Category 9: AI-Specific Bias Vulnerabilities

Contents

This directory contains detailed implementation schemas for all 10 indicators in the AI-Specific vulnerability category.

Overview

AI-specific vulnerabilities exploit over-reliance on AI/ML systems, automation bias, algorithmic opacity, and unique human-AI interaction patterns.

Indicators

1. [**9.1**] **AI Recommendation Over-Trust** - Uncritical acceptance of AI outputs
2. [**9.2**] **Automation Bias** - Preferring automated decisions over human judgment
3. [**9.3**] **Algorithmic Authority Deference** - Treating AI as infallible authority
4. [**9.4**] **Model Drift Blindness** - Missing performance degradation over time
5. [**9.5**] **Adversarial Manipulation Susceptibility** - AI systems fooled by crafted inputs
6. [**9.6**] **Machine Learning Opacity Trust** - Black box acceptance without understanding
7. [**9.7**] **AI Hallucination Acceptance** - Treating confident but wrong AI outputs as truth
8. [**9.8**] **Human-AI Team Dysfunction** - Collaboration failures in hybrid teams
9. [**9.9**] **AI Emotional Manipulation** - AI exploiting human emotions
10. [**9.10**] **Training Data Bias Propagation** - AI amplifying dataset biases

Implementation Schema

Each indicator follows the **OFTLISRV** framework with AI system monitoring.

Key Metrics

AI Override Rate

```
AOR = Human_overrides / AI_recommendations
```

Very low (<5%) or very high (>50%) indicates dysfunction.

Automation Bias Score

```
ABS = AI_accepted_errors / Total_AI_errors
```

Model Performance Drift

```
MPD = (Accuracy_current - Accuracy_baseline) / Accuracy_baseline
```

Key Data Sources

- **AI/ML Systems**: Prediction logs, confidence scores, feature importance
- **SIEM**: AI-generated alerts, ML model outputs
- **User Decisions**: Overrides, acceptances, modifications of AI recommendations
- **Model Metrics**: Accuracy, precision, recall over time
- **Incident Data**: False positives/negatives from AI systems

Detection Approach

Automation Bias Detection

```python
# Track acceptance vs verification
ai_recommendations = get_ai_outputs(window=7_days)

for recommendation in ai_recommendations:
    if recommendation.accepted and not recommendation.verified:
        if recommendation.confidence < 0.8:  # Low confidence
            flag_automation_bias(user_id)

    # Check if errors are caught
    if recommendation.actual_result == 'false_positive':
        if recommendation.accepted_without_override:
            automation_bias_errors += 1
```

Model Drift Detection

```python
# Monitor model performance over time
current_metrics = model.evaluate(recent_data)
baseline_metrics = load_baseline_metrics()

drift = {
    'accuracy': current_metrics.accuracy - baseline_metrics.accuracy,
    'precision': current_metrics.precision - baseline_metrics.precision,
    'recall': current_metrics.recall - baseline_metrics.recall
}

if any(abs(d) > 0.1 for d in drift.values()):  # >10% degradation
    alert_model_drift(model_id)
```

AI Hallucination Detection

```python
# Detect confident but wrong outputs
predictions = get_ai_predictions()

hallucinations = [
    p for p in predictions
    if p.confidence > 0.9 and p.actual_result == 'error'
]

if len(hallucinations) / len(predictions) > 0.05:  # >5% rate
    flag_hallucination_risk(model_id)
```

Baseline Establishment

AI-specific indicators require: - Initial model performance metrics - Human-AI collaboration patterns - Override rate baselines - Model retraining schedules

Common Event Types

- `ai_recommendation_followed` → 9.1, 9.2, 9.3
- `model_prediction_error` → 9.4, 9.5, 9.7, 9.10
- `ai_explanation_missing` → 9.6
- `human_ai_disagreement` → 9.8
- `ai_generated_content` → 9.9

Risk Levels

- **Low** (0-0.33): Healthy skepticism, appropriate AI use
- **Medium** (0.34-0.66): Some over-reliance, verification still occurs
- **High** (0.67-1.00): Blind trust in AI, critical thinking suspended

Mitigation Strategies

Technical

- Confidence thresholds for auto-acceptance
- Mandatory human review for high-impact decisions
- Model performance monitoring dashboards
- Explainable AI (XAI) implementations
- Adversarial testing programs

Organizational

- AI literacy training
- Understanding ML limitations
- Human-in-the-loop requirements
- Regular model audits
- Diverse AI development teams

Process

- Model retraining schedules
- Performance degradation alerts
- Override documentation requirements
- Alternative hypothesis testing
- Bias audits of training data

Special Considerations

LLM Integration

For RAG-enhanced CPF systems using LLMs: - Validate LLM outputs against ground truth - Monitor for hallucinations in psychological assessments - Maintain human expert oversight - Version control for prompt engineering

Adversarial Robustness

- Test models against adversarial examples
- Monitor for evasion attempts
- Implement ensemble methods
- Regular red team exercises

Related Resources

- **Dense Foundation**: `/foundation docs/core/en-US/` - AI vulnerability formalization
- **CPF LLM Blueprint**: Main CPF paper - RAG integration methodology
- **Dashboard**: `/dashboard/soc/` - AI performance metrics
- **Research**: Human-AI teaming in cybersecurity