INL/CON-05-00191
PREPRINT

Cyber Assessment Methods for SCADA Security

15th Annual Joint ISA POWID/EPRI Controls and Instrumentation Conference

May Robin Permann
Kenneth Rohde

June 2005

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance

Idaho National Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint should not be cited or reproduced without permission of the author. This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights. The views expressed in this paper are not necessarily those of the United States Government or the sponsoring agency.

Cyber Assessment Methods for SCADA Security

May Robin Permann
Staff
Information & Communications Systems
Idaho National Laboratory
Idaho Falls, ID 83415

Kenneth Rohde
Computer Security Researcher
Cyber Security Technologies
Idaho National Laboratory
Idaho Falls, ID 83415

KEYWORDS

Supervisory Control and Data Acquisition, SCADA, Cyber Security, Testing, Assessment

ABSTRACT

The terrorist attacks of September 11, 2001 brought to light threats and vulnerabilities that face the United States. In response, the U.S. Government is directing the effort to secure the nation's critical infrastructure by creating programs to implement the National Strategy to Secure Cyberspace (1). One part of this effort involves assessing Supervisory Control and Data Acquisition (SCADA) systems. These systems are essential to the control of critical elements of our national infrastructure, such as electric power, oil, and gas production and distribution. Since their incapacitation or destruction would have a debilitating impact on the defense or economic security of the United States, one of the main objectives of this program is to identify vulnerabilities and encourage the public and private sectors to work together to design secure control systems that resolve these weaknesses.

This paper describes vulnerability assessment methodologies used in ongoing research and assessment activities designed to identify and resolve vulnerabilities so as to improve the security of the nation's critical infrastructure.

INTRODUCTION

The National SCADA Test Bed (NSTB) program is sponsored by the Department of Energy – Office of Electricity and Energy Assurance (DOE-OE) to improve the security of cyber assets in the energy sector. The Idaho National Laboratory (INL) SCADA Test Bed is a venue for assessing the security of various SCADA system configurations. NSTB work at the INL focuses on outreach and awareness, vulnerability assessment and mitigation, standards development and best practices, and the creation of new security assessment tools. Information obtained through this program is shared with vendors and/or industry in order to enhance security by helping control system vendors and customers secure their own systems against external and internal cyber attacks.

Ongoing research and assessment activities have revealed an effective methodology for identifying vulnerabilities and developing assessment methods to secure SCADA systems. This assessment methodology, which resulted from lessons learned when testing vendor systems, is presented in this paper for the purpose of helping vendors, utilities, and others assess and enhance security measures on their own SCADA systems.

ASSESSMENT METHODOLOGY

The steps involved in this assessment methodology are: developing an assessment plan, configuring the test environment, assessing the system, reporting requirements, and using assessment metrics for scoring. The purpose of each step and suggestions for implementing them are discussed in the following sections.

ASSESSMENT PLAN DEVELOPMENT

Security assessments should be bounded by a detailed assessment plan that specifies a schedule and budget, targets and goals, expected deliverables, hardware and resource requirements, rules of engagement, and a recovery procedure. The team assigned to perform the assessment should be involved in the development of the assessment plan.

Assessing the vulnerability of an information system can be a never-ending task as one digs deeper and deeper into the system, exposing vulnerabilities and finding new exploits. Specifying the time and resources to be spent on any one task keeps the level of effort in check. It is important, therefore, that the assessment plan specify targets of evaluation (TOEs) to be tested. The Common Criteria for Information Technology (IT) Security Evaluation (2) defines a TOE as an IT product or system with IT security functions. This includes operating systems, computer networks, distributed systems, and applications. For this application, a TOE is typically a subset of the SCADA system.

As an example, a TOE for a SCADA system might be the alarms and commands to and from the field components. One method of attack to change alarms and commands would be to analyze the network traffic to and from the Human-Machine Interface and develop a man-in-the-middle (MITM) style of manipulation. In an MITM attack, an attacker is able to read, insert, and modify messages between two parties without either party knowing that the link between them has been compromised. A functional description of the alarms and commands and their importance as well as ways of accomplishing an MITM attack on the system being tested could be included in the test plan to help testers focus the attacks. This assumes that an insider or an accomplished hacker has already penetrated the perimeter and gained access to the SCADA network. A flow chart, such as the one in Fig.1, could also be included in the assessment plan to better illustrate this process.

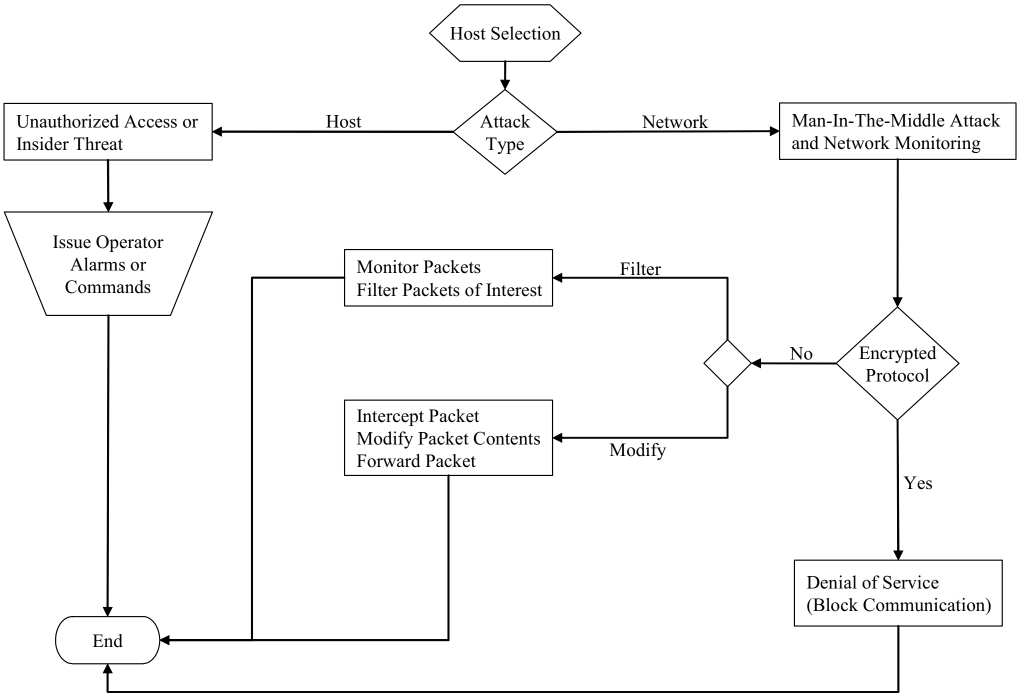

Fig. 1: Modifying Alarms and Commands

Based upon priorities and resources, an allocated testing period would define the level of effort to give this task. The test plan would help the testers get started by suggesting methods of attack based on the combined knowledge of the cyber and SCADA team members. Data requirements describe the information that must be gathered before attempting the TOE, and in many ways define the difficulty of the attack and characterize the attacker. For this example, a detailed composition of the message format and the complete process for changing both an alarm and a command would be valuable information for achieving this TOE. Finally, a definition of a successful attack lets the test team know when they have accomplished the task.

Generally, TOEs should be prioritized in the assessment plan based on how critical they are to the operation of the whole system, and a corresponding level of effort for testing should be defined for each one. Prioritizing the list of TOEs allows the assessment team to specify which ones must be evaluated, with the flexibility of having additional TOEs to assess should time remain. Flexibility in the assessment process, however, is extremely important, and schedule or resource adjustments should be allowed to account for unexpected findings during testing.

Along with accomplishing the TOEs, the team may also be expected to produce other deliverables such as intermediate and final reports. These expectations should be identified in the assessment plan.

The rules of engagement define the knowledge, funding, and timeframe of the attacker(s) as well as their skills and capabilities. Defining the attacker aids in the development of the types of attacks and where they are based. Possibilities include a nation state, a well-funded terrorist, a hacker who has penetrated the firewall or otherwise gained access to the SCADA network, and/or an insider.

A recovery plan should be delineated in case vulnerability testing corrupts the system. This could be implemented with tape backups, ghosting, or mirrored disks that are removed before testing.

A well-defined assessment plan provides detailed direction to the team, eliminating confusion and wasted time. The time allocated for testing is well utilized and productive.

TESTING ENVIRONMENT CONFIGURATION

The ideal environment to assess the SCADA is a safe (non-production) configuration. However, it is also desirable to have all connecting components and functionality available in order to confidently assess full system interoperability. This includes everything from the Inter-Control Center Communication Protocol (ICCP) link to the Remote Terminal Unit (RTU) connectivity. ICCP is the international standard, International Electrotechnical Commission (IEC) Std 870-6 Telecontrol Application Service Element Two (TASE.2), for real-time data communication between control centers. An RTU is used in SCADA systems at a remote location to collect code and transmit data back to the control station, as well as receive and implement commands from the control center. Ideally, these components are configured with actual signals for the assessment, although emulators and simulators may be necessary.

The ability to establish a typical SCADA installation allows the assessment team to more accurately test and make recommendations on the configuration and deployment of a production system. Understanding the system configuration is key when deciding where the biggest risks in the system may reside and, therefore, which TOEs are top priority. Mirroring the connections to external systems is vital when replicating this configuration. The assessment team must know where and how a SCADA system element typically connects to such things as the Internet, ICCP servers, and RTUs. Any of these elements being accessed from outside the SCADA network, such as the Historian database, should be tested based on its position in the network. Firewall and intrusion detection system (IDS) configurations should be duplicated in order to test the effectiveness of the perimeter security they provide. IDSs can be tested to determine if they detect intrusions into the SCADA system. In general, the firewall(s) enforce the security policy for the SCADA system and the IDS is a auditor to ensure that the rules are enforced. Accepted and normal operating procedures need to be known in order to assess the likelihood of an attack being successful. The goal of many attacks is to alter information and/or commands coming into or going out of the SCADA system. A completely functional system, or reasonable representation, provides a realistic assessment environment since complete determination of the results or consequences of an attack is dependant upon all of the interconnected functionalities within the system. For example, assessing a system without a redundant backup system with automatic failover doesn't show how a system which is usually implemented with redundancy responds to an attack.

There are several features that are desirable when assessing a SCADA system. A working ICCP link for I/O testing and as many representative RTU protocol connections (i.e., Transmission Control

Protocol and the Internet Protocol [TCP/IP] or protocols that are supported and used in the field) allow for testing of the control specific communication protocols. If applicable, a corporate network separated by a firewall from SCADA should be simulated. If a historian database is typically placed in the corporate network, that configuration should be implemented in the target system. Firewalls and intrusion detection systems that are either recommended by the vendor, representative of the typical installed system, or recommended by cyber security professionals offer a configuration to be validated so that a recommendation can be given for all installations. Firewall configuration and other security features must be tested by the vendor to validate that they do not affect the operability of the SCADA system.

VULNERABILITY ASSESSMENTS

Penetration testing must be conducted from a machine that is not part of the SCADA system unless otherwise defined in the assessment plan (i.e., an insider threat). This replicates a typical attack scenario where the attacker must penetrate the system from a remote computer. Placement of the attack computer depends on what is being tested and where the attack scenario originates.

Sophisticated attacks are often system specific and tailored to the target computer's architecture. Therefore, the attacker needs a similar computer to create and test malicious code. For example, if 64-bit processors are used in the target SCADA system, the assessment team may need equivalent hardware and software for developing exploits specific to that architecture. Although in a test environment there is direct access to the target system, its configuration should not be altered. A separate machine is needed to install required software such as compilers, debuggers, and other tools. It may not be feasible to obtain test computers for each represented hardware or software configuration on the SCADA system, but doing so would improve the assessment time and results.

Dedicated assessment equipment is also necessary for the assessment to be conducted without interruption. An attack machine with all the tools needed to perform the vulnerability assessment should be available for this work. In addition, a reliable Internet connection for research in the test area makes work more efficient.

It is also important to return the system to its original state before each new test. This could be implemented with the same process as the backup plan. This insures that all effects are from the current exploit.

The following steps proved to be a useful method of assessing a SCADA system. These may vary depending on the assessment plan TOEs and rules of engagement.

1. Perform reconnaissance to gather information on the target system if not previously defined in the assessment plan
2. Scan the SCADA network for open ports and vulnerabilities
3. Achieve the TOEs defined in the assessment plan.

The following resources and advice are valuable in the assessment process:

- Prioritized vulnerabilities to assess based on the probability of obtaining the target and its significance
- Dedicated semi-private work area
- Broadband (reliable) internet access for research
- Vendor help and support
- Backing up the target (SCADA) system
- Rebooting the system after every attack to ensure all of the effects are presented; some effects may not be apparent until the system has been rebooted.

ASSESSMENT TOOLS

Below is a list of the open source and commercial tools useful for assessing SCADA systems. Specific tools are listed because they are commonly available and are the best or only option without developing a tool for the specific application.

It is important to note that these scans and exploits are being run in a controlled lab environment. Any of these used on a production system could cause it to malfunction or stop operating. Legal restrictions on using scanning tools and exploits across a public network or your own computing systems vary. Therefore, check with your company legal personnel and get permission from the appropriate company authorities before running any of the following attacks.

NMAP

Nmap is an open source multi-purpose network scanning tool used mainly as a port-mapper. It identifies open ports on each machine. This information is then used to identify the services that are actually running on each port. This is one part of the information-gathering phase of a vulnerability assessment. The list of open ports on each machine gives a place to start testing for open holes into the SCADA system, but further analysis is required to not only verify the Nmap results but to also query "unknown" ports returned by Nmap. Care should be taken when running this tool on a production machine as it could cause aberrant behavior. Nmap scans should be performed on a test or backup system, if available (3).

NESSUS

The Nessus open source remote security scanner performs nearly 6000 security checks against a target system, detecting vulnerable services running on the scanned hosts and providing a warning level and recommended fix for each possible vulnerability. Nessus is very flexible and can be configured to perform only the appropriate tests. The Nessus output can be used as another starting point for assessing a system. The Nessus report identifies possible vulnerabilities that the assessment team can then try to exploit thereby verifying or disproving the SCADA system's susceptibility to that particular threat. It is not enough to report all vulnerabilities discovered in a Nessus scan as many of them may not actually affect the system and/or are false positives reported by Nessus. Conversely, Nessus may not detect all vulnerabilities; therefore, it is only a starting point for assessing the system. It does not, for example, check for vulnerable installed software that runs locally on a computer (e.g., vulnerabilities with versions of Internet Explorer). Other vulnerability scanning tools are used for this task (see STAT Scanner below) (4).

STAT SCANNER

The commercially available STAT Scanner by Harris Corporation is useful in evaluating Windows-based systems in greater depth. The package provides superior detection of vulnerabilities on Microsoft operating systems, applications, and components. It has a low rate of false positives, excellent reporting capabilities, and is relatively inexpensive. For the Microsoft Windows 2003 and Windows XP operating systems, the software requires access to the local administrator account on the host and requires the messenger, server, and remote registry services to be enabled.

STAT Scanner provides powerful reporting capabilities and is also available for other platforms (5).

ETHEREAL

Ethereal is a widely used open source network protocol analyzer that allows communications monitoring between the individual SCADA system components. This is useful for discovering which components communicate, if communications are encrypted, and intercepting information (including user names and passwords) being sent in plain text (6).

ETTERCAP

Ettercap is an open source suite for MITM attacks on switched networks. It features sniffing of live connections, content filtering "on the fly," and other capabilities such as active and passive dissection of many protocols (even ciphered ones) and password grabbing from specific applications. This allows a MITM attack to be created for viewing communications between the SCADA system components in Ettercap and/or Ethereal, and testing the system against Ettercap's other attack capabilities (7).

METASPLOIT

The Metasploit Framework is an advanced open-source platform for developing, testing, and using exploit code. It is a powerful tool for penetration testing, exploit development, and vulnerability research. Metasploit provides a set of application programming interfaces (APIs) that is used by cyber testers for packaging customer exploit code in a reusable and automated fashion (8).

DEBUGGERS

Debuggers allow one to evaluate a program during execution or a program's state at the moment it crashed. The main functions of this tool are to:

- Execute a program, given any input parameters which affect its behavior
- Allow the user to step through the program instructions, or stop and start the program at any given points
- Allow the user to examine and change values at stop points or after it crashes.

SCADA system software can be analyzed in a similar manner to search for unsecured programming practices that allow an attacker to gain access through a known vulnerability into the system. Debuggers can also be used to find security holes such as possible buffer or heap overflows, places where the program crashes, and other code that is not secure. Buffer overflows can be analyzed, for example, by testing programs in a debugger with different input parameters. Some examples of debuggers are GDB, IDA Pro, and those built into compilers such as the Microsoft Visual Studio.

SOFTWARE DEVELOPMENT TOOLS

Software development tools facilitate software development, giving the programmer a user-friendly interface for writing, debugging, and compiling the code. These tools are useful in vulnerability assessments for analyzing code for safe programming practices and for generating exploit code with which to test the SCADA system. The most common software development environments are the GNU compiler/debugger packages for UNIX- and Linux-based operating systems, and Microsoft's Visual C for Windows operating systems.

PUBLISHED EXPLOITS

Rather than reinvent the wheel, it is most efficient to use any published exploits for the components in the target system. This is a quick approach for verifying vulnerabilities in a system. Many well-known exploits have associated exploit code available on the internet that can be downloaded and run against the system. However, caution should be used when entering sites that house exploit code as well; visiting such sites presents a danger of attack by the site or by the downloaded code.

IN HOUSE CODE DEVELOPMENT

When no public exploits are available for targeted systems, the ability to develop exploits in house is very valuable because it allows the researchers to not only report discovered vulnerabilities, but provide working exploits back to the customer. An example of this is the discovery of a vulnerability in a vendor's software application. This application is not widely available to the public, yet it still suffers from some common software vulnerabilities. Simply stating that these vulnerabilities exist is much less authoritative than providing documents of the vulnerability along with working exploit code.

In order to develop in-house code for security assessments, the assessment team must have access to personnel with extensive experience in assembly code, debugging, reverse engineering, and computer hardware. Creating an assessment team with these resources creates a complete and comprehensive set of skills for thorough security testing.

FUZZERS

Fuzzers are an automated way of finding vulnerabilities by feeding the target application with a wide range of invalid input. Input that causes the application to respond abnormally or crash is then used to identify vulnerabilities. The fuzzer can look for specific kinds of vulnerabilities and can range from manual mode to fully automated. Vulnerabilities typically identified are application level overflows and format string vulnerabilities. They are best suited for services using documented protocols, standard servers, web applications, and protocols with many field combinations (9).

STATIC CODE ANALYSIS

Static code analysis disassembles binary executables, looking for vulnerabilities. A disassembler can be used to translate the executable into a higher level language that is easier for the evaluator to understand. This is a very time consuming process, but provides a very deep analysis of the application code. This process is best for finding protocol level overflows, complex vulnerabilities and integer vulnerabilities in protocol parsers, unknown protocols, code using unsafe functions, and critical code sections.

IDA PRO

Interactive Disassembler (IDA) Pro is a Windows or Linux hosted multi-processor disassembler and debugger. This tool can be used in vulnerability research for code analysis and software reverse engineering. It is one of the most advanced tools for hostile code analysis, vulnerability research, and reverse engineering. This tool allows assessment teams to evaluate software on the SCADA system for simple vulnerabilities (10).

SUBJECT MATTER EXPERTS (SMEs)

SMEs are some of the most valuable resources in the vulnerability assessment process. Experts in both cyber security and the target SCADA system are crucial to a thorough security assessment.

Subject matter experts in how the system operates and its most critical functions help guide the assessment process. Using a SCADA system specialist to define how the system works can substantially reduce the time required for an attacker to gather the information required to plan the attack.

Cyber security experts know how to use and develop vulnerability tools, test for and verify the existence of vulnerabilities, and recommend fixes and secure practices. For instance, an expert in running the STAT Scanner software is a valuable resource because the expert’s opinion can be used to characterize the risk reduction between system versions and recommend ways to secure the current version.

BOOKS

In addition to some of the software development tools, books on Windows Administration, Linux Administration, Scripting, and Command-line References are very helpful in testing for vulnerabilities. Security books can provide a process and identify vulnerabilities to test for in particular operating systems and applications. The particular books selected depend on the individual components in the system under review and the knowledge of the assessment team. Due to the constant evolution of software and vulnerabilities, reference materials should be kept up to date with the latest hacking techniques and information on the current operating system and application versions being used in the SCADA system. Potential resources for vulnerability testing are the "Hacking Exposed" series of books (11). The information in some books can also be found on the Internet, such as the Windows command line reference (12).

HARDWARE

Often times SCADA systems use operating systems and hardware that are not in wide usage. Due to this fact, it is important that the assessment team have access to some of these less common platforms so that they can build their own testing environment for use during an assessment. An example might be an Alpha Server running Tru64 UNIX.

LOGS

System logs should be saved during testing because they can be used to indicate intrusions. Information gathered during testing can later be used for discovering these attacks on a production system.

REPORTING

Detailed reporting of all tools used against the SCADA system and the associated steps, findings, and system response is invaluable. This includes archiving the test tools and scripts used. Documenting this information as quickly as possible assures that no information is forgotten and saves time when trying to duplicate the attack. This information can then be used in the future for writing reports and validating which tests were conducted and whether the system was susceptible to them. Reports also provide a way to reproduce an attack when confirming that the hole was fixed in the next version of the software. The goal for this process should be to document to the level where someone skilled in the area could duplicate the results.

The following reporting guidelines make reporting and future verification straightforward:

- Write detailed weekly reports
- Keep the report writer involved during the assessment, at least at high level, so report writing is easier and more accurate
- Keep documentation of testing
- Archive testing tools and scripts for reproducibility
- Use configuration management
- Reboot and revert to the original configuration between each test.

Proprietary or otherwise sensitive data collected on specific systems must be protected against external release. Internal controls should include physical isolation during testing, assurance that anyone obtaining vulnerability information acknowledges applicable agreements, and security on computer systems holding the data. The organization conducting the assessment should have an administrative security program that clearly defines protection controls and implements security background checks of all personnel with access to the data.

METRICS AND SCORING

It is important to have a way to quantitatively measure the security of a system to determine its risk level, measure risk reduction due to security enhancements, and evaluate how it compares to other systems. Since there is no tool to perform these tasks, the best way to do it is to seek the opinion of cyber security experts. This is somewhat subjective, however. The Department of Homeland Security (DHS) is currently working on a risk-based decision methodology that can be used to calculate risks associated with potential terrorist acts that utilize control systems (13).

Information Technology (IT)-based scoring systems do exist. Experience using these tools shows that it is important to decide on a scoring system before the assessment is performed. This ensures that the necessary data can be gathered for more meaningful scoring.

SUMMARY

Methodologies for performing vulnerability assessments of SCADA systems are being developed through ongoing research, with the goal of improving the security of the nation's critical infrastructure. This experience can be leveraged to refine the assessment process and provide industry with better options to secure their section of the infrastructure.

Lessons learned in a laboratory environment can be taken advantage of in other environments as well. Those sites which operate with backup and test systems can perform vulnerability assessments on these systems. Many hacking tools and resources are available for sale or as free downloads off the Internet. These tools do require some computer skills and therefore should only be performed by a qualified IT/cyber security professional.

REFERENCES

1. Bush, George, “National Strategy to Secure Cyberspace”, The White House, Washington, February, 2003.

2. Canada, France, Germany, Netherlands, United Kingdom, and United States, "Common Criteria for Information Technology Security Evaluation", Version 2.1, August, 1999.

3. “Nmap – Free Security Scanner For Network Exploration & Security Audits”, http://www.insecure.org/nmap/, April, 2005.

4. “Nessus Open Source Vulnerability Scanner Project”, http://www.nessus.org, Tenable Network SecurityTM, March 22, 2005.

5. Meyer, Sleightom, Stein, Neal, “Harris Corporation’s STAT® IT Security Tool Now Checks For 3,000 Individual Computer Vulnerabilities”, http://www.stat.harris.com/news/pr/PR_Scanner_3%20000_Vulnerabilities_8_30_04.pdf, Melbourne, Florida, August 30, 2004.

6. “Ethereal: The world's most popular network protocol analyzer”, http://www.ethereal.com/, Network Integration Services, Inc, March 31, 2005.

7. Ornaghi, Alberto, Valleri, Marco, “Ettercap”, http://ettercap.sourceforge.net/, April, 2005.

8. “Metasploit Project”, http://www.metasploit.com/indexa.html, April, 2005.

9. “Bug Hunting”, http://www.phenoelit.de/stuff/Bugs.pdf, April, 2005.

10. “The IDA Pro Disassembler and Debugger”, http://www.datarescue.com/idabase/, Liège, Belgium, April, 2005.

11. McClure, Stuart, Scambray, Joel, Kurtz, George, Hacking Exposed: Network Security Secrets & Solutions, Ed. 4, McGraw-Hill/Osborne, Berkeley, California, 2003.

12. “Microsoft Windows XP Professional Product Documentation”, http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/ntcmds.mspx, April, 2005.

13. Beitel, George, Alessi, Sam, “Control Systems Risk Decision Methodology”, INL/EXT-05-02585, Rev. 1, US Department of Homeland Security, Idaho Falls, Idaho, March, 2005.

ACKNOWLEDGEMENTS

This work was supported by the U.S. Department of Energy under DOE Idaho Operations Office Contract No. DE-AC07-05ID14517.