2k per month would sit idle more than 90% of the time.

Everything downstream of the screen is untouched, so real findings still receive the full analysis.

3

read-only tools, and no ability to change the repository it inspects

Only One Outcome Can Hide a Real Vulnerability

The agent's choice is dismiss or defer, with a deliberately high bar for dismiss. A wrong deferral costs a few cents of downstream analysis. A wrong dismissal is the one failure mode that matters.

	SCREEN DISMISSES	SCREEN DEFERS
FINDING IS A FALSE POSITIVE	BENEFICIAL **Saves downstream work** The classifier never has to run.	WASTEFUL **Harmless** Loses some cost savings and nothing else.
FINDING IS A REAL VULNERABILITY	DANGEROUS **The one failure that matters** Possible only by dismissing on evidence that turns out to be wrong.	BENEFICIAL **Caught downstream** The normal pipeline handles it as before.

01

The verdict is asymmetric

The agent may dismiss only when it can point to **affirmative, cited evidence** that a finding is benign, such as a value traced to a source no attacker controls.

02

The task is closed

The agent is not asked to understand the repository in general. It is asked whether **one specific security claim** survives inspection of the relevant code.

Every Operational Failure Becomes a Deferral

Uncertainty · Timeout · Exhausted budget · Tool error · Provider outage · Model regression · Prompt regression → **Defer** → Original pipeline, unchanged

DEFERRAL IS THE IDENTITY

The screen may fail to save money, but it does not introduce a new way to lose recall.

A deferred finding enters the downstream pipeline with exactly the inputs it would have received had the screen not existed. All added recall risk sits in dismissals, and every dismissal carries citations that can be audited against the code.

OBSERVED IN PRODUCTION

14 of 696

invocations (2.0%) returned no verdict through timeouts and loop ceilings. **All failed closed** to full triage, and no run crashed.

One Hard Finding Through Both Screens

Before touching tenant traffic, both designs ran end to end on a matched pair of findings: a security-sounding function name (CWE-327, weak hash) whose benign purpose can be established only by inspecting its caller.

One-shot screen Agent screen

INPUT TOKENS

One-shot	187,862
Agent	4,718

97.5% fewer input tokens

WALL-CLOCK TIME

One-shot	592.9 s
Agent	18.4 s

Roughly **32× faster**

COST AT FRONTIER RATES

One-shot	$0.2122
Agent	$0.0218

$0.0026 for the agent at open-weight rates

SCREEN OUTCOME

One-shot	Agent
Deferred	False positive

The decisive evidence lay outside the fixed window.

THE AGENT'S RECORDED INVESTIGATION · 3 TURNS · 4 TOOL CALLS

```
1 Read(app/authcode.py)  // open the flagged file
2 Grep(make_authcode)  // find who calls it
3 Read(app/support_ticket.py)  // read the caller
4 submit_false_positive_investigation()
```

“used solely as a content checksum / deduplication key … no secret is involved, nothing verifies it, it grants no access”

Two Cautions From the Pilot

THE OLD SCREEN WAS STRONGER THAN EXPECTED

2 of 3

attempts to construct a finding that would force a deferral were still resolved from the local window, because file and symbol names often reveal intent.

EXPLORATION COSTS MORE ON EASY FINDINGS

1.6×

the cost of the one-shot screen, like for like, on cases already resolvable from the fixed window. The design pays only if extra dismissals on harder findings outweigh that.

More Turns per Finding at a Third of the Cost

The agent screen went live on five customer tenants on 6 August 2026 and on Nullify's own dogfood tenant on 10 August. These are its first nine days.

682
findings screened

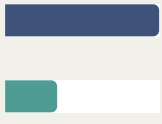
445
dismissed, or **65%** of screened

$0.0103
mean billed cost per call

3.9
mean turns per investigation

Billed Cost per Call

Same window · one-shot row is tenants still on the legacy screen



The agent takes **2.5× the turns** and still costs a third as much per finding.

TENANT	SCREENED	DISMISSED	RATE	$/CALL
company-c	263	194	74%	$0.0108
company-e	185	137	74%	$0.0098
company-h	149	91	61%	$0.0117
internal-test-org	52	5	10%	$0.0076
company-f	21	6	29%	$0.0069
company-g	12	12	n/a	$0.0098
Total	**682**	**445**	**65%**	**$0.0103**

WHAT 445 DISMISSALS WERE WORTH

$7.18
spent on screening

→ **$273.90**
of downstream classifier spend avoided

→ **$38**
of avoided work for every dollar of screening

Each dismissal skips the context-gathering agent and five classifier stages, worth **$0.55–0.76** per finding. Net saving: **$266.72**, about **$30 per day**, or roughly **$11,000 a year** across these six tenants alone.

Operating 35–48× Above Break-Even

Screening pays for itself whenever the dismissal rate exceeds the cost of screening divided by the downstream cost a dismissal avoids.



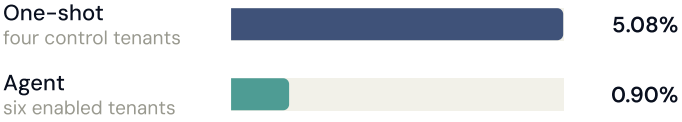
The screen stays net-positive down to a **2% dismissal rate**, which is to say with 97% of its current dismissals disallowed.

27 of the 445 dismissals (6.2%) cite a flagged file as absent because of a pre-existing repository-checkout defect. The fix merged on 12 August but had not deployed within this window, and those findings need re-screening. Excluding them, the dismissal rate is 61%.

More False Positives Caught and No Real Vulnerabilities Lost

False Positives That Leaked Past the Screen

Deferred, then ruled false positive by the full classifier · same window, same classifier path



Catch rate rose from **94.9%** to **99.1%**, a 5.6× reduction in miss rate.

AGENT CATCH RATE BY TENANT

TENANT	CAUGHT	RATE
company-c	194 of 195	99.5%
company-e	137 of 138	99.3%
company-h	84 of 84	100%
company-f	6 of 7	85.7%
internal-test-org	5 of 6	83.3%
company-g	12 of 12	100%
Pooled	**438 of 442**	**99.1%**

Only four findings leaked in total, so read the pooled figure as “at least 98%”. Ground truth is the classifier's own verdict, so a finding both stages get wrong is invisible to this measurement.

HAND AUDITS AT THE EXACT SCANNED COMMIT

46
dismissals audited

44
fully sound

0
real vulnerabilities dismissed

Week one · 11 of 11
Every genuine multi-step investigation sampled was correct.

Day nine · 29 of 31
Stratified across 6 tenants, 22 repositories and 24 rules. Both faults reached the right verdict for a wrong stated reason.

Adversarial · 4 of 4
Every dismissal contradicted by another machine signal resolved in the screen's favour.

How the Screen Wins

5
median tool calls per dismissal

55%
opened a file other than the flagged one

MOST COMMON EVIDENCE

- Tracing a value to a constant
- Proving a guard or type-range invariant
- Establishing execution context, such as a manually run CLI, a test file or a build script

Dismissals a Forty-Line Window Would Have Deferred

- `Prisma.raw()` receives a hardcoded table alias, never the user's search term.
- A `jwt.decode()` call only picks a code branch, and the real `verify()` guards both sides.
- An alleged SSRF URL comes from a local `.env` file with no path from CLI arguments.
- Path-traversal “keys” are drawn from a hardcoded array.
- A Rust “insecure deserialization” finding is plain `serde_json::Value` parsing.

Match the Model to the Question Being Asked

Forty-six clean audits are evidence that the mechanism works. They are not a recall guarantee, and the paper is explicit about what remains open.

What Is Still Open

RECALL

Asserted by contract and audit, not yet by a labelled benchmark. A labelled recall corpus is the gate before the caveat comes off. One pattern to keep auditing for: absence of evidence stated as confident evidence.

PROMPT INJECTION

A larger surface than before. An agent that reads whole files can be addressed by their contents. An adversarial subset in the recall corpus is the corresponding gate.

SAMPLE SIZE

Nine days and six tenants. Only four findings leaked, so the 99.1% carries wide uncertainty on the downside. The audit should be repeated at scale after the checkout fix deploys.

SCOPE

The result is specific to screening. Nothing here shows that open-weight agents can replace frontier models for full triage of real findings. That is a separate, larger bet.

The Next Experiment Is the Other 83%

25×

the screen's tokens, used by the context-gathering agent on the same repository

400 s

the ceiling that often had to terminate it

The screen stops early because it answers a closed question. That stage runs on because it was given an open-ended mandate. If cheap-model-plus-tools is cheap only when the task can tell the agent when to stop, the same technique generalises to the larger slice of spend.

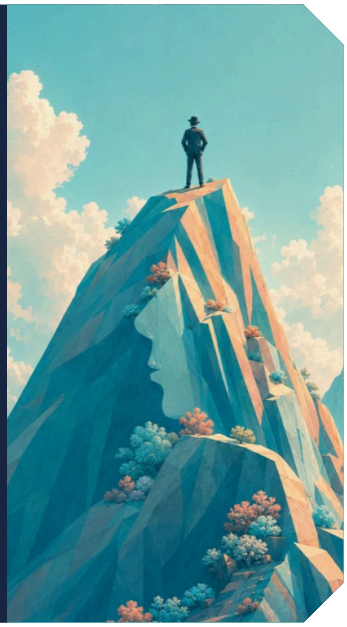
THE BROADER LESSON IS ARCHITECTURAL, NOT MODEL-SPECIFIC

A small model that can read the caller beats a frontier model that cannot.

01 Match the model tier to the question being asked.

02 Give the model the tools needed to answer that question.

03 Define the task tightly enough that the evidence decides when the investigation stops.



Source: “Let the Cheap Model Look: Replacing a One-Shot Frontier-Model False-Positive Screen with a Tool-Using Open-Weight Agent”, J. Lam, Nullify, v0.1.0, 18 August 2026. All figures are taken from the paper or derived directly from its numbers.