EXECUTIVE SUMMARY

Cyber-Security Integrity for Electric Grid Facilities Management

ESTCP Project EW-201608

DECEMBER 2019

Daniel 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

Distribution Statement A
This document has been cleared for public release

This report was prepared under contract to the Department of Defense Environmental Security Technology Certification Program (ESTCP). The publication of this report does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official policy or position of the Department of Defense. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the Department of Defense.

EXECUTIVE SUMMARY

Project: EW-201608

TABLE OF CONTENTS

Page

1.0 INTRODUCTION 1
2.0 OBJECTIVES 1
3.0 TECHNOLOGY DESCRIPTION 2
3.0 ADVANTAGES AND LIMITATION OF THE TECHNOLOGY 4
4.0 PERFORMANCE ASSESSMENT 5
4.1 ARMY RESEARCH LABORATORY DEMONSTRATION 5
4.1.1 Focus group discussion 5
4.1.2 Levels of expertise required to execute the LLNL BATs 5
4.1.3 Concolic testing 5
4.2 ELMENDORF AIR FORCE BASE DEMONSTRATION 5
4.2.1 Focus group discussion 6
4.2.2 Methods of integrating the BAT tools into current process 6
4.2.3 Future tools including concolic testing 6
4.3 SUMMARY ASSESSMENT 7
5.0 COST ASSESSMENT 7
5.1 COST MODEL 8
5.2 COST DRIVERS 8
5.3 COST ANALYSIS AND COMPARISON 8
6.0 IMPLEMENTATION ISSUES 9
6.0 TEXTGENERAL OBSERVATIONS 9
6.0.1 Legal Issues 9
6.0.2 Installation 9
6.0.3 Pre-Disassembly 10
6.0.4 De-Obfuscation 10
6.0.5 Analysis Time 10
6.0.6 Feature Accessibility 10

LIST OF FIGURES

Page

Figure 1. Binary Analysis Workflow Within ROSE 3

LIST OF TABLES

Page

Table 1. Cost Model for single deployment of Binary Analysis Tools 8

ACRONYMS AND ABBREVIATIONS

BAS	Building Automation System
BAT	Binary Analysis Tool
DoD	United States Department of Defense
DoE	United States Department of Energy
ELF	Executable and Linkable Format
ESTCP	Environmental Security Technology Certification Program
E&W	Energy and Water
GUI	Graphical User Interface
LLNL	Lawrence Livermore National Labs
PE	Portable Executable
SCADA	Supervisory Control and Data Acquisition
U.S.	United States

1.0 INTRODUCTION

Firmware used in the control and monitoring of the United States (U.S.) electric grid and Building Automation Systems (BAS) are rarely, if ever, analyzed by end-users for potential vulnerabilities prior to being deployed. 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 which 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.

2.0 OBJECTIVES

The objective of this project and demonstration was to validate that the suite of Lawrence Livermore National Laboratories (LLNL)-developed binary analysis tools 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, DoD could achieve a more robust security posture, thus making an adversary's intentions to degrade control system capabilities exceedingly more difficult.

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.

The Federal government has understood the importance of securing critical infrastructure assets since the Presidential Decision Directive 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.

3.0 TECHNOLOGY DESCRIPTION

Over the last 20 years 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 (BAT) 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 of these technologies were designed with three 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 the Power Engineer
- DoD Reverse Engineering Expert/ Firmware Vendor, who both perform deep dive analysis if the protection engineer recommends further analysis.

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

Figure 1 describes the 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).

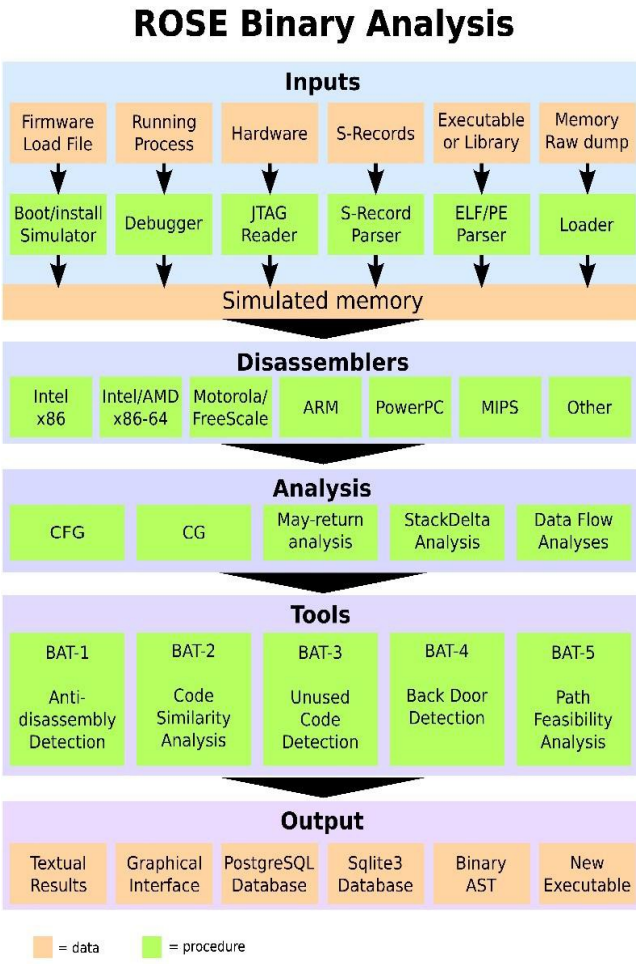

Figure 1. Binary Analysis Workflow Within ROSE

The ROSE framework was initially developed, over a period of more than 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 United States Department of Energy (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.

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 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.

These BATs have been developed using ROSE, but not released in the open source framework of ROSE. BATs 1-3 were developed in year 1, with a Graphical User Interface (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.

3.0 ADVANTAGES AND LIMITATION OF THE TECHNOLOGY

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.

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.
2. The software can be run on basic computer hardware. There is no need for specialized computing hardware.

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. The primary cost limitation is the potential cost of a protection engineer to analyze firmware if the BAT tool indicates further analysis is needed.

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.

To aid in the transfer of this technology, 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 life cycle assurance approach' for ongoing management and maintenance of device firmware. 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.

4.0 PERFORMANCE ASSESSMENT

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

4.1.1 Focus group discussion

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

1. Levels of on-site expertise required to fully leverage the capabilities of the binary analysis tools.

2. Concolic (i.e. concrete and symbolic) testing (4.1.2 Concolic Testing below).

4.1.2 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 industrial control systems 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.

4.1.3 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.

4.2 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 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.

4.2.1 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. Methods of integrating the BAT tools into the existing process.
2. Future tools including concept of concolic testing.

4.2.2 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.

4.2.3 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.

4.3 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.

5.0 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.

5.1 COST MODEL

Table 1. 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.

5.2 COST DRIVERS

The cost drivers will be the training and turnover of personnel on-site.

5.3 COST ANALYSIS AND COMPARISON

Operational use of our software has only hardware, training, and time requirements as outlined in our 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 pursue 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. Our 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 attach via firmware, estimates of return on investment are especially difficult to quantify.

6.0 IMPLEMENTATION ISSUES

6.0 TEXTGENERAL 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.

6.0.1 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 dominant costs are for hardware, training, and operator time. User documentation has already been assembled at a fixed cost and is also no-cost to the government. What we learned from the demonstrations at two DoD sites were that groups there were supporting of testing of firmware compared to the current state-of-the-art which include 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 testing firmware, the work to support testing was advantageous as compared to the risk of cyber-attack on base facility equipment.

6.0.2 Installation

The installation of the BAT suite has improved significantly since the mid-year review. While still a time-consuming process, it should have minimal if any impact to production as 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.

6.0.3 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 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.

6.0.4 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 demand for a method to de-obfuscate or unpack 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.

6.0.5 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.

6.0.6 Feature Accessibility

The 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.