Red Teaming of Advanced Information Assurance Concepts

Bradley J. Wood
Red Team Program Manager
Distinguished Member of Technical Staff
Sandia National Laboratories
bjwood@sandia.gov

Ruth A. Duggan
Red Team Project Leader
Senior Member of Technical Staff
Sandia National Laboratories
rduggan@sandia.gov

Abstract

Red Teaming is an advanced form of assessment that can be used to identify weaknesses in a variety of cyber systems. It is especially beneficial when the target system is still in development when designers can readily affect improvements. This paper discusses the red team analysis process and the author's experiences applying this process to five selected Information Technology Office (ITO) projects. Some detail of the overall methodology, summary results from the five projects, and lessons learned are contained within this paper.

1 Introduction

Red Teaming is an advanced form of information surety assessment. This approach is based on the premise that an analyst who attempts to model an adversary can find systemic vulnerabilities in an information system that would otherwise go undetected.

Sandia National Laboratories Information Design Assurance Red Team (IDART) has developed a process for performing these assessments. This methodology is derived from the surety design, evaluation, and vulnerability assessment processes used at Sandia National Laboratories for the US Department of Energy.

The IDART methodology combines assessment techniques that were previously used for studying high-consequence systems such as nuclear weapons, weapon use control systems, information systems, and nuclear reactor designs. That core methodology is extended to cyber systems by research, development, and application of information design assurance techniques and tools.

The goal of red teaming is to provide feedback to designers for improving the system under scrutiny. Timely red team analysis provides awareness of weaknesses so that risk mitigation options can be explored prior to system deployment. More information about IDART can be found at the web site at the address: http://www.sandia.gov/idart

2 Approach

The IDART analysis is based on certain assumptions:

- The threat exists.
- Diversity is essential for a thorough analysis.
- Critical Success factors and performance metrics can be identified.
- Multiple viewpoints provide a range of possible attacks.
- Vulnerabilities exist.

2.1 Adversary Model

A red team is a model adversary. The type of adversary that a given red team models depends on the perceived threat and the research sponsor's goals and objectives.

For this work, the Baseline Adversary is a small nation state with political or military objectives [1]. This adversary is modeled using these assumptions:

- The adversary is well funded. The adversary can afford to hire consultants or buy other expertise. This adversary can also buy any commercial technology. These adversaries can even afford to develop some new or unique attacks.
- This adversary has aggressive programs to acquire education knowledge in technologies that also may provide insider access.
- This adversary will use classic intelligence methods to obtain insider information and access.
- This adversary will learn all design information.
- The adversary is risk averse. They will make every effort to avoid detection.
- This adversary has specific goals for attacking a system.
- This adversary is creative and very clever. They will seek out unconventional methods to achieve their goals.

Sophisticated national technical means are the only resources unavailable to this class of adversary. This adversary was originally developed from theoretical models. However, recent stories in the press and elsewhere indicate that this adversary may actually exist [2] [3].

2.2 Team Building

Red teams are formed with the assumption that diverse teams deliver the best results. This process is illustrated in Figure 1. Here, individual red teams are formed for each project or system under scrutiny. These teams consist of a core team of information system analysts. Consultants, experts, and other analysts, as needed, augment this core team.

Not all red team members are information systems analysts. Additional expertise is sought, depending on the system under consideration. For example, a red team examining a biological and chemical agent detection system could include experts on biological and chemical warfare agents, physicians, and even meteorologists. A red team that evaluates a petroleum storage system might include petroleum engineers as well as information systems analysts.

2.3 Assessment Process

IDART uses the same basic process for each red team assessment. This process is illustrated in Figure 2.

Source Information – Each assessment begins with a rigorous attempt to gather all source information on the system of interest. This typically includes, but is not limited to these types of data:

- Design & development documents.
- Test requirements and results documents.
- Network design, development, and configuration documents.
- Results from various forms of fieldwork. This could include live on-site network discovery and active network reconnaissance.
- Software quality assurance documents.
- Source code.
- Interviews with developers.

Formal Understanding – The purpose of gathering this information is to develop a formal understanding of the system of interest. This formal understanding consists of the following:

Description – a comprehensive system-level description of the information system of interest;.

Objective Purpose – a thorough understanding of the objective purpose of the system of interest, including its mission and relevance in the defender’s enterprise;. and

Critical Success Factors – a consensus opinion on the critical success factors for both the system of interest and the red team.

Multiple Formal Views – As time and resources permit, the Sandia IDART tries to understand the system better by developing five views of the system:

- A functional/conceptual/logical view of the system to answer the question, “How does it work?” is developed.
- The next view of the system encompasses physical and spatial understanding of what things exist and where they are.

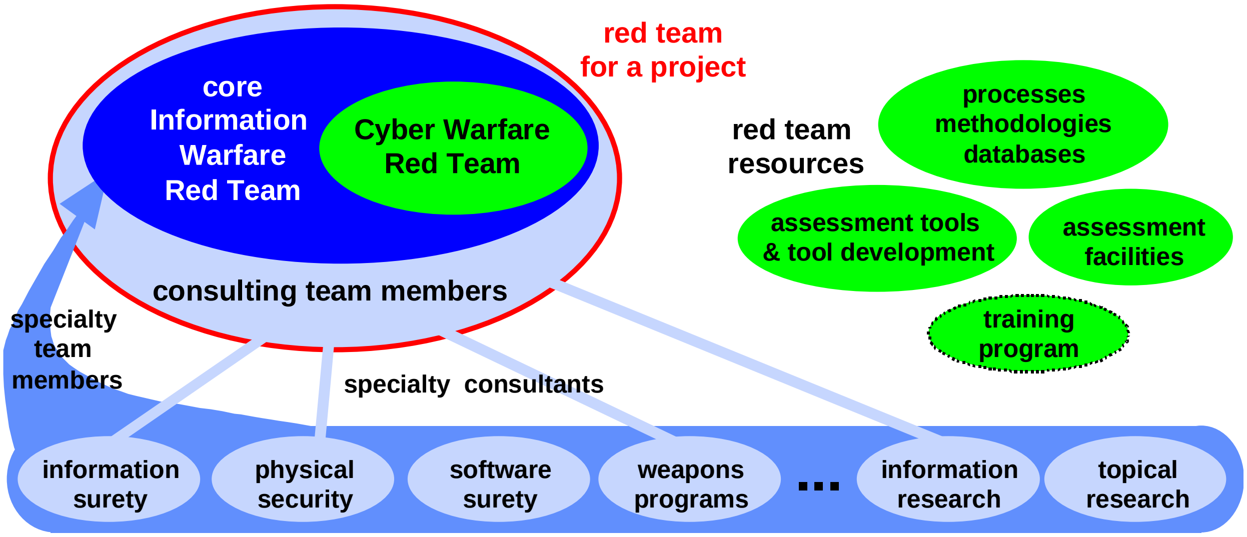

Figure 1 -- Red Team Development Process

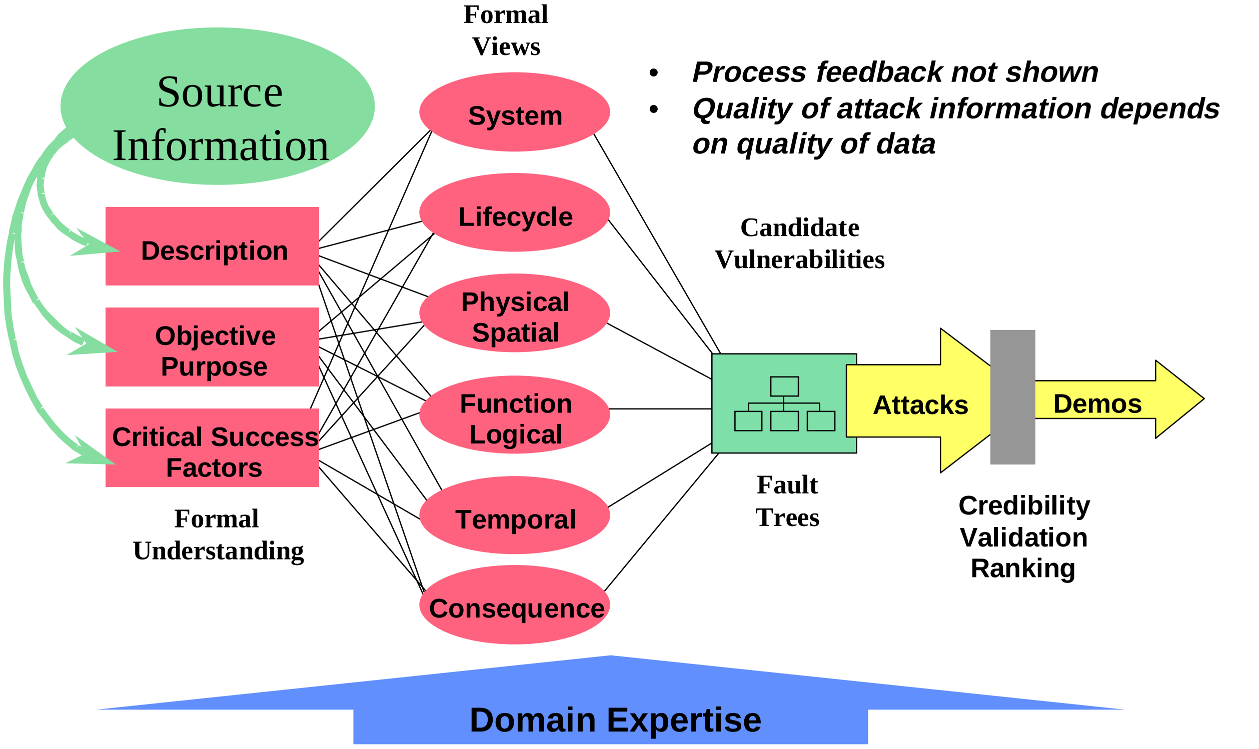

Figure 2 – Assessment Process

- A temporal understanding of the many possible sequences of events is explored.
- If appropriate, a view of the system lifecycle, including where and how it is designed, built, enhanced, modified, used, and repaired is also developed.
- Finally, a consequence-based view is used to determine what can go wrong.

Candidate Vulnerabilities – The red team then institutes a series of brainstorming sessions among the red team members to develop a list of candidate vulnerabilities. These are highly speculative and represent the red team's perception of reality at this point in the assessment. The red team attempts to document the results of these brainstorming sessions in an effort to generate a complete record of the process. No vulnerabilities are dismissed at this point, simply because the team does not want to overlook or dismiss any attack. All potential attacks are reported to the sponsor. The results of this process are usually reported in either Fault Trees or Attack Trees.

Attacks – Members of the core red team then take the list of candidate vulnerabilities and select certain vulnerabilities that will be further developed into conceptual attacks. Vulnerabilities are chosen based on their perceived value to the adversary and their ability to achieve the adversary's objective purpose. Only high-value vulnerabilities are developed into candidate attacks.

Candidate attacks are reported in Attack Tables. For each attack, the red team typically reports:

- Description – Detailed description of the attack.
- Attack % Complete – To what extent has this attack already been developed by other parties? For example, an attack script that can be downloaded from the Internet might be 90% complete, whereas a new conceptual attack may only be 10% complete.
- Probability of Success – If the attack is carried out, what is the probability that the attack will achieve its objective purpose?
- Attack Mean Time to Recover (MTTR) – How long would it take defenders to recover from this attack?
- Cost to Develop – usually measured in man-hours.
- Time to Develop – How long would it take the adversary to develop this attack in a laboratory environment?

- Time to Implement – Once developed, how long would it take the adversary to mount this attack in the field?
- Skills to Develop – What unique skills are required to develop this attack?
- Resources to Develop – What type and quality of resources such as facilities, equipment, and materials are needed to develop this attack?
- Skills to Implement – What unique skills are required to carry out this attack? Often, less-skilled staff is required to implement an attack than is required to develop the same attack.
- Resources to Implement - What type and quality of resources such as facilities, equipment, and materials are needed to execute this attack?

Credibility Validation & Ranking – This list of potential attacks is then delivered to the sponsor with the expectation that the sponsor will apply their own risk management processes to prioritize the attack list. Ideally, the customer will have some established intelligence or risk management function to perform this function. Otherwise, the red team will recommend which attacks should be developed into some sort of activity.

Activities – Optionally, the sponsor may elect to demonstrate any or all of the red team attacks as part of a larger activity. These red team activities usually take the form of some sort of demonstration, field exercise, or experiment.

2.4 Reporting Process

The red team usually generates three types of reports as shown in Figure 3.

Quick Look Report – The Quick Look is the result of a high-level speculative analysis of the system. This report is usually generated after the Candidate Vulnerabilities are identified (see Section 2.3). This report is usually delivered halfway through the assessment schedule.

Detailed Surety Assessment – This report attempts to document all of the attacks that were identified by the red team. The goal of this report is to deliver a complete set of all attacks and vulnerabilities postulated by the red team as well as details required for developing any of the high-value attacks. This report is usually delivered before the “Credibility Validation & Ranking” step described in the Red Team Assessment Process (Section 2.3).

Activity Plans – In cases where the red team is expected to proceed with some sort of activity based on their attack list, the red team will then develop detailed plans for each activity. These plans may include:

- Experiment Plans – Details of a scientific experiment designed to gather data that either supports or refutes an experimental hypothesis.
- Demonstration Script – Details of a demonstration that illustrates how an adversary might mount a given attack.
- Exercise Plan – Details of a generic red team exercise.

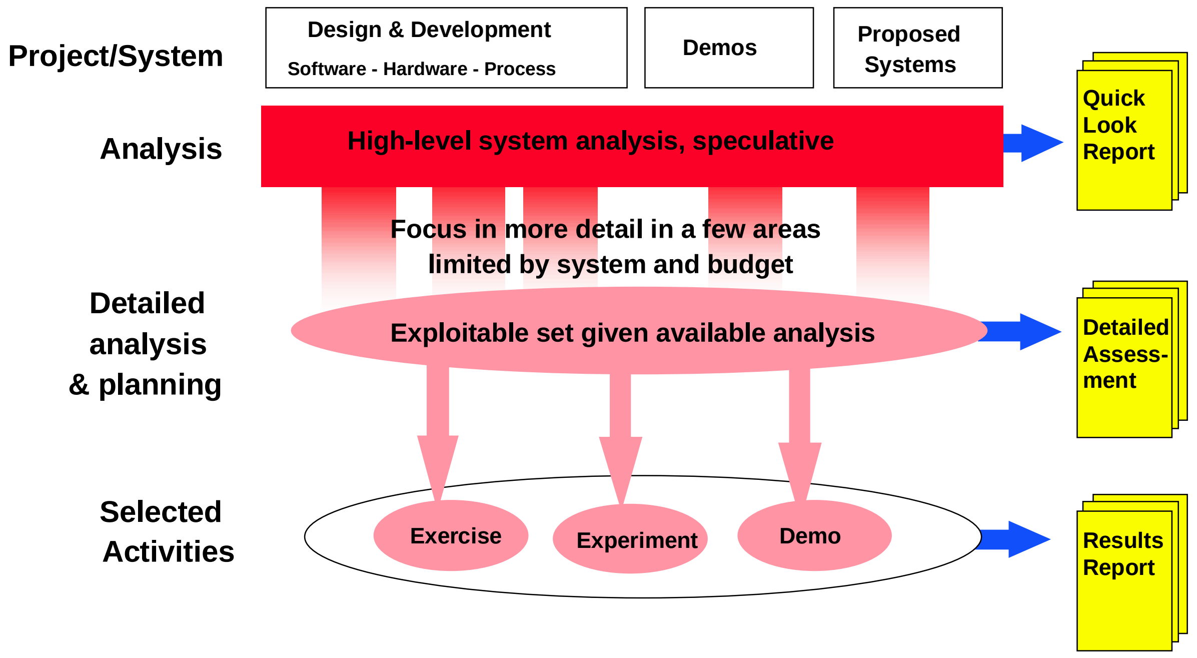
Project/System
Design & Development
Software - Hardware - Process
Demos
Proposed Systems
Analysis
High-level system analysis, speculative
Quick Look Report
Focus in more detail in a few areas limited by system and budget
Detailed analysis & planning
Exploitable set given available analysis
Detailed Assess-ment
Selected Activities
Exercise
Experiment
Demo
Results Report

Figure 3 -- Red Team Reporting Process

This may be appropriate if a sponsor needs a red team to model a given adversary for a particular event such as a war game or training exercise.

Activity Reports – Detailed results from a red team activity. These may include, but are not limited to the following data:

- Test Data – Test results or other relevant data collected during the activity.
- Red Team Diary – chronology of each step taken by the red team and the system's response as observed by the red team.
- Attack Trees and other planning aids used by the red team.

3 Application of the Process

The Red Team Assessment Process was applied to five different DARPA ITO projects, including:

- MIT Lincoln Labs Intrusion Detection Evaluation
- Generic Software Wrappers
- Agile Security Architecture
- Ensemble
- Intruder Detection and Isolation Protocol (IDIP)

These projects were selected by consensus between the Sandia Red Team and the DARPA ITO. The selected research projects represent a variety of development stages and concepts.

Four projects were reviewed through the Quick Look Assessment and Surety Assessment Phase. One project (Agile) was determined to be no longer viable and was dropped from the full assessment process prior to a full Quick Look Report. Attack tables were generated only for those projects that have a technology implementation component.

For the remaining four ITO projects, the following were reported:

- **System Description**
- **Critical Success Factors** for the system
- **Metrics** for success of the systems
- **Surety Assessment** results
- **Recommendations** including proposed experiments
- **Conclusions**
- **Results of the Interaction** between the developers and the Red Team
- **List of References** for each project

The actual assessment results are detailed in a separate limited-distribution report [4].

4 General Results

4.1 Data Collection

The amount of data available was comparable to the stage of development that each project had achieved. Much of the data was available on the project web sites or was collected through interviews. The data collection process revealed the importance of a concept of operations since many of the projects depended heavily on the correct or appropriate application of the end product of the project.

4.2 Critical Success Factors

Critical success factors were determined for each project. Some of the key success factors include the following:

- The attack scenario data sets used in each project must be appropriate to the design and continually updated to represent possible real-world threats. The data sets must be diverse and of sufficient quantity to represent current real-world threat vectors.
- Simulation networks must be consistent with real world networks both in diversity and complexity to be representative of true operating conditions.
- System performance and overhead requirements must be minimized on each platform and should not significantly degrade the network performance of an application.
- Sequencing of security systems is critical to interactions with COTS software.
- Security systems must be portable to multiple platforms in today's computing environment and must show value in group communication settings. They should be flexible and easy to configure.
- Security system automated responses shall not deny service to valid users such that a critical operation is denied. There must be a way to recover from the adversary using the security system's own responses as a denial of service attack.
- One security system should not be able to circumvent another security system.
- The security system shall be self-protecting and resilient against attacks on the security system itself.

4.3 Metrics

The red team identified metrics relevant to each project. Not only must performance metrics for each project be defined, they must be consistently applied so that appropriate comparisons can be made. The advertised reliability/fault tolerance gained by use of a security system should be validated against different network loads and various group setting scenarios. Some of the metrics used include:

a) Number of exploits and attack scenarios.

b) Number of services and protocols represented in the test network and therefore, in the data.

c) Number of resources required to run each security system feature on a specific platform.

d) Number of operating system platforms that can run successfully use a security system.

e) System overhead percentages for each security feature.

f) The degree to which a security system increases the applications capability and enhances the reliability and fault tolerances of a network.

g) The security system does not add depreciably to the end-to-end network delays as indicated by throughput calculations and measurements

h) Security mechanisms provided by a security system are shown not to be weak against different compromising scenarios

i) Percentage of:

   - access attempts that are effectively denied by a security system
   - denial of service attacks that succeeded in denying critical operations
   - attacks that succeed in crashing the security system itself and the degree of recoverability

j) Distribution of times:

   - for response to be complete from time that the attack was assessed
   - for response to be complete from time first event of attack was detected
   - to alert of unauthorized access from time first event of attack was detected
   - to recover operations for valid users after a denial of service attack
   - to recover critical operations
   - to re-establish the normal operational state after a successful attack against the system

4.4 Surety Assessment Results

Through review of the project web-sites and limited discussions with the project principal investigators, the following observations are made:

Limited Attack Scenario Data Set - The number and types of attacks that are currently used for testing and evaluating security systems are limited and are not representative of current real-world threat vectors. The attacks used for testing are typically single-step exploits and do not include multi-step attack scenarios.

Consistent Application of Comparison Metrics – A set of appropriate comparison metrics should be established and consistently applied such that distinctions are made for the attack sets a particular IDS system is designed to cover.

Simulation Network Suitability - The test network used to test systems should be representative of the complex user-environment of today's computing environment.

No Adversarial Model – Test networks currently have no clear mission and therefore, an attacker has no clear goals. The adversarial model is non-existent. Several adversarial models should be developed to show what currently can be detected and what cannot against various threat levels.

4.4.1 Potential Design Weaknesses

COTS and security system efforts duplication - If security checking functionality is duplicated between the COTS application and the security system, performance issues may arise. Further, if interference occurs, then security lapse issues may arise.

COTS applications and OS evolution - When COTS software products release new versions, issues of security system compatibility may need to be addressed. A potential exists for lapses regarding security functions when patches are made to COTS or changes made to the operating system.

4.4.2 Security Issues

Communication Transitions - Protocols that initiate a changeover from non-secure to secure communications and secure to non-secure communications must be thoroughly understood so as not to introduce security breaches. It is particularly important to understand how a system determines if all communications are within the confines of the firewall, and whether the protocol can be spoofed into believing that all communication is within the firewall when actually a node maybe outside the firewall.

Insider Threat - Many security systems assume no insider threats.

Security system characterization - The fact that a security feature exists invites the possibility that an attacker could intentionally provoke a security system in order to characterize it. This is particularly true for response systems. The attacker could present attacks of sufficient severity to invoke response through relays and cut-outs to prevent tracking the attacks back to their origin. Once response characteristics are known, an attacker can use that information for attacks.

4.4.3 Performance Issues

Denial of Service - Memory fragmentation or full memory may cause denial of service issues.

4.4.4 Implementation Issues

System interfaces - Solutions involving multiple systems and incorporating various degrees of non-homogeneity within system interfaces are possible issues. Careful design of the interfaces between security system modules is important to minimize the potential of introducing significant overheads. Also, reconfiguration of an active system can be difficult since it involves delicate synchronization that could, if not handled appropriately, prove so disruptive as to shut down the application.

Lack of user documentation - To the novice operator, there still exists perplexing and confusing problems with path, environment, compiling, and execution of security systems. This condition is exacerbated by a lack of technical and user documentation

5 Lessons Learned

The experience resulted in some valuable lessons learned in the application of red teaming to development projects:

Timing is everything. With cyber development projects, red teaming can occur at design stages or within development stages. For red teams to have an effective impact, they must be a player throughout design and development as a reality check. Documentation is the primary element needed to support red teaming at design stages. Deployable elements are needed for red teaming in development stages.

Red teams are a tool, not a threat. Red teams reveal potential weaknesses in cyber systems so that mitigation can be developed as early as possible prior to deployment of the system. If red teaming is perceived as a threat or audit, the information sharing is limited and the results less successful than those resulting from a teaming environment.

Integrated red teaming requires planning. For red teaming to have value, it must be an integral part of cyber design and development. As such, it must be planned and scheduled as part of the project like other elements such as code reviews. It does take some time and resources to support red teaming dialogue, data collection, and experiments. This must be planned for in order not to burden a development project unexpectedly.

Red teaming of cyber development projects is mutually beneficial. Understanding the threat and possible attack vectors for new systems keeps red teams ahead of state-of-the-art. Projects benefit from early detection of potential problems with time to address mitigation prior to deployment or at least an understanding of the risks.

6 Conclusions

The key to successful red teaming at the development stage is the understanding that red teaming is not an audit. Rather, red teaming provides, but rather a mechanism for improving the deliverables associated with DARPA cyber development projects, similar to a proof of concept test. Success also depends on the free sharing of information in a timely fashion. The projects evaluated and the red team both benefited from using this process.

7 Acknowledgements

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

8 References

[1] Schudel, G. and Wood, B., *Modeling Behavior the Cyber Terrorist,* submitted for consideration by the 2000 IEEE Symposium on Security and Privacy, October 1999. This information may also be available on the World Wide Web at *https://ests.bbn.com*

[2] *Networks Attack from Russia?*, reported by Reuters news service, October 6, 1999

[3] Vatis, M. A., *NIPC Cyber Threat Assessment*, Statement for the Record by Michael A. Vatis, Director, National Infrastructure Protection Center (NIPC), Federal Bureau of Investigation, before the Senate Judiciary Committee Subcommittee on Technology and Terrorism, Washington DC, October 6, 1999

[4] Wood, B. and Duggan, R., *Survivability Information Systems Red Teaming Analysis Results*, final report to the DARPA ITO office, October 1999. Copies of this limited report may be obtained through DARPA.