Realistic and Affordable Cyberware Opponents for the Information Warfare BattleSpace

Martin R. Stytz, Ph.D.
Air Force Research Laboratory
Wright-Patterson AFB, OH 45431
mstytz@att.net,
martin.stytz@wpafb.af.mil

Sheila B. Banks, Ph.D.
Air Force Research Laboratory
Orlando, FL 32828
sheila.banks@afams.af.mil

Michael J. Young, Ph.D.
Air Force Research Laboratory
Wright-Patterson AFB, OH 45431
michael.young@afams.af.mil

Abstract

As military environments increase in the complexity, fidelity, scope, and number of participants, the reliance of the military upon information superiority to facilitate successful operations increases. In conjunction with this increase upon accurate and timely information the vulnerability of military forces to information attack also increases. Additionally, information management capability improvements inevitably increase the value of the information management networks and software; thereby, directly increasing the incentive for attacking or pirating the network and software capabilities. Therefore, as the information management capabilities of military forces increase, there is a corresponding need for improved security for the software and network systems and this need for improved security will increase as the value of the software and network systems increases. This improvement in information management capabilities must be accompanied by a corresponding increase in the ability to manage the protection of military information systems, which is a topic that has received scant attention.

In this paper, we address a portion of the information management protection problem by proposing an information warfare red team capability. This red team will enable standardized evaluation of defenses and allow command echelons to experience information warfare attacks. In the paper, we discuss the technologies that are emerging that enable the development of a cyber red team and a methodology for the development of the cyber red team. We motivate and highlight the need for the cyber red team for defense training and defensive systems evaluation, discuss the requirements for the information warfare cyber red team, and elaborate our vision for its need and capabilities within the information warfare, or cyber, battlespace. Given the state of technology, we argue that a symbiosis will be required to achieve a cyber red team and that there must be a corresponding dividing line between the human's responsibilities and the cyber red team's functionality. We discuss the developmental approach we foresee for the information warfare cyber red team and the testing and validation steps that are required in order to successfully complete each phase. The paper concludes with a brief summary and suggestions for future research.

1. Introduction

"Train the way you will fight" is the mantra for the United States' military and this philosophy has served the warfighter well as evidenced by the many successful operations executed around the world over the last decade. This need for realistic training has been recognized and proven by the warfighter and continues to be the beacon that guides US military training at the operator level and is pervasive in its aggressive application. However, this philosophy has not been as nearly as pervasive in command post and commander exercises, especially in the arena of information warfare. While there may be a variety of reasons for this

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE
JUN 2003

2. REPORT TYPE

3. DATES COVERED
00-00-2003 to 00-00-2003

4. TITLE AND SUBTITLE
Realistic and Affordable Cyberware Opponents for the Information Warfare Battlespace

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory,Wright Patterson AFB,OH,45431

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**41**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

unacceptable situation, one clear reason is the lack of realistic, cost effective, and formidable cyberwarfare opponents that can be used and re-used to create a realistic information warfare battlespace whose main characteristics are relatively constant from training experience to training experience. The seriousness of this shortfall is highlighted by the coming capabilities in networks and computing, which portend a time when warfighters will have access to and increase their dependence upon the unprecedented detail available concerning a situation within a battlespace. Recent technological advances, improvements in computer-generated forces, and research in information assurance and software protection, coupled with this increased dependence upon information, indicate that the information warfare battlespace will be a key theatre of conflict in future combat across the spectrum of combat. This new combat arena has many unique characteristics: the extreme speed with which events occur, nearly instantaneous change of attack vectors and attack, the high degree of technical expertise needed at all levels of command, the lack of metrics to measure the effectiveness of defense techniques, and the difficulty in developing situation awareness and mental models of the cyberbattlespace due to extremely rapid changes in the environment, the difficulty in achieving a level of prediction for cyberbattlespace activity, and the extreme susceptibility of the combatants and civilians to intended and unintended effects of the results of operations within the cyberspace battlespace. As a result of all of these factors, training is more difficult to perform than for other forms of commander training and access to real-world facilities is limited due to the potential for grievous and even irreversible harm. Paradoxically, the need for cyberbattlespace training appears to be increasing at all echelons of command. We propose addressing these needs by providing a powerful and comprehensive cyberbattlespace training environment that is robust and adaptive to the technologies and their employment in the cyberbattlespace arena. The key to achieving this vision is the development of high-fidelity cyberbattlespace opponents and computer-controlled (generated) actors that can function as a cyber red team that has a number of uses including the following: evaluation of defenses, execution of information warfare exploits against friendly force, implementation of opposing forces attack strategies for analysis, and development of new attack vectors based upon their experiences.

The impetus for the development of a cyberbattlespace red team, or information warfare opposing force (IW OPFOR), is the increasing reliance by US forces upon information dominance and the ever increasing value of US software assets. Application software and its data are increasing in their value because of the cost of acquiring the data, the cost of building the software, and the cost of assembling, testing, and validating ever more complex systems or for preparing command echelons for dealing with cyberbattlespace issues. Until recently, the need for application software protection and security has been addressed through efforts to provide security using network level resources or operating system level resources[5] and even some forms of software licensing. These meager security measures left software and data at risk. Obviously, traditional software protection schemes, licenses, network-centric defenses, and operating system-based defenses are inadequate to the task of protecting ever more valuable software and data in the cyberbattlespace. Even with the use of strong encryption technology, network-centric and operating system-based defenses have failed to secure applications from a variety of different types of attacks. The record and number of successful attacks demonstrates that only placing security technology in the network and the operating system, while necessary, is not sufficient to provide protection and security for the application software and data involved in military systems. This situation also demonstrates the need for the development of expertise and experience in the conduct of information warfare defensive operations by all command echelons. Application security, which is the field of security related to the security of application software and data, will become an ever more important part of the warfare environment as military environments and operations increase in their reliance upon information and as more participants become ever more closely linked via networks to form a powerful, information-based, military force.

However, before a even a minimally useful information warfare opposing forces (IW OPFORS) can be fielded for defense testing or command echelon training, a research and development effort is needed. Research targeted at advancing cyberbattle understanding, human behavior modeling, intent inferencing, information display, data mining, reasoning, and knowledge base expansion and re-use must be conducted. In this paper, we examine the enabling technological advances in these fields that would now permit the assembly of an autonomous cyber red team to conduct simulated cyberbattle and evaluate cyber defenses and discuss how these key technologies can be combined to develop a semi-autonomous force (SAF) information warfare red team. The envisioned red team can be used to generate and conduct information warfare exploits and provide a practical, controlled, safe, repeatable and challenging environment for commanders and all others who rely upon information assurance to conduct operations. An additional important result of this effort is that the improvement in the scientific assessment capabilities of cyberdefenses due to the repeatability of attacks and the resulting accumulation of statistical evidence of the effectiveness or ineffectiveness of a cyberdefense or cyberdefense suite and avenues for improvement will be possible. As a result of this research effort, the commanders' information warfare cyberbattlespace can increase in fidelity and all command echelons will be better prepared to face the information warfare and cyberbattlespace challenges that we will likely encounter in future conflicts.

In this paper, we discuss the development of an information warfare red team capability, the technologies that now enable the development of a cyber red team, and a methodology for the development of the cyber red team. We motivate and highlight the need for the cyber red team, discuss the requirements for the information warfare cyber red team, and further elaborate our vision for its need and capabilities within the information warfare battlespace. We discuss the symbiosis that is required to achieve a cyberwarfare red team and discuss the dividing line between the human's responsibilities and the cyber red team's functionality and capabilities. We present a preliminary developmental approach that can be used to achieve the information warfare cyber red team and the testing and validation steps that are required in order to successfully complete each phase. This paper is organized as follows. The next section contains an introduction to the nascent technologies that can enable the information warfare cyber red team. Section Three contains a discussion of the requirements that we foresee for the cyber red team. Section Four contains a description of the methodology and approach we propose for development and evaluation of the information warfare cyber red team in the information warfare battlespace. Section Five contains a summary and suggested research directions.

2. Background

In this section, we review the central technologies that are emerging and that enable the development of computer generated actors (CGAs) that can function as a powerful, effective, high fidelity information warfare red teams and opposing forces that can provide command echelons with experience in detecting, responding, and managing information warfare exploits. The technologies include improved software protection techniques, better modeling of human cognitive processes, better techniques for capturing and expressing knowledge, and better technologies for building systems that can, in conjunction with a human supervisor, execute cyberattacks.

2.1 Software Protection Technologies

Traditionally, information assurance and the security of a computation and its data have been provided by the network defensive systems and in the authentication mechanisms in the host operating system[26-39]. Despite intense and ongoing efforts to strengthen these two types of defensive systems they cannot assure the security of software and data on the host computer and as a result users place their application software and data at risk whenever they use a computer. Recently, the concept of information assurance has broadened from the traditional dyad of defensive systems to a triad, a triad that employs defensive systems embedded in the application software. The technological components of application software defense, also called software protection, are a mix of techniques whose individual and composite objective is to deny the pirate or intruder the capability to misuse, reverse engineer, tamper with, or steal application software or data. Software protection is the last ring of defense for application software and data, with the first two defensive rings being the protection technologies residing in the network resources and the other being the protection technologies residing in the operating system.

There are three chief software protection technologies whose steadily improving capabilities are enabling better software defenses and as a result the cyber red team must be able to detect, analyze, and disable these defenses in order to conduct effective attack exploits when conducting cyber red team operations. The first of these technologies is software watermarking. Software watermarking is used to protect the intellectual property contained in a program by embedding a secret message within the software. Watermarking a program is similar to adding a copyright notice to a textual document to assert ownership rights. A second technology is code obfuscation. Code obfuscation is used to protect a secret algorithm hidden in a piece of software by reorganizing the software such that the reorganization makes the software more difficult to read, understand, and reverse engineer. The secret can be of many forms, for example the overall design and structure of the software, important algorithms used in the software, or data (such as cryptographic keys) hidden within the software. The latter is, for example, common in digital rights management systems. The amount of protection afforded by code obfuscation depends upon the sophistication of the available obfuscation algorithms, the size and structure of the program being protected, the experience and determination of the attacker, and the power of the tools the attacker has at his disposal (such as decompilers, static analyzers, debuggers, etc).

When applying software protection technologies, the evaluation of the strength and effectiveness of software protection techniques require a well-defined threat-model for the software being protected. A threat-model describes the tools and techniques that an adversary is likely to employ and come in two major variations: manual attack models and automated attack models. Manual attack models assume that the software is inspected and modified by hand by a programmer skilled in reverse engineering techniques. Automated attack models, on the other hand, assume the use of tools that autonomously attack and break software protection schemes. To estimate the degree of protection achieved by a suite of software protection techniques, a variety of statistics associated with the program are typically computed. A few of the more important measures are the following: McCabe's cyclomatic measure[1], Halstead’s measure for number of operators and operands[2], Shyan’s object oriented code metrics[3], Henry and Kafura measure for information flow across classes and methods[4], and Harrison and Magel’s measure for nesting level[5]. While these and a few other measurements of software complexity are typically used as part of an estimate of the degree of protection achieved for a program, there are currently no accepted metrics by which software protection algorithms should be evaluated. For example, most papers on software watermarking do not empirically or theoretically evaluate these algorithms against attacks, nor do they specify what attacks the technique is designed to counter. Based on experience to date, however, a software watermarking algorithm can be usefully evaluated according to the following criteria: 1) Data Rate: What is the ratio of size of the watermark that can be embedded to the size of the program; 2) Embedding Overhead: How much slower/larger is the watermarked application compared to the original; 3) Resilience Against Manual Attacks (stealth): Does the watermarked program have statistical properties that are different from typical programs; 4) Resilience Against Semantics-Preserving Transformations: How well will the watermark survive transformations such as code optimization and code obfuscation? If not, what is the overhead of these transformations and is it unacceptable to the attacker; 5) Resilience Against Collusive Attacks: Given two or more different copies of the same application can the location of the fingerprints be determined; and 6) False Positive Rate: Given a random value to the watermark would the random value be determined to be a true watermark.

2.2 Knowledge Representation

In addition to improvements in software protection technologies there has been a corresponding increase the knowledge and understanding used by an attacker when executing an attack exploit against software or computer network security capabilities[8-25]. This knowledge includes taxonomies of attack exploits, taxonomies of defensive techniques and technologies, attack vectors and avenues, and improved understanding of how to structure defenses. Additional knowledge that has been acquired includes vulnerability categories, attack categories, network intrusion detection taxonomies, vulnerabilities that arise from architecture and design choices as well as implementation choices and strategies, development of some metrics and measures for measuring security capabilities, and development of techniques for vulnerability analysis. Improvements in our information warfare understanding and knowledge also now give us the capability to determine network systems vulnerability, constructing network defenses in depth, assembling defenses that are adaptive and capable of providing intrusion detection, methodologies for developing and analyzing operating system defenses, and methodologies for developing and analyzing network system defenses. We also now have a large and ever increasing library of exploits that can provide insight into attack strategies and tactics as well as insight into hacker methodologies and approaches employed to hijack computer systems, networks, and software applications. In addition, we are also beginning to understand how to assemble coordinated defenses, test and determine the effectiveness of defensive systems at all levels, conduct rapid analysis of attacks being executed, determine status of defenses and types of attacks being executed, tracking hackers and crackers, understanding types of cyberbattlespace attacks, and gathering and analyzing attack forensics at all levels across the entire cyberbattlespace spectrum. Further advances are reported regularly in *IEEE Security and Privacy* magazine and hold out the promise for improved defenses and understanding of attacks upon software and network systems as well as means for insuring the privacy and security of information. Clearly, we have accumulated a broad an ever-expanding body of knowledge about cyberbattlespace attacks. These improvements in basic understanding of information warfare attacks and defenses call for improved means for organizing the information and making it available for computer-controlled systems that can execute a variety of attacks and aid in the analysis of the effectiveness of defenses. To assemble an effective SAF cyberwarfare red team, this knowledge must be organized in an expandable and powerful manner. Two key technologies that can aid in the knowledge organization effort are the eXtensible Markup Language and the Unified Modeling Language.

The eXtensible Markup Language (XML) is a meta-language that supports the customized definition of the components of the language (syntax, data types, vocabulary, and operators) needed to support the interchange of data for a particular application environment[7]. Each application-specific definition is contained within a Document Type Definition (DTD). The DTD describes a vocabulary and syntax for the data to be transmitted. XML provides a basis for the development of data transmission formats that are transmitter and recipient independent and that are completely self-describing and self-contained. In our opinion, XML permits a deeper level of specification by providing data definitions and formats that are flexible, independent, and comprehensive.

The Unified Modeling Language (UML)[8] plays a key role in our approach to assessing and evaluating cyber red team behavior. UML is a standardized graphical language that can be used to develop and compose blueprints (architecture specifications) of software systems. The UML documents the conceptual and physical representations of a system and permits modeling and visualization of a system from a variety of viewpoints. UML provides a capability for capturing the knowledge about a subject and for expressing the knowledge. UML contains a large and useful set of predefined modeling and documentation constructs and supports custom representations of information through its inherent mechanisms for extensibility. UML also provides constructs for specifying and documenting the building blocks and components of a system and for documenting a complete system whether it is a federate or federation.

2.3 Software Technologies

Another set of crucial technological advances that enable the development of the cyber red team at this time is the improvement in software technology. Allow us to briefly review some of the more central technologies here. Clearly, the cyber red team's software must be extensible, adaptable, and flexible. There are several technologies that will be important to these needed capabilities software components, frameworks, and gauges. A prime goal for the software development community has been enabling the assembly of complex software systems from simpler *software components* to enable software reuse and cost-effective maintenance of legacy software systems. A software component is a nontrivial, nearly independent, and replaceable part of a system that fulfills a clear function in a well-defined architecture. Components typically rely upon a separate infrastructure for interface definitions, message passing, and data transfer. A *framework* performs these functions. A framework is a software skeleton for an application that can be customized. Frameworks provide a support infrastructure, interface standards, and execution scaffolding for components and objects. Conceptually, a framework serves as a backplane that interconnects the components and objects that form the application, in our case the cyber red team. Frameworks are an ideal conceptualization for enabling re-use and experimentation because they allow functionality to be captured at multiple levels of abstraction and enable re-use at multiple levels of encapsulation. Finally, *software gauges* are constructs used to enable rapid assembly of systems. A software gauge is a display system for data collected using a software probe. A software probe collects information by intercepting data in transit between components/objects in a system. A software gauge allows a designer or implementer to view the configuration of a system at multiple levels of abstraction and to conduct experiments with different configurations. Gauges can be used to assess the suitability of two components for interaction before, during, and after software insertion and can also help the designer and the cyber red team manager determine if the cyber red team's knowledge is being used correctly and to maximum advantage. These are not all of the advances that we can exploit when constructing the cyber red team, *IEEE Software* has regular reports and papers that describe, discuss, and evaluate improvements in software technology, one such technology being extreme programming.

2.4 Human Behavior Representation

Lastly, but of crucial importance, are the improvements in our abilities to construct computer-controlled systems that can faithfully emulate human performance and behaviors with an ever increasing degree of fidelity (as discussed in the series of *IEEE Simulation Interoperability Workshops* and the series of *IEEE Conferences on Computer-Generated Forces and Behavior Representation*, as well as *IEEE Intelligent Systems* and *AAAI Magazine*.) As the research reported in these and other human behavior journals and conferences attests, our ability to gather, categorize, and employ knowledge about a military domain to construct computer-controlled entities that are similar to and difficult to distinguish from humans is improving. This human behavior and intent modeling capability provides an important technological component of the cyber red team. The improving capabilities for human behavior modeling and the ongoing projects for human intent inferencing indicate that our ability for accurately portraying any type of adversary in the cyberbattlespace will continue to improve and that over time exploits will become more powerful, human-like, and require less human oversight to develop and execute. These technologies for modeling human activity and insuring that the computer-controlled adversary is unpredictable but nevertheless faithfully complies with attack knowledge and hacker/attacker tactical concepts indicates that we should construct an automated cyber red team that can employ the knowledge about cyberbattlespace attack and defense in an accurate, high fidelity manner. And, the cyber red team should be expandable and modifiable in its software as well as knowledge base so that advances in portraying human activity using computers can be readily inserted into the cyber red team. Also, and of major importance, the performance of the cyber red team should remain consistent given a set of knowledge, thereby permitting the scientific measurement of the effectiveness of cyber defenses and the degree of improvement obtained when a new defense is developed. The technology for development of a cyber red team is emerging and will enable us to construct a cyber red team whose behavior and capabilities are nearly indistinguishable from a human red team, especially when the cyber red team is managed by a human in real-time during executing of an exploit, and whose capabilities for conducting exploits will improve over time since we can readily add knowledge and capabilities to the cyber red team.

3. IW OPFOR Requirements

While it is clear that US military command and control forces will benefit from realistic information warfare training in the information warfare battlespace, the question of the characteristics and capabilities of the simulated opponents, the information warfare semi-autonomous force (SAF) cyber red team, naturally arises. Firstly, the cyber red team must be able to employ any form of reasoning and have a capability for adaptive learning in order to provide realistic and unpredictable exploits, attacks, and tests against friendly command and control structures. The cyber red team must be able to autonomously analyze the results of its actions and modify its behavior in response to the results of the analysis so as to maximize its ability to provide realistic tests of friendly command and control defenses at minimum cost and at maximum speed. However, we cannot expect the cyber red team to learn a wide variety of productive attacks on its own nor should we expect it to be able to provide specific types of exploits on demand, as it were, in order to provide a specific training experience or command

and control defenses test. Therefore, the cyber red team must be able to be readily programmed with new plans and forms of attack as well as specific actions that form part of an attack. The cyber red team will also require human assistance and management. Any portion of an exploit must be visible to the human monitor (or team manager) and be able to be changed so that human monitors of cyber red team operations can alter the cyber red team's activities in order to provide the creativity, deep insight, and intuition that can not currently be provided in computer-generated forces (and hence the need for a semi-autonomous cyber red team). This symbiosis of human and machine capabilities can lead to a powerful capability for conducting exploits, devising variations or even new exploits, and even analyzing the results of the cyber red team's activities during an exploit. To support its own inherent analytic capabilities and its human operators, there must be automatic logging of attacks, actions, and responses and there must be a methodology for automatic scoring/assessment of attacks and exploits. To insure precise and accurate communication between the SAF information warfare cyber red team and its operators, as well as to insure that the operators can precisely control the SAF, an ontology is needed in order to provide a common terminology and frame of reference. The ontology will provide the precise communication between the cyber red team and its human operators that is required. Finally, the information warfare SAF cyber red team must be able to conduct multiple simultaneous, independent, coordinated attacks against friendly forces within the information warfare battlespace. This requirement is particularly important because it opens up the possibility for conducting advanced, new types of attacks that can stress friendly defenses in a manner that has not been experienced in the real world and can help prepare defenses and operators for unexpected types of exploits within the information warfare cyberbattlespace.

There are several technologies that must be the focus of the research effort in order to achieve the desired characteristics and qualities for the cyber red team. Briefly, one research effort must provide the tools that are needed to successfully divide and support the tasks to be performed by the SAF monitor and the SAF cyber red team, thereby achieving a symbiosis between the computer system and the human managers. In general, the division of workload questions that must be addressed concern the SAF manager needs, the nature of SAF manager decision making, and the technologies needed by the SAF cyber red team in order to conduct rapid, effective exploits against friendly forces in command and control training situations and when assessing new defenses or defensive combinations. Clearly, another critical component of the overall information warfare SAF cyber red team system is a command, control, and situation awareness console for use by the SAF cyber red team manager. Indeed, given the speed with which activity occurs in the cyberbattlespace, predictive cyberbattlespace awareness appears to be necessary and should be an early focus of a sustained research program. A third research focus should be the development of a hybrid decision-making capability for the SAF cyber red team. These decision-making capabilities include the assembly, categorization, cross-referencing, and analysis of a broad suite of knowledge bases that will be needed to enable the SAF cyber red team to engage in a broad variety of exploits against network and software defenses. A fourth technology focus should be a capability for the SAF cyber red team to analyze its actions, to change its operations, and to learn from its experiences all under the guidance of the SAF manager.

4. Cyber Red Team Development and Evaluation Approach

Our approach to cyberbattlespace SAF red team development acknowledges the difficulty of fully validating any software product, and so is experimentally based and employs two main strategies. The first strategy is successive, iterative refinement and development of technical capabilities in accordance with their definition as specified using UML and XML. The second strategy is successive refinement of the UML and XML specifications based upon experimental outcomes and analysis of technical activities. These two strategies are supported by UML use cases that identify usage of the cyber red team, expected cyber red team performance, sequences of uses, the required inputs for each case, and the minimum percentage of correct responses that the cyber red team must achieve in order for the cyber red team to be considered to be performing acceptably.

The overall process that we have developed is specified in Figure 1. As is normal, the first step in the process is the identification of specific requirements for the cyber red team. The identification of requirements is crucial because in the later stages of cyber red team development these requirements form the foundation for the specification of desired cyber red team performance. The second phase contains two steps that occur in parallel and reinforce each other. One step is the development of the ontology and XML data descriptions (DTDs) indicated by the requirements. The other step is the development of use cases based upon the requirements. With the use cases, ontology, and XML DTDs in-hand and in light of the requirements for the cyber red team, the use case diagrams for the cyber red team are developed. The fourth step is the development of the scenarios, tests, and experiments needed to evaluate the cyber red team (using interaction diagrams), development of the cyber red team's components (knowledge bases, human behavior models, software, class diagrams, component diagrams, etc.), and development of the minimum acceptable performance that the cyber red team must exhibit in the scenarios, tests, and experiments. The fifth step is the integration of the components for the cyber red team. The sixth step is the execution of the experiments defined to test the cyber red team. The seventh step is analysis of the experiments and refinement of the ontology, DTDs, use cases, and other components of the cyber red team and feedback into earlier development steps for the cyber red team.

Allow us to briefly highlight some of the key aspects of the cyber red team knowledge acquisition and development process that we propose. The process begins with a list of each type of attack that was identified during analysis, which forms the basis for the requirements. The analysis should also develop a narrative description of the threat models to be considered. One narrative description should be written for each type of attack (attack case) that is uncovered during the analysis process and should be related to the threat models. A narrative description of each attack case describing the objective, required attacker functionality, required resources, and mode for each attack should be developed as well. Each attack case should be documented as an UML-based use case. All notations for each use case should be written in XML in order to insure that the system is as open and expandable as practicable. Necessary XML DTDs to support annotations for the attack cases should be written, the elements for each DTD shall be defined, the tags for each DTD should be defined, the attributes (as necessary) for each element should be defined, and the entities shall be defined in each DTD. One set of UML-based sequence diagrams and statechart diagrams for each identified attack case that identifies and defines the sequence of activities that an attacker would employ for the attack case should be developed.

The knowledge acquisition and development process for defense mirrors that for attack. The process begins with a list of each type of defense that was identified during analysis, which forms the basis for the requirements. The analysis should also develop a narrative description of the threat models to be considered. One narrative description should be written for each type of defense (defense case) that is uncovered during the analysis process and should be related to the threat models. Each defense case should be documented as an UML-based use case that should describe and define the defensive activities that a defender could use to defend against each attack case. All notations for each use case should be written in XML. Necessary XML DTDs to support annotations for the attack cases should be written, the elements for each DTD should be defined, the tags for each DTD should be defined, the attributes (as necessary) for each element should be defined, and the entities should be defined in each DTD. Finally, a set of DTDs for XML that describe the format(s) for the knowledge used to execute each attack and each defense should be developed, including necessary and appropriate elements, attributes, and entities.

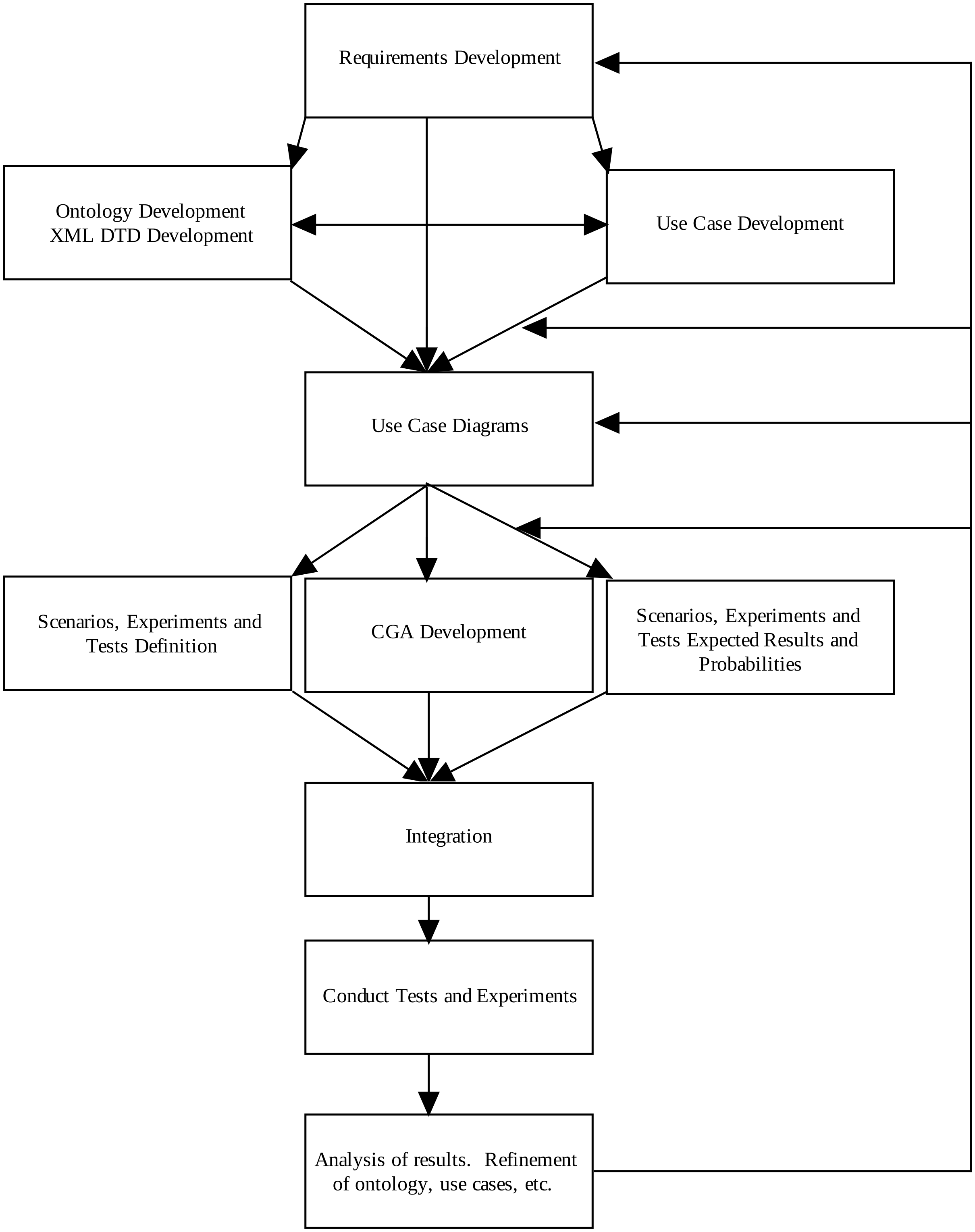

Requirements Development
Ontology Development
XML DTD Development
Use Case Development
Use Case Diagrams
Scenarios, Experiments and Tests Definition
CGA Development
Scenarios, Experiments and Tests Expected Results and Probabilities
Integration
Conduct Tests and Experiments
Analysis of results. Refinement of ontology, use cases, etc.

Figure 1: Proposed Methodology

5. Conclusions and Research Suggestions

In conclusion, as military environments increase their use of information to achieve dominance in the real world battlespace and information capabilities become more widely used;

the sensitivity of their data and their vulnerability to compromise will increase as will the motivation for attacking military software and networks to gain access to the information embedded within. In the past, the protection provided by network and operating system based security capabilities was adequate. However, network and operating system based security technologies can no longer protect military information management and control capabilities and they are not able to stave off compromise and protect the increasingly sensitive information contained within military environments. We are becoming ever more reliant upon information for operations in the real world battlespace, but our capabilities for defense in the cyberbattlespace have not been developed to the degree needed to secure information. To address this shortfall, we propose the development of a cyber red team that can conduct exploits against defenses to evaluate their effectiveness in a standardized manner and that can also be used to prepare command echelons for activity in the cyberbattlespace.

In this paper, we discussed a number of items related to the cyberbattlespace and the cyber red team. We discussed the needs for the SAF cyber red team and discussed our vision for its operational capabilities within the information warfare or cyberbattlespace. We discussed the requirements for the SAF cyber red team, presented an overview of the technologies that are just now enabling the development of the cyber red team, the potential for using UML and XML to support the research effort, and attempted to illuminate desired capabilities and activities. We discussed our view of the symbiosis that is required to achieve an effective cyber red team. We also discussed our vision for the information warfare SAF cyber red team manager's command and control console. Finally, we discussed the developmental approach that we foresee for the information warfare SAF cyber red team.

There are many other research questions that must be addressed, one of which is the cost of protection as related to its benefits. Cost can be broken down into three parts, efficiency cost (what is the performance penalty of a technique or combination of techniques), implementation cost, and maintenance cost (ie., what effect upon software maintenance does the technique incur?). Another need is for the development of a spectrum of protection technologies and evaluation of their associated costs and benefits so that developers can make informed decisions about the degree of protection needed for software based upon the sensitivity of the software and the costs involved in applying the indicated application techniques. Another major research question is improved protection metrics. Some metrics that must be refined further are <u>resilience</u> (a measure of how difficult is it to defeat a technique), <u>obscurity</u> (a measure of how difficult it is to determine if a particular protection technique has been employed, aka stealthiness), and <u>expected longevity</u> (a measure of the length of time that a protection technique will afford a worthwhile degree of protection) of each protection technology, the costs and benefits of different mixtures of protection techniques, and the level of protection required for a given application and military environment. A further research question is the development of a methodology for determining and assessing the importance of cyber red team requirements and then implementing and evaluating them in priority order. A future research need is the development of scenarios for the cyber red team, scenarios that we believe should be developed automatically and validated automatically, or to the maximum extent possible in order to minimize cost and improve the capability of the cyber red team to execute new exploits as they are devised in the real world. This is a difficult research issue and will require long-term research commitment to achieve a basic but useful capability. Research in automatic scenario generation is already underway for simulation systems, and we can possibly bootstrap our efforts by using their work as a foundation for a corresponding cyber red team capability. Finally, we must insure that we develop scalable systems, the cyber red teams must be able to execute multi-opponent exploits, scalable exploits, and exploits against computer and network systems across the entire spectrum of computing power and network capability.

We believe that the foundation to successfully address these key research questions has been laid even though some work in these areas remains. We now have the technologies in hand to start the development of the cyber red team and can begin to reap the benefits of a standardized, computer-controlled evaluation of cyber defenses and the preparation of command echelons for the coming cyberbattlespace.

References

1. McCabe, T. J. (1976). "A Complexity Measure." *IEEE Transactions on Software Engineering*,2(4), pp. 308-320, December
2. Halstead, M. H. (1977) *Elements of Software Science*. Elsevier North-Holland.
3. Shyam R. Chidamber and Chris F. Kemerer. (1994) A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, vol. 20, no. 6, pp. 476- 493, June.
4. Henry, H and Kafura, D. (1981) "Software Structure Metrics Based On Information Flow,".*IEEE Transactions on Software Engineering*, vol. 7, no. 5, pp. 510-518, September.
5. Harrison, W.A. and Magel, K.I. (1981) A complexity measure based on nesting level. *SIGPLAN Notices*, vol. 16, no 3, pp. 63-74.
6. St. Laurent, S.; St. Laurent, S.; & St. Laurent, S. (1999) *XML: A Primer*. 2nd edition. IDG Books Worldwide: Foster City, CA.
7. Booch, G.; Rumbaugh, J.; and Jacobson, I. (1999) *The Unified Modeling Language User Guide*, Addison Wesley, Reading, MA.
8. Abbott, R. P., Chin, J. S., Donnelley, J. E., Konigsford, W. L., Tokubo, S. and Webb, D. (1976) A. "Security Analysis and Enhancements of Computer Operating Systems," *NBSIR 76–1041, Institute for Computer Sciences and Technology, National Bureau of Standards*, April..
9. Aslam, T. (1995) "A Taxonomy of Security Faults in the Unix Operating System." *Computer Science*. West Lafayette, Indiana, Purdue University.
10. Bisbey, R. and D. Hollingworth (1978). *Protection Analysis: Final Report*. Marina del Ray, California, University of Southern California.
11. Bishop, M. (1995). *A Taxonomy of UNIX System and Network Vulnerabilities*. Davis, California, University of California.
12. Bishop, M. and Bailey, D. (1996) "A Critical Analysis of Vulnerability Taxonomies," *Proceedings of NIST Invitational Workshop on Vulnerabilities*, July.
13. Bloom, B.S. (Ed.) (1956) *Taxonomy Of Educational Objectives: The Classification Of Educational Goals: Handbook I, Cognitive Domain*. New York ; Toronto: Longmans, Green.
14. Carver, C. A. (2000). "An Intrusion Response Taxonomy and its Role in Automatic Intrusion Response." *IEEE Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY, IEEE.

15. Cheswick, W. R. and S. M. Bellovin (1994). *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, MA, Addison-Wesley Publishing Company.

16. Cohen, F. B. (1995). *Protection and Security on the Information Superhighway*. New York, NY, John Wiley & Sons.

17. Howard, J. D. (1997). "An Analysis of Security Incidents on the Internet 1989-1995," Chapter 6: A Taxonomy of Computer and Network Attacks." *Computer Science*, CERT.

18. Krsul, I. (1998). "Software Vulnerability Analysis," *Computer Science*. West Lafayette, IN, Purdue University.

19. Landwehr, C. E., Bull, A. R., et al. (1994). "A Taxonomy of Computer Program Security Flaws," *CM Computing Surveys* , vol. 26, no. 3.

20. Lindquist, U. and E. Jonsson (1997). "How to Systematically Classify Computer Security Intrusions," *1997 IEEE Symposium on Security and Privacy*, Oakland, CA, IEEE.

21. Neumann, P. G. (1978). "Computer Security Evaluation," *1978 National Computer Conference*, Arlington, VA, *AFIPS Conference Proceedings*.

22. Perry, T. and P. Wallich (1984). "Can Computer Crime be Stopped?" *IEEE Spectrum*, vol. 21, no. 5.

23. Russell, D. and G. T. Gangemi (1991). *Computer Security Basics*. Sebastopol, CA, O'Reilly & Associates, Inc.

24. Stallings, W. (1995). *Network and Internetwork Security Principles and Practice*. Englewood Cliffs, NJ, Prentice Hall.

25. Viega, J.; Kohno, T.; Potter, B. (2001) "Trust (and Mistrust) in Secure Applications," *Communications of the ACM*, Vol. 44, No. 2, pp. 31-36, February.

26. Alexander, I. (2003) "Misuse Cases: Use Cases with Hostile Intent," *IEEE Software*, vol. 20, no. 1, January, pp. 58-66.

27. Amoroso, E.G. (1994) *Fundamentals of Computer Security Technology*. Prentice Hall: Englewood Cliffs, NJ.

28. Collberg, C.; Thomborson, C.; and Low, D. (1998) "Manufacturing Cheap, Resilient, and Stealthy Opaque Constructs," *Principles of Programming Languages 1998, POPL'98*, San Diego, CA, January.

29. Denning, D.E. (1999) *Information Warfare and Security*, Addison-Wesley: Reading, MA.

30. Garfinkel, S. and Spafford, G. (1991) *Practical Unix Security*. O'Reilly & Associates: Sebastopol, CA.

31. Gollmann, D. (1999*) Computer Security*. Wiley: Mew York.

32. Howard, M. and LeBlanc, D. (2002) *Writing Secure Code*. Microsoft Press: Redmond, Washington.

33. Jalal, F. and Williams, P. (1999) *Digital Certificates*: *Applied Internet Security*. Addison-Wesley: Reading, MA.

34. National Security Council. (1999) *Trust in Cyberspace*. National Academy Press: Washington, DC.

35. Schneier, B. (1996) *Applied Cryptography*, John Wiley and Sons: New York.

36. Stallings, W. (1999) *Cryptography and Network Security: Principles and Practice*. Prentice Hall: Upper Saddle River, NJ.

37. Summers, R. (1997) *Secure Computing: Threats and Safeguards.* McGraw Hill: New York.

38. Shrobe, H. (2002) "Computational Vulnerability Analysis for Information Survivability," *AI Magazine*, vol. 23, no., 4, Winter, pp. 81-91.

39. Waltz, E. (1998) *Information Warfare: Principles and Operations*. Artech House: Norwood: MA.

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

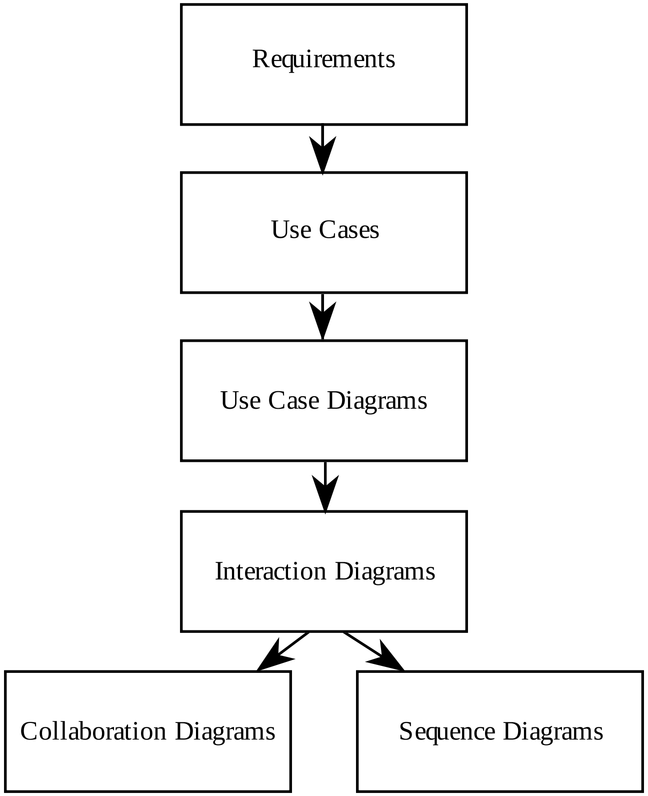
Cyber Red Team Requirements

- 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

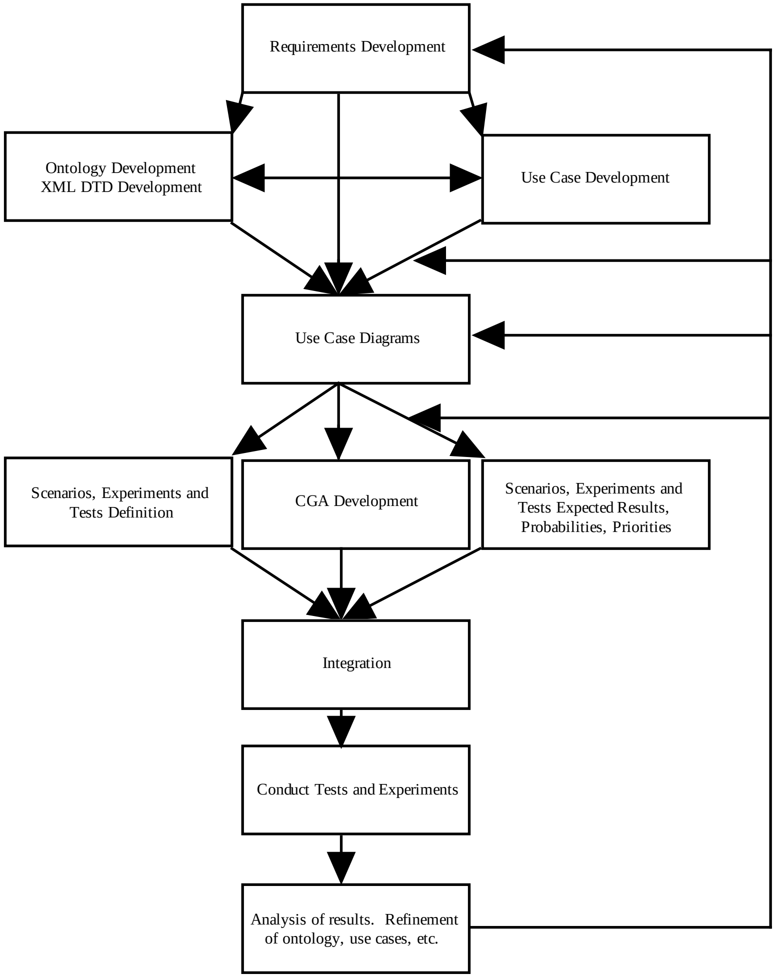
- Two mutually supportive strategies
  - Successive refinement and development of capabilities/implementation
  - Successive refinement and development of UML and XML descriptions
- UML use cases identify what the CGA must do, required inputs, and minimal acceptable performance
  - XML captures this 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
Ontology Development
XML DTD Development
Use Case Development
Use Case Diagrams
Scenarios, Experiments and Tests Definition
CGA Development
Scenarios, Experiments and Tests Expected Results, Probabilities, Priorities
Integration
Conduct Tests and Experiments
Analysis of results. Refinement of ontology, use cases, etc.

- 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

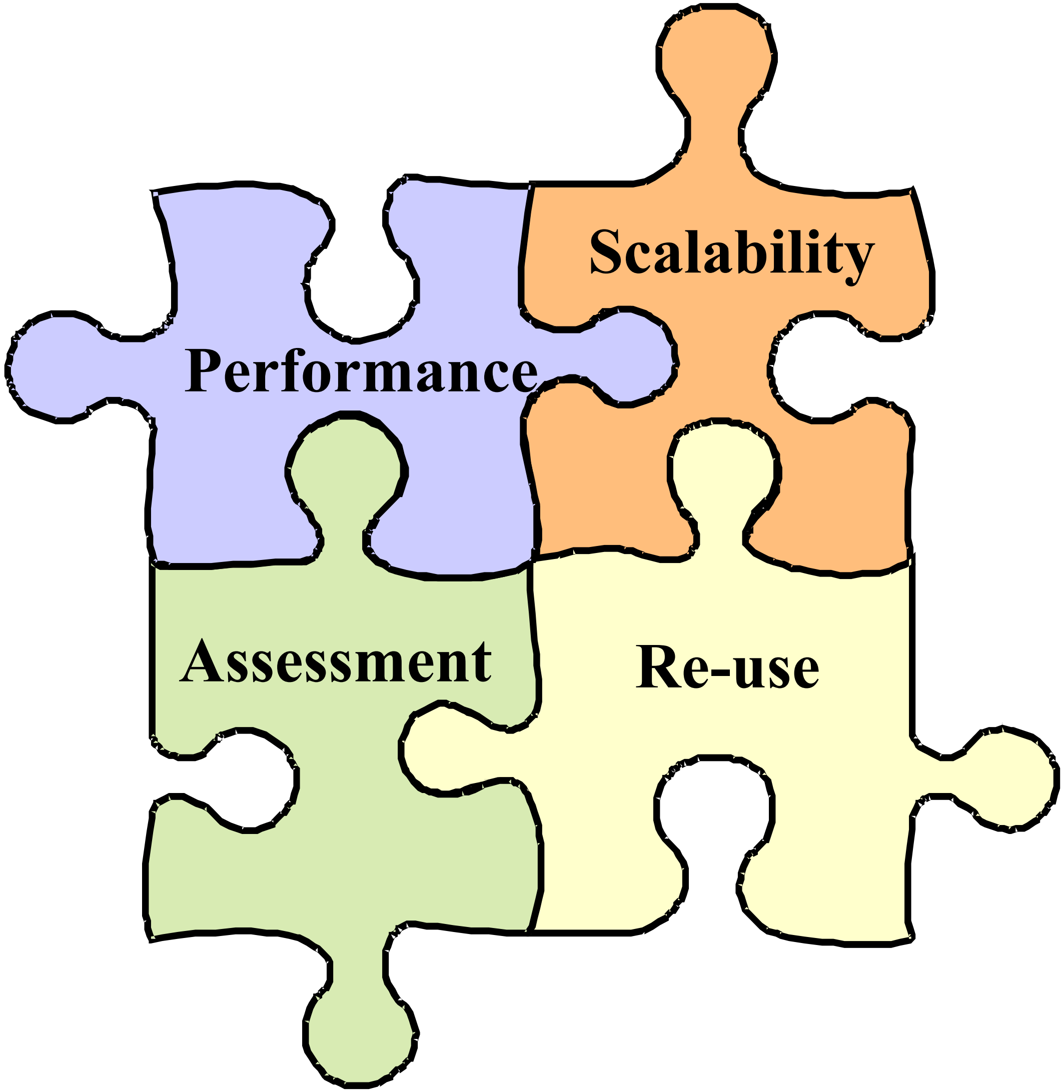
Overall Methodology (cont.)

- 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