

An Analysis of Whitelisting Security Solutions and Their Applicability In Control Systems

Andrew Ginter, Chuck Rohs
Industrial Defender
Calgary, Alberta, Canada
{aginter, crohs}@industrialdefender.com

Abstract: Whitelisting is described by its advocates as “the next great thing” that will displace anti-virus technologies as the host intrusion prevention technology of choice. Anti-virus has a checkered history in operations networks and control systems – many people have horror stories of how they installed anti-virus and so impaired their test system that they simply couldn't trust deploying it in production.

While anti-virus systems detect “bad” files that match signatures of known malware, whitelisting technologies identify “good” executables on a host and refuse to execute unauthorized or modified executables, presumably because such executables may contain malware. This is a least privilege approach of denying everything that is not specifically approved.

In this paper the Industrial Defender team performs an independent analysis of a variety of whitelisting solutions for their applicability to control systems. The paper closes with some recommendations related to this technology and areas for further research.

Keywords: Anti-Virus, Host Intrusion Prevention System, Whitelisting

1 Introduction

Whitelisting is a term that can be applied to a number of technologies. In general, a black list identifies things that are “bad” or “not allowed” and a white list describes things that are “good” or “allowed”. This paper focuses on application whitelisting – a comparatively new technology to prevent the execution of unauthorized applications, including viruses, worms and other malware.

To understand whitelisting technology, let's start by looking briefly at anti-virus technology, which is the dominant anti-malware application control technology deployed today. Anti-virus uses a black list of signatures that match known malware. Anti-virus technology routinely searches your hard disks, memory, USB sticks, and various kinds of communications for code or data that matches a signature and takes various kinds of action when a signature is matched. The long-standing criticism of anti-virus technology is that it only responds to high-volume malware and then only after a signature has been created. Malware has to infect a critical mass of machines before a signature is created. Low-volume or targeted malware might never have a signature created for it.

Application whitelisting identifies applications that are allowed to execute on a particular machine. If any application tries to execute but is not on the whitelist, the whitelisting application takes various kinds of advisory or preventive actions. Allowed, or whitelisted, applications are generally identified by at least an application name and a cryptographic hash, and sometimes with additional attributes such as file size, path name or file owner.

1.1 Control Systems

This paper examines the applicability of whitelisting solutions to control systems. Process control systems generally control large, dangerous physical processes, such as power grids, chemical plants, and transit systems. While owners and managers of such physical processes may be rewarded by the productivity or reliability of the physical process, safety is king.

When a control system is designed, there are typically many layers of safety systems, and all of the designs are reviewed, tested and documented for safe operation. This is a costly process. Once a control system is certified safe and deployed, any change that calls the correct and safe operation of the control system into question can trigger a costly re-certification. As a result, operations personnel are very reluctant to make any “significant” change to a deployed control system, such as deploying anti-virus software or patching the operating system or the application. In addition, many users of control systems rely on control system application vendors to support deployed systems and keep them running safely and correctly. Those users generally will do nothing to the system that puts it into a state where the vendor will no longer support it.

1.2 Anti-Virus on Control Systems

Historically, anti-virus technologies had a rough ride in the control systems space. Anti-virus slowed things down enough to impair time-critical functionality of control systems. Control systems vendors did not support the use of anti-virus with their applications and few customers were willing to deploy unsupported technology on safety-critical equipment. As a result, anti-virus was and is generally not applied to a control system after deployment.

More recently, users and standards bodies have seen value in protecting with anti-virus technologies those parts of control systems that run on conventional computers. Control systems vendors have designed-in and now support the use of at least some anti-virus applications with their latest software versions. As a result, anti-virus technologies are being specified, designed and reviewed into new and updated control systems deployments, even if anti-virus is still not routinely retrofitted into already-deployed control systems.

1.3 Whitelisting

People are starting to apply whitelisting to control systems. Emerson recently included the CoreTrace Bouncer whitelisting suite in the Ovation Security Center product. We are

aware of two end users who are evaluating the use of whitelisting to protect already-deployed control system components.

In this paper we examine whitelisting solutions from four application whitelisting vendors: Lumension, Bit9, Savant and Stegosystems. We examine features of these solutions for applicability to control systems, and we examine the run-time impact of whitelisting technologies. Throughout the paper, we compare whitelisting to anti-virus technologies.

2 Application Whitelisting Features

The application whitelisting technologies we reviewed had a wide spectrum of features. This section of the paper describes major features and their impact on and applicability to protecting those parts of control systems that run conventional computers and operating systems.

2.1 Application Control

Basic application control features are common to all the whitelisting solutions we evaluated. After some installation and configuration steps, your control system computer is protected by an authorization daemon. That function might live as a driver in the kernel, or might live as a high-priority application process in user space. Either way the daemon intercepts a variety of requests to the operating system kernel. One way or another, the daemon is activated whenever there is an attempt to “execute” something. Execution might be loading a process image, it might be loading a shared library, OCX or DLL, or it might be reading a script in preparation for interpreting it.

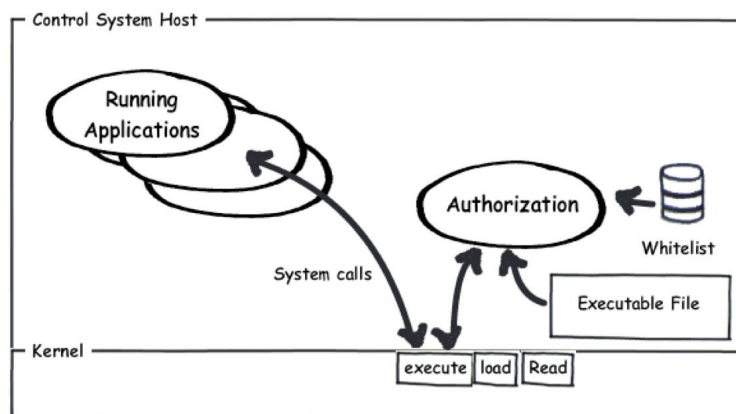


Figure 1 – Basic Whitelisting

In all these cases, the authentication daemon calculates something like a cryptographic hash on the file about to be executed and compares the hash to the whitelist rules for the host computer. Some vendors just check the hash, others check the user making the

request, the time of day or other context, but ultimately the authentication daemon renders a verdict – yes, the executable can execute, or no, it is not allowed. If the application is not allowed, the authorization daemon can generally be configured to report the violation and continue, deny execution, suspend execution to consult the user, or suspend execution to consult a management server for more advice. Application control lets you control what software runs on a computer and if you want, forbidding all non-whitelisted applications from running, including zero-day and targeted malware that the whitelisting system has never seen before.

When whitelisting systems first emerged for desktop applications, the complaint against them was that desktop software was constantly being updated, that end users in many organizations are allowed to download or purchase and install software components themselves, and that managing a whitelist is simply too much work. These concerns are largely addressed by modern enterprise whitelisting technologies.

However, these criticisms miss the point for control systems. The most critical control system components are servers and operator workstations. End users are not allowed to change the software on either of these components. Further, many control system components cannot be patched as regularly as enterprise components because of the cost of safety recertification, and so would benefit disproportionately from additional intrusion prevention protections.

So whitelisting technology protects a control system owner / operator from running unapproved software, and it protects them from running software that has been tampered with. Most infections attack via one or more vulnerabilities, and then establish a permanent presence on the targeted machine by creating, replacing, or modifying executable files. Whitelisting shuts down execution of these unauthorized files.

Whitelisting application control lets you not just prohibit the execution of malware, application control lets you control execution of any application on the host, including new applications introduced through removable media like Flash sticks and CDROM's. Controlling the execution of new applications that are introduced with removable media has real value in control systems. Not only can you prevent dangerous components like games or network scanning tools from running on control system hosts, you can shut down old-style viruses that propagate via new media. All of the vendors reviewed have basic application control that extends to applications running from or copied from removable media. Many vendors have more sophisticated removable media controls as well.

Whitelisting alone is no silver bullet though. For example, commercially available whitelisting products cannot prevent a memory-only attack like the SQL Slammer worm, which propagates in a single UDP packet and writes nothing to disk. You need Data Execution Prevention technology or memory-scanning anti-virus or other technology to completely prevent buffer overflow attacks. Whitelisting alone does not prevent the creation of executable files containing malware during an attack, or tampering with existing executables. Whitelisting only prevents the execution of unauthorized files.

2.2 Tamper Prevention

Which brings us to our next feature – the Savant technology suite is unique among the suites reviewed in that Savant includes a tamper-prevention feature to prevent creating or changing executable files. The technology prevents the creation of unauthorized executable files and prevents modifying existing applications, shared libraries, scripts and other executable files. Again, the feature tends to be implemented with kernel hooks. Every one of several kinds of common kernel calls are intercepted and evaluated against rules for that kind of call, application, user, etc.

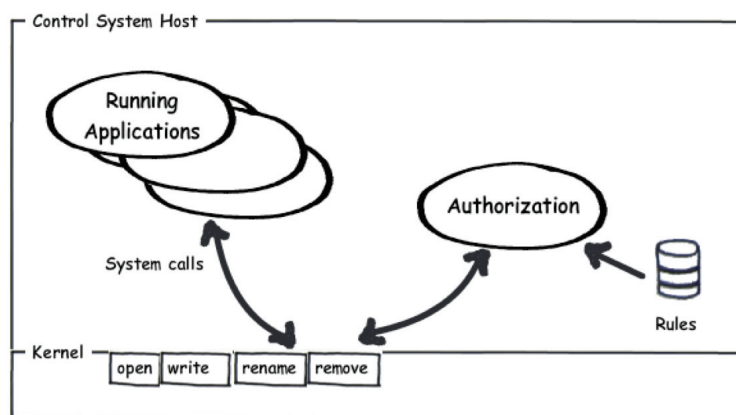


Figure 2 – Tamper Prevention

Tamper prevention keeps malware that may have exploited a buffer overflow or other memory-only vulnerability from establishing a permanent presence in executable files on the control system host. To simplify patches or authorized software delivery mechanisms, you can define rules that flag certain executables as “trusted”. Trusted executables such as the Microsoft update client are allowed to create or modify applications.

The feature is useful in minimizing the impact of a partial compromise, like a buffer overflow, on non-critical hosts like office desktops. Software that gains a foothold on a host through a particular vulnerability frequently downloads and installs a “root kit” component to ensure continued access to the compromised host. Tamper-proofing prevents the creation of new unauthorized executables, even before whitelisting has a chance to forbid their execution.

What this means for safety-critical control systems is less clear. If you get an alert saying your safety-critical system has successfully stopped the creation of an executable file with the name of known malware, what are you going to do? First off, you work hard to isolate infected systems and stop the infection from propagating. Then when cleaning up, even if executables were not changed, you have to ask the question “do I still trust this system?” On less critical systems, you might do enough homework to convince

yourself that the malware did not compromise any critical data or application configuration information on the system, and that the tamper-proofing prevented any changes to executables. For example – if the fault was in the file sharing service, you might conclude that the control system host was not compromised at all by the attempt to store an unauthorized executable to it. But a more cautious administrator might simply say “I can't trust the configuration and safe execution of this system any more,” scrub the system and rebuild it from trusted sources.

2.3 Management Server

All of the mature whitelisting technologies have a management server. The management server provides a number of functions, the most important being establishing policies and creating and a whitelist database. Authorization clients contact the management server periodically and pick up new rules.

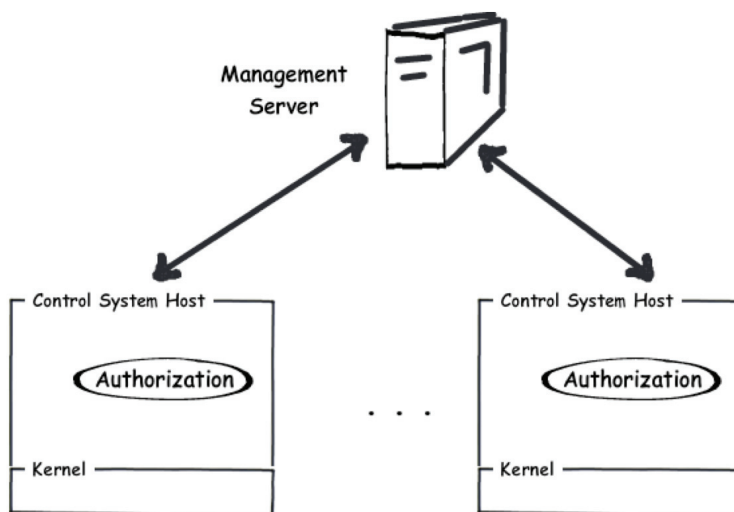


Figure 3 – Management Server

The thing to watch for with management servers is that the authentication servers and any other whitelisting software on the control system hosts have to be able to cache enough information to continue to function correctly for some long period of time without the management server. Operations users generally do not want to introduce another single point of failure into their control system. All of the technologies reviewed that had management servers had this caching capability. Controlled hosts had enough information in their rules database and their local whitelist database to operate correctly, indefinitely.

Another thing to watch for when deploying whitelisting systems is introducing a dependency on a management server that serves the entire enterprise. Enterprise servers

tend to be located at the far end of a WAN connection and may not be managed as a safety-critical system needs to be managed. Do you want some desktop-oriented policy change on the management server to cause your control system hosts to start preventing execution of safety-critical control applications? Sometimes this issue can be addressed by careful policy and rule design as well as careful delineation of which administrators can change which rules. Sometimes you are better off with an operations-controlled management server, or for your biggest sites, a site-controlled management server.

2.4 Cloud Management, Reputation-Based Whitelisting

Bit9 and Lumension have each introduced what amounts to “cloud” based Internet-hosted applications and checksums database. These services answer the question “how trustworthy is this software?” and are a big step in the direction of bringing whitelisting to the masses.

The average home user, and for that matter the average enterprise user with admin privileges, has no idea what software is safe to install or use on their computer. When the whitelisting software detects unauthorized software, for example when you download a specialized tool from the Internet that you found via a search, the whitelisting software can immediately query the database to determine how “safe” the new application is and whether continuing with the installation is wise.

This of course, is not how control systems are used, but there are two ways that a central database of whitelist information can help control systems. First, when provisioning a control system computer and calculating whitelist checksums or signatures for all executables on the new host, you can check those signatures against the central database. Any signatures that are not recognized, or that are recognized as malware, are flagged. This gives you warning that your provisioning environment itself may be compromised and is in turn compromising the equipment you produce using that environment. A second benefit is from periodic scans of deployed control system hosts. Just like anti-virus, you can use most whitelisting products to scan a host and produce a list of unauthorized components. To prevent impairing the operation of the control system, you may need to schedule the scan during downtime for the host, but like a periodic full anti-virus scan, a whitelist scan can tell you about unauthorized executables that have found their way on to a host, even if no software has yet tried to run those executables.

3 Whitelisting Vendors

3.1 Lumension

Lumension develops and markets a number of technologies, including patch management, vulnerability management, anti-virus, removable device control, application control/whitelisting, and compliance/risk management products. The application control product is highly scalable, with a powerful central management server. The server lets you create policies controlling which machines, which groups of machines, which users, and which groups of users, run which approved software and under which conditions.

Lumension also has a cloud whitelist database – a central Internet-accessible database of applications and signatures. This year Lumension announced a partnership with Microsoft where Microsoft would register whitelisting checksums with Lumension for all Microsoft products and patches. Lumension has this information from Microsoft and similar information from other software vendors in Lumension's Internet-accessible database. You can use the database to verify the provenance of software on a computer before it is deployed, or periodically reverify it after deployment.

The Lumension whitelist is based on SHA-1 cryptographic hashes, with support for a variety of SHA-2 hashes about to ship. The hash is recalculated when the application is loaded into memory prior to execution. Lumension whitelisting is integrated with Lumension's patch management offering and soon their anti-virus offering. A system for managing patches and whitelisting authorized patches as they are installed reduces the effort of deploying patches. Lumension is unique among these vendors in that they are about to release an anti-virus product designed to co-exist with the whitelisting product. The integrated anti-virus component scheduled for release in 2010 gives you greater depth of defense. Integrated whitelisting and anti-virus lets you do traditional anti-virus malware scanning on sockets, file I/O and memory in addition to the filesystem-based whitelist application control.

3.2 Bit9

Bit9 is focused exclusively on application whitelisting and whitelisting management technologies and claims the deepest and most feature-rich of whitelisting products. The product line includes the “parity” whitelisting technology, a “drift” component to help manage systems that are not completely locked down, removable device control, and the reputation-based Internet management server. The whitelist is based on SHA-256 cryptographic hashes and has auto-discovery and patch management features for ease of managing locked-down systems.

We evaluated a version of Parity software that used the reputation-based Internet management server and installation was a snap. Bit9 reports that with information for 7.5 million applications in the database, it is the largest such database in the world. You can use the database to verify the provenance of software on a system before that system is deployed, or periodically re-verify the software after deployment. We imagine deployment and configuration of a Bit9 enterprise management server would be a bit more involved, and deploying a management server is the approach that makes the most sense for a control system deployment.

3.3 Savant

Savant Protection is also focused exclusively on whitelisting technology, but is younger than Bit9. The Savant whitelisting product suite is very simple – end-point protection and an enterprise manager, and the enterprise manager is optional. We evaluated a client-only install of the Savant package.

The Savant package creates per-machine whitelist hashes using a patented algorithm that means hashes cannot be forged or re-used across machines because they are different on every machine. Savant is the only vendor evaluated to bundle tamper prevention into

their product. Tamper-prevention can prevent a large class of in-memory malware from establishing a permanent foothold on a compromised system. The Savant client package was painless to install and use.

3.4 Stegosystems

Stegosystems is a technology start-up and we evaluated a pre-release version of their first product. Stegosystems technology uses patented and patent-pending approaches to signing and verifying software. Stegosystems verifies executable code using a steganographic information-hiding approach instead of a cryptographic hash. A steganographic watermark is checked at runtime to prevent the execution of unauthorized code. Unlike traditional steganography, the watermark is stored external to the protected file, much like the other whitelisting products evaluated, so the watermark does not modify the executable file in any way.

The Stegosystems software is unique among the applications evaluated in that it repeatedly scans the executable while it runs to ensure that the water mark is still valid and that the executable is unchanged while running. The Stegosystems vision is to enhance this capability to the point where almost any set of unauthorized instructions running on a computer can be detected and dealt with. Time will tell how close they come to this mark, but their next version scheduled in January sounds impressive. They plan to re-verify the watermark on all functions pointed to by the stack of every authorized application, pretty much every time any system call is made, even if those functions reside in data segments where buffer-overflow attacks leave their code. This should let the Stegosystems software detect and shut down even all-memory infections like the Slammer worm.

The Stegosystems installation was straightforward, but the pre-release software lacks sophisticated features like a management server, a central database and auto-watermarking of executables on installation.

4 Test Results

To dig a bit deeper, we carried out a few tests on these whitelisting systems. We first installed the software and verified that it worked. That is – we corrupted a whitelisted application and observed that the software detected the corruption and refused to execute the modified file. We then ran four tests, 100 times each and timed the tests to see what impact each whitelisting technology had on execution times. In addition, during execution we had a network sniffer running to see if we really had configured the technology correctly into a stand-alone mode that was not dependent on a management server for correct operation. The four tests were as follows:

- We used our own Industrial Defender RTAP control system toolkit to start up a minimal control system environment and immediately shut it down
- We tested running very small executables – we fired up the Windows DOS shell and did a “dir” of a folder
- We tested an I/O-bound job – copy a 60MB file
- We tested a CPU-bound job, compressing a large file

In addition to the four whitelisting technologies, we ran these four tests on a vanilla Windows XP SP3 installation to get baseline numbers, and on a copy of Symantec anti-virus 2010 for comparison purposes. Measurements in the table are the average of the 100 test runs in milliseconds per iteration. Outlier values, such as the first iteration of the copy command where the 60 MB file was read into memory were discarded from the analysis.

Vendor	Tests			
	RTAP	"dir"	60MB copy	compress
Vanilla XP	3019 ms	12 ms	425 ms	13707 ms
Symantec AV	3089 ms	39 ms	434 ms	14169 ms
Lumension	3048 ms	29 ms	437 ms	13832 ms
Bit9	3064 ms	25 ms	436 ms	13900 ms
Savant	3037 ms	14 ms	438 ms	13803 ms
Stegosystems	3110 ms	40 ms	438 ms	13844 ms

Table 1 – Test Results

A lesson learned from the test that is only indirectly evident in the table of results has to do with compatibility. We had originally wanted to run all of the tests on the same machine – an older single-core XP workstation with 3 GB of memory. It turned out that the numbers we got for the “copy” test just made no sense. Two of the vendors took an surprisingly long time to copy the 60 MB file, a finding the vendors’ support organizations were unable to reproduce on their equipment. In hindsight, we recall having trouble with this workstation in other circumstances because of what seemed at the time like a poorly written SATA driver. We repeated only the “copy” test on a modern four-core box with 3.5 GB memory and those results are reported in the table.

The lesson is that whitelisting software, like anti-virus software, uses kernel hooks to intercept certain operating system actions. Applications that do this could exhibit compatibility problems when co-existing with some other kinds of software. Modern whitelisting applications, like modern anti-virus applications, have such issues largely behind them, but there are occasional exceptions.

The network sniffer turned up no communications with the management server that could be correlated to running the tests. We did observe a couple of packets each for the whitelisting solutions that had management servers, but not during the execution of individual tests. The packets appear to have had some other cause – possible checking in periodically for new whitelisting policies to download?

As for the performance results in the table above, one should not read too much into them. This is not a bake-off. Nobody is going to say one whitelisting system is a better control systems protection fit than another because one did the “dir” test 4 ms faster

than the other. The right conclusion to draw is that even the biggest latencies here are close to imperceptible to the human eye, and almost all whitelisting latencies are lower than those introduced by routinely-deployed anti-virus technology.

This is good news. If latencies of this size do not prevent the deployment of anti-virus control systems, they should not prevent the deployment of whitelisting systems, at least on new installations with the blessing of application vendors. Further, since whitelisting systems are not susceptible to false positives like anti-virus is, and do not require wholesale scanning of file systems the way early anti-virus solutions did, there are fewer impediments to deploying whitelisting on hard-to-patch, already-deployed control systems that may desperately need additional protections.

5 Conclusion

Whitelisting technology could add real value to control system security:

- Whitelisting catches many kinds of zero-day and targeted malware that anti-virus is blind to.
- Whitelisting provides the additional benefit of controlling what applications can be installed and run on a controlled host, even if those applications would have passed an anti-virus inspection, and even if those applications came from removable media.
- On the whole, whitelisting appears marginally faster than anti-virus already, and predictions are that this gap will increase over time. The number of different attacks being released into the wild is increasing exponentially, and with those attacks is an increase in the number of anti-virus signatures needed to identify the attacks. Searching files for an increasing number of anti-virus signatures can only take longer as time goes by.
- When coupled with an anti-virus solution, a data execution prevention solution and/or tamper prevention, whitelisting is even more attractive. The combinations reduce the risk of corrupting important executables in the first place, and reduce the amount of malware clutter left in the file system.

The long-standing criticisms of whitelisting for home use or in enterprise settings do not apply in professionally-managed, slow-changing operations environments. Further, many control system components cannot be patched as regularly as enterprise components because of the cost of safety recertification, and so they would benefit disproportionately from additional intrusion prevention protections.

Whitelisting is not a silver bullet though – it is a worthwhile layer in a defense-in-depth strategy.

In terms of applicability to safety-critical environments:

- Whitelisting introduces new code into critical execution paths. This calls into question the continued safe operation of a complex control system if whitelisting is deployed after safety certification.
- Whitelisting introduces measurable new latencies into all the execution paths we tested, again raising questions about safety.

That said, both the new execution paths and the execution latencies are comparable to paths and latencies introduced when deploying anti-virus technology, and anti-virus is deployed routinely into many new control systems, with the blessing of those control systems vendors. Further, the history of anti-virus was that not only did anti-virus introduce new control paths, the technology risked false positives and from a control system perspective had serious performance problems. Modern whitelisting applications should not suffer from false positives introduced by constantly changing rules or signatures, and do not suffer from serious performance problems. It may be that whitelisting can safely be applied to even legacy control systems.

6 Recommendations

Our few simple tests showed that the significant security value of application whitelisting comes at an architectural and execution latency cost comparable to the cost of modern anti-virus technology. We recommend that control systems users and control systems regulators request of control systems vendors the same level of support for whitelisting technology as is routinely offered by those vendors for anti-virus technology. That is:

- Support at least one or two whitelisting vendors for each control systems product, and
- Publish guidance for the installation and configuration of whitelisting technologies on supported control systems products.

When selecting whitelisting technologies, we recommend that control system vendors pay careful attention to failure modes and ensure that the combination of whitelisting technology and management system does not introduce new single points of failure or sources of unpredictable timings.

We also recommend additional research into the possibility of applying whitelisting technologies to protect already deployed control systems. As indicated earlier, we know of two end users carrying out such evaluations and we commend those efforts. Some control systems components are very old, running on obsolete operating systems like Windows NT. At some point the risk of triggering a safety shutdown due to a malware infection exceeds the risk of triggering a safety shutdown because of an unexpected interaction between whitelisting software and the control system application. How well do old control systems applications tolerate something like whitelisting, and how great a risk of outage due to malware would justify the risk and certification cost of deploying whitelisting after-market?

About the Authors – Andrew Ginter is the Chief Security Officer at Industrial Defender, responsible for security research and product architecture. Andrew's background includes management of software development teams at Hewlett-Packard and Agilent responsible for the development of control systems software and control system / enterprise middleware.

Chuck Rohs is the Security Product Architect at Industrial Defender, responsible for new security technologies. Chuck's background includes the CTO role at Au-zone, leading the development of hardware and software products, including hardened substation gear, banking products and cell phones.

References:

- [1] SSL 3.0 specification. <http://wp.netscape.com/eng/ssl3/>.
- [2] S. Kent and R. Atkinson. IP authentication header. IETF RFC 2402, November 1998.
- [3] An Liu, Peng Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks," in Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), SPOTS Track, pages 245--256, April 2008.
- [4] N. Gura, A. Patel, and A. Wander. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In Proceedings of the 2004 Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004), pages 119–132, August 2004.
- [5] D. Malan, M. Welsh, and M. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In Proceedings of IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON), pages 71–80, 2004.
- [6] H. Wang and Q. Li. Efficient Implementation of Public Key Cryptosystems on Mote Sensors. In Proceedings of International Conference on Information and Communication Security (ICICS), Dec. 2006.
- [7] Rivest, R.; A. Shamir; L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". Communications of the ACM 21 (2): 120–126. 1978.
- [8] D. Hankerson, A. Menezes, and S.A. Vanstone, Guide to Elliptic Curve Cryptography, Springer-Verlag, 2004.
- [9] MICAz motes.
http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf.
- [10] Imote2 motes.
http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf

- [11] University of Technology Seminar on Network Security, in TML laboratory report series, December 2000.
- [12] D. Wallner, E. Harder, and R. Agee. Key Management for Multicast: Issues and Architectures. IETF Request For Comments, RFC2627, 1999.
- [13] C. Wong, M. Gouda, and S. Lam. Secure Group Communications Using Key Graphs. In Proceedings of the ACM SIGCOMM '98, pages 68–79, Vancouver, B.C, September 1998.
- [14] D. Naor, M. Naor, and J. Lotspiech. Revocation and Tracing Schemes for Stateless Receivers. Advances in Cryptology-CRYPTO'01, LNCS 2139:41–62, 2001.
- [15] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, and D. Dean. Self-Healing Key Distribution with Revocation. In Proceedings of 2002 IEEE Symposium on Security and Privacy, Berkeley, California, USA, May 2002.
- [16] Liu, D., & Ning, P. (2003). Establishing Pairwise Keys in Distributed Sensor Networks. Proceedings of ACM Conference on Computer and Communications Security (CCS).
- [17] Kun Sun, An Liu, Peng Ning, Rogue Xu and Douglas Maughan. “Securing Network Access in Wireless Sensor Networks”, In Proceedings of the Second ACM Conference on Wireless Network Security (WiSec 09), March, Zurich, Switzerland