CYBER-SECURITY INTEGRITY FOR ELECTRIC GRID FACILITIES MANAGEMENT

Energy and Water Projects
EW-201608

December 2019

Final Report

Lawrence Livermore National Laboratory
Pacific Northwest National Laboratory
The Boeing Company

DISCLAIMER

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(19-12-2019)*	**2. REPORT TYPE** SERDP Final Report	**3. DATES COVERED** *(From - To)*
4. TITLE AND SUBTITLE CYBER-SECURITY INTEGRITY FOR ELECTRIC GRID FACILITIES MANAGEMENT		**5a. CONTRACT NUMBER** DE-AC52-07NA27344
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) D. Quinlan, L. Banks, K. Masica, R. Matzke (Lawrence Livermore National Laboratory) N. Multari, T. Merz, W. Hutton, M. Engles (Pacific Northwest National Laboratory) R. Hammond, J. Cazalas (Boeing)		**5d. PROJECT NUMBER** EW-201608
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lawrence Livermore National Laboratory P.O. Box 808 Livermore CA, 94551		**8. PERFORMING ORGANIZATION REPORT** LLNL-TR-800620
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Environmental Security Technology Certification Program 4800 Mark Center Drive, Suite 16F16 Alexandria, VA 22350-3605		**10. SPONSOR/MONITOR'S ACRONYM(S)** ESTCP
		11. SPONSOR/MONITOR'S REPORT NUMBER(S) EW-201608

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

13. SUPPLEMENTARY NOTES

14. ABSTRACT
A set of automated tools for the analysis of binary executables has been developed within an open source software analysis framework. This technology aims to mitigate cyber risk in the context of facility maintenance and protection of DoD base and facility substations as well as building automation systems. A process for the analysis of firmware upgrades to substation equipment has also been defined, which would result in improved supply chain integrity for mission-critical energy delivery systems.

15. SUBJECT TERMS
CYBER-SECURITY INTEGRITY, ELECTRIC GRID, FACILITIES MANAGEMENT

16. SECURITY CLASSIFICATION OF: U			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Quinlan, Dan
a. REPORT UNCLASS	**b. ABSTRACT** UNCLASS	**c. THIS PAGE** UNCLASS	UNCLASS	39	**19b. TELEPHONE NUMBER** *(include area code)* 925-423-2668

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

Abstract

Firmware used in the control and monitoring of the U.S. electric grid and Building Automation Systems (BASs) are rarely, if ever, analyzed by end-users for potential vulnerabilities prior to being deployed. This is of concern to Department of Defense (DoD) Energy and Water (E&W) operations who maintain many such control systems and BASs. Should a firmware vulnerability in a control system be exploited, it could lead to disastrous consequences such as failure of systems, destruction of equipment, and potential compromise of DoD base infrastructure.

This project successfully developed and demonstrated a suite of automated tools for the analysis of binary executables to identify potential vulnerabilities prior to deployment within electric control systems at DoD base facilities.

The suite of tools consists of a set of five Binary Analysis Tools (BATs). These tools leverage the ROSE infrastructure developed over the last 20 years at Lawrence Livermore National Laboratory. Each BAT analyzes a binary for the potential presence of specific cyber vulnerabilities. They were designed for three different user-types to perform actions based on the BAT analysis output. These user-types include:

- DoD Power Engineer, who uses the BATs before performing firmware updates
- DoD Protection Engineer, who analyzes specific firmware based on BATs warnings
- DoD Reverse Engineer/Firmware Vendor, who perform deeper analysis if needed

The BAT tool suite was demonstrated at the Army Research Laboratory and at Elmendorf Air Force Base. The demonstrations illustrated how each BAT functions, the outputs they produce and how each of the three user-types could leverage the tools in their respective environments. The concept of potentially escalating a firmware analysis question to the responsible vendor was met with understanding and agreement. As most sites will not have advanced binary analysis expertise it was agreed that the firmware vendor should be responsible for explanation of any anomalies encountered in the analysis.

The cost model for the BAT tool suite addresses the requirements to support firmware analysis at individual DoD sites. Since the software is made available to DoD at no cost, the BAT tool compute platform and end-user training will dominate implementation cost. The cost model includes a conventional desktop computer, installation, and ongoing system administration.

The introduction of the usage of the BAT technology also introduces the need for added time for a Power Engineer to apply firmware updates. This added time comes from the need for a Power Engineer to first use the BAT technology to analyze the firmware update before applying it, and if necessary, the time needed for an additional Protection Engineer to review the findings of the BAT analysis. Because cyber-risk mitigation is often only noticeable when a system fails due to cyber-intrusion, using the BAT technology may seem like an unnecessary step to firmware upgrades.

Several publications were produced during this project:

- *ESTCP Operational Usage Report* – The Boeing Company
- *ROSE Binary Analysis Tools User Guide* – Lawrence Livermore National Laboratory
- *Firmware Management Best Practices for Energy Infrastructure Embedded Devices* – Lawrence Livermore National Laboratory
- *Including Firmware in Vulnerability Assessment for Energy Infrastructure Embedded Devices* – Lawrence Livermore National Laboratory

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

BAS	Building Automation System
BAT	Binary Analysis Tool
DoD	Department of Defense
DoDI	Department of Defense Instruction
DoE	Department of Energy
EO	Executive Order
ESTCP	Environmental Security Technology Certification Program
E&W	Energy and Water
ICS	Industrial Control Systems
JFAC	Joint Federated Assurance Center
LLNL	Lawrence Livermore National Labs
PDS	Power Distribution System
PNNL	Pacific Northwest National Laboratory
RTAC	Real-Time Automation Controller (by Schweitzer)
SCADA	Supervisory Control and Data Acquisition
SEL	Schweitzer Engineering Laboratories

ACKNOWLEDGEMENTS

Appreciation and gratitude for help and support are extended to the following persons who have contributed to the success of this project and the generation of this final report.

The **ROSE development team** at Lawrence Livermore National Laboratory. Their continued commitment in supporting the needed capabilities in the ROSE infrastructure makes possible the development of tools such as provided in this project.

Mr. Robb Matzke, principal developer of Binary Analysis Tools. Big shout out to Robb's patience and thoroughness in explaining (and re-explaining) the how's and why's for the developing functionality of the Binary Analysis Tools.

Mr. Ken Masica, Lawrence Livermore National Laboratory, for serving as lead engineer and main evaluator of multiple potential demonstration sites as well as coordinating the efforts for our main demonstrations throughout the project.

Dr. Marcus Schordan, Lawrence Livermore National Laboratory, for unending optimistic support of ROSE software assurance capabilities.

Dr. David Manz and **Dr. Nick Multari**, Pacific Northwest National Laboratory (PNNL), for managing the Team at PNNL including **Dr. Terry Merz**, **Dr. Will Hutton**, and **Dr. Matt Engles**. This team provided critical concepts for the Demonstration, System Test plans and expert DoD perspective on software (source and binary) vulnerability concerns.

Mr. Ryan Hammond and his team at Boeing consisting of **Zach Thornton** and **Josh Cazalas**. Their ongoing feedback on Operational Usage including system installation and tool usage led to several usability improvements for the end-user experience.

Special thanks **Dr. Greg Pope,** at Lawrence Livermore National Laboratory, for ongoing support and resource oversite across the many projects utilizing the ROSE infrastructure.

1.0 INTRODUCTION

1.1 BACKGROUND

1.1.1 Problem Statement

Current routine maintenance of large-scale electrical grid infrastructure and building automation systems (BAS) includes maintenance of both physical systems and the software systems that control them. Security of the physical systems is well understood and protected with perimeters of defense, but the software that periodically updates the control systems (software that is known as firmware) is automatically installed on these devices, often without any formal consistent or scientifically valid evaluation.

This firmware used in the control and monitoring of the U.S. electric grid and BASs experiences the same vulnerabilities plaguing software written for other purposes. Additionally, the specialized and proprietary nature of Supervisory Control and Data Acquisition (SCADA) firmware deployed throughout electrical grid control systems has historically limited the efforts of vulnerability researchers to detect and propose fixes for serious flaws in this critical firmware. Vendors' solutions are typically limited to certificate signing of firmware, which does not inspect the firmware being signed for vulnerabilities. Since no direct analysis of firmware integrity is done prior to installing the firmware on substation and BAS devices, the supply chain could be at greater risk from firmware errors than from mechanical systems because mechanical systems are certified and protected through perimeter defenses.

This problem is of particular concern to Department of Defense (DoD) Energy and Water (E&W) operations who employ many such SCADA control systems and BASs. The problem is exacerbated by the increased connectivity of SCADA/BASs which often employ some type of network/internet connectivity. Should a firmware vulnerability in a DoD control system be exploited, it could lead to disastrous consequences such as failure of systems, destruction of equipment, loss of service, and potentially loss of human life.

1.1.2 Current Technology State in DoD

Software assurance, and in particular embedded software has become a key area of interest within the Department of Defense. Through the Department of Defense's Office of the Deputy Assistant Secretary of Defense for Systems Engineering, an initiative was created that takes a focus on software assurance. This initiative is referred to as the Joint Federated Assurance Center (JFAC). The JFAC consists of a federation of DoD organizations that have a shared interest in software and hardware assurance. As such, the JFAC's area of cognizance includes detection, analysis, and remediation services; information about emerging threats and capabilities; and software and hardware vulnerability assessment tools, services and best practices. As such, the JFAC identified the following issues as it relates to software engineering within DoD (Hurt, 2016):

- Software assurance tools and techniques across DoD is inconsistent.
- Expertise of best practices is isolated in various programs.
- Cost of Software Analysis (SwA) tools and lack of general knowledge about their use hampers widespread adoption.
- SwA tools are not optimized for remediation of vulnerabilities by engineers.

1.2 OBJECTIVE OF THE DEMONSTRATION

The goal of the technology demonstration was to validate that the suite of LLNL-developed binary analysis tools (BATs) would identify firmware integrity issues that may indicate malware points-of-presence or vulnerabilities in the firmware binaries. The purpose in creating tools that can detect vulnerabilities and compromises is to introduce a capability to support operators in the inspection of firmware binaries prior to installation on DoD control systems. By intercepting the firmware prior to installation on a control system, the DoD could achieve a more robust security posture, thus making an adversary's intentions to degrade control system capabilities exceedingly more difficult.

1.2.1 Technology Demonstrated

A set of automated tools for the analysis of binary executables has been developed within an open source software analysis framework. This technology aims to mitigate cyber risk in the context of facility maintenance and protection of DoD base and facility substations as well as building automation systems. A process for the analysis of firmware upgrades to substation equipment has also been defined, which would result in improved supply chain integrity for mission-critical energy delivery systems.

There are 5 major functionalities, each incorporated into a separate binary analysis tool (BAT tool) that the technology to be demonstrated includes:

1. BAT-1: Anti-Disassembly Detection
2. BAT-2: Code Similarity Analysis
3. BAT-3: Unused Code Detection
4. BAT-4: Back Door Detection
5. BAT-5: Path Feasibility Analysis

1.2.2 Acceptance

The successful demonstrations of the technology showed how the 5 BAT tools mentioned above can help detect potential vulnerabilities within control system firmware. Such capabilities supplement the cyber risk-mitigation for Department of Defense – Energy and Water (DoD E&W) systems and supply-chain integrity strategies currently in place. The demonstration also illustrated the security gaps that currently exists within DoD E&W systems and processes. Illustrating both these gaps and a technology that fills these gaps helped to understand the need to incorporate this technology within the DoD.

1.2.3 Technology Transfer

In addition to demonstrating the technologies, LLNL produced two guides: *Firmware Management Best Practices for Energy Infrastructure Embedded Devices* and *Including Firmware in Vulnerability Assessment of Critical Energy Infrastructure*. These two guides stress the need to adopt a 'security lifecycle assurance approach' for ongoing management and maintenance of device firmware. The *best practices* guide discusses third-party and vendor provided tools to assist with firmware storage and organization as well as a process that include use of the BAT tools to lessen the risk of cyber vulnerabilities. The *assessment* guide discusses vulnerability assessment tools and techniques available that can help mitigate potential firmware security threats and stresses the importance of including firmware security assessments in base security plans as well as in an overall Cyber Security Management System.

The use of the newly developed BAT tools within these practices can be applied at all facilities that have E&W substations with similar firmware-controlled devices. The need for such a capability along with this process of tailoring to facility needs, as well as the mobility of the technology and process make transfer of this technology a straightforward process.

1.2.4 Findings and Guidelines

Given the DoD's desire to integrate cybersecurity standards into their research and procurement processes, a successful demonstration may provide to DoD a tool that improves supply chain integrity by analyzing device firmware before installing/upgrading. The use of the technologies and the tailoring of the procedure for inspecting firmware integrity could easily be integrated into the research and procurement processes.

1.2.5 **Validation**:

Because the DoD does not currently perform direct analysis of firmware integrity prior to installing it on substation devices, this demonstration showed how the use of these firmware analysis tools improve cyber risk analysis for E&W control systems over current practice.

1.3 DRIVERS

The Federal government has understood the importance of securing critical infrastructure assets since the Presidential Decision Directive (PDD) 63 was released in May of 1998. Since then, a myriad of Federal laws, regulations, and policies have laid the governance foundation for agencies to address and implement security programs, processes, and procedures relative to critical infrastructure assets. For example, the National Defense Authorization Act for Fiscal Year 2017, section 1650, mandates the evaluation of cyber vulnerabilities within the DoD critical infrastructure.

Recently, the focus sharpened on supply chain vulnerabilities, particularly software/firmware and the lifecycle thereof. As these concerns evolved, a need to address software assurance crystalized, and as such is defined in Department of Defense Instruction (DoDI) 5200.44 as:

> *The level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software throughout the lifecycle.*

These efforts moved beyond the need to mitigate supply chain concerns through vendor vetting alone, but also focused on the need to vet software/firmware through an inspection process prior to integration on any critical infrastructure system.

The DoD, in particular, was an early adopter of software assurance, during the development and sustainment phases as a function of its overall Software Assurance Initiative. The policies documents in particular that address the assurance of software within DoD include:

- DoD-specific actions:

- In October 2002, the President’s Critical Infrastructure Protection Board (PCIPB) created the National Security Agency (NSA) -led IT Security Study Group (ITSSG) to review existing IT acquisition processes.
- In July 2003, the Assistant Secretary of Defense for Networks and Information Integration [ASD(NII)] established the Software Assurance Initiative to examine software assurance issues
- In December 2004, Undersecretary of Defense for Acquisitions, Technology and Logistics [USD(AT&L)] and ASD(NII) established a Software Assurance (SwA) Tiger Team to:
  - Develop a holistic strategy to reduce SwA risks within 90 days
  - Provide a comprehensive briefing of findings, strategy and plan
- In May 2004, GAO reviewed how DoD mitigates risks from foreign suppliers of software, recommending that DoD make changes to its acquisition processes to better manage this risk (http://www.gao.gov/new.items/d04678.pdf).
  - In November 2004, GAO began study of Software Development Off-shoring to address risks of off-shoring to critical infrastructure, and steps being taking by federal government
- In November 2012, DoD issued DoD Instruction 5200.44: Protection of Mission Critical Functions to Achieve Trusted Systems and Networks addressing mission resiliency and software assurance.

- <u>Executive Orders</u>: EO 13800, EO 13636, EO 13587
- <u>Legislative Mandates</u>: National Defense Authorization Acts for Fiscal Years 2011-2017, Fixing America's Surface Transportation Act of 2016, Cybersecurity Information Sharing Act of 2015, Energy Independence and Security Act of 2007, Energy Policy Act of 2005
- <u>Agency Policy</u>: DoD Cyber Strategy, DoE Cyber Strategy, Department of Energy Office of Electricity Delivery and Energy Reliability Energy Sector Cybersecurity Framework Implementation Guidance
- <u>Regulations & Guides</u>: FERC Order 703, National Institute of Standards and Technology Framework for Improving Critical Infrastructure Cybersecurity, M-16-04 Office of Management and Budget Cybersecurity Strategy and Implementation Plan for Federal Civilian Government

2.0 TECHNOLOGY DESCRIPTION

2.0 TECHNOLOGY OVERVIEW

2.0.1 Description

Over the last 20 years Lawrence Livermore National Laboratories (LLNL) has developed an open-source framework called ROSE which among several capabilities, includes an infrastructure for static and dynamic analysis of source code and binary code, as well as supporting the development of custom analysis tools. Leveraging this framework, LLNL will develop and demonstrate a set of 5 Binary Analysis Tools (BATs) specifically for the purpose of automated cyber-risk analysis of firmware binaries. Each BAT analyzes a binary for the presence of specific potential cyber threats. Each BAT functions as follows:

- BAT-1 is used to detect the presence of anti-disassembly technologies, which are often correlated to advanced forms of malware.
- BAT-2 is used to analyze differences between two firmware releases. This is useful when analyzing a firmware update to determine what is being updated (and if the update contains the presence of malicious code).
- BAT-3 analyses a binary for the presence of unused code, which is often a mechanism to obscure the behavior and complicate the analysis of both source code and binaries.
- BAT-4 analyses for the existence of backdoors and the inputs required to exploit them.
- BAT-5 evaluates specific paths to protected resources in the firmware, which is meant to support an expert reverse-engineering analyst.

Each of these technologies were designed with 3 specific user-audiences in mind to perform specific actions based on the BAT output. These include the following:

- DoD Power Engineer, who uses the BATs before performing firmware updates
- DoD Protection Engineer, who analyzes specific firmware after BATs warns Power Engineer
- DoD Reverse Engineering Expert/ Firmware Vendor, who both perform deep dive analysis if protection engineer recommends further analysis.

Additionally, graphical user interface framework, named gROSE, has been developed for using these tools with different views aimed at these different classes of users.

2.0.2 Overall Schematic (or Flow of the System)

Figure 1 describes the Binary Analysis Tools (BAT) infrastructure and workflow. The following BATs have been developed within this project:

BAT-1: This is a tool for detecting the presence of anti-disassembly technologies. These technologies are extremely well correlated to both advanced forms of malware and IP protection—neither of which should be in firmware updates to power utility equipment.

BAT-2: This is a tool for analyzing differences between two firmware releases. The technology scales to handle realistic binaries and can be coupled with an analysis to evaluate each difference between any two binaries (such as those represented by a firmware update).

BAT-3: This is a tool for detecting the presence of unused code. The existence of unused code can be a mechanism to obscure the behavior and complicate the analysis of both source code and binaries.

BAT-4: This is a tool to detect the existence of backdoors and the inputs required to exploit them (supporting a vendor to do this analysis at the request of DoD).

BAT-5: This is a tool for evaluation of specific paths to protected resources in the firmware (supporting the expert reverse-engineering user to do analysis either internally or externally at the request of DoD).

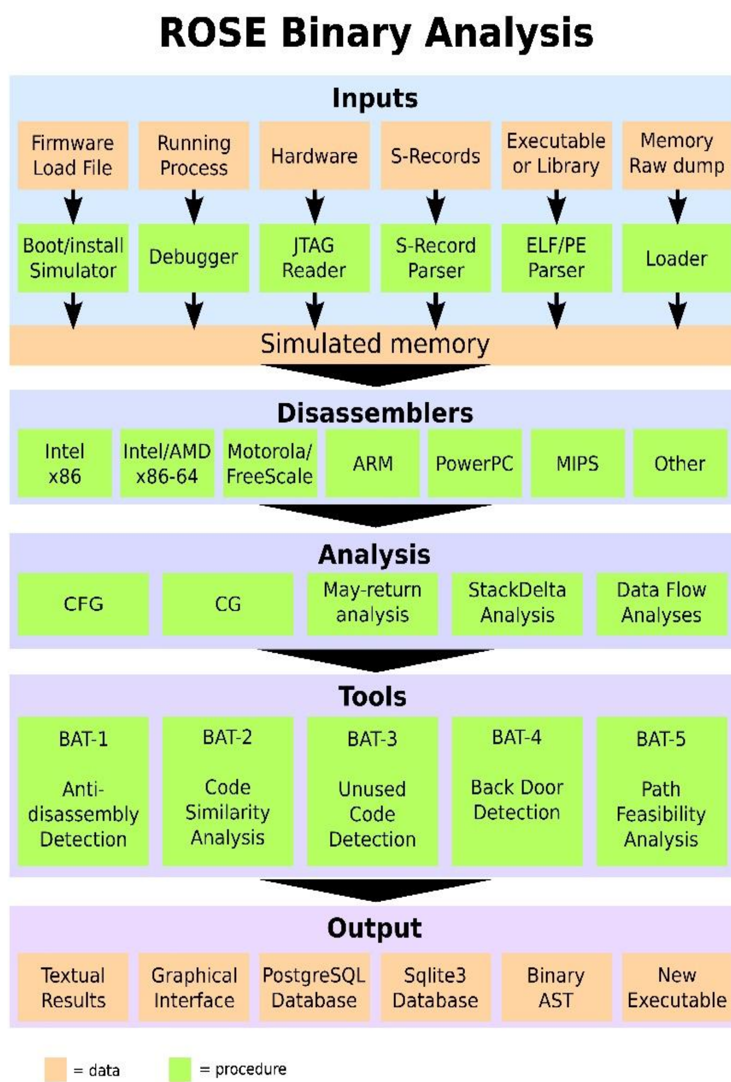

Figure 1: Binary analysis workflow within ROSE

2.0.3 Chronological Summary

The ROSE framework was initially developed, over a period of over 20 years, as an open source compiler infrastructure to build source-to-source program transformation and analysis tools for large-scale C, C++, UPC, FORTRAN, OpenMP, Java, Python, PHP, and binary applications. The primary goal of the ROSE project was to optimize applications within the DoE for High

Performance Computing and detect specified software features required to address complex computer architectures. Binary analysis support in ROSE has been developed for over the last 14 years.

In response to the DoD's Environmental Security Technology Certification Program's (ESTCP) "*Cybersecure Connectivity for Energy System Components and Military Installation Energy Infrastructure*" topic area, LLNL proposed a project to use the ROSE framework to develop BAT tools to address the DoD's need for firmware integrity analysis. These Binary Analysis Tools (BATs) have been developed using ROSE, but not release in the open source release of ROSE. BATs 1-3 were developed in year 1, with a GUI interface framework developed to support them in year 2, BATs 4 and 5 were developed over years 2 and 3. Significant testing was done over all three years.

Because the ROSE framework is already open source, the BAT technology could easily be transferred to open source tools as needed. Also, both ROSE and the BAT technology use a BSD license, which easily permits adoption and commercialization opportunities by third parties and therefore facilitates the transferability of the proposed technology.

2.0.4 Future Potential for DoD

This technology aims to improve supply-chain integrity for mission-critical energy delivery systems and automated building control systems. Where there is presently no process for the analysis of firmware upgrades to DoD substation and building automation equipment, LLNL has defined several, also showing how to integrate these into a more general security assessment of base infrastructure. LLNL has tailored the process to the requirements of the facility management of electrical substations and automated building control systems, providing DoD with enhanced security of its base electrical facilities and building controls systems. This work is replicable in all facilities that have electrical substations and building control systems with similar firmware-controlled devices. Both the technology and the process tailoring has built missing cyber defenses to complement the physical security of DoD's substation and building automation equipment.

2.1 TECHNOLOGY DEVELOPMENT

Development of the five binary analysis tools leveraged the functionality implemented in the ROSE framework as part of the Supply Chain Integration for Integrity (SCI-FI) project. The SCI-FI project developed a suite of open source capabilities to address supply chain integrity needs for end utilities, vendors, and chipset manufacturers. As part of this effort, the ROSE infrastructure was extended to enable analysis of source code and software binaries for malicious functionality. The success of these additional capabilities led to the concept of building an initial set of binary analysis tools to automate detection of specific types of vulnerability indicators in firmware.

Work began on integrating functionality for the detection of disassembly techniques into the first binary analysis tool, BAT-1. Although not part of the estimated development effort, it became evident that a basic graphical user interface would be needed to better show the use of the tools, and to allow the basic analysis mode of the tools to be the default behavior, while allowing more advanced analysis features to be activated when needed. Thus, the gROSE graphical user interface was developed and enhanced with increasing functionality to accommodate BAT tools 1, 2, and 3 during the project.

Throughout the project, firmware samples needed for testing and analysis were collected. One initial risk of the project, and firmware analysis in general, is the ability to 'unpack' the firmware in order to analyze the binary code. This capability was developed to allow for full automation of analysis and was incorporated as an automated feature in the gROSE user interface.

Based on the Demonstration Test Plan, a System Test procedure was developed by the PNNL team to be used for ongoing regression testing during tool development and to be used as part of an internal test/demonstration utilizing firmware from the PowerNet testbed at PNNL.

As initial work on BAT-1 concluded, incorporation of functionality for BAT-3, detection of unused code, began. The gROSE interface was very well received within the team and the concept of using gROSE to wrap different levels of functionality within the interface to address three different user types began. The team defined the three types of users:

1) Power Technician – the user that has the task of downloading and installing new firmware onto a power system device and is not assumed to have any binary analysis skills.
2) Power Engineer – an advanced user that would be able to drill a level deeper into the binary code and have the ability to identify where specific issues have been found and share with a more knowledgeable binary analyst – likely reaching out to the firmware vendor.
3) Cyber Analyst or Reverse Engineer – this user would be able to more fully understand the lower-level details the firmware analysis. The gROSE tools allows annotations to be preserved with a more hands-on analysis to allow documenting issues that can be re-examined with future version of similar firmware.

During 2017, BAT-2, code similarity analysis, was developed and initial functionality added to the gROSE user interface. Functionality to toggle different views relevant to user-type was also added to gROSE along with advanced capabilities to assist a cyber analyst user-type to navigate through disassembled code. Coinciding with SERDP and ESTCP 2017 Winter Symposium, the team visited the binary analysis research group at Army Research Laboratory with an objective of showing advanced features of the BAT tools. Valuable feedback on demonstration content and future tool ideas was captured and utilized in continuing development of the analysis tools and user interface. The gROSE interface was modified to simplify views for the field technician user-type along with an easier visualization of BAT-3's analysis of unused code. Basic scenarios for green light/red light analysis conditions were developed for future site demonstrations.

In 2018, specific firmware was gathered for Schwitzer devices specific to the ARL site as well as firmware for Automated Logic Corporation (ALC) devices in use at Elmendorf AFB. This included previous versions of these specific firmware specimens to needed to build collections that enable statistical difference analysis of new firmware versions.

Full system testing of the BAT tools began at PNNL based on the drafted System Test Plan. It was determined that the system testing procedure developed in the Plan should be implemented within the regular testing regime implemented by LLNL's BAT tool development process. This greatly streamlined the development/test cycle and allowed the PNNL team to focus on needed documentation.

Work in the 2018 year also included development efforts for BAT-5: Path Feasibility Analysis which provided needed functionality for BAT-4: Backdoor Detection. The initial draft of the Operational Usage Report was completed by the Boeing team providing critical input to the installation and usability of ROSE and the BAT tools.

The BAT tool suite was completed in 2019 with BAT tools 1, 2 and 3 fully integrated in the gROSE user interface. BAT 4 and 5 are more advanced tools targeting the reverse code engineer user-type. These last two tools are run from the command line with various options to guide the path analysis and backdoor detection capabilities.

This final year of the project saw demonstrations of the Binary Analysis Tools at both the Army Research Laboratory and Elmendorf Air Force Base. Demonstrations included:

- An overview of the project
- Demonstration of the firmware analysis tools
- An overview of best practices relative to firmware analysis

Focus-group discussions followed the formal demonstration of the tools to capture feedback on the BAT tool functionality and usage as well as future uses of the tools and potential additional capabilities.

The Boeing team contributed two documents: "Detecting Firmware Backdoors" an updated "Operational Usage Report" based on modifications made from feedback from the initial version of this document. The combined team completed the BAT tool User Guide. Building on the Best Practices Guide, the Livermore team authored: "Including Firmware in Vulnerability Assessment of Critical Energy Infrastructure."

2.2 ADVANTAGES AND LIMITATION OF THE TECHNOLOGY

2.2.1 Performance Advantages

The most important performance advantage these BATs provide is increased system security of DoD E&W control systems. Because no system is currently in place to inspect firmware within critical device infrastructure, this technology can be used to build missing cyber defenses to complement physical security of DoD substation equipment.

Additionally, because there is no automated tool that examines all five aspects of a firmware binary that the BAT technology does, the BAT technology is a huge performance advantage over current technologies in that the BAT technology is automated and targeted for use by power engineers who regularly upgrade firmware as needed, where as other current tools are often manual tools targeted at reverse engineering subject matter experts.

2.2.2 Cost Advantages

There are at least 2 important cost advantages to the BAT technology.

1. The software is covered by an U.S. Government unlimited use rights and a BSD license. The BSD license allows free redistribution of software in either source or binary format.
2. The software can be run on basic computer hardware that can be, but need not be, connected to DoD base network infrastructure. There is no need for specialized computing hardware.

2.2.3 Performance Limitations

The introduction of the usage of the BAT technology also introduces the need for added time for a Power Engineer to apply firmware updates. This added time comes from the need for a Power Engineer to first use the BAT technology to analyze the firmware update before applying it, then the time needed for an additional Protection Engineer analyst to review the findings of the BAT technology, if needed. The time for the Power Engineer to use the BAT technology may be close to negligible but should an additional Protection Engineer be needed to review the findings of the BAT technology; this adds considerable time to the process of applying firmware updates.

2.2.4 Cost Limitations

The primary cost limitation is the potential cost of a protection engineer to analyze firmware if the BAT tool indicates further analysis is needed. This may require additional staffing/contracting to properly support the LLNL proposed firmware update process.

2.2.5 Potential Barriers to Acceptance

One potential barrier to acceptance by Power Engineers is that the immediate benefits of BAT technology usage are difficult to see, and therefore the tool may be seen as a hindrance to the existing process and go unused. Because cyber-risk mitigation is often invisible (only noticeable when a system fails due to cyber-intrusion), using the BAT technology may seem like an inefficient additional step to firmware upgrades. Additionally, if the technology frequently produces inaccurate or confusing results, it may be deemed counterproductive. Finally, requiring the usage of a tool by Power Engineers that could increase the time needed to install firmware updates that could be critical may be seen as an additional frustration by field users. However, interactions with operational personnel indicate that there is more than enough delay between the collection of firmware and its installation, which can support an automated analysis of the software.

3.0 PERFORMANCE OBJECTIVES

This demonstration test is designed as a quantitative study focused on probability distribution, more specifically a binomial distribution. An evaluation of the test results using a probabilistic model is appropriate because this demonstration seeks to identify the percent probability each BAT tool can identify a given cyber-risk in a firmware binary. See Table 1 below.

Table 1: BAT Tool Performance Objectives

Performance Objective	Metric	Data Requirement	Probability Success Criteria ------------- Was the trial successful?	Confidence Interval ------------- How much confidence do we have in the outcome of the trial?	Confidence Level ------------- How much confidence do we have that the BAT demonstrated the objectives?
Quantitative performance objectives					
BAT1	% of code fragments	Records of anti-disassembly code in binary	≥ 75% Detection of Anti-Disassembly Code	≥75%	95%
BAT2	% of code fragments	Records of firmware modification in binary	≥75% Detection of Firmware Differences	≥75%	95%
BAT3	% of code fragments	Records of unused code in binary	≥75% Detection of Unused Code	≥75%	95%
BAT4	% of code fragments	Records of back doors in binary	≥75% Detection of Backdoor Exploitable Code	≥75%	95%
BAT5	% of control paths found	Records of control paths and protected resources	≥75% Detection of paths to protected resources	≥75%	95%
Qualitative Performance Objectives					
Usability	Narrative: Case Study	Interviews Documents, reports, observations	User recommendations, comments, observations, and documentation		

4.0 FACILITY/SITE DESCRIPTION

Multiple DoD sites were visited in the process for selecting a demonstration site. A basic description of candidate infrastructure systems with additional detailed information (e.g., vendors, products, system diagrams, documentation, etc.) pertaining to the electrical power distribution system (PDS) and the building automation system (BAS) at the military base was obtained for each site. For the PDS and smart metering systems, we sought a modernized system including protective relays and automation controllers with embedded microprocessors that execute operating system and control code firmware accessible and compatible with analysis tools within the ROSE framework. Similarly, for the BAS we sought a system using open industry standards and control protocols utilizing a secure operational approach and again employing embedded controllers applicable to ROSE firmware analysis.

It was also preferable, although not necessary, for the BAS at the demonstration site to use the BACnet protocol. BACnet is an open, object-oriented building automation industry protocol standard created to foster interoperability between BAS equipment vendors. For the purposes of this project, the use of the BACnet was a desirable criterion for selecting a suitable BAS system. Working with open and standards-based systems vs. closed and proprietary systems lowers project risk regarding availability of the specification if needed for clarification purposes during analysis of the technology.

After the preliminary on-site military base evaluations were concluded, a single fully qualified candidate did not emerge that could meet all the objectives outlined in the LLNL ESTCP project proposal in terms of the type of energy infrastructure equipment and associated software and firmware resident on the systems. Therefore, two sites were selected for demonstration purposes

4.0 FACILITY/SITE LOCATION AND OPERATIONS

4.0.1 Army Research Laboratory, Adelphi, MD

Army Research Laboratory (ARL) was selected as the candidate for demonstration of firmware analysis relating to the power distribution system as it houses a modern and accessible power distribution and an advanced metering infrastructure (AMI). ARL also has an active group of researchers in the area of binary analysis which will allow more in-depth demonstration of the BAT tool advanced features targeting the reverse engineer user-type.

Figure 2: Army Research Center NSRL Building, Adelphi Maryland

4.0.2 Elmendorf Air Force Base, Anchorage Alaska

Elmendorf Air Force Base was selected for demonstration of firmware analysis for BAS devices as the site has an extensive BACnet-based building automation infrastructure, an impressive buildings operations center implemented to provide a centralized system management capability, a cooperative building technician staff that provide base-wide system monitoring and both preventative and emergency maintenance of the BAS infrastructure, and an engaged and cyber-aware management that demonstrated a willingness to partner and incorporate the tools and techniques to augment their existing approach to BAS security.

Figure 3: Elmendorf Air Force Base, Anchorage Alaska

4.1 FACILITY/SITE CONDITIONS

4.1.1 Army Research Laboratory, Adelphi, MD

The Government Accountability Office's report to the Armed Services Committee of the U.S. Senate dated October of 2018 (www.acq.osd.mil/se/briefs/2018_21341_Shanahan_HwA.pdf), highlights considerable cyber concerns relative to Department of Defense's tactical mission systems. In response, ARL has turned its research resources towards these systems.

While ARL's current focus is on mission focused tactical systems, the cyber issues found in ICS firmware are quite similar to those found in many other firmware dependent devices. Given the nature of the problem, by extension, the LLNL binary analysis tools could be effective in the analysis of any device firmware. This discussion illustrated potential future research opportunities.

4.1.2 Elmendorf Air Force Base, Anchorage Alaska

Elmendorf AFB merged with the co-located U.S. Army at Fort Richardson in 2010. The combined base is referred to as the Joint Base Elmendorf–Richardson (JBER). However, the Air Force supports the entire energy infrastructure of the installation. They handle all the ICS issues on the Richardson (Army) side and inherited an older dial-up Energy Management Control System, including 71 disparate systems. These older systems are being integrated into the Automated Logic based Energy Infrastructure in a phased approach as funding and resources become available.

Due to the geographical location and criticality of mission, Elmendorf AFB is by necessity very self-sufficient. They exist in a tough climate and are not easily reachable; they cannot wait for assistance and additional resources when something breaks. For instance, they do not contract construction/repair of runways as they must remain operational at all times. Thus, they have gravel, asphalt and crews ready to pave and fix runways as needed. They estimate that they do more paving and repairs to runways during one season then most bases do in five years.

This need of self-sufficiency and a relatively isolated environment has contributed to Elmendorf's success of being a proving ground for new innovative processes and tools. They shared that their experience in demonstrating a project such as our firmware analysis system could be helpful in gaining acceptance with a wider community within DoD.

5.0 TEST DESIGN

This demonstration test is designed as a quantitative study focused on probability distribution, more specifically a binomial distribution. An evaluation of the test results using a probabilistic model is appropriate because this demonstration seeks to identify the percent probability each BAT tool can identify a given cyber-risk in a firmware binary.

5.0 CONCEPTUAL TEST DESIGN

The demonstration test design meets the following criteria for the utility of a binomial distribution:

- The demonstration test had a fixed number of trials for each BAT
- Each trial is independent of the others
- There were only two possible outcomes for each trial (did technology succeed y/n?)
- The probability of each outcome remains constant from trial to trial

This demonstration test used a binomial distribution as its statistical model to determine the probability of successful detection of a cyber-risk by each tool and the Clopper-Pearson confidence interval of each measurement to ensure the measures fall within an acceptable rate of error.

5.1 SAMPLING PROTOCOL

The demonstration test design meets the following criteria for the utility of a binomial distribution:

- The demonstration test has a fixed number of trials for each BAT
- Each trial is independent of the others
- There are only two possible outcomes for each trial (did technology succeed y/n?)
- The probability of each outcome remains constant from trial to trial

This demonstration test consisted of binomial distribution as its statistical model to determine the probability of successful detection of a cyber-risk by each tool and the Clopper-Pearson confidence interval of each measurement to ensure the measures fall within an acceptable rate of error.

Binomial Distribution

The binomial distribution formula is:

$b(x; n, P) = {}_nC_x * P^x * (1 - P)^{n-x}$

Where:

b = Binomial probability
x = Total number of successful trials (pass/fail)
P = 0.5 (probability of success of a two-tailed, none-directional: pass/fail)
n = 15 (number of trials)
${}_nC_x = n!/x!(n-x)!$
Clopper-Pearson Confidence Interval (Mayfield, 2013):

$$\sum_{k=0}^{k} \binom{n}{k} {p_{UB}}^k (1 - p_{UB})^{n-k} = \frac{\alpha}{2}$$

$$\sum_{k=x}^{n} \binom{n}{k} {p_{LB}}^k (1 - p_{LB})^{n-k} = \frac{\alpha}{2}$$

Interval from p_{lb} to p_{ub} where:

p_{lb} is the confidence interval lower bound
p_{ub} is the confidence interval upper bound
n is the number of trials
k is the number of successes in n trials
α is the percent chance of making a Type I error, 1-α is the confidence

5.2 TEST ARTICLES AND TEST APPROACH (SAMPLING RESULTS)

5.2.1 BAT-1 - 10 Trials with Inject Intervention, 5 Trials with no Intervention

- Description: Detection of anti-disassembly technology

- Purpose: Anti-disassembly technologies are typically used to obfuscate the functionality of binary code. These technologies are often correlated with both advanced forms of malware and IP protection, neither of which are appropriate for firmware updates to power utility equipment. The purpose of this demonstration is to show the use of the BAT1 tool in detecting the presence of anti-disassembly technologies and thereby mitigate cyber risks and reduce vulnerability to power grid disruptions.

- Metric: Percentage of correctly discovered anti-disassembly technologies.

Test Input: The test article interventions include:

  - Bad Instruction
  - Block Scatter
  - Call-Branch
  - Function Pointer
  - No-Op
  - Opaque Predicate
  - Overlap Instructions
  - Stack Delta
  - Strange-Call
  - Strange-Return

Test Methodology: LLNL used disassembled firmware (from the suite of disassembled firmware at LLNL and PNNL) and presented this firmware to the BAT1 enabled ROSE tool. Next, LLNL will inject the same firmware with anti-disassembly technologies from the list above. LLNL then rewrote the binary to include anti-disassembly and reran the BAT enabled ROSE tool. The results of both BAT1 tests were juxtaposed and analyzed.

5.2.2 BAT-2 - 10 Trials with Inject Intervention, 5 Trials with no Intervention

- Description: Function-level differences between 2 binaries.

- Purpose: BAT2 tools analyzes differences between two firmware releases. The purpose of this demonstration is to show the effectiveness of the BAT2 tool in detecting function-level differences between 2 binaries.

- Metric: Percent of discovered code similarity at the function level

- Test Input: Firmware with differences, and firmware without differences

- Test Methodology: The test team will take a specific firmware version and present this firmware to the BAT2 enabled ROSE tool. The test team will take the selected firmware version and modify it as follows:
    - Remove some functions and references to those functions.
    - Insert new functions and some references to those functions.
    - Replace an existing function with some other function.
    - Change the order of some functions but otherwise leave them the same as much as possible

  The test team then used this altered version of the firmware to the BAT2 enabled ROSE tool. The results of both BAT2 tests were juxtaposed and analyzed. The tests also included a sensitivity test to determine the minimum degree of change before the change is detected and identified.

5.2.3 BAT-3 - 10 Trials with Inject Intervention, 5 Trials with no Intervention

- Description: Detecting unused assembly code.

- Purpose: Much like anti-disassembly, the existence of unused code in a binary can be a mechanism to obscure the behavior and complicate the analysis of binary code. However, extraneous segments of unused code require a very different type of analysis as opposed to an anti-disassembly technique as is the focus of BAT1. In its most simple case, unused code segments must be identified by first determining valid start points in the code. Then, following paths from these start points, identify code segments that are not reached – this is unused code. The purpose of this demonstration is to show the use of the BAT3 tool in detecting the presence of unused code thus revealing potential cyber risks.

- Metric: The percentage of discovered unused code within the firmware binary.

- Test Input: A binary containing previously known and identified unused code and a binary with no unused code.

- Test Methodology: The test team used selected firmware and presented this firmware to the BAT3 enabled ROSE tool. The test team rewrote the same firmware to have unused code. The test team used this firmware with unused code to the BAT3 enabled ROSE tool. The results of both BAT3 tests were juxtaposed and analyzed.

5.2.4 BAT-4 - 10 Trials with Inject Intervention, 5 Trials with no Intervention

- Description: Detecting the existence of firmware/binary backdoors. A "backdoor" is a bypass mechanism that causes execution to reach a protected resource without going through the usual authentication and/or authorization step. BAT-4 goes hand-in-hand with BAT-5 in identifying reachability from a backdoor to specified point in the firmware. Typically, reachable points of interest will be functions that control/modify protected resources of a device.

- Purpose: The existence of “backdoors” in a binary is usually indicative of a bypass mechanism placed either by developers, or by malware to bypass internal protections. The purpose of this demonstration is to show the use of BAT-4 to assist a software analyst in determining whether a protected resource can be reached without going through an authentication and/or authorization step.

- Metric: The percent of backdoor mechanisms that are successfully identified.

- Test Input: A binary with backdoors and to have pre-knowledge of what backdoors are present within the binary.

- Test Methodology: The test team used collected firmware and manually identified a protected resource and an authorization step in the firmware, as required these were inserted into the firmware samples. The firmware and locations were presented to the BAT-4-enabled ROSE tool to determine whether it identified the control flow path that bypasses the authentication and/or authorization.

5.2.5 BAT-5 - 10 Trials with Inject Intervention, 5 Trials with no Intervention

- Description: Path evaluation relative to protected resources in firmware.

- Purpose: Within device firmware and binary executables, knowing the feasible control flow paths from one point of execution to another, and the preconditions that cause such a path to be taken is valuable knowledge to a software analyst in order to understand potential threat vectors and how to defend against them. This tool is meant to statically enumerate paths, evaluate their feasibility, and indicate path preconditions

- Metric: The percent of possible paths detected, the correct determination of whether the path is feasible, and the correctness of preconditions for paths between two selected endpoints in the control flow graph.

- Test Input: Firmware with pre-knowledge of existing and feasible paths between two selected endpoints.

- Test Methodology: The test team selected from collected firmware binaries and defined pairs of instructions (endpoints) in those binaries. Ground truth was obtained by manual analysis of the paths. The firmware and chosen endpoints were presented to the BAT5-enabled ROSE tool, the results of which were evaluated alongside the manual analysis.

5.3 DEMONSTRATION SCENARIOS

To show the typical use of the tools, common scenarios were presented as part of the demonstration. First, the most likely scenario that the firmware to be installed is that it contains no unexpected anomalies per analysis results from the BAT tools and installation/upgrading of device firmware can proceed as normal. This is the 'green-light' scenario.

5.3.1 Power Technician – Green Light

A family of related firmware, typically previous version of the device firmware, have been analyzed and are put into a 'collection' via the gROSE user interface. The new version of firmware is analyzed and a statistical comparison to the collection is made. If the resulting analysis is statistically the same as the collection, i.e. within one standard deviation from accumulated results, then the firmware is determined to not need further investigation. Figure 4 shows the flow of this analysis with a sample firmware named 'false'.

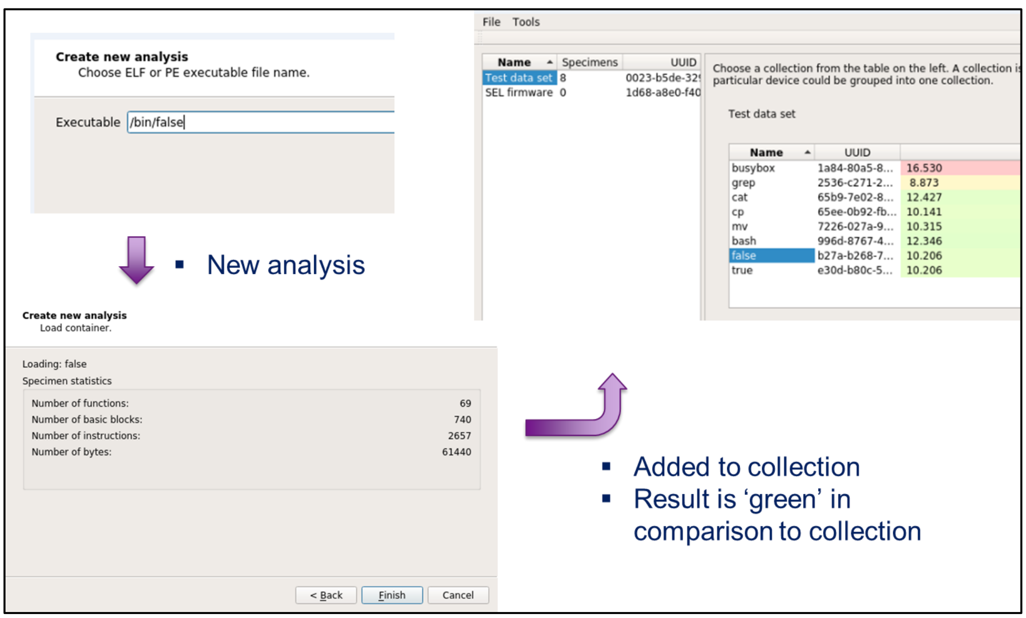

Figure 4: Example 'green light' analysis scenario.

5.3.2 Power Engineer – Red Light

If the analysis of the firmware sample is statistically different then a collection of previous versions of the firmware, the gROSE interface will tag the firmware with a red background indicating further analysis should be conducted before potentially installing/updating the device. Figure 5 shows advanced capability to analyze a firmware sample named 'busybox'. As the automated analysis within the gROSE interface highlights the sample red, an advanced user can drill into function(s) that have high counts of detected anti-disassembly methods within the binary.

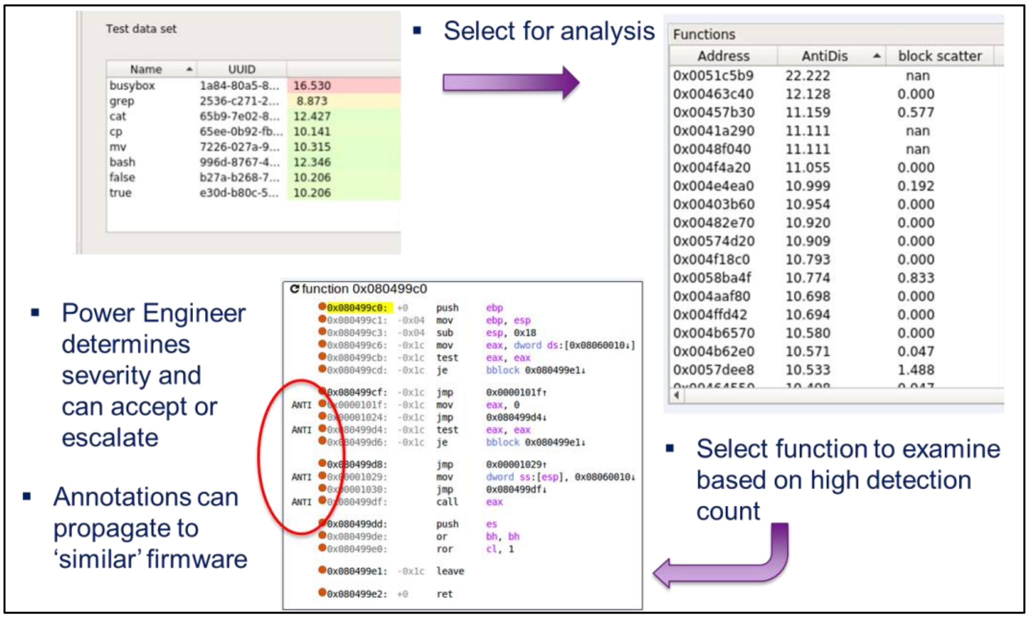

Figure 5: Example 'red light' analysis scenario.

6.0 PERFORMANCE ASSESSMENT

6.0 ARMY RESEARCH LABORATORY DEMONSTRATION

On May 21, 2019, the LLNL led team conducted a Technology Capability Demonstration at the Army Research Laboratory (ARL) in Adelphi, Maryland. The demonstration consisted of:

- overview of the project
- demonstration of the firmware analysis tools
- overview of best practices relative to firmware analysis
- focus-group discussion around the tools and impressions and potential future uses of the tools

The ARL team's participation was organized by Michael Weisman, PhD, and Charles Kamhoua, PhD.

6.0.1 Focus group discussion

Following the project overview and the demonstration, a mini focus group question and answer session was held with five ARL researchers. The main topics resulting from the focus group included:

1. ARL's shift and concentration on tactical systems, and specifically firmware contained therein.
2. Levels of on-site expertise required to fully leverage the capabilities of the binary analysis tools.
3. Triangulation of additional system data points relative to the outputs of the binary analysis tools.
4. Concolic (i.e. concrete and symbolic) testing (6.0.5 Concolic Testing below).

6.0.2 ICS Cyber vulnerabilities vs. tactical systems—firmware is contained in both types of systems

The Government Accountability Office's report to the Armed Services Committee of the U.S. Senate dated October of 2018 (www.acq.osd.mil/se/briefs/2018_21341_Shanahan_HwA.pdf), highlights considerable cyber concerns relative to Department of Defense's tactical mission systems. In response, ARL has turned its research resources towards these systems.

While ARL's current focus is on mission focused tactical systems, the cyber issues found in ICS firmware are quite similar to those found in many other firmware dependent devices. Given the nature of the problem, by extension, the LLNL binary analysis tools could be effective in the analysis of any device firmware. This discussion illustrated potential future research opportunities.

6.0.3 Levels of expertise required to execute the LLNL BATs

The consensus of the focus group was that a fair degree of understanding around firmware analysis may be required to use the BAT's effectively on site. In general, OT operators found in the ICS space may not have received cybersecurity training and may not understand the nature of the BAT outputs. However, as currently planned, the release of the BATs does include a comprehensive user guide as well as training. It may be necessary to include training relative to general cybersecurity as a component of the BATs release or more specifically, training around the nature of the events alerted on by the BATs.

6.0.4 Triangulation of data points

To understand the scope of a cyber-attack, triangulation of data points coming from various devices as well as potential outputs from the devices themselves was expressed as a desirable capability by ARL researchers.

Identifying a potential anomaly relative to a binary is a data point that requires researchers to understand various other aspects of an attack. Being able to assign a temporal timeline to an attack as well as understanding specifically what a device is processing could help researchers identify the maturity of an attack, which in turn could help all stakeholders identify the scope of the attack. Understanding the scope ultimately allows organizations to do an impact analysis.

In addition to having the ability to put the binary analysis outputs into a triangulated context, ARL researchers expressed a desire to understand the "physics" of the devices. These characteristics could be obtained through probing of sensors.

However, for a binary to generate an output, such a binary would require a redesign and re-write. Such a re-write would be beyond the scope of the LLNL binary analysis tools.

6.0.5 Concolic testing

Concolic testing relative to firmware code surfaced in the course of the discussion as a potential research opportunity. Concolic testing, as understood by the participants involved a software verification technique that worked with program variables as symbolic variables, in the context of a specific, concrete execution path.

When considering the potential cyber threats to firmware, the purpose of concolic testing would be to maximize code coverage with the intention to identify anomalies as opposed to validating the correct operations of the program. This type of testing would be a particularly interesting approach towards firmware analysis for the purposes of identifying cybersecurity issues.

6.1 ELMENDORF AIR FORCE BASE DEMONSTRATION

On July 7, 2019, the LLNL led team conducted a Technology Capability Demonstration at the Elmendorf Air Force Base in Anchorage, Alaska. The demonstration consisted of:

- An overview of the project.
- Demonstration of the firmware analysis tools with focus on pass/fail scenarios.
- An overview of best practices relative to firmware analysis.
- Focus-group discussion around the tools, impressions, and potential future uses of the tools.

The Elmendorf AFB team's participation was organized by John Elsholz, Energy Management Control System (EMCS) Supervisor, and Bill Farabaugh, Chief of Infrastructure Systems. Participants included several members of the EMCS staff, and representatives the base Energy Infrastructure support contracting agency, Meridian Systems.

6.1.1 Focus group discussion

Following the project overview and the demonstration, a mini focus group question and answer session was held with four U.S. Air Force technicians, and two on-site support personnel from Meridian Systems. The main topics resulting from the focus group included:

1. Necessity of Elmendorf AFB self-sufficiency.
2. Methods of integrating the BAT tools into the existing process.
3. Future tools including concept of concolic testing.
4. New infrastructure and energy networks being implemented.

6.1.2 Elmendorf AFB self-sufficiency

Elmendorf AFB merged with the co-located U.S. Army at Fort Richardson in 2010. The combined base is referred to as the Joint Base Elmendorf–Richardson (JBER). However, the Air Force supports the entire energy infrastructure of the installation. They handle all the ICS issues on the Richardson (Army) side and inherited half an older dial-up Energy Management Control System, including 71 disparate systems. These older systems are being integrated into the Automated Logic based Energy Infrastructure in chunks as funding and resources become available.

Due to the geographical location and criticality of mission, Elmendorf AFB is by necessity very self-sufficient. They exist in a tough climate and are not easily reachable; they cannot wait for assistance and additional resources when something breaks. For instance, they do not contract construction/repair of runways as they must remain operational at all times. Thus, they have gravel, asphalt and crews ready to pave and fix runways as needed. They estimate that they do more paving and repairs to runways during one season then most bases do in five years.

This need of self-sufficiency and a relatively isolated environment has contributed to Elmendorf's success of being a proving ground for new innovative processes and tools. They shared that their experience in demonstrating a project such as our firmware analysis system could be helpful in gaining acceptance with a wider community within DoD.

6.1.3 Methods of integrating the BAT tools into current process

There was much discussion around different ways in which the tools could be integrated in the process for installing/upgrading firmware on energy devices. Elmendorf AFB noted that these tools could be used on a local level but also noted the very large number of devices on their site. It became clear that the analysis of firmware does not need be done during the actual update process, but instead (and preferably), done independently and in advance of the actual upgrade event. There was also mention of a sandbox network environment, implemented by Elmendorf's control systems staff, that has the potential to be used to trial the BAT tools' capabilities.

6.1.4 Future tools including concolic testing

Discussion of the current and future tools started with noting that very few tools exist for looking at the 1's and 0's in a test/validation environment. Source code is typically the focus during development, and this fails to include the inspection of compiled dynamically linked and/or shared libraries—one of the supply-chain risks that this project addresses. There was much in-depth discussion on how BAT1 (i.e. anti-disassembly detection) worked based on statistical analysis of a related family of firmware. There was also increase interest in BAT4 (i.e. path feasibility analysis) and its use of an SMT (Satisfiability Modulo Theories) solver to prove feasibility from Point A to Point B in a binary while also providing the parameter values needed to instantiate the path.

While discussing details of the currently developed BAT tools, conversation led to the description a set of new tools being prototyped for a different project at LLNL. These tools address a set of the most common vulnerabilities found in binary code. It was agreed that these new tools would be a good addition to the current set of BATs as they could indicate a pass/fail (e.g., green flag/red flag) on one analysis pass of a firmware sample without needing history of related samples.

The LLNL-led team also described the concept of concolic testing for binary code that is an approach to identify non-specific general vulnerabilities—those not specific to any one vulnerability. It combines static and dynamic analysis. The group found the capabilities of this tool very interesting as it is similar to an extremely smart fuzz tester—using a static analysis of the binary firmware as a mechanism to define inputs, tracking inputs as concrete executions in the binary, using symbolic analysis to define types of scenarios to explore tests to make the target application fail, and recording the failures. This tool is different from all the other BAT tools in that it is not just a static analysis tool.

6.1.5 New infrastructure and energy networks being implemented

This project set out to demonstrate both a new technology and a process for applying this technology to mitigate cyber risk within the context of facility maintenance and protection within DoD base electrical substations and building automation systems. In going forward with incorporating of the remainder of the Richardson Army base systems into the Elmendorf AFB system it has been decided to maintain a 'single campus' for the BAS system with Automated Logic as the vendor. The Elmendorf team also shared that they will be implementing newer power distribution systems with SEL devices. It was also discovered that our host, John Elsholz has a seat on a DoD panel for cyber security maintenance.

6.2 SUMMARY ASSESSMENT

Both the ARL and Elmendorf AFB demonstrations were successful in showing the viability of the BATs as well as their effectiveness in detecting potential issues in the firmware that warrant further investigation prior to installing into an electric grid device. The demonstrations successfully illustrated how each BAT functions, the outputs they produce and how each user group would leverage the tools in their respective environments. The concept of potentially escalating a firmware analysis question to the responsible vendor was met with understanding and agreement. As most sites will not have advanced binary analysis expertise it was agreed that the firmware vendor should be responsible and willing to explain any anomalies encountered in the analysis.

Much of the follow-on discussions were focused on how these tools could be used in each specific base while integrating into a firmware update procedure. It was generally agreed that a site-by-site implementation is not scalable and not the correct approach to instantiate a capability for the wider DoD community. Although Elmendorf AFB was amenable to trialing the BATs system in a sandbox environment as a proving ground for applicability to a wider audience.

Follow on research opportunities were identified in the area of firmware analysis. Several new tool concepts were discussed, and much interest was conveyed to the development of tools that could detect specific vulnerabilities in a one-pass analysis (e.g., in combination to the currently required family history of specific firmware for a statistical analysis). There was more in-depth discussion on the concept of concolic testing of firmware for detecting general vulnerabilities.

The demonstration at Elmendorf AFB was particularly successful in that the group included two of our three user roles. The demonstration and discussion lasted for several hours with full engagement from everyone involved. The Elmendorf team was genuinely excited to test a capability for analyzing firmware for potential vulnerabilities prior to installing on their energy control infrastructure.

7.0 COST ASSESSMENT

Our cost model addresses the requirements to support testing at individual DoD sites. Since the software is made available to DoD at no cost, only the computer and training are dominate values in the cost model.

7.0 COST MODEL

Table 2. Cost Model for single deployment of Binary Analysis Tools

Cost Element	Data Tracked During the Demonstration	Estimated Costs
Hardware capital costs	Estimates made based on component costs for demonstration	$10K
Installation costs	Labor and material required to install	$2K
Consumables	Estimates based on rate of consumable use during the field demonstration	$0K
Facility operational costs	Reduction in energy required vs. baseline data	N/A
Maintenance	• Frequency of required maintenance • Labor and material per maintenance action	System Administration of Desktop Computer
Hardware lifetime	Estimate based on components degradation during demonstration	5 years
Operator training	Estimate of training costs	$7K

This cost model includes a conventional desktop computer, installation, and ongoing system administration. We expect the computer to last about five years, and training in how to use the software be about $7K (operator time and 2 days training on site). BAT tools are provided free to the government, upgrades as they occur, will be free as well. BAT tools are being used in other projects and as a result have been extended in several ways specific to those DoD projects.

7.2 COST DRIVERS

The cost drivers will be the training and turnover of personnel on site that would drive the training costs.

7.3 COST ANALYSIS AND COMPARISON

Operational use of the software requires only hardware, training and time requirements as outlined in the cost model. The protracted use of our software to test firmware has only a time requirement, and as a result is simple to evaluate. In addition, the software can be setup to run on firmware repositories to provide more automated testing. Evaluation of results depends on the depth to which operators wish to peruse flagged warnings. The evaluation of such warnings is intended to quickly escalate the evaluation to more expert reverse engineers available at other facilities within DoD.

The largest contributing factor to the cost is operator turn-over which would drive the training cost only. The testing process for firmware does not replace or supplant any existing technologies, since firmware is notoriously not tested before being loaded onto critical hardware systems. Since the use of testing for malicious firmware is low compared to the extraordinary and incalculable costs of a malware-based attack via firmware, the estimate of return on investment is especially difficult to quantify.

8.0 IMPLEMENTATION ISSUES

8.0 GENERAL OBSERVATIONS

This section outlines observations of the operational usage of the tools. Facets of operational usage have been identified as pertinent to organizational costs in terms of time and training. This section details usability in terms of those identified facets with qualitative commentary from Boeing Cyber Engineers, evaluation in the context of current industry process, and commentary on competing tools.

8.0.1 Legal Issues

The Digital Millennium Copyright Act (DMCA) provides for the analysis of software when done for security purposes, so that evaluation of firmware using our tools would be provided for, but mostly regulated by this act. The software is available via BSD license and at no cost to the government. Reproduction is the cost of making a copy or downloading a new version. The dominate cost is hardware, training and operator time, user documentation as already been assembled at a fixed cost and is also no-cost to the government.

What was learned from the demonstrations at two DoD sites is that there is strong support for testing of firmware compared to the current state-of-the-art which defaults to no testing. The cost was low because both sites already had computers readily available that could run the software, and the only obstacle was initial training and time to run the software on firmware sample (which can run overnight if desired). Given the especially low barrier to analyzing firmware, the work to support ongoing analysis was advantageous compared to the risk of a cyber-attack on base facility equipment.

8.0.2 Installation

The installation of the BAT suite has improved significantly since the midyear review. While still a time-consuming process, it should have minimal if any impact to production since the analysis is expected to occur in an offline environment on a research computer that meets the expected computing requirements. The tools support installation from binaries on two common builds and variants of the Linux operating system.

This installation is dependent on a significant number of specific external Linux libraries that fluctuate in availability and functionality making support for Linux variants more challenging. If these libraries are unavailable in the installation environment, such as in spaces requiring explicit scrutiny and approval for each dependency, functionality may be reduced or entirely halted until all dependent libraries in the supported version are accessible. Due to these constraints the tools were only successfully installed with full functionality on Linux Mint and Ubuntu systems.

Because installation costs are incurred in a singular instance prefacing usage of the BAT suite, the installation procedure is not directly pertinent to operational costs and is of low criticality in considerations of this report.

8.0.3 Pre-Disassembly

Due to the varied number of firmware binary architectures, BAT usage requires pre-processing of a binary prior to disassembly. Binary disassembly is a manual process in the BAT suite for firmware types other than Windows Executables, known as Portable Executable (PE) files, or Linux Executables, known as Executable and Linkable Format (ELF) files, due to current ROSE tool limitations. More assembly types have been added, such as firmware for Schweitzer Engineering Laboratories (SEL) devices, but this may only cover an estimated 6% of firmware architecture types. However, these firmware types are assumed, in terms of total existing binary coverage, to cover the majority of firmware in organizational use. New firmware types are expected to require manual disassembly.

This pre-processing is an intense process in terms of both requisite time and skill and likely necessitates the involvement of a seasoned reverse engineer. Pre-processing, as a prerequisite setup step for disassembly and analysis of a firmware, is of critical import to usage of the BAT suite. However, because pre-processing is assumed to be performed once per binary, the procedure does not affect operational usage and remains transparent to the field engineers.

Though manual disassembly and code base contributions are not abnormal tasks within open source communities, manual pre-processing of candidate firmware could render industry endorsement of the BAT suite less likely. Though not pertinent to recurring operational costs, perceived high setup costs can pose a deterrent to the adoption of the BAT suite for vendors employing a variety of firmware binaries not automatically pre-processed by the tools.

If field engineers must manually pre-process candidate firmware or escalate analysis to a protection or cyber engineer due to the demands of pre-processing, rather than the analysis remaining transparent to them, this could make adoption more difficult. The assumption that pre-processing imposes an infrequent demand for each vendor depends on whether the particular vendor is prone to new equipment purchases or if the equipment selected is subject to frequent major version changes to the installed firmware.

8.0.4 De-Obfuscation

Unlike BAT-1 which detects firmware binary obfuscation, general usage of BATs 2-5 could be affected negatively unless a binary has been firstly de-obfuscated. As firmware developers continue to implement security measures to protect intellectual property and to ensure cyber security, obfuscation of firmware binaries is expected to become more common, creating a demand for a method of de-obfuscating or unpacking firmware binaries as a prerequisite for general use of the BAT suite.

As with pre-processing, however, whenever obfuscation is necessary, it should not affect operational usage as this process would occur offline, and this process should only occur once per binary. However, the mode of obfuscation is not contingent to the type of the firmware, as multiple obfuscation techniques can be employed by a particular vendor or even for a particular binary. Furthermore, the selected obfuscation techniques can be altered more easily than a vendor can change the type of firmware, which, depending on the alterations, may make de-obfuscating subsequent versions of a binary as challenging as de-obfuscating the initial version. Because the requisite technical skillset to manually perform de-obfuscation is singularly held by the cyber engineer, organizations may find it cost prohibitive to analyze obfuscated binaries without a feasible automatic or outsourced option.

8.0.5 Requisite Knowledge

To analyze a binary, the BAT suite must have knowledge of the CPU architecture for which the firmware was compiled, whether ARM, MIPS, PowerPC, x86_64, etc. This information is not identified by ROSE and must be entered manually by the user, requiring a high degree of operational understanding to know the specific architecture of the firmware. This step, however, does not affect the operational time required for a field engineer's usage of the tools.

It is possible that a typical software user, such as a field engineer, lacks this type of knowledge, necessitating additional training or staffing particularly if multiple architecture types are being serviced. Cyber engineers, with greater familiarity of specific architecture information, require less, if any, additional training.

8.0.6 Analysis Time

The time requisite to analyze a firmware binary depends both on the size of the binary and on the processing capabilities of the machine hosting the BAT suite. On a personal workstation hosting the BAT suite, a firmware binary exceeding several megabytes in size often requires multiple days for the BAT suite to perform analysis. When executing the BAT suite on server-grade hardware as recommended in the tool's documentation, the processing time for the same binary can be reduced to multiple hours.

Lengthy processing times can contribute to aversion to adoption or employment of the tool in environments where powerful servers are not already installed for other tasking. This aversion could be offset by recommending or making available the use of cloud computing environments.

8.0.7 Feature Accessibility

The Graphical User Interface (GUI) for the BAT suite, a visual representation of relevant determinations of the BAT suite, is intended for use by both advanced operators, such as cyber engineers, and standard operators, such as field engineers. The most recent BAT suite GUI has made the analysis of firmware binaries and identification of detected issues much more feasible.

- The GUI contains a display that details the determined obfuscation issues found correlated to specific assembly sections and percentages of assembly containing obfuscation. Cyber engineers can use this display to identify specific areas of interest for de-obfuscation.
- The GUI contains a display that details function-specific similarities between binaries. Engineers can compare the panels to identify points of divergence in specific functions when performing analysis, illustrating the introduction point for potentially problematic sections.
- The GUI contains a simplified display for operators like field engineers who do not perform detailed analysis as a function of their job. The simplified view presents only a few data items, such as a single numeric value for obfuscation percentage with a color indicator for predetermined obfuscation thresholds and a similarity percentage for comparison between two binaries.

9.0 REFERENCES

Adm. Mike Rogers, C. U. (2015). *Statement of Admiral Michael S. Rogers, Commander USCYBERCOM before the House Committee on Armed Services.* Washington, D.C.: U.S. House of Representatives.

Department of Defense. (2012). *DoD Instruction 5200.44: Protection of Mission Critical Functions to Achieve Trusted Systems and Networks.* Washington DC: Department of Defense.

Donavan, S., & Scott, A. (2015). *Cybersecurity Strategy Implementation Plan (CSIP) for the Federal Civilian Government: Memorandum for Heads of Executive Departments and Agencies.* Washington DC: United States, Executive Office of the President.

Government Accountability Office. (204). *GAO-04-678: Defense Acquisitions, Knowledge of Sortware Suppliers Needed to Manage Risks.* Washington DC: Government Accountability Office.

Hurt, T. D. (2016, October 24-27). DoD Joint Federated Assurance Center (JFAC) Update. *19th Annual NDIA Systems Engineering Conference*. Springfield, VA: DOPSR.

James Clapper, D. N. (2015). *Statement for the Record Worldwide Cyber Threats House Permanent Select Committee on Intelligence.* Washington, DC: U.S. House of Representatives.

Manz, D. O., Masica, K., Miller, C. H., Munro, J. K., Pleszkoch, M., & Quinlan, D. (2016). *Supply Chain Integration For Integrity: Final Report.* Richland: Pacific Northwest National Labortory.

Masica, K. (2019). *Firmware Management Best Practices for Energy Infrastructure Devices.* Livermore: Lawrence Livermore National Laboratory.

Masica, K. (2019). *Including Firmware in Vulnerability Assessment of Critical Energy Infrastructure.* Livermore: Lawrence Livermore National Laboratory.

Mayfield, P. (2013, UNK UNK). *SigmaZone*. Retrieved November 22, 2017, from http://www.sigmazone.com/binomial_confidence_interval.htm

National Security Council and National Security Council Records Management Office. (1998). *PDD-63 - Critical Infrastructure Protection, 5/20/1998.* Little Rock: Clinton Presidential Library & Museum.

Obama, B. (2011). *Executive Order 13587 -- Structural Reforms to Improve the Security of Classified Networks and the Responsible Sharing and Safeguarding of Classified Information.* Washington DC: White House.

Obama, B. (2013). *Executive Order 13636: Improving Critical Infrastructure Cybersecurity.* Washington DC: White House.

Trump, D. J. (2017). *Executive Order 13800: STRENGTHENING THE CYBERSECURITY OF FEDERAL NETWORKS AND CRITICAL INFRASTRUCTURE.* Washington DC: White House.

U.S. Department of Energy. (2015). *Energy Sector Cybersecurity Framework Implementation Guidance.* Washington DC: U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability.

APPENDIX A: POINTS OF CONTACT

POINT OF CONTACT Name	ORGANIZATION Name Address	Phone Fax E-mail	Role in Project
Dan Quinlan	Lawrence Livermore National Laboratory	quinlan1@llnl.gov	Principal Investigator
Lawrence Banks	Lawrence Livermore National Laboratory	lebanks@llnl.gov	Project Manager
Ken Masica	Lawrence Livermore National Laboratory	masica1@llnl.gov	Lead Engineer
Nick Multari	Pacific Northwest National Laboratory	Nick.Multari@pnnl.gov	PNNL Team Lead
Ryan Hammond	Boeing Company	Ryan.D.Hammond@boeing.com	Boeing Team Lead
Michael Weisman	Army Research Laboratory	michael.j.weisman2.civ@mail.mil	ARL POC Computation and Informational Sciences
John Elsholz	Elmendorf Air Force Base	john.elsholz@us.af.mil	JBER POC Energy Management Control Systems (EMCS) Supervisor
Bill Farabaugh	Elmendorf Air Force Base	william.farabaugh@us.af.mil	Chief of Infrastructure Systems