PRODUCT REQUIREMENTS · NOVEMBER 2026

PRD: "Report this email" with built-in learning

Author: Shrayanshi Tomar · **Status:** working prototype (this repo) · **Illustrative; no affiliation with any vendor**

1. PROBLEM

Security-awareness programs train people to spot phishing, then give them a "report" button that returns nothing. The employee never learns whether they were right or why, so the moment of highest attention is wasted. Security teams, meanwhile, drown in reports that are mostly harmless.

2. USERS AND JOBS

User	Job to be done	Pain today
Employee (non-technical)	"Tell me if this is safe and why, fast"	Silence after reporting; no feedback
Security / IT analyst	"Send me only reports that need a human"	Triage queue full of newsletters and false alarms
Awareness program owner	"Show that training changes behaviour"	Click rates only; no signal on reporting quality

3. PRINCIPLES (WHAT THE PROTOTYPE DEMONSTRATES)

1. **Explain, don't just score.** Every flag names the sign and gives one sentence on what to look for next time.
2. **Never reassure without evidence.** If the sender cannot be verified and the message asks you to open something, the answer is "check first", not "safe".
3. **Show where links really go.** Displayed text vs. true destination is the single most teachable moment.
4. **Privacy by default.** Content is analysed in memory and never stored; only counters are kept. Links are never fetched.

4. SUCCESS METRICS

Metric	Definition	Prototype result (44 synthetic messages)
Missed-threat rate (north star, lower is better)	phishing rated "no warning signs" / all phishing	keyword baseline 43.5% → v1 8.7%
False-alarm rate	legitimate mail rated "likely phishing"	0% (both)
Ambiguous handled	uncertain mail sent to "check first"	66.7% → 100%
Lesson open rate	reports where the user expanded the lesson	tracked live in the UI

Production leading indicators: report rate per 100 employees, time from receipt to report, repeat-mistake rate (same sign missed twice), share of reports analysts can close without opening, user rating of the explanation.

5. SCOPE

v1 (built): signal engine, review band, link inspector, lessons, counters, eval harness, container deploy. **Next:** (a) mailbox add-in that reads real headers instead of a pasted form, (b) analyst queue with the explanation attached, (c) per-user "signs you tend to miss" recap that feeds the next training, (d) sender-behaviour history to catch compromised internal accounts, (e) an LLM engine behind the same input guard for wording and edge cases, compared in the same scorecard.

6. KNOWN FAILURES (KEPT VISIBLE ON PURPOSE)

- **Compromised internal account (k01):** authenticated sender, own-domain link, no language cues. Needs sender-behaviour history (first-time share pattern, unusual hour), which content rules cannot see.
- **Pure social engineering (k02):** "call me on this new number" has no technical signal. Needs a payment or contact-change policy prompt and analyst review, not more rules.

7. RISKS

Risk	Mitigation
Small author-written set; rules tuned against it	Treat numbers as harness validation. Next: held-out set from real reported mail
Over-warning trains people to ignore it	Track legit friction and false alarms as first-class metrics
Attackers learn the rules	Combine with reputation and behaviour signals; rules are the explainable layer, not the only one
Pasted-form input is not real header data	v2 reads headers directly from the mail client

8. ROLLOUT IDEA

Pilot with one team, compare report rate and analyst time-to-close against the current process for four weeks, then expand if missed-threat stays low and analysts close more reports without opening them.