



# Analytical Evaluation Framework

Tim Shimeall  
CERT/NetSA Group  
Software Engineering Institute  
Carnegie Mellon University

August 2011



# Disclaimer

---

© 2011 Carnegie Mellon University

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.



# Describing Network Analytical Capabilities

---

Develop descriptions that support fair evaluation of current or potential capabilities to address network defense needs and operational cycles

- “How does it fit” not “Is it good”
- Input to acquisition, not decision for them
- Methodical and impartial, not objective

Supportive of network security, but applicable somewhat beyond just network security

- Harvest analyst expertise
- Consideration of carry-over effects

# Phase 1: A Language Model

---

Nouns – forms of data handled by the capability

- Inputs
- Processing
- Results

Verbs – primitive actions supported by the capability

- Data handling
- Process
- Analytic
- Presentational

Adverbs – characteristics of the capability

- Process
- Product

Prepositions – scope or limitations of the capability

# Assessing Data

---

What is the primary data handled by the capability?

What is secondary data handled by the capability?

What is supportive data handled by the capability?

What primitive operations are associated with each?

How well are the operations implemented? What is missing?

# Example: Sourcefire IDS

---

Primary input: Packet data

- Collect, Abstract, Parse, Alert, Store, Query, Export

Secondary input: Network map

- Select, Group, Aggregate

Supportive input: Signatures

- Import, Alert, Store, Export

# Input/Processing/Output

---

Input: what data does the capability consume?

Sourcefire consumes network packets

Process: what data is used for control or direction of the capability?

Sourcefire uses signatures and network configuration information

Output: what data is produced by the capability?

Sourcefire produces alerts, and selective packet capture

# Network Level of Abstraction

---

Many capabilities are focused on particular range of protocols and behaviors

IP layer: packet-based analysis, does not get into local behavior and only infers application behavior (e.g., SiLK)

Application layer: message-based analysis, does not deal with transport mechanics (e.g., analysis of email patterns)



# Assessing Operations

---

What locus of operations forms the “core” functionality of the capability?

What are secondary operations?

What are supportive operations?

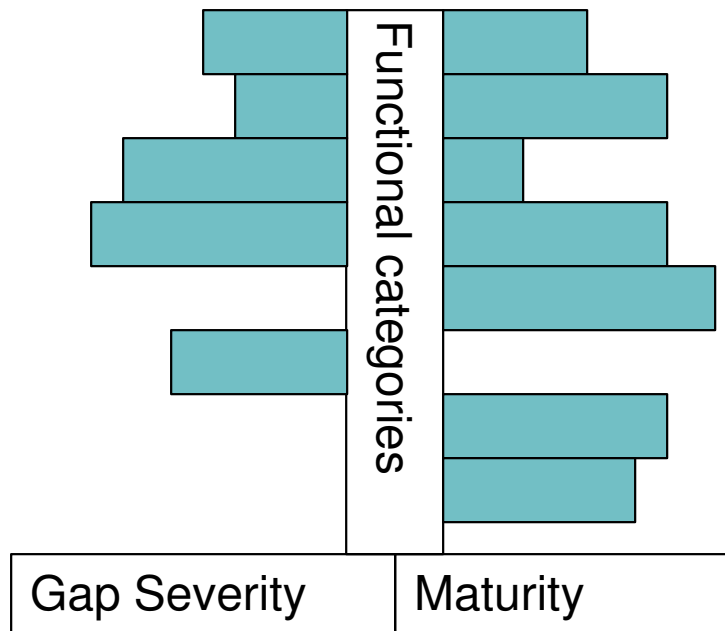
How well are those operations implemented?

How scoped is the intended application?

Rating scheme: 0-5, plus n/a, not eval, absent

# Summarizing Operational Gaps/Maturity

---



Balance functional maturity vs. capability gaps

All tools have gaps

Goal is to see how peaks and valleys match

# Process Adverbs

---

Sourcefire IDS:

Operational

Qualitative

Tactical

Concise

# Product Adverbs

---

## Sourcefire IDS:

- Not Data-diverse
- Immediate
- Responsive
- Interoperable
- Documented
- Supported
- Trained
- Robust
- No Workflow
- No AAA

# Prepositions

---

Under Conditions (e.g., edge vs. transit)

At Size / scale (e.g., enclave vs. enterprise, days vs. months)

Of Scope (e.g., CND vs. network ops)

Within Coverage (e.g., sparse vs. complete)

In time (e.g., interactive vs. batch vs. continuous)

# Phase 2: Process Descriptions

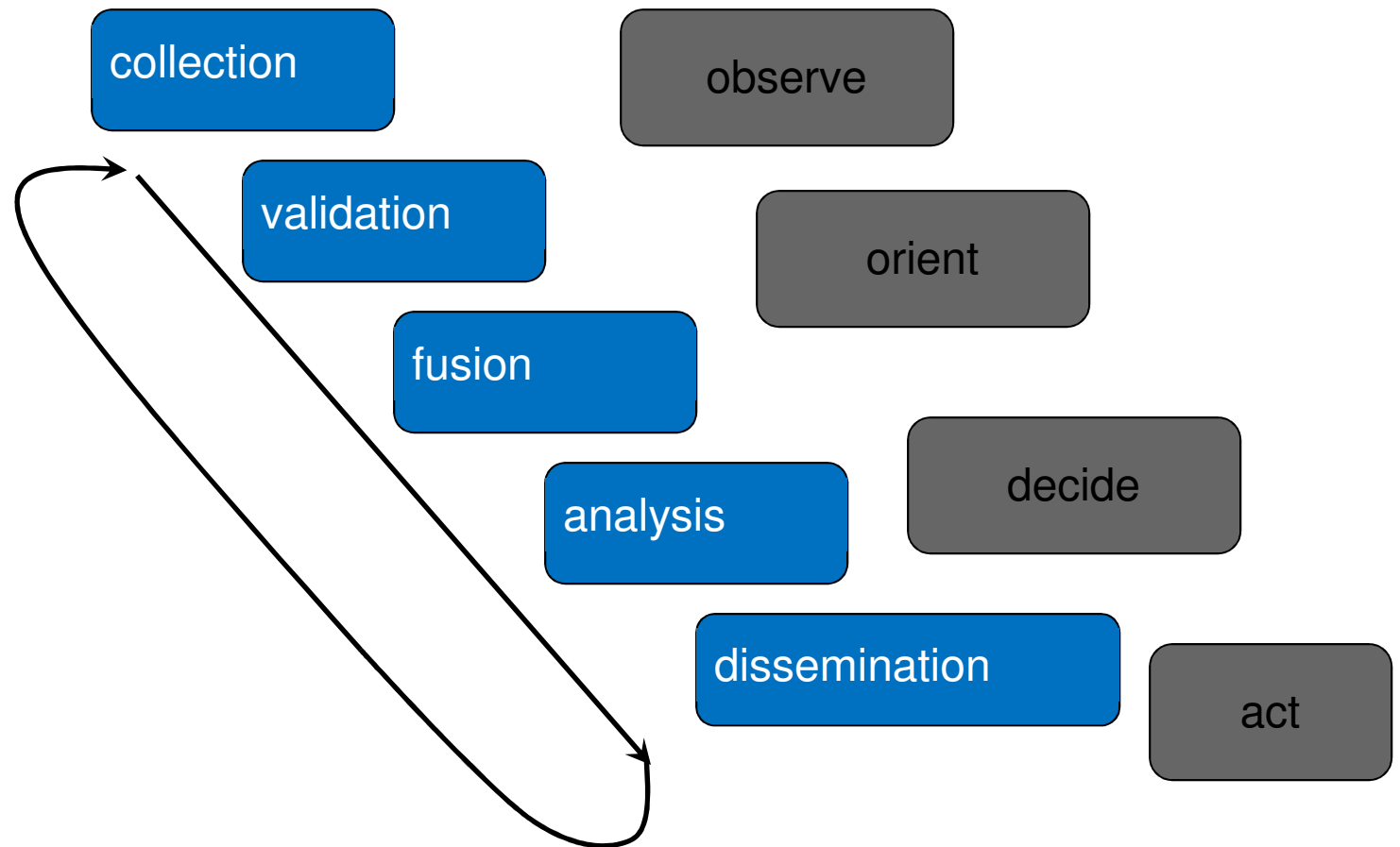
---

What form of reasoning should the model support?

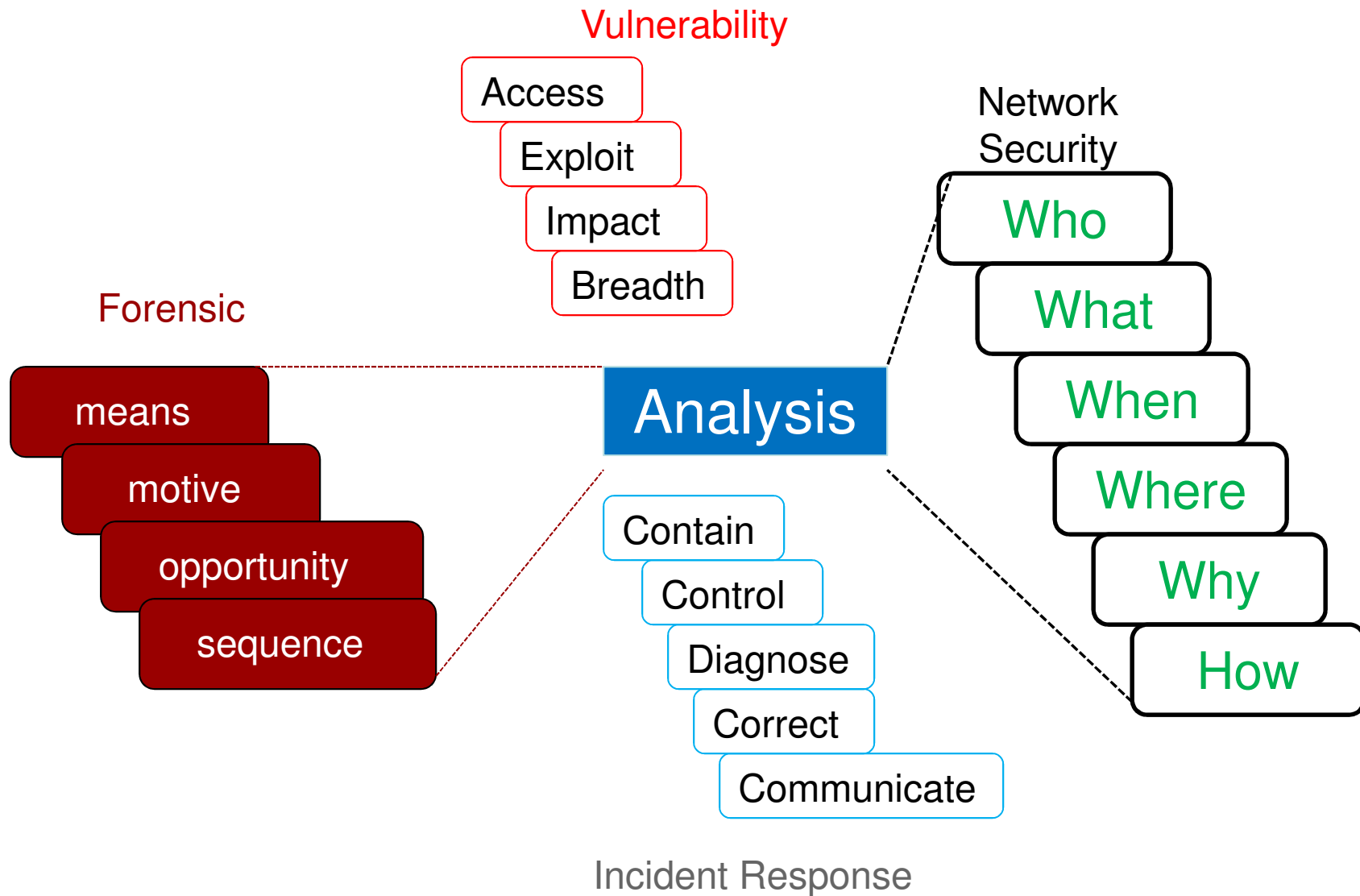
- Fused-source intelligence
- C2/OODA?
- Forensic?
- Bayesian hypothesis testing?
- Abductive pattern matching?

# Network Analysis Approaches

---



# Analysis Decomposed





# Next Steps

---

Expand initial visual results into fair comparisons

- Spider diagrams
- Input/Process/Output tables
- Network level tables
- Operational maturity/gaps

Define requirements for evaluation process using model

- Team?
- Approach?
- Process?
- Outcomes?
- Threats?

Tie capabilities to process needs

- Threshold approach (score needs to be X)
- Conditional approach (capability must include Y)
- Descriptive approach (need to support operations Z)

Reasoning Support