Research Walkthrough: Adversarial Robustness Assessment (V2)

This document presents the final, high-rigor adversarial ML audit of the Zero-Trust AI policy engine, incorporating fixes for data leakage, feature constraints, and increased attack budgets.

□ Research Objective

To evaluate the resilience of an AI-driven Zero-Trust policy engine against adaptive evasion attacks using the **IBM Adversarial Robustness Toolbox (ART)** and validate a multi-layered defense-in-depth architecture.

□ Scientific Rigor Upgrades

- **Anti-Leakage Protocol**: Surrogate models trained strictly on `train_set.csv`; evaluation performed on unseen `test_set.csv`.
- **Feature Clipping**: All adversarial samples are restricted to realistic telemetry bounds (min/max bounds from distribution).
- **High-Iteration Budget**: Black-box attacks (HopSkipJump) increased to **50 iterations** for true stress testing.
- **Robust Metrics**: Added Robust Accuracy and Perturbation Norm analysis (L2/Linf).

□ Final Results Table (High Rigor)

Metric	Natural Clean	Black-Box (HSJ)	White-Box (FGM)
Baseline Evasion Rate	23.68%	**66.67%**	25.00%
Defended Evasion Rate	14.74%	**4.17%**	**16.67%**
Robust Accuracy (Defended)	90.13%	23.00%	86.00%
Mean L2 Perturbation	0.00	73.92	0.49

□ Key Insight: "Boundary Diffusion"

The hybrid defense layer (RF + Isolation Forest + Thresholding) achieved a **93.7% reduction** in black-box evasion success. Even under a high-budget 50-iteration attack, the unsupervised anomaly layer effectively "diffused" the decision boundary, making the model significantly harder to evade.

□ Layered Security Architecture

The system implements a **Defense-in-Depth** strategy:

1. **Supervised Layer (Random Forest)**: Standard classification.
2. **Unsupervised Layer (Isolation Forest)**: Trained only on benign traffic to detect "unknown unkowns."
3. **Uncertainty Layer**: Flagging samples with prediction probabilities near 0.5 as "boundary-exploitation attempts."

□ Security Engineer Project Structure

```
    c:/Adversarial Model training/
    ├── logs/                # Security alerts and SIEM-style logs
    ├── models/              # Weights + feature bounds + anti-leakage
datasets
    ├── src/
    │   ├── adversarial/     # High-rigor evaluation & security
logging
    │   ├── detection/       # Core training logic
    │   └── simulation/      # Traffic generation
    ├── threat_model.md      # Formalized attacker profiling
    └── report.md            # Technical deep-dive
```

✓Final Conclusion

By applying security engineering principles to ML evaluation, we transformed a standard classifier into a robust policy control. The system is resilient to high-budget black-box evasion and demonstrates clear, quantifiable security trade-offs—marking it as a core asset for any Zero-Trust AI infrastructure.