Realistic and Affordable Cyberwarfare Opponents for the Information Warfare Battlespace

Martin R. Stytz, Ph.D.
AFRL
WPAFB, OH
martin.stytz@wpafb.af.mil
mstytz@att.net

Sheila B. Banks, Ph.D.
AFRL
Orlando, FL
Sheila.banks@afams.af.mil

Michael J. Young, Ph.D.
AFRL
WPAFB, OH
Michael.young@wpafb.af.mil

Motivation

- "Train the way we fight"
  - Realistic training at all levels
- Increasing reliance on information superiority for the battlefield
- Need to train for information warfare operations for commanders at all levels
  - Need for effective training is increasing
- The needed tools are not available
- The technological advances in computer generated forces, information assurance, and software protection technologies can be exploited to provide the tools
- But research is needed in several areas

Overview

- The Arena
- Background
- Requirements
- Development Approach/Methodology
- Suggested Further Research

Information Warfare Arena

- Events occur at high speed, much faster than human thought processes
- Rapid change in attack vectors
- Need for technical expertise for command and control
- Current lack of metrics to measure defense effectiveness
- Difficult to develop and maintain situation awareness
- Difficult to predict future activity in cyberbattlespace
- High degree of vulnerability to intended and unintended effects of cyberspace actions
- Hence - training is difficult and access to real-world facilities is limited due to potential for unintended harm

Need/Objectives

- Information warfare cyber red team
- Prepare all command echelons for cyberbattlespace
- Cost effective
- Suitable for training and testing
- Flexible, innovative exploits across the entire cyberbattlespace
- Ease of assembly and modification of the cyber red team
- Indistinguishable from human conducted exploits

Solution Overview

- Provide cyberbattlespace training environment
- Develop high-fidelity models of opponents expressed as computer controlled actors
  - Satisfy training and testing needs
  - Cost effective
  - Provides repeatability and basis for statistical analysis
  - Human overseer
- Information Warfare Opposing Force (IW OPFOR)
  - The computer controlled red team

Background

- Discuss enabling technologies
- Security technologies
- Computer generated actor (CGA) technologies
  - Knowledge representation
  - Human behavior representation
- Software Technollogies

Network-Based Attacks

- Commonly known vulnerability
- Traditional attack vector
  - Provides entry point for application attacks as well
- Deny service or false information
- Success requires a combination of speed and knowledge about software construction
- Information Assurance programs attempting to reduce vulnerability
- Costly to provide opponents or to test

Software Protection

- Long history but not as well known
- Application software and data are increasing in importance and value
- Network and operating system security cannot meet current and future software protection needs
  - Currently, no inherent protection; encryption not sufficient
  - History of successful exploits highlights vulnerabilities
- Need for improved application security will arise from the ever increasing value of simulation software and its data and inability to close all network/operating system vulnerabilities
- Main technical objectives
  - Make the task of compromising the software so difficult that attackers give up
  - Make the task of compromising the software so time consuming that attackers give up

Software Protection Requirements

- **Protect**
  - Application security without development or performance penalty
  - Array of validated protection techniques tailored to the criticality of the code, the operational and threat environments, and computational power
  - Scalable and customizable protection
- **Detect**
  - Self monitoring of protected software for
    - Malicious activity
    - Code integrity
- **React**
  - Array of autonomous self defense measures for protected codes
- **Major tools**
  - Obfuscation, watermarking, computational degradation

Obfuscation

- Employed at the source and binary levels
- Employs counter-intuitive programming logic to hide control and data flows
- Preserves the semantics of the program
  - Same observable behavior
  - Understanding and reverse engineering the obfuscated program must be more time consuming than performing the same tasks for the unobfuscated program
- Challenges
  - Determining which transforms to apply
  - Determining where to apply transformations
  - Determining the level of security achieved

Software Watermarking

- Idea is to embed a watermark into a program such that:
  - The watermark can be detected
  - It is unlikely that the watermark occurred unintentionally
  - Performance is not adversely affected
  - Stealthy
- Two types: static and dynamic
  - Static - computed at compile time and permanently embedded in the software
    - Easier to develop but less resilient
  - Dynamic - computed at runtime and changes from execution to execution
    - Resilient but performance impact difficult to predict
- No good techniques at present

Performance Degradation

- Reduce the accuracy of computations in such a manner that the pirate can not detect them
- Relies upon authentication and watermarks/metrics to enable the software to determine if it has been subverted

Knowledge Representation

- Improvement in understanding of knowledge needed to attack network or software and defend them
- Increased knowledge about attack exploits and attack strategies, vulnerability categories, and metrics
- Improved understanding of network and information warfare as well as attack strategies and tactics
- Gradual improvement in understanding of defensive needs
- Have the knowledge needed to assemble elementary and gradually improving computer-controlled attack systems for training and testing

Software Technologies

- Several enabling technologies have been devised
- Software components
  - Enable reuse and maintenance
  - Independent, tied together by other software
- Frameworks
  - Tie together components, objects, aspects, etc
  - The skeleton of the system
- Software gauges
  - Enable runtime evaluation and modification of the system
  - Permit cyber red team to assess performance automatically as well as help human overseer assess effectiveness of attack and change strategy or tactics dynamically
  - Consist of a probe to gather data and a display to evaluate data

Software Technologies (cont.)

- Two key software technologies to assist in the development of cyber red team
  - eXtensible Markup Language (XML)
  - Unified Modeling Language (UML)
- XML can be used to express the knowledge needed
  - Independent of user
  - Self-describing and self-contained
  - Extensible and flexible
- UML can be used to capture knowledge use sequences, attack strategies, and defense strategies as well as systems and federations of attacking systems

Human Behavior Representation

- Improved ability to construct systems that emulate human behaviors and performance
  - Ever increasing fidelity is key and iss the current trend
- Improved ability to gather, categorize, and employ specialized knowledge
  - Military as well as cyberbattlespace
- Better intent and human behavior models
- Expandable and modifiable
- Attaining consistent performance
  - Enables consistent testing as well as repeatable training

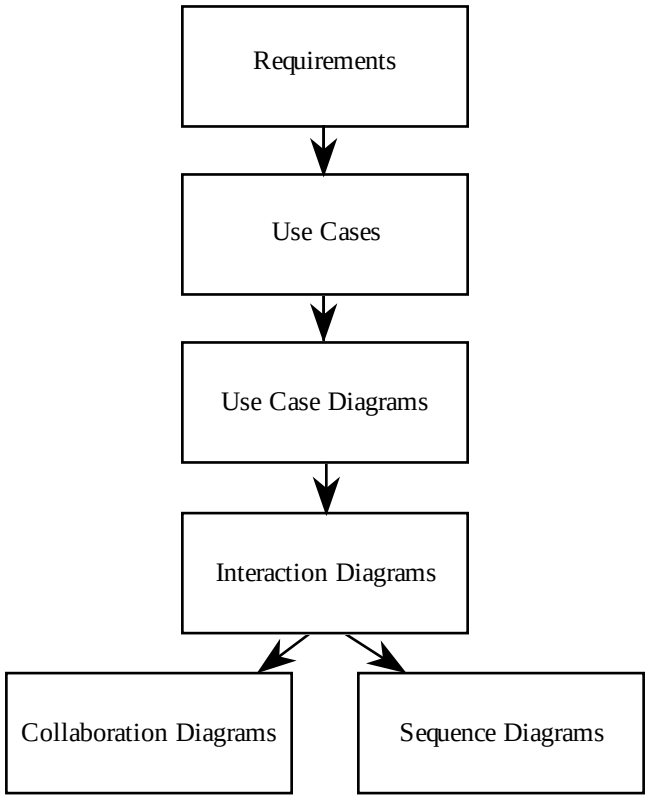
Cyber Red Team Requirements

- Employ any reasoning technique or hybrid combination
- Adaptive learning and autonomous behavior modification
- Unpredictability of exploit
- Autonomous analysis of actions
- Readily programmed with exploits and assessment criteria
- All actions in an exploit visible to human overseer
  - Symbiosis
- Ontology
  - Description of knowledge and standard meaning
- Conduct multiple, simultaneous, coordinated, mutually supporting exploits

IW OPFOR Development Strategy

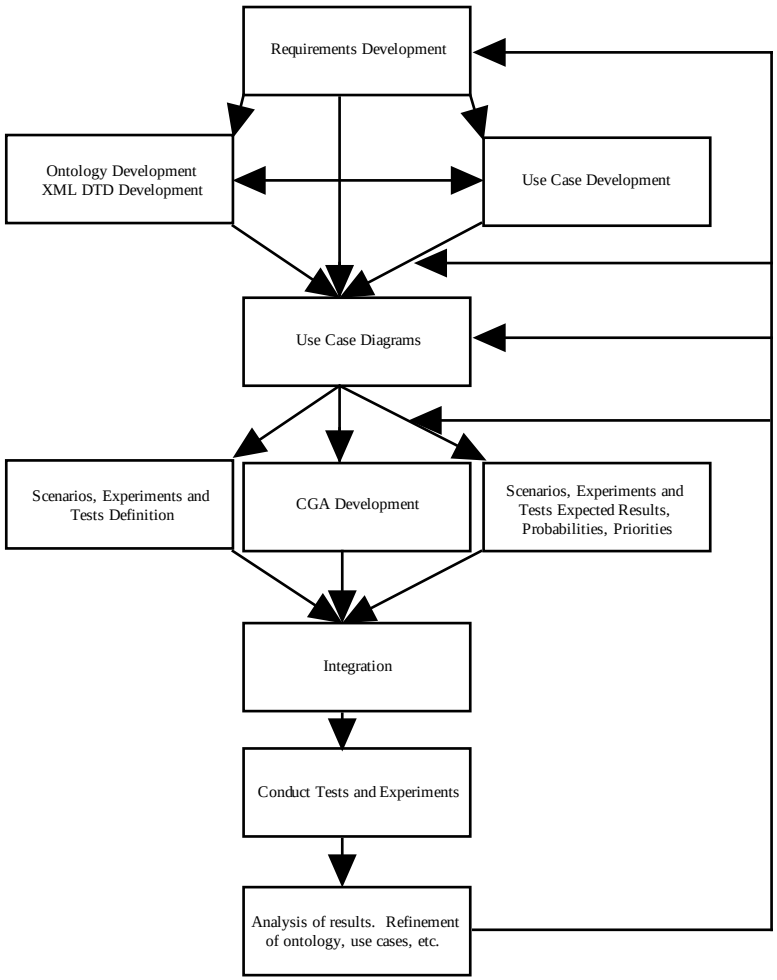
- **Two mutually supportive strategies**
  - Successive refinement and development of capabilities/implementation
  - Successive refinement and development of UML and XML descriptions
- **UML use cases identify what the CGA must do, required inputs, and minimal acceptable performance**
  - XML captures this behavior requirement in a machine readable format so that performance can be validated semi-autonomously
    - XML for annotations and knowledge base, helps refine behavior description
  - Convert from standard knowledge base representation to implementation before execution
- **Once execute CGA, measure its behavior against requirements, then**
  - Refine UML/XML behavior specifications to conform to uncovered requirements
  - Refine CGA software and knowledge bases so that they achieve required behaviors
  - Continue refinements until behaviors and documentation are sufficient and correct

IW OPFOR Design Process



- **UML Based**
- **Start with requirements**
- **Iterative, top-down approach**
- **Identify the use cases needed to satisfy the requirements**
- **Early focus on correctly defining the most abstract parts of the CGF**
  - **Selectively elaborate diagrams when design choices are complex**

Overall Methodology



- Requirements development begins process
- Parallel development of needed ontologies, DTDs and use cases
- Use case diagrams to document required performance and behaviors, XML for annotations(s)
  - One for each of the required set of behaviors for the CGA
- Parallel development of
  - Tests, scenarios, and experiments
  - CGA components
  - Required performance
- Integration of components
- Testing and analysis of cyber red team
- Refinement: components, use cases, DTDs, ontologies, knowledge bases, etc.
- Feedback

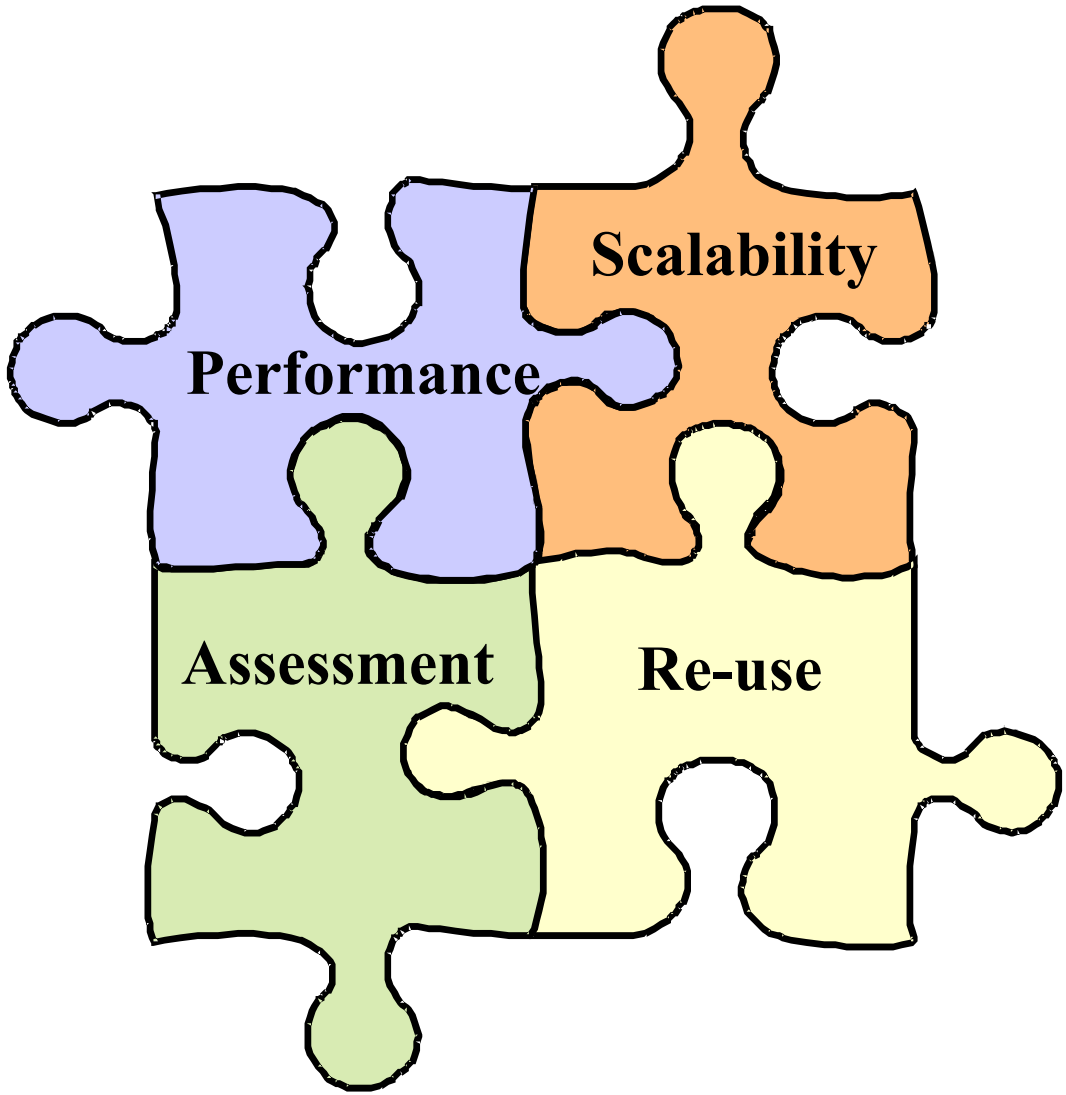
Overall Methodology (cont.)

- Need to identify each type of attack/exploit category early in process
  - Narrative description
- Mirror process for defense
- Convert each narrative into UML use case and sequence diagrams
- Parallel development and evaluation of overseer's console

Immediate Research Areas

- Tools to divide tasking and support human
- Workload Division
- Situation awareness/command&control console
  - Predictive cyberbattlespace awareness
- Hybrid decision-making capabilities
- Autonomous analysis capability and learning
- Development of defense and attack cases and documentation in XML/UML

Research Issues



Future Research Topics

- Further research
  - Decompilers
  - Disassemblers
  - Compilers
  - Watermarking resilience
  - Obfuscation
  - Debuggers
  - Multiprocessors
  - Cost assessment
  - Automatic developer logging and profiling
  - Software development methodology modification
  - Virtual machine attacks
  - Multiprocessors and coordinated network attacks
  - Benchmarks, metrics, and test suites
  - Data
    - Attack and analysis of attack on data

Conclusions and Future Work

- Increasing reliance upon information to maintain battlefield superiority makes it a target and requires better testing of defenses
- No good current capability, but have enabling technologies that can be exploited
- Discussed an approach to develop a cyber red team, IW OPFOR, that addresses the training and testing need for command forces
- Variety of research needs to make the vision a reality
  - Symbiosis between computer and human
  - Acquire knowledge and assemble IW OPFOR
  - Spectrum of technologies
- Need to develop metrics for cost benefit analysis
- Scenario development for IW OPFOR
- Ability to build the IW OPFOR exists, the need exists, the benefits are clear