“Mission Assurance in a Distributed Environment”

14th ICCRTS – C2 and Agility

Track 8 - C2 Assessment Tools and Metrics

Authors
Chad DeStefano and Thomas A. Clark
AFRL / RISF

Document cleared for public release with Case Number 88ABW-2009-2512
Overview

- Problem
- Objective
- Defining Mission Assurance (MA)
- DEEP Description
- Applying MA to DEEP
- Future Work
  - Metrics and Experimentation
- Summary
Problem

• Shift from individual hackers to sophisticated teams operating at will in complete stealth
  – Website defacement, Denial of Service (DoS) attacks, identify theft are overt, and nearly immediate to detect
  – Persistent access designed to influence in subtle or perhaps violent ways is becoming the new threat

• Continued shift to network-centric C2 with information processing distributed over computer networks at geographically dispersed locations presents technical challenges
  – The biggest threat is to our core mission planning and processing systems, examples:
    • Target coordinate, inventory decrement manipulation
Objective

- Define and illustrate mission assurance concepts within a distributed application operating in a notionally contested environment
  - Use the Distributed Episodic Exploratory Planning (DEEP) as an exemplary planning environment
  - Identify DEEP components that can be enhanced to maintain operations under duress
    - Initial “fight-through” capability
  - Formulate a test environment to conduct experimentation and determine metrics
Defining Mission Assurance

• Use standard information assurance (IA) tenets as a baseline
  – Attribution - holding a user accountable for their actions
  – Authentication – ensuring only privileged users access appropriate information
  – Availability - ensuring information and services are available when required
  – Confidentiality – ensuring information destined for an individual or group is exclusive
  – Integrity – information is kept unmodified by unintended sources

• IA Extensions
  – Availability a function of prioritized mission tasks mapped to network capabilities
  • So degraded states can be specified and measured
  – Trust must be built on top of attribution, authentication, confidentiality and integrity
  • So that contributors to mission success will be given increased responsibility
  – Mission workflow must be formally specified as business processes

• Exploring Trust
  – Trust is integral regarding either human or machine interaction
  – DEEP does not address trust formally yet (trust is assumed)
BOGSAT

- Bunch of Guys/Gals Sitting Around a Table

Constrains planning

- Quality
  - Finite experience
- Speed
  - Limited automation
- Creativity
  - Finite diversity

Improve planning quality, speed, and creativity

- Experienced-based
  - Orient and decide faster than adversaries with better plans
- Mixed-initiative
  - Syntheses of the strengths of both human and machine
- Net-centric
  - Expert team formation with greater diversity and creativity
**DEEP Example**

**Scenario/Situation**

- **Multi-Case Agents**
  - Initial Plan: $A_{51} + L_{32} + M_{451}$
  - Experience-Based Planner

- **Critic/Advis or Agents**
  - Refined Plan: $p_1$
  - Plan Advising & Critiquing Agents

- **Plan Explorer (M&S)**

**Distributed Experience Bases**

- $A = \text{air}$
- $S = \text{space}$
- $C = \text{cyber}$
- $I = \text{integrated}$
- $L = \text{land}$
- $M = \text{maritime}$
• Specifically
  – **Distributed AI Blackboard** for multi-agent, non-deterministic, opportunistic reasoning “at the edge”
  – **Experience-Based Reasoning** to capture experiences (successes and/or failures)
  – **Episodic Memory** for powerful analogical reasoning
  – **Multi-Agent System** for mixed-initiative planning
  – **ARPI Core Plan Representation** for human-to-machine dialog
  – **Constructive Simulation** for exploration of plausible future states
**DEEP Approach**

1. Engaged CMDR: “I have a situation!”
2. Planning Agents
3. Case Base
4. CBR System
5. Suggested
6. Adjusted
7. Judged
8. Execution Selection Critic Agent
9. Simulated
10. User Interface

**Candidate Plans:**
- Objective 1
- Objective 2

**Plan Execution**

**Objective**
- Objective 1
- Objective 2

**Situation**

**Selected:**

**Adaptation Agents** ("Repairers")

**Critic Agents** ("Evaluators")
Applying MA: Modeling the Process

- Protecting internal and external applications requires a model of the overall business process
- In DEEP, the business process is modeled at the application level and we can determine:
  - The sequence of prioritized events/activities
  - Event dependencies
  - Events that are not as important to the core business as others
- Knowing this information allows us to make decisions on redundancy, contingency plans, resource management for IA, and the impacts of resource losses
- In some cases, DEEP handles intrusions intrinsically
  - Plans have to survive a critical review process that would eliminate plans that were not fit for the objective
  - Critic agents do not have authority to modify plans
Agent Control Center (ACC)

- Agents are an integral part of DEEP, so proper synchronization and control is important.
- The ACC automatically and manually controls agents and monitors the system and network, it should:
  - Monitor traffic, move agents, shutdown agents, restart agents, ping agents, conduct behavior analysis based on connection patterns, and assess agent interaction as a foundation for determining trust.
  - Some of these functions are provided by the Java Agent Development Framework (JADE) used to develop the DEEP agents.
  - Detect network issues like congestion and attempt to automate system restart on an operable network.
• Data concerns
  – Modification (both minute and large)
  – Deletion
  – Theft
• Solutions
  – Encryption
    • All traffic should be encrypted
    • Data repositories should be encrypted
  – Hold data integrity using signature techniques to ensure data has not been modified
  – ACC could monitor traffic and alert based on irregular data movement
  – Redundant stores of data and rollback capability to ensure steady recover in the event of intrusion
  – Authentication to data repositories (limit access to a need to know basis – blackboard has panes / layers concept)
The human in the loop can pose problems for the mission as well

- Classic “insider threat”
- Insiders may have access to critical data and knowledge of how to use it
  - Very tough problem to solve

Solutions

- Enable authentication procedures
- User privileges – blackboard using authentication and proper registration to specific zones of information
• Networks that applications operate on also provide an attack vector
  – Examples of issues include limited bandwidth, loss of bandwidth (DoS, kinetic attack)
  – Solutions
    • Control center and network examining tools should detect loss of communication and attempt to regain functionality.
      – Software component movement or restart with state
    • Use of another mode of communication
Future Work

• Better establishment of metrics / experimentation
  – Experimentation
    • Emulation of rogue agent behavior sending out messages it shouldn’t
    • Conducting a DoS attack at critical pressure points
    • Emulation of component loss
    • Data modification – Can DEEP intrinsically handle data changes during the process?
  – Metrics (area of interest)
    • Must be able to achieve the above issues
    • Rollback must be faster than full restart
Future Work

• Establish a generic framework to apply to other programs
• Integration of AFRL IA in-house technology
• Multi-agent control
• Trust (can we employ wisdom of the crowds voting mechanic or control procedures to ensure trust?)
Summary

- Providing mission assurance is not an option, but a requirement for surviving in a contested network environment
- Emphasize building applications and systems that are reliable, self-sustainable and trustworthy
- Applying mission assurance using DEEP allows for experimentation as well as the creation of a generic model of mission assurance
Thank You and Questions

Chad.destefano@rl.af.mil
Backups
• Business Process Execution Language (BPEL)
  – Web service standard for specifying interactions
  – Model executable and abstract processes
• Business Process Modeling Notation (BPMN)
  – Graphical representation of business processes in a workflow
• Unified Modeling Language (UML)
  – Use standard UML diagrams to model the system
  – Component, sequence, activity diagrams