Modernizing Test and Evaluation for Insider Risk Analysis

Bob Ditmore

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM22-0772

Initial Testing Approaches

- Insider Risk Management Programs may take a penetration-type testing approach
  - Testing can be *Overt* or *Covert*

Overt	Covert
• + Trains staff • + Exercises Insider Risk scenarios • + Tests process and procedures with team knowledge • - Requires staff time to execute • - Injects activity into an operational network	• + Tests process and procedures and team readiness • + Exercise Insider Risk scenarios • + Third party can execute (with ELT consent) • - Injects activity into an operational network

Both assume sensors and analytics are in place

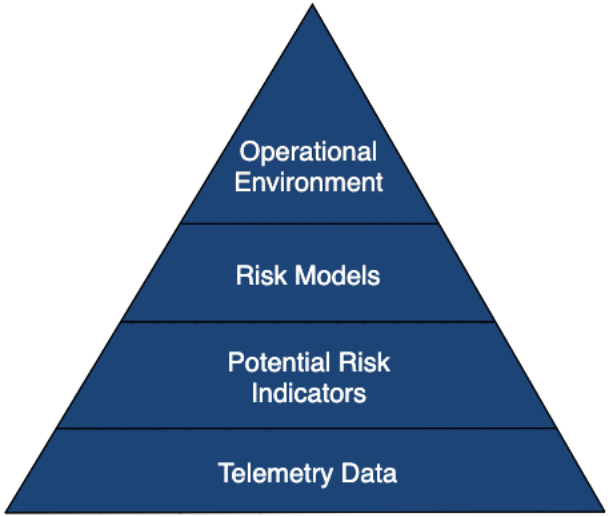
Testing Sensors and Analytics

- Risky to test on an operational environment
  - Can disrupt organizations mission
    - Overwhelm network resources
    - Spike resource utilization
  - Can impact insider risk analysts
    - Potentially flood analyst with false positives
    - Cause alert fatigue
    - Potentially cause analysts to miss true positives due to excessive ‘noise’

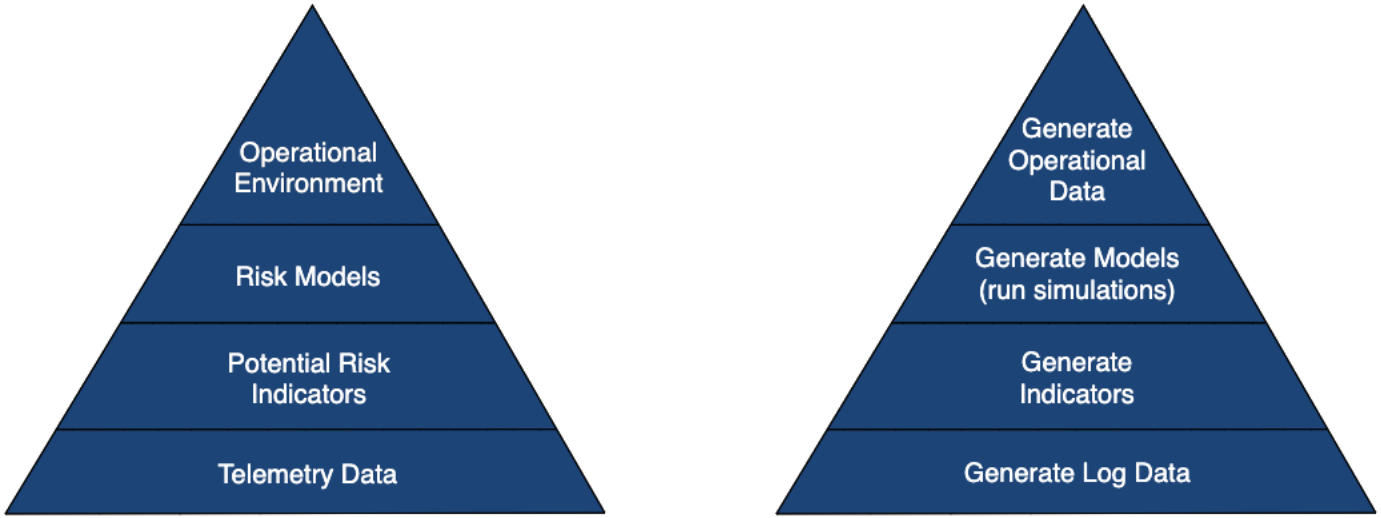
What questions are we trying to test

- At the sensor (data source) level and analytic level
  - Will my indicator work?
    - Will it indicate something an analyst can action?
  - Are my analytics effective and efficient?
    - What data/fields does my analytic need?
    - Can it produce timely results?
  - Is there a better sensor for my analytics?
  - Will my AI/ML algorithm find scenario *X*?
    - Based on scenarios in the IRMP's scope
    - Using the data sets available to the application

Questions can be asked a different levels – 1



Questions can be asked a different levels – 2



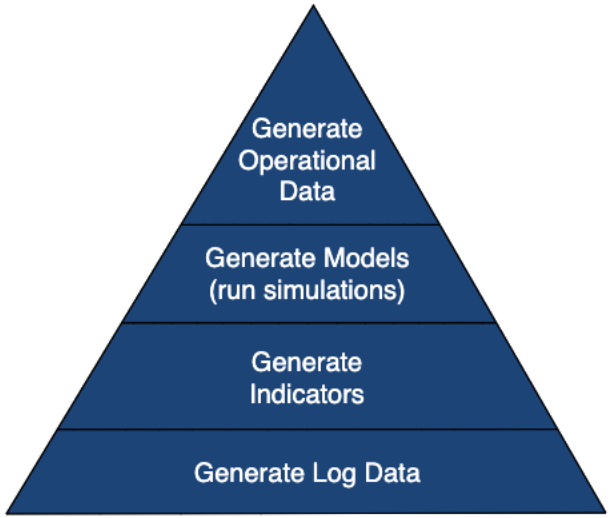
Issues with these approaches

Hard to determine if we have ground truth

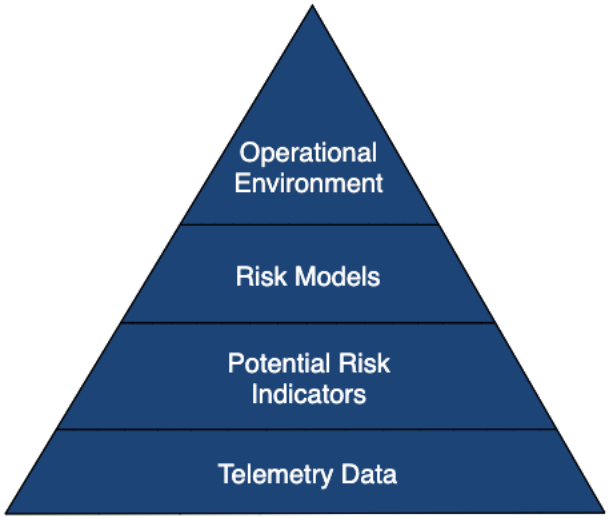
Hard to maintain realism of underlying systems

Hard to *what if* other scenarios

Hard to scale



Test and Evaluation Environment Needs



Environment

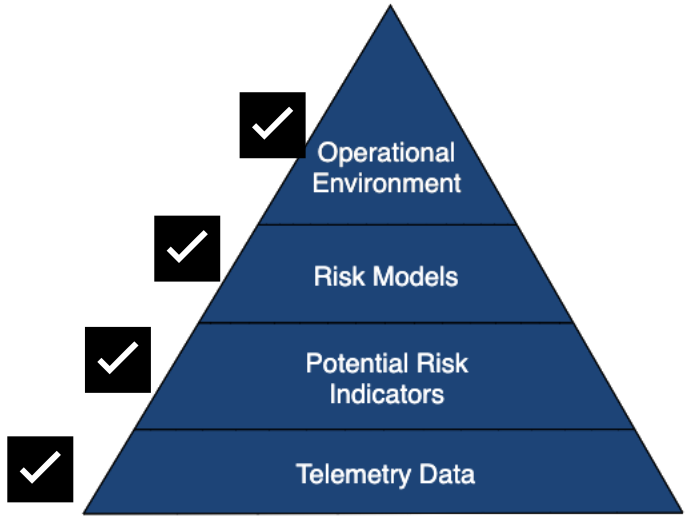
- Ground Truth
- Systems Realism
- Ability to *What-If* different scenarios
- Scalability

- Reasonable cost
- Used different telemetry data

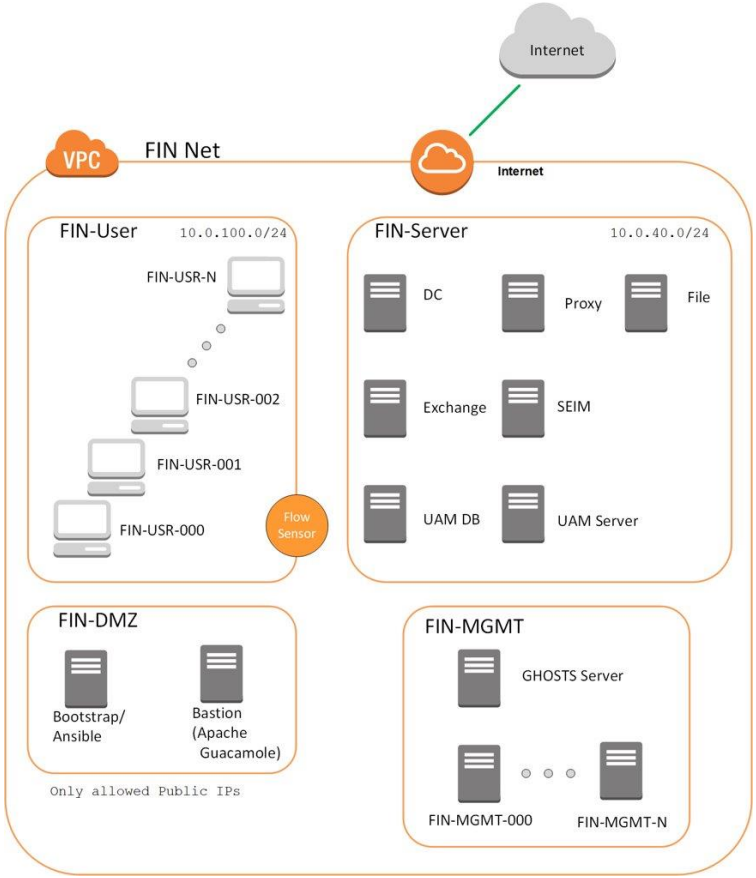
Our approach

NeedleStackCloud

- Infrastructure as Code (IaC)
  - Rapid setup and teardown of environments
    - Went from days/weeks to minutes
  - Easy to what-if different environments
  - Easy to scale environments
  - Easy to run multiple environments simultaneously
- Cloud Based approach
  - Provides scaling at a reasonable cost
  - Bonus: ability to share environments
- Non-player Character (NPC) Approach
  - Ground Truth



NeedleStackCloud



HashiCorp Terraform

https://www.terraform.io

Apache Guacamole™

https://guacamole.apache.org

ANSIBLE

https://www.ansible.com

https://github.com/cmu-sei/GHOSTS

Contact Information

Bob Ditmore

Team Lead – Insider Risk, CERT Division

Software Engineering Institute

Carnegie Mellon University

rmditmore@sei.cmu.edu

https://www.sei.cmu.edu/our-work/insider-threat/index.cfm