LOGIIC HPS

April 2012

Final Project Report

REVISION HISTORY

Version	*Author*	*Date*

DISTRIBUTION

This report is approved by U.S. Department of Homeland Security and the LOGIIC Executive Committee for unlimited public distribution.

ABSTRACT

The LOGIIC[1] consortium was established by members of the oil and gas industry in partnership with the Cybersecurity Research and Development Center (CSRDC) of the U.S. Department of Homeland Security, Science and Technology Directorate (DHS S&T) to review and study cybersecurity issues in Industrial Control Systems (ICS) impacting safety and business performance as they pertain to the oil and gas sector. LOGIIC has sponsored research initiatives that involve the interests of oil and gas sector stakeholders.

Protection of ICS process control data, networks, applications, and host operating systems, particularly in multi-vendor environments, is a critical, ongoing requirement for the oil and gas sector. The threat to a control system's availability and integrity is real. Attack methods and tactics are diverse. These are evidenced by the recent Stuxnet and Duqu attacks. A loss of control over a critical control process potentially could result in loss of life, personnel injury, environmental impact, facility damage, production loss, and economic cost. System maintenance and protection increasingly centers on patching vulnerable automation software and operating systems, which often are implemented only infrequently. Many of these systems have reached end-of-life, are unsupported by the vendor, or lack economic basis for replacement. This situation presents a formidable challenge to asset owners demanding process automation reliability.

The availability of advanced and integrated host system protection technologies, complemented by a new alternative in the classic approach to maintaining system security, is worth considering as an alternative or supplemental approach to existing host protection strategies. In 2011, the LOGIIC program launched a Host Protection Strategies project to evaluate host-based cybersecurity technologies for use in a process control environment. Specifically, application whitelisting (AWL) technology was evaluated against a set of established criteria that support continuity of operations in critical system environments by providing strong cybersecurity.

Application whitelisting is a security technology that will maintain a list of executable files, and will flag or deny the execution of a file that is not on the list, depending on policy settings. AWL technology provides a mechanism to preserve a system by protecting against unauthorized file execution. Advanced memory and device protection is available in different capacities depending on the AWL product selected. Although AWL provides strong protection against file execution, it cannot guarantee protection against zero-day threats that may incorporate advanced memory attacks and leverage unpatched, third-party vulnerabilities. AWL should be considered as one tool in a comprehensive security plan for the operational environment. AWL requires planning to deploy the solution, and additional resources to maintain the solution. It may offer some risk mitigation for older systems with no anti-virus and/or systems that cannot apply operating system maintenance in a timely manner.

AWL is a host protection strategy that may provide many system-level benefits. The assessments conducted under this project indicate that AWL was effective at stopping malware in the environments that were tested, including those where AV updates are not readily available or where other security controls are not feasible. AWL does require planning, implementation, and maintenance. Including AWL in an organization's overall

[1] LOGIIC - Linking the Oil and Gas Industry to Improve Cybersecurity.

security plan should support the overarching objectives of the operational environment with clear cost, incident prevention, and long-term protective benefits.

This report presents findings of the LOGIIC program regarding application whitelisting, key attributes to its use in an ICS environment, and general conclusions about its implementation.

ACKNOWLEDGEMENTS

The project to evaluate host protection strategies was developed and guided by the members of the international LOGIIC forum, who devote their time and expertise to conduct projects that will lead to improvements to cybersecurity in the oil and gas industry, and to the control systems community in general. LOGIIC would like to thank the U.S. Department of Homeland Security, Science and Technology Directorate, for providing leadership, vision and commitment to enhancing cybersecurity. The Automation Federation serves as the LOGIIC host organization and provides a needed home and legal framework for our efforts. We would also like to express our appreciation for the work of our team of subject matter experts who refined the evaluation strategy, performed the system evaluations, and developed the project reports. Since the inception of LOGIIC, the scientific research organization SRI International has provided coordination, project management, and subject matter expertise.

The work performed on this project by SRI International and its subcontractors was funded under contract by the U.S. Department of Homeland Security (DHS) Science and Technology (S&T) Directorate. The content is solely the product and responsibility of the LOGIIC program and does not necessarily represent the official views of DHS.

Table of Contents

Table of Tables

1 INTRODUCTION

The LOGIIC[2] program was established to review and study cybersecurity issues as they pertain to the oil and gas sector, and has sponsored research initiatives that involve the interests of oil and gas sector stakeholders. LOGIIC initiatives are applicable to many industries with control systems.

Protection of process control data, networks, applications, and host operating systems, particularly in multi-vendor environments, is a critical ongoing requirement for the oil and gas sector. The threat to a control system's availability and integrity is real, and attack methods and tactics are diverse. These are evidenced by the recent Stuxnet and Duqu attacks. A loss of control over a critical process potentially could result in loss of life, personnel injury, environmental impact, facility damage, production loss, and economic cost.

In 2011, the LOGIIC program launched a Host Protection Strategies project to evaluate host-based cybersecurity technologies for use in a process control environment. Specifically, application whitelisting (AWL) technology was evaluated against a set of established criteria that support continuity of operations in critical system environments.

This report presents application whitelisting (AWL), key attributes of its use in a process control environment, and overarching conclusions about its implementation. Project details such as the technical approach, test methodology, and evaluation criteria are also presented. Conclusions in this report is focused on guidelines that Industrial Control System asset owners should factor in when considering implementation of host protection strategies, particularly AWL, in process control environments. The objective of this report is to convey important factors when considering AWL and support a dialogue between asset owner, automation vendors, and AWL product vendors.

1.1 Intended Audience

The intended audience for this report is the Industrial Control System technical and security communities; automation vendors, and security vendors.

[2] LOGIIC - Linking the Oil and Gas Industry to Improve Cybersecurity

2 PROJECT BACKGROUND

The growth in cyber threats, attempted and successful, malicious or unintentional combined with operational reliance on increased system reliability and availability create the need for a better approach to host protection. System maintenance increasingly centers on patching vulnerable automation software and operating systems, many of which have reached end-of-life, are unsupported by the vendor, or lack economic basis for replacement. This situation presents a formidable challenge to asset owners requiring process automation and environment reliability.

The LOGIIC Host Protection Strategies (HPS) Project sought to assess the use of application whitelisting to meet control system security objectives in a changing threat landscape. This project consists of three phases:

1) Technology landscape identification,
2) Test architecture design and test planning design, and
3) Evaluation in automation labs to verify the efficacy of AWL to prevent various attacks including:
    - Malware attacks
    - 0-day attacks
    - Direct host attacks

The LOGIIC team original goals were to lower complexity, cost, and administrative overhead by implementing AWL, without adversely impacting system reliability or performance. The objectives of the project included:

- Determining how AWL will integrate with or potentially replace current AV solutions
- Assessing how AWL solutions impact maintenance effort (e.g. AWL product maintenance, OS and application patching, and AV signature updates)
- Testing the feasibility of a single AWL solution that can support multi-vendor Automation systems, when possible
- Enabling deployment of AWL solutions into automation environments for the purpose of obtaining automation vendor accreditation
- Verifying the effectiveness of AWL solutions particularly to manage Stuxnet-type and other zero-day attacks
- Verifying that AWL does not introduce new security risks into the automation environment and evaluate possible risks associated with executing critical processes (e.g. change management)
- Identify how AWL solutions can support various Legacy components (e.g. OS, process control systems)

The project team decided to focus on AWL to answer key questions in the HPS project. To meet the project objectives, a technical approach and evaluation methodology was developed. An AWL vendor selection process was established, candidates were evaluated, and selections were made based on established criteria. Automation vendors offered test architectures to evaluate selected AWL products. AWL architectures were dictated by each AWL vendor. Assessments occurred throughout the fall of 2011. Analysis of findings supported overarching conclusions regarding the use of AWL in a process control environment. Those conclusions are presented in this report.

Since the start of the project in early 2011, automation vendors began accrediting AWL solutions and AWL vendors showed an increased interest in the process control market. The public-private partnership created by the LOGIIC project increased interest and action regarding AWL by both automation vendors and AWL vendors.

3 APPLICATION WHITELISTING OVERVIEW

In this section, we provide a general description of what application whitelisting (AWL) is, how it works, how it compares to other security mechanisms, and its capabilities and limitations.

3.1 AWL Definition

A simple basic definition of application whitelisting could be stated as:

> *Application whitelisting is a security technology that will maintain a list of executable files, and will flag or deny the execution of a file that is not on the list, depending on policy settings.*

This is in contrast to traditional antivirus (AV) technology, which is mainly based on maintaining a "blacklist" of known bad file patterns or signatures that represent viruses or other malware. The recent exponential growth of the number of entries in the AV blacklists and also the rate at which new entries are added, led to the emergence of whitelisting technology. AWL is also different from traditional configuration and patch management. While it maintains a "whitelist" inventory of files, it does not have the ability to prevent execution of files with malware.

It should be noted that while AWL protects against execution of unknown executable files, that does not mean that all forms of program code execution are covered. Some applications import and run code that does not originate from an executable file. For example, a web browser could run ActiveX or JavaScript code sent from a web server, or a word processing or spreadsheet application could run macros embedded in data files. AWL does not prevent execution of server or client scripts.

The idea of maintaining a "list of what is good" is a traditional concept in security, as is keeping a "list of what is bad". Compare for example how physical access to a facility is handled. The facility security staff will have a "whitelist" of employees, consultants, contractors, service providers and so forth, who are cleared and badged for access to the facility. The facility or processing unit may also maintain a "blacklist" of personnel who should not be granted access. This could be someone who should normally be on the cleared list, but whose access has been suspended because they did not complete required training, or have been disqualified for some other reason. The blacklist could also include people that law enforcement agencies have warned the facility owner about.

A visitor who is not on either list will need to be subjected to some review process before an access decision can be made, depending on policy. It generally is not a good idea for a non-public facility to rely on a blacklist alone, because there could be people with bad intentions whose names are not on the blacklist. There are also challenges with relying on a whitelist, such as a) ensuring that a program on the whitelist is periodically validated before being added to the list; b) ensuring that an impostor cannot pose as a program on the white list; and c how to handle new requests to the whitelist As described further below, these basic challenges occur in the planning, deployment, and maintenance of AWL.

Sometimes we use the term "default-permit" when we have a list of what is "bad" and our access policy is to permit everything that is not on the list. Similarly, an access policy where everything that is not defined on the "good" list is denied is called "default-deny". See the table below for a comparison of the concepts.

	What is bad ("black")	What is good ("white")
Policy (default) stance	Default-permit	Default-deny
Facility access example	No-access list (terminated staff, known criminals, etc.)	Access permission previously arranged for staff, others, etc.
Computer security example	Antivirus	Application whitelisting
Main motivation	Easily finds bad things without impacting those not on the bad list	Tighter security because anything not explicitly listed as good is questioned
Main problem	All bad things may not be on the list (leads to "false negatives") permitting access/execution when it should not occur (e.g. malware executes or bad guys get access)	All good things may not be on the list (leads to "false positives") preventing access/execution when it should occur (e.g. business disruption)

Table 1: Whitelisting Analogy

3.2 AWL Capabilities

The basic operation of an AWL solution is fairly straightforward:

- Keep a whitelist of known good executable files
- Ensure the executable files have not been altered (e.g. with malware)
  - For each executable file on the system, compute a checksum and see whether it is in the whitelist.
- Prevent execution of files not on the whitelist
  - If it is not in the whitelist, report it or deny execution, depending on the policy setting.

The main intended benefit of AWL is that it will detect any malware, known or unknown, because the malware will not be in the whitelist. This is conceivably a major advantage compared to traditional AV technology. With AV every new piece of malware must be added to the AV blacklist before it can be detected which often doesn't occur for days after the malware is identified and included in a DAT file. It is important to note that not all possible attacks manifest in the form of changes to an executable file, see the discussion on limitations in following section.

3.2.1 Differences between AWL Products

AWL products differ from each other in the management of the basic AWL functions:

- Storage of whitelist information like file name, hash, and signature
- Communication between the AWL clients (individual hosts like HMI, EWS, etc.) and the centralized AWL server can be either by a persistent network connection, periodic network connections, building and distributing new whitelist from AWL server to AWL client, etc.
- Maintaining the whitelist for automation vendor updates, OS patches, and other third party applications including trusted directories, trusted updaters, or trusted users
  - There are a very large number of known good executable files, and new entries must be added frequently as updates, patches, and new applications are released.

3.3 AWL Limitations

While AWL can be a useful tool, it will not stop all attacks. There are limitations related to how AWL is designed and implemented, and fundamental limitations to the approach:

- Basic AWL only detects changes to executable files. There are attacks that do not change executable files, such as when a whitelisted application (e. g. a web browser or word processing application) is tricked into doing something bad on behalf of the attacker.
- Any file that is not on the whitelist is flagged, even if there is nothing bad about that file. This could cause problems in terms of added workload for operators or even denial-of-service (DoS) for valid control system functions.
- AWL cannot detect nor prevent exploiting a vulnerability of a whitelisted program.
- If an attacker can get an executable on to the whitelist, the executable will not be prevented from executing. This could happen under these circumstances:
    - Pre-Existing (malicious) executables are included to the whitelist during AWL installation.
    - Existing (malicious) executables are added to the whitelist during whitelist update processes
    - Attacker creates seemingly legitimate digital signatures using stolen credentials (when digital signatures are used to verify executable files)
- Some AWL (vendor) solutions don't effectively whitelist complex applications, such as modern DCS automation software. This can require authorization of a complete folder of executables which may make these applications vulnerable to malicious attacks.

3.4 Testing Objectives

Testing objectives were defined before the project team developed the Technical Approach. These are the testing objectives:

1. Determining how AWL will integrate with or potentially replace current AV solutions
2. Assessing how AWL solutions impact maintenance effort (e.g. AWL product maintenance, OS and application patching, and AV signature updates)
3. Testing the feasibility of a single AWL solution that can support multi-vendor Automation systems, when possible
4. Enabling deployment of AWL solutions into automation environments for the purpose of obtaining automation vendor accreditation
5. Verifying the effectiveness of AWL solutions particularly to manage Stuxnet-type and other zero-day attacks
6. Verifying that AWL does not introduce new security risks into the automation environment and evaluate possible risks associated with executing critical processes (e.g. change management)
7. Identify how AWL solutions can support various Legacy components (e.g. OS, process control systems)

4 TECHNICAL APPROACH

Evaluation of AWL technology involves the investigation of key AWL functionality provided by a number of commonly available technologies that are marketed to the process control community. Evaluation criteria include technical attributes, product capabilities, and usability with process control systems. A series of assessments were conducted within a defined scope and established rules of engagement.

The scope of each assessment was based on the objectives of the LOGIIC HPS Project, and include blend of technical red teaming and functional test processes.

4.1 Assessment Methodology

Although assessments focus on a specific segment of an entire operational system, operability in a process control system environment was a core principle of the assessment and the guiding objective of the assessment team. To ensure proper scientific approach was utilized the evaluation was conducted under the context of the standard risk equation, $R(f) = TxVxC$, where risk is identified as a function of plausible threat, an existing vulnerability in the system, and resulting consequences. Measuring the performance of the technology was determined by the use of defined, realistic scenarios that were rooted in the existence of a plausible threat, existing vulnerability, and observed consequence. In this case, *(T)* Threat, is defined as the existence of an insider or outsider with the ability to launch a specific piece of malware towards an asset that is protected by AWL technology. Capabilities of threat in general, are complex, and are typically composed of resources, motivation, computing power, knowledge of the system, etc. In each red team attack scenario conducted, a plausible threat was denoted, and requirements for a threat to be successful were documented. *(V)* Vulnerability, is the existence of a weakness in the system that would provide a threat a mechanism for successful attack. Vulnerabilities can exist in the software, configuration, or implementation and how it's used in the operational environment. Lastly, *(C)* Consequence, is the observed or measured result of a successful threat that exploits a vulnerability. Severity of consequences and the results of the overall risk are considered in the analysis and summation about effectiveness of the technology in a control system environment.

Traceability and reproducibility were key aspects of this assessment. A clear understanding of why an exploit was successful and the ability to repeat that event were necessary for thorough evaluation.

4.2 Assessment Approach

When investigating application whitelisting as a technology for potential use in the control systems environment, there are several main considerations. As described in Section 2, there is first the need to clearly define AWL, what it is, and what it is not. The evaluation considers several constants in the control system environment including the need for 24/7/365 uptime, operational situational awareness, unobstructed access to the system during incidents, and the life-safety criticality of data and control decision integrity. After evaluating

the technology, several conclusions can be drawn about its efficacy in a control systems environment and its ability to meet the core objectives set forth by the LOGIIC project.

Each assessment was composed of multiple phases defined by the LOGIIC project team, which included testing with and without other security tools such as AV. The following are examples of key attributes evaluated:

- File Execution Protection
  - Various media and attack vectors
  - Protection against Stuxnet
- Whitelisting Processes
  - Server Installation (complexity, skill set, effort, duration, etc.)
  - Client Installation (complexity, skill set, effort, duration, etc.)
  - Administration and Management of AWL
  - Ease of Use (particularly with AWL Vendor Architecture)
  - Ease of Tuning (particularly with the AWL Vendor Architecture)
- AWL ability to work in various process automation environments (particularly with network architecture)
  - Works in a stand-alone, air-gapped Environment
- Integration with other host protection solutions
  - Ability to work with AV solutions accredited by the automation vendors
- Disruption of automation processes and/or new risks introduced
  - Rebooting Required with AWL Installation
  - Security of the AWL Server
- Memory Protection (commonly an add-on product with AWL)
  - Protection against Conficker
- Device Control (commonly an add-on product with AWL)
  - Ability to block various devices (e.g. USB)

Testing did not include an evaluation of malware injection methods (e.g. no network testing was performed).

4.3 Analysis of Findings

Analysis of assessment findings included consideration of multiple data sources:

- Baseline information gathered from technical scans, vendor documentation and discussion, and network reconnaissance
- Performance during technical red teaming and exploit response
- Observations during the assessment
- Usability testing
- Completion of functional test matrices
- AWL and automation vendor roadmap discussions were also considered

Criteria for evaluation were established by the LOGIIC team including weights. Technical responses to red teaming and attacks weighed equally against functional capabilities and usability testing. Independent analysis reports were completed for each assessment. All reports, technical findings, and usability findings, were considered when forming overarching conclusions about AWL technology in a process control environment.

5 FINDINGS

The findings enumerated in this section offer conclusions about AWL technology as implemented in a process control environment. Key attributes are presented that should be considered when making decisions about implementing the technology in segments of the control domain or in specific system environments.

5.1 AWL Functionality

Preventing the execution of unverified (i.e. have not been added to whitelist) files is the core functionality of AWL. While some AWL products examine the structure of files differently, this core functionality remains the same across all products. The complexity of this protection lies within the product itself. When whitelisting files AWL must establish trust in some capacity. Trust is handled through trusted files, trusted users, or trusted installers/programs. How this trust is maintained or handled within a system depends on the security objectives of the asset owner and/or the automation vendor through recommended practices. The amount of system lockdown and defined trust must still be established by a human that administers the system. This trust should be a balance between maximum security control and operational effectiveness. Blocking unverified file execution is consistent across the AWL products.

The distinguishing factor among AWL vendor products is primarily in the human interface with the products ranging in complexity and easy to use. The project team rated AWL products based on ease of use by comparing the complexity of each function between the AWL products that were tested. Another distinguishing factor of AWL vendor products is the functionality (e.g. device control, memory protection) that provides additional protection. For example, some vendors view additional functionality such as device control, as an integral part of their product. Others separate this functionality into a separate product.

Memory protection is perhaps the most complicated aspect of AWL. Memory protection implies protection of processes and address space in memory. Inherently, this is not file execution control. By definition, files would not execute until they reside on a disk. Many new attacks target memory, and have an increasingly higher sophistication in doing so. While memory protection is technically not AWL, a decision was made to include this in the project. Preventing different types of memory attacks, such as DLL injection and reflective DLL injection[3], requires a protective mechanism to exist in the kernel. Such protection (e.g. memory protection) often requires a reboot upon installation. Real-time (memory) protection can produce a significant resource (CPU) load on the system. Some products protect memory through periodic scans while others monitor persistently. A balance must be struck between operational effectiveness and maximum security when configuring and tuning memory protection.

More complex attacks that involve memory space and process threads require a different level of protection. Testing of memory protection identified 1). Many differences in effectiveness against certain type of memory

[3] Reflective DLL injection is the act of a library self-loading into host process through the use of a Portable Executable.

based attacks and 2). Differences in the degree of effort required to configure to memory protection. Zero-day attacks and future exploits are likely to include memory as a target. The threat to memory cannot be overlooked. Conficker, for example, is an exploit first recognized in 2008. Yet after three years, many older operating systems remain vulnerable and unpatched. Because Conficker employs an attack on memory, protection provided by AWL is highly dependent on the vendor's specific product. Some vendors were successful at protecting against Conficker while others failed.

5.2 Systems Best Suited for AWL

Frequency of changes in the automation systems impacts the effectiveness of AWL on specific systems. The nature of whitelisting and maintaining a "clean" image of trusted executable files (e.g. no malware) on a system indicates minimal software changes are desired and all software changes must be controlled. By that nature, systems that change infrequently or perform the same functions routinely are best suited for AWL. Regardless of the AWL product, changes to a policy, and therefore a whitelist, are not easily accomplished without connectivity back to an AWL server or a highly controlled software packaging and distribution process offered by some AWL vendors. Likewise, the ability to group like systems and deploy a standard policy across those groups, means that individual systems with highly specified security needs have an additional level of management within the AWL server. Given these factors, deployment and configuration of AWL on a network of systems requires consideration and planning.

Connectivity and design of the automation network should be considered. Many AWL vendors require connectivity to an AWL management server to install, whitelist, and configure a system. Remote sites (particularly with low-bandwidth) and/or systems that are air-gapped require an administrator to setup a server at a site, and install and configure the clients for some AWL vendor solutions. This could be done on a portable laptop, and the server then reintroduced into the environment when an AWL policy change is required (e.g. when automation software changes occur). This does require an administrator to perform these functions at the remote site. Some vendors also require a proprietary hardware server. This would require a server at every air-gapped site. Once systems are whitelisted, they are protected, but it is important to remember that any policy change (for automation application or OS maintenance) to a system requires a connection to the server for some AWL vendor solutions. Changing a whitelist or policy on a system requires significant effort if there is no connection to the AWL server.

The operating lives of systems in an automation and process control environment are expected to be significantly longer than standard IT configurations. Older operating systems may be at greater risk because they have more vulnerabilities which make them more difficult to protect. AWL may be a mechanism to protect older systems that cannot be patched or easily upgraded. Some AWL vendors support older operating systems, while others do not. Depending on the product selected, AWL may serve as a stopgap measure to protect the system until it can be replaced.

5.3 Installed Base, New Project, and the Role of the Automation Vendor

Protecting legacy or existing systems requires the addition of security as a bolt-on feature. In older systems, this option can be challenging. Some AWL products support older operating systems, but the fragility of a system and the critical functions of automation software may make the introduction of AWL too risky. System resource requirements for AWL, the requirement to reboot the automation system when installing some AWL solutions and the existence of very dated or unknown third party software could make AWL more difficult to integrate.

Understanding the core functions of the automation software and the ability to fine-tune those functions in AWL is absolutely necessary. This indicates the important role of the automation vendor. A detailed understanding of key functions, sub-processes, and patching and updates, is necessary to configure the interaction between automation software, administrators, and controllers, with AWL. Extensive testing on automation software and previous releases is paramount.

Automation vendor accreditation of AWL carries weight in a decision by asset owners to employ AWL. Automation vendors need to take an active role in the support and maintenance of AWL solutions. Employing AWL on an installed base without automation vendor accreditation means the asset owner must accept the resulting risk, management of the systems, and potential impact on maintenance, service agreements, and system restoration commitments. It is highly recommended to use AWL on automation systems based on the accreditation and support by the automation vendor, if possible. It is unlikely that an asset owner would have a choice to include multiple AWL products on an automation system. Automation vendors likely will not support the same AWL solution which may require asset owners to support multiple AWL solutions. If AWL is accepted as a critical element of layered security and defense, it is likely that automation vendors will accredit AWL products for new configurations and on currently support control system software. Support and accreditation of older configurations may occur later and/or be provided as a paid service.

5.4 Automation System Inventory, Change Management, and Backup

5.4.1 Automation System Inventory

A side benefit of employing AWL is the improved ability to catalog systems and their changing attributes. Some AWL products perform better than others in providing an organized inventory of changing system characteristics. By the definition of AWL they produce a view of systems on the network and basic attributes.

5.4.2 Automation System Change Management

Some products offer system drift (e.g. unplanned changes in configuration) and change management reporting, to include the logging of software changes. A limited number of AWL products go as far as to allow for the sorting and searching of system catalogs to display details such as a list of systems running Microsoft Office, its version, the last time it executed on a system, and the userid that launched the application. Inventory and change management can be a significant hurdle when implementing new security controls. Choosing the right AWL product can streamline inventory and change management activities particularly if this is an obstacle in your implementation.

5.4.3 Automation System Backups with AWL

Backup and restore functions are critical in automation system environments. AWL must be managed as part of automation system backup and restore functions to preserve the integrity of the whitelist. AWL must also be tuned to allow for automated backup and restore functions offered by automation vendors to run without issue. Control system and operating system patches on the host system require planning and tuning of AWL, typically in the form of a trusted installer, user, or directory.

5.4.4 Security Assessment of AWL solutions

AWL vendors were effective at preventing cybersecurity attacks against the AWL solution itself. The AWL master consoles were each attacked using malware and encryption between AWL server and client were tested but no known vulnerabilities were identified.

5.5 Antivirus, Delayed Patching and Zero-Day Attacks

The relationship between AWL and AV is not completely clear. Some AWL vendors offer their solution as part of a broader information security platform that may or may not already include AV. Other vendors specialize in AWL and retain a limited scope and footprint on the system. In either implementation, AWL should be viewed as a component or tool of a comprehensive, layered security plan. There are, however, some general conclusions that can be made regarding AWL and AV:

- AWL blocks the execution of any file not on the whitelist. AV protects against known threats through identified signatures in malicious code.
- AWL is a proactive, protective mechanism. AV, depending on its configuration, can be both proactive (when using real time threat protection which often turned off with automation systems) and reactive.
- AV is not uniformly applied across control systems. In fact, many automation vendors provide limited support for AV. Some automation vendors will likely provide limited support for AWL. Therefore, the engagement between existing AV and new AWL software is highly dependent on the existing automation architecture and security methodology of the asset owner.
- Real-time threat protection within AV is often disabled on control systems due to the resource (CPU) load on the system.
- AV DAT files may be obtained multiple times per day for the most current protection. Many automation systems are only able to update AV files on a periodic basis because of network challenges.
- Updates to the AWL core software generally occur once per month or at longer intervals, but the AWL solution remains effective even if these updates are not applied.

- AV can do little to protect against malicious code without a recognizable signature. This elevates the need for regular DAT file updates. If the AV signatures are not updated frequently AWL is one mechanism to prevent malicious code execution between signature file updates.

The key question is: *Why is AV required if AWL is protecting the system*?

- AWL will block file execution of any file not whitelisted, but it will not clean these files off the system. Therefore a risk exists that a malicious file may later be whitelisted / trusted, or moved to another system via USB, network share, etc.
    - Some AWL vendors offer products to clean these files off the systems.
- As defined by the AWL policy, AWL prevents not-allowed executable files from running. The prevention of file execution depends entirely on AWL policies, and AWL does not identify specific malware.
- AV cannot prevent an executable from running unless it contains a signature that AV recognizes as malware. Configured correctly and assuming an AV signature exists for the specific malware, AV will immediately identify malware and attempt to clean, delete, or quarantine the file.

Many consider AWL as a mechanism to address systems that are not regularly updated with OS, application, or AV patches. While regular patching is always recommended, there are many reasons in a process control environment that these system updates do not occur frequently. Impacts to critical processes, risks of downtime, time required to test patches, and sporadic network connectivity, do not facilitate regular patching. Although not a substitute for patches or a reason to delay, AWL can assist in mitigating risks on systems that have not been patched. This is particularly the case when a vulnerability can be exploited by an executable payload on the system. Similar to the aging of a system, AWL can assist in mitigating the risks to systems as more vulnerabilities are discovered. During the time it takes to test and implement patches on a system, AWL provides a mechanism of protection over an unpatched, vulnerable asset.

AV and AWL have very different objectives on the system. Choosing both provides an added layer of defense. On legacy systems without AV or the ability to update AV signature files regularly, AWL may be a clear choice for protection. This table includes solution combinations, benefits, challenges/problems, and the automation vendor position on the solution combination:

Solutions	Benefits	Challenges/Problems	Automation Vendor Position
AV only	Change management process for automation systems are simple with AV (unlike AWL)	AV signatures are difficult to maintain if there is not persistent network connection; Manual processes are used to maintain signature in these environments	Most automation vendors have an accredited AV solution
AWL only	AWL may be the only viable solution when AV signature files cannot be maintained; AWL provides protection when OS and application maintenance is performed infrequently	Significant change control processes are required to ensure a clean automation image is whitelists (e.g. no malware); AWL installation and configuration is difficult if there is limited automation vendor support	Many automation vendors are working to accredit AWL solutions; their support for AWL is limited

Solutions	Benefits	Challenges/Problems	Automation Vendor Position
AWL with AV	AWL provides some protection from some zero-day exploits until an AV signature is deployed; AV protects against malware that may have been whitelisted; See above Benefits	AWL and AV solutions may hang automation systems if they don't work together well; a suite may be required to prevent this problem; See above Challenges/Problems	Automation vendors recommend the combined solutions

5.6 Resources and Requirements in Managing AWL

Many AWL implementation and management requirements and overhead are the responsibility of the asset owner. Although some automation vendors have accredited AWL solutions some are providing limited support for these solutions, particularly for ongoing maintenance. AWL implementation and management complexity and overhead appear to have weighed heavily on the selection of AWL solutions and accreditation choices made by the automation vendors.

5.6.1 AWL Installation and Configuration

Labor resources are required in planning the initial AWL installation, as well as carrying out the setup and configuration. Planning includes a close working relationship with the automation vendor and the AWL vendor. Some automation vendors are providing limited support for AWL solutions. Network structures, AWL server locations, and AWL client configurations should be considered. Depending on the connectivity and structure within the asset owner's network an AWL administrator may be able to simply roll-out an installation from the AWL server, or may need to visit each client (e.g. on air-gapped systems). After installation, a reboot may or may not be needed, depending on the product selected and required features (e.g. memory protection requires reboot). Depending on the criticality and complexity of the automation system a post-installation health check may be required along with system monitoring following AWL installation. Planning the AWL installation and rollout can take considerable amount of time depending on the complexity of the AWL product, the amount of automation vendor support for integration of the AWL solution, and staffing skill levels. Several data points suggest that installation time may average from a few minutes to one hour per AWL client system which is mostly dependent on the AWL solution selected.

5.6.2 AWL Server Management

After installation, it is likely that more than one AWL security policy or group of systems will be managed from the AWL server. A typical configuration of AWL would include different policies for different groups of systems. They might all be managed from one AWL server, but it's rare for all AWL systems to use the same policy. An AWL administrator is required to perform fine tuning of AWL policies deployed and enforced on one or more automation systems. This requires knowledge of automation vendor processes, as well as a good working relationship with the automation vendor for support. Automation vendors are likely to provide varying degrees of support for AWL based on their comments to LOGIIC. The tuning of AWL policies within a system or group of systems should not be underestimated. AWL

tuning goals include ensuring operational processes execute while also deploying the security standards of the organization. Decisions on AWL operational modes, trust, user rights, as well as individual process and file allocations, are required. This highlights the need to choose an AWL product with a usable, intuitive interface that eases those decisions.

AWL server administrators are tasked with maintaining the highest level of security while still facilitating all critical operational processes. Like many security controls, over time AWL requirements are sometimes loosened to accommodate specific operations. AWL administrators perform periodic checks to ensure the automation systems are well protected and strong policies are enforced. Management of the AWL server is highly dependent on the size of the architecture, number of systems and systems that require a unique AWL policy, and complexity of the automation software. Asset owners should anticipate that resources will be required for continual management of any AWL solution.

5.6.3 AWL Maintenance and Upgrades

AWL is another piece of software installed on an automation system. It requires maintenance and upgrades like all other software. One advantage of AWL over AV is the length of time between maintenance and updates of AWL software. Daily DAT files are not required for AWL. Most AWL vendors issue hotfixes once per month, an update once per quarter and a major version upgrade once per year. AWL software maintenance likely would be deployed in conjunction with current automation vendor and OS maintenance cycles. AWL servers should also regularly be backed up, even if simply ghosting the system.

Note: To better understand how AWL impacts Change Management of Automation Systems, see Systems Best Suited for AWL

AWL technology is not a fix-it-and-forget-it solution. However, if it is deployed on a system that changes infrequently, management of and interaction with AWL is greatly reduced.

5.7 Memory Protection

Memory protection is separate from AWL, but it is either included with or offered as an optional product of AWL solutions. Memory protection is another important consideration in answering the AV/AWL question. AV does not normally provide memory protection. AV only does so depending on the signatures and behavior of the exploit. AWL can provide memory protection depending on the product and its configuration. In much of the LOGIIC testing, AV did not protect against a number of memory attacks, and depending on the AV product, it could be subverted easily by simply restoring quarantined files.

One AWL vendor in our lab test required signature updates to protect memory from the latest malware. Future threats and zero-day attacks make the decision about the value of AV and/or AWL more difficult to determine. It is difficult to protect against the unknown, but it can be assumed that attacks will become increasingly sophisticated and employ advanced attacks on memory. Stuxnet is a highly sophisticated threat and employed four zero-day attacks on Microsoft in addition to self-preservation and stealth capabilities while it awaited the presence of a suitable Siemens target. Protection may or may not include AV *and* AWL, but it should include the fundamentals of security such as access control, authentication, intrusion prevention, situational awareness, and incident response. Basic policies that address minor issues such as USB handling may prove to be invaluable in protecting against the next zero-day attack.

5.8 Device Control

Device Control, the ability to block various devices (e.g. USB), is a separate from AWL, but it is either included with or offered as an optional product of AWL solutions. Device Control solutions vary greatly in their implementation which makes some solutions impractical for many companies. Some solutions block device drivers which may be difficult to implement and maintain.

5.9 AWL General Consideration

These are some AWL general considerations:

General Issues	Considerations that impact AWL
Protection to prevent execution of compiled files (programs)	The majority of AWL tools focus on preventing execution of files that have not been whitelisted. There are attacks that do not add or modify executable files, such as when a whitelisted application (e. g. a web browser or word processing application) is tricked into doing something bad (e.g. executing malicious code) on behalf of the attacker.
Detecting when the executables have been altered	AWL vendors verifying that executables have not been modified which could introduce malware. Some AWL solutions use a hash signature to ensure executables have not been modified.
Network communication between the AWL client (e.g. HMI) and the AWL infrastructure (e.g. server)	Like other services, many AWL vendors have dependencies on a communications between the AWL server and AWL client which can pose a challenge for control systems. Regardless of the AWL product, changes to an AWL policy, and therefore a whitelist, are not easily accomplished without connectivity back to an AWL server or a highly controlled software packaging and distribution process offered by some AWL vendors. Likewise, the ability to group like systems and deploy a standard AWL policy across those groups, means that individual systems with highly specified security needs have an additional level of management within the AWL server. Note: Firewall rules need to enable connectivity between the AWL server and AWL clients (e.g. HMI) when persistent connectivity is required.
Ability to build and distribute new whitelist from AWL server to AWL client (e.g. HMI)	Depending on the complexity of the network (e.g. air-gapped systems) special processes will need to be developed to distribute new whitelists from the AWL server to the AWL client
Maintaining the whitelist	The whitelist must be maintained for automation vendor updates, OS patches, and other third party applications including trusted directories, trusted updaters, or trusted users. Asset owners will need to determine the frequency that software updates must be applied because of the complexity required to maintain the whitelists.

General Issues	Considerations that impact AWL
Tuning and other processes required to introduce a new applications to the AWL environment	Any file that is not on the whitelist is flagged, even if there is nothing malicious about the file. This could cause problems in terms of added workload for operators (i.e. responding to alerts) or even denial-of-service (DoS) of valid control system functions.
Limited support from automation vendor for AWL solutions and maintenance	Some automation vendors are not likely to maintain the whitelists required for their applications and some may provide limited support to asset owners.
Integrity of whitelist applications	If an attacker can get a bad executable on to the whitelist, the bad executable will not be prevented from executing. This could happen under these circumstances: • Existing (malicious) executables are added to the whitelist (pre-existing or during AWL installation or during a whitelist update processes) • Attacker creates seemingly legitimate digital signatures using stolen credentials (when digital signatures are used to verify executable files)
Complexity of applications	Some automation software is very complex which may require authorization of a complete folder or executables. This can allow malicious code to be introduced into the folder or to corrupt existing executables.
AWL as replacement for AV	AWL can be a counter measure for automation systems that don't support AV or the AV signature files cannot be maintained.
Device control	AWL vendors have different device control solutions using different basic approaches. Some vendors include device control as part of application whitelisting while other have it as a separate product.
Memory Protection	Configurations required for memory protection vary which can cause heavy resource (CPU) utilization and can be challenging to configure and maintain depending on the product.
Ensuring the AWL client doesn't have malicious code before creating the whitelist	Before beginning to deploy an AWL solution the asset owner must be confident that the system has no malicious applications on the device. All executables on the device will be able to execute without challenge once the whitelist is created.

General Issues	Considerations that impact AWL
Multi-vendor environments (automation application diversity)	Special consideration should be made that a one size fits all rarely exists. Multiple AWL solutions may have been deployed to match the automation applications used at your company. Accreditation of AWL solutions by automation vendors is critical to many asset owners.
Frequency of change	AWL solutions works best for systems which don't change often. Therefore, systems that change infrequently or perform the same functions routinely are best suited for AWL.
Legacy systems and applications	AWL support may not extend to some legacy systems. AWL can be used for currently and future supported Windows systems. Systems that will be in-service for many years will likely benefit the most from AWL in later years of service (e.g. when they are considered legacy systems).

5.10 Attributes to Consider when selecting AWL solutions

The decision to employ AWL and the subsequent product choice is greatly dependent on the host systems and the goals of the asset owner. While there are many specifics in this report, such as the differences between AWL solutions, variations in performance of AWL against a particular exploit, there are many similarities between AWL products that result in overarching characteristics to be considered. The following table describes some of the characteristics to consider when selecting and planning deployment of AWL:

Potential Issue Impacting AWL Selection	Considerations when selecting AWL
Asset owner main security objectives	The asset owner should define their security objectives to determine if AWL is the best solution for their company and/or specific automation systems
Criticality and risk of automation system	Assess the criticality and risks of automation systems to determine if the additional cost and effort required to deploy AWL is justified
AWL products accredited by the automation vendor	Identify the AWL solution(s) accredited by automation vendors and determine if the solution(s) meet your needs and possible effort to accredit other AWL solutions or consequences of using unaccredited solutions

Potential Issue Impacting AWL Selection	Considerations when selecting AWL
AWL product features and characteristics	Determine the importance of various AWL product features and characteristics relative to your assets, including memory protection, device control, etc.; Memory protection solutions vary in effectiveness, maintenance (e.g. require signature updates and/or custom rules), and/or may require excessive automation resources (e.g. CPU); Device Control varies in its implementation which may make it impractical to maintain
AWL product vs. security software suite required to integrate with AV	Understand the viability of using a AWL product with your existing AV solution to determine if a security suite should/must be used; some AWL solutions don't work well with some AV solutions; An AWL/AV suite often works together better than heterogeneous solutions
Organizational cost objectives	Understand AWL Total Cost of Ownership including: • Software costs based on AWL solution architecture • AV transition costs if replacement/suite is required • Labor costs based on complexity required to use and administer the AWL solution • Cost associated with maintaining the AWL solution in air-gapped and remote systems (particularly with low-bandwidth) Note: AWL has costs that should be considered that are similar to the introduction of any other new tool or technology.
Structure of automation network and architecture	Comparing your BPCS connectivity, remote sites, and staff skills to AWL Architecture and support complexity; Air-gapped and remote systems (particularly with low-bandwidth) may create significant challenges for some AWL solutions which require a persistent network connection to the AWL server
Age of automation system and typical Asset Life Cycle for BPCS systems	Older automation systems benefit more from AWL capabilities because newer systems have built-in security controls that prevent malware even if AWL is not present
Legacy support by AWL vendors	Identify legacy automation system needs and compare to AWL solution offering; AWL legacy system support varies and should be identified when selecting the AWL solution

Potential Issue Impacting AWL Selection	Considerations when selecting AWL
Organizational capability of automation staff	Consider likely organization capabilities required to support AWL; AWL solutions vary in the complexity required to use and administer them; also the ability to administer and package AWL updates centrally varies which may have significant impact at air-gapped and remote sites (particularly with low-bandwidth); identify AWL training required for the organization
Resource load on automation system	Evaluate the likely resource load on the automation system (CPU usage and memory) which varies by AWL product and/or Automation Application
AWL solution requirements	Determine how the AWL solution is architected (e.g. AWL server hardware, software distribution, etc.) and how that will impact your deployment; AWL solution architecture (e.g. need for persistent connectivity with AWL clients) can be challenging in air-gapped and remote sites (particularly with low-bandwidth)
AWL tuning, maintenance, and software updates	Understand the complexity required to tune, maintain, and update the AWL software; Understand the need and complexity of AWL software maintenance and AWL end-of-life timing
AWL working with complex automation software	Determine how effective the AWL solution will work with the automation solution based on the complexity of the automation architecture; Some AWL solutions cannot work with the automation software without granting full access to the automation file folder
AWL support by multiple automation vendors	Identify the AWL solutions accredited by the primary automation vendors used in your company; Try to reduce the diversity of AWL solutions when practical; there are likely limitations to your ability to use a single AWL server to support your enterprise
AWL Client Reboot requirements	Understand if a reboot is required to install the AWL client and how that will impact critical automation systems; A reboot is typically required to have robust memory protection capabilities

5.11 Stuxnet

Although overall cybersecurity is an objective of AWL in operational environments, threats that specifically target or affect control systems are the highest priority within the automation community. Stuxnet and Duqu are examples, and both exploits were tested during the project. In every assessment, publicly available Stuxnet exploits were launched on systems with AWL. In every test case, AWL successfully blocked the execution of the exploit, regardless of the presence of AV. Exploit code that requires non-whitelisted file execution is exactly the threat that AWL protects against. Memory attacks and zero-day attacks may be more difficult to protect against, but in the case of Stuxnet, AWL performed very well.

6 CONCLUSIONS

AWL technology provides a significant layer of protection to preserve a system by protecting against unauthorized file execution. Other layers of defense often provided as part of or as an option of AWL are advanced memory and device protection.

AWL does not introduce any new security risks based on our test results, but there is a significant change control effort. Some critical processes (e.g. change management) are much more complex with AWL. AWL should be considered as one tool in a comprehensive security plan for the operational environment. Operational environments and resource limitations create the desire for "one size that fits all" AWL solution for a single company. AWL requires structured management, including management of change, back-up and restoration, and development of staff competencies. An initial study is required to select the best AWL solution that fits a specific company environment. Although AWL doesn't have significant impact on the resource utilization memory protection can cause heavy resource (CPU) utilization and can be challenging to configure and maintain depending on the product.

Many attributes of AWL should be evaluated when defining an AWL implementation. In addition to technical capabilities and functionality, the usability of AWL with automation vendor software is extremely important. Support by the automation vendor for the AWL solution can ensure AWL provides maximum security with no impact to critical automation processes. Particular attention should be paid to impacts AWL may have on critical processes during the development, testing, patching, maintenance and operational phases. AWL may provide many benefits but requires engagement of both Automation Vendor Support and the Asset Owner as well as an on-going investment of time and personnel competencies development.

No one is able to predict the future complexity of malware and other possible attack vectors. Cybersecurity professionals must be vigilant to continually monitor the evolving attack methods and deploy layered defenses. AWL is one example of a technology choice that is currently available to the public.

APPENDIX A – ACRONYMS

Term/Acronym	
AV	Anti-Virus
AWL	Application Whitelisting
CIFS	Common Internet File System
DAT	Data File (in this case an AV signature file)
DLL	Dynamic Link Library
DMZ	Demilitarized Zone
DoS	Denial of Service
GUID	Globally Unique Identifier
HMI	Human Machine Interface
HPS	Host Protection Strategies
IDS/IPS	Intrusion Detection System/Intrusion Prevention System
LOGIIC	Linking the Oil and Gas Industry to Improve Cybersecurity
OS	Operating System
USB	Universal Serial Bus
VM	Virtual Machine