Security Assurance Levels:
A SIL Approach to Security

Dr. Nate Kube, Bryan L. Singer
Wurldtech Security Technologies
Vancouver BC Canada
nkube@wurldtech.com, bsinger@wurldtech.com

Abstract: A Safety Integrity Level (SIL) is a statistical representation of the reliability of the Safety Instrumented System (SIS) when a process demand occurs. SIL's are correlated to the probability of failure of demand (PFD), which is equivalent to the unavailability of a system at the time of a process demand. Given the de facto acceptance of SIL and the widely recognized interrelationships and interdependencies between safety and security, many have pushed for the adoption of a security level concept similar to safety in the security environment. Several efforts have been directed at this including security assurance levels defined in documents such as NIST 800-53. This paper takes a critical look at the SIL concept, its overall strengths and weaknesses as applied to security, and proposes general models for use within the security arena.

Keywords: Security Assurance Level (SAL), Safety Integrity Level (SIL)

1 Introduction

The continued deployment of Ethernet enabled devices in the industrial automation and control systems community has exposed a number of new potential risks previously unrealized in both IT and process control safety and security. Industry can draw upon previous history to exact a similar model. The Safety Integrated Levels (SIL) as defined in ISA – 84 [5] and internationalized in ISO 61508 [3] and IEC 61511 defined a four layer model for dealing with process safety requirements in two separate categories: hardware safety integrity and systematic safety integrity. This probabilistic model utilizes Failure Model and Effects Analysis to project risk and damages during a system failure, and provides a clear model by which systems can be assessed or implemented to achieve the desired level of safety risk reduction. While many draw parallel between security and safety, this is one intersection in which the lessons are clear and provide an effective model for security as well.

The following paper outlines the history of SIL, testing rigors as applied to control devices, models by which security can be evaluated similar to SIL, and implications for implementing this model both from an asset owner's and a vendor's perspective. The authors' intent is not to invalidate any existing safety models, but rather to show how extensions upon the SIL concept, whether applied as a separate standard or as a basic extension, provide an excellent framework upon which vendors can design, test, and market more resilient components. Asset owners are then free to determine and meet desired levels of risk mitigation within their environment, and achieve confidence in the final solution that reasonable measures have been taken to prevent systematic failures.

The authors define security in this document in a very broad sense. No distinction between external or internal or a malicious or non-malicious threat is made. Rather, security is defined in context of an availability and resilience problem. Any failure of a device or system to maintain its as-designed run-state under nominal or sub-optimal network conditions results in a breakdown in security and as such is understood as in scope. Security is most often a function of protecting business operations and maintaining as safe environment, regardless of the potential cause for failure.

2 Security Assurance

Software security and secure software are often discussed in the context of software assurance. Assurance here is defined as a positive declaration to give confidence and alleviate doubt, also a promise or pledge; guaranty; surety. Software assurance is broader than software security, encompassing the additional disciplines of software safety and reliability. A key objective of software assurance is to provide justifiable confidence that software is free of vulnerabilities. Another is to provide justifiable confidence that software functions in the "intended manner" and that the intended manner does not compromise the security and other required properties of the software, its environment, or the information it handles. A third objective of software assurance is the ability to trust, with justified confidence, that software will remain dependable under all circumstances. These include:

- The presence of unintentional faults in the software and its environment
- Exposure of the operational software to accidental events that threaten its dependability
- Exposure of the software to intentional threats to its dependability, both in development and in operation

According to the Committee on National Security Systems (CNSS) Instruction No. 4009, "National Information Assurance Glossary," software assurance is defined as, "the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at anytime during its lifecycle, and that the software functions in the intended manner.

Software assurance addresses these attributes:

- **Trustworthiness**: no exploitable vulnerabilities exist, either maliciously or unintentionally inserted
- **Predictable execution**: justifiable confidence that software, when executed, functions as intended
- **Conformance**: planned and systematic set of multidisciplinary activities that ensure that software processes and products conform to requirements and applicable standards and procedures

3 Background on Safety Integrity Levels (SIL's)

Rising industrial safety incidents into the 1990's prompted a critical eye towards industrial practices, and resulted in ISA creating ANSI/ISA-84.00.01-1996 [5], the first safety standard developed by SP84 committee. This standards document focused heavily on risk reduction through a systematic risk reduction process. This process was later internationalized in IEC 61508 [3] for functional safety and IEC 61511 for process safety.

The SIL requirements define techniques and measures required to analyze the likelihood of and prevent systematic failures due to design flaws in the system or individual components, a system being recognized as the sum of its component parts. Achieving requirements in both categories is required for a given SIL rating, and may be met through either the development process or through sufficient historical failure mode analysis. This model defines not only what the components individually must be able to demonstrate, but also what the system as a whole must achieve.

This matter of specification and evaluation provides a framework by which asset owners can define a desired level of safety and then be able to procure or implement only vendor supplied components that will contribute to or exceed the target safety level. *A priori*, it is important to note that SIL is not a guarantee of safety, but rather due care measure to identify and reduce safety risk. There are and will continue to be "black swan" moments where seemingly trivial system components or systematic failures may still induce safety incidents and therefore must be computed into the overall risk management equations.

Two factors served as principle influences towards development of SIL:

- Adoption of the belief that this continuum is a scale of risk making risk analysis an essential feature in the development of safety-related systems.
- Huge increase in the use of software (and complex hardware like microprocessors) thus shifting the balance between random and systematic faults. Previously, it was normal to assume (often implicitly) that safety could be achieved through reliability, and to deduce a value for the reliability of a system by aggregating, often through a fault tree, the random failure rates of its components. With software, in which all faults are systematic, there is no possibility of deducing system reliability by a method that is restricted to the consideration of random failures.

Safety integrity is defined as the average probability of a safety instrumented system satisfactorily performing the required safety instrumented functions under all the stated conditions within a stated period of time. Specifically, to what extent can the end user expect the safety processes in question to perform safely, and in the case of a failure, fail in a safe manner?

There are four levels of safety integrity: level 4 has the highest level of safety integrity; level 1 has the lowest. The higher the safety integrity level, the higher the probability that the required safety function will be carried out. IEC 61508 [3] equates SIL's with probabilities of unsafe failures in 2 tables, one for on-demand systems whose demand rate is low, and one for systems with continuous operation or a high demand rate (Tables 1 and 2 below).

Low demand operation is defined as no greater than 1 demand per year (approx 10^4 hrs). Thus the tolerable probabilities of failure in the low-demand cases are increased by a factor of 10^4 in order to arrive at the equivalent SIL values for continuous systems.

Safety integrity level	Probability of failure to perform its safety functions on demand
4	≥10−5 to <10−4
3	≥10−4 to <10−3
2	≥10−3 to <10−2
1	≥10−2 to <10−1

Table 1 – Safety integrity levels of low demand mode of operation

Safety integrity level	Probability of dangerous failures per hour
4	≥10−9 to <10−8
3	≥10−8 to <10−7
2	≥10−7 to <10−6
1	≥10−6 to <10−5

Table 2 – Safety integrity levels for continuous /high demand mode of operation

4 Establishing SIL Requirements

Establishing SIL requirements varies depending on the functional safety standard in use. IEC 61508 is based on the model shown in Figure 1. Consider a plant or equipment under control (EUC) which is used to provide a utility or benefit. Complementary to the EUC is a control system that together with the EUC executes processes which may pose hazards that must be identified and analyzed.

Risk Based Safety Analysis (RBSA) - the task of evaluating a process for safety risks, quantifying them, and subsequently categorizing them as acceptable or unacceptable – is conducted on the processes executed by the EUC and its control system. RBSA dissects a process into its functional components, with each being evaluated for risk. By combining these risk levels, a comparison of actual risk can be made against the risk budget. When actual risk outweighs budgeted risk, risk-reduction facilities are considered and when risk-reduction facilities are provided in addition to the EUC and its control system, and these take the form of electronic systems, the standard applies.

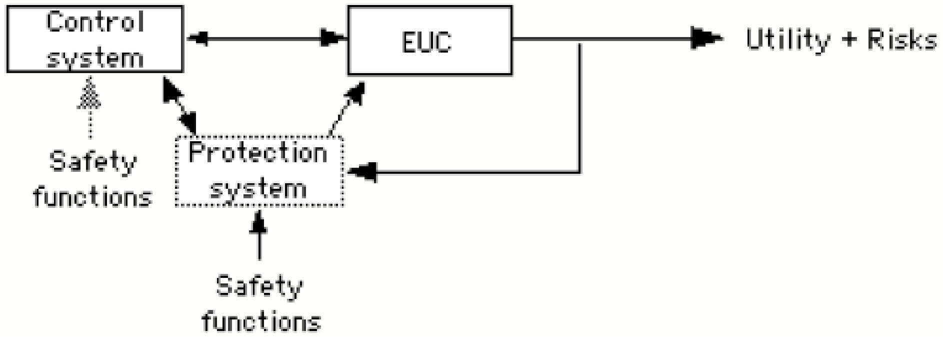

Figure 1 – IEC 61508 System Model

The required risk reduction associated with each hazard is specified as a safety requirement and, according to the standard, each requirement must have two components, the functional requirement and the safety integrity requirement. The later takes the form of a SIL. The totality of the safety requirements for all hazards forms the safety requirements specification.

Safety requirements are satisfied by the provision of safety functions which are implemented in safety related systems. The SIL's of the safety requirements become those of the safety functions that will provide them, and then of the safety-related systems on which the safety functions are to be implemented. Hence, the SIL of a safety-related system reflects the risk reduction that the system much achieve.

Suppose, the tolerable risk for a process is deemed to be 10^{-9} dangerous failures per hour and the EUC is calculated to have a probability of 10^{-2} dangerous failures per hour, the difference must be achieved by one or more safety functions. If the risk reduction was provided by protection system separated from the EUC and its control system, the protection system would need to have a probability of 10^{-7} dangerous failures per hour (assuming of course that the EUC and the protection system failures were independent, see Figure 2). From this, and Table 2, the safety-related system would have to be of SIL 2.

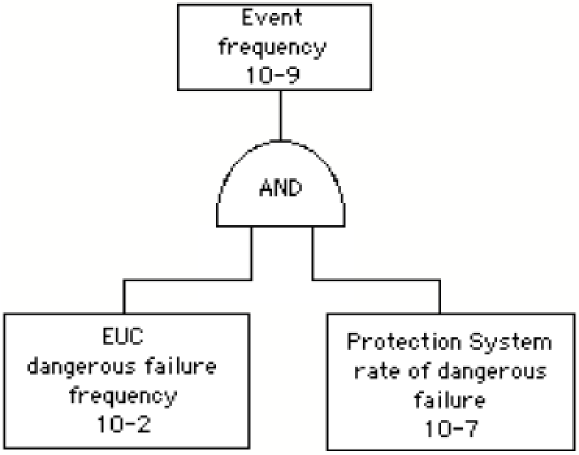

Figure 2 – The principle of a protection system

5 Meeting SIL Requirements

Safety integrity comprises both hardware safety integrity and systematic safety integrity. In determining safety integrity, all causes of failures (both random hardware failures and systematic failures) which lead to an unsafe state are to be considered; for example, hardware failures, software induced failures and failures due to electrical interference. Quantitative factors in combination with qualitative factors, such as development process and safety life cycle management, are employed.

The SIL requirements for hardware safety integrity are based on a probabilistic analysis of the device. To achieve certification for a given SIL environment, the device must have less than the specified probability of dangerous failure and have greater than the specified safe failure fraction. These failure probabilities can be calculated by performing a Failure Modes and Effects Analysis (FMEA). The actual targets required vary depending on the likelihood of a demand, the complexity of the device(s), and the types of redundancy.

The SIL requirements for systematic safety integrity define a set of techniques and measures required to prevent systematic failures (bugs) from being designed into the device or system. These requirements can either be met by establishing a rigorous development process, or by establishing that the device has sufficient operating history to argue that it has been proven in use.

Note: some of the above types of failure, in particular random hardware failures, may be quantified using such measures as the failure rate in the dangerous mode of failure or the probability of a safety instrumented function failing to operate on demand. However, others, such as software failures, cannot be accurately quantified but can only be considered qualitatively.

Although it is recognized that, given the current state of knowledge, many systematic causes of failure can only be assessed qualitatively, a safety integrity level is defined numerically so as to provide an objective target to compare alternative designs and solutions.

It is important to note that no individual product can carry a SIL rating. Individual components of processes, such as instrumentation, can only be certified for use within a given SIL environment. Once components are selected for a process, the actual SIL of the process is determined by methods including: Simplified Calculations, Fault Tree Analysis, and Markov Analysis.

6 Implications of Safety Integrity Levels

The SIL concept results in an interesting side benefit, device reliability measurements. Products must be able to demonstrate that they will be available for their designed function at a given rate in order to achieve compliance in order to operate in a SIL rated environment (note that products are not SIL rated, they are rated to exist in a SIL rated environment). SIL ratings are calculated as a Probability to Fail on Demand, and they factor in "random" events such as Mean Time Between Failure and Mean Time to Repair. Based on industry calculations, the SIL levels determine the functional requirements of the system from risk exposure and reduction perspective.

Vendors must supply components that meet or exceed the failure criteria as defined in order for the component to be allowed into a SIL rated environment. This implies some level of confidence or assurance must be supplied by the vendor, and evidence of that claim must be sufficient. As per SIL, this assurance can either be demonstrated for the past performance on this device or through a rigorous testing process. Either of these requires that the vendor can demonstrate and understanding and perform adequately to understand the functional components of the system, apply an appropriate level of testing and results measurement on the device, suitably stress the component with a sufficient amount of test criteria, and apply both negative and positive testing of functionality. This testing must emphasize not only the performance of the device, but also its ability to perform its designed function while in abnormal conditions or states in order to assure safe function.

7 Limitations of SIL as Applied to Security

Considering the safety of an environment requires a holistic analysis of any potential systematic failures that could result in unsafe operating conditions. A comprehensive view of safety requires analysis of, at a minimum, the following:

- Physical Constraints – walls, firebreaks, physical access control devices, piping, jumpers, and other physical protection mechanisms
- Logic in the EUC - High, low, high-high, low-low, or other instructions that limit the functional behavior of the equipment under control
- Electrical / Electronic / Programmable Electronic Safety-Related Systems – the actual safety systems that provide the overall safety level of the EUC

Assurance of security requires that the design engineer must ensure that all of these work together in an overall cohesive environment to assure security. A SIL perspective focuses only on the device functionality at level 3, and care must be paid to ensure that functionality at the physical or controllers for the EUC are not able to override this safety functionality.

Case Study - An analysis of a safety failure on a filler in a batch processing environment demonstrated that the E/E/PES safety systems were improperly designed and that the actual logic hooked to a cabinet switch existed within the PLC. A failure on the PLC while the cabinet was open allowed the device to start up even though this physical limit switch was disabled. This is a failure where the E/E/PES was overridden in its functionality by logic within the controller.

From a safety perspective, design constraints are well known and understood today, provided that the rigorous processes dictated by IEC 61508 and 61511 are followed correctly. Systematic failures can be limited or reduced to a level of statistically low probability. The challenge in security, however, is that in addition to the low level of random system faults, there is compounded problem of directed and intentional attacks where systematic failure can be induced at multiple levels. No longer can the single point analysis of a given devices probability of Failure on Demand be relied upon when a constructed attack can result in failures at any given level of the operation.

Another weakness of the SIL concept is that in general, no real distinction is made between achievement and assurance of safety or integrity: there is an implicit assumption that following the process delivers the required integrity. As such, the general state of safety or lack thereof becomes a self-fulfilling prophecy – “I need SIL 4, I’ve followed the SIL 4 process, so I’ve got SIL 4.” There is no definition of what the SIL’s mean independent of the process. Because a software-based system cannot be shown to have met them, SIL’s are intended to define the requirements for rigorous testing and analysis to be used in the development process.

This is a serious short coming with respect to safety assurance. None of the standards presents any evidence why we should be able to infer product integrity from a process nor is any real distinction made between achievement and assurance of safety; there is just an implicit assumption that following the process delivers the required integrity.

Given that it is impossible to prove the absence of software faults, Dykstra - fundamental limitation of testing, and that it takes an impracticably long time to derive complete confidence in reliability from testing, a number of problems arise for the developer who needs not only to achieve but also to demonstrate software safety. For example: How do we define criteria against which to make claims of achieved safety? When a SIL has been used to define the level of safety to be achieved, it follows that that SIL should be the criterion against which a claim for the achieved safety would be made (and judged). But if numerical values for the expected failure rate of software cannot be derived with confidence, it is not possible to adduce proof of such a claim.

8 Defining Device Security

An industrial system is defined as a collection of functionally related and either independent or dependent configurations, for the purposes of fulfilling a desired design objective for an industrial process. Given this, the most basic component of a system is a device, generically held here to be any functionally independent component for a specific purpose such as a controller, RTU, I/O, sensor, etc. Similar to SIL, in which the achieved security rating requires both the system and all of its devices must demonstrate the same level of resilience to failures.

Testing a device requires not only positive testing of its function (that all functions work correctly), but also negative testing to demonstrate that any potentially negative inputs, outputs, device states, conditions, or other exceptions are properly handled. Successful testing requires the following:

- The device components, communications architectures, and processing capabilities must be known
- The designed function including all positive and negative logic states must be understood
- Communications protocols and device communication design must be documented and understood with all permutations exercised
- Exception handling (fail-safe) protocols and functionality must be tested in suitable set of permutations

9 Considerations for Equipment Under Control (EUC)

Some basic extensions to the SIL concept are immediately required when considering security. Evaluating security reliability for EUC requires a balanced understanding of the following:

1. Physical Security Constraints
2. Software and Logical Security Constraints (integrity of the device and operations)
3. Design and relationships with E/E/PES, all immediately connected devices, controllers, configuration modules, etc.
4. Design of the E/E/PES

Faults must be analyzed in terms of design constraints of the above against the each of the other layers. For example: a security fault can occur if a skilled attacker causes a mixer to start despite the safety logic or if a device fault at level 2 or if a controller can cause something physical like a swing arm or robot to move beyond maximum permissible range and cause a safety problem.

9.1 Examples of Failure

Failure Level	Equipment	Consequences
Uncontrollable	Centrifuge is able to be set to unsafe speed that generates a failure	High speed moving parts and if a failure is generated, it is impossible to control the randomness of the physical even
Difficult to Control	Set speed on a high-speed filler is too high	Chance of an uncontrollable failure exists, but likely only this one machine will fail and the damage will not be physically compounded beyond the device
Debilitating	Network Based Denial of Service limits availability of HMI's and operator stations to interact with the process	Most safety systems will still likely work, but stopping the process safely requires exercised protocols, e-stops, etc.
Distracting	Virus deploys on an industrial network that affects operator screens, but no Denial of Service	Email bombs, pop-up windows, etc. affect operator ability to interact, but they do not crash the system
Nuisance Only	Harassing phone call attempts at social engineering	Could keep an operator from being at his or her operator station, but systematic failure probability is low

Table 3 – Failure Examples

10 A Model for Device Security Levels - From SIL to SAL

Demonstrated above, the safety SIL concept provides a reasonable model for expressing security levels as well, but a number of key limitations of SIL drives that a security assurance model requires a paradigm shift. *Prima facie*, security failures are often multi-level; they can and nearly always do violate design constraints. For example, Always False Instructions to bypass safety logic, sharing passwords, turning off security settings, etc.

Knowledge of the system will allow an attacker to deny, disable, or deter security, safety, and other protection mechanisms to facilitate and force-multiply the damage of the attack. Additionally, there is no possibility of deducing system security reliability by methods which only consider random failures, or of proving the absence of system faults. These two factors combine to make probabilistic models of security to determine acceptable levels of failures per hours a complicated and limited benefit model. As with safety, security is best achieved by an inherently secure process design combined, if necessary, with a number of protective systems which rely on different technologies

(IDS, IPS, etc). Any security strategy considers each individual security system in the context of the other protective systems.

A similar example exists within the auto industry. They have taken the approach of modeling safety in terms of the controllability of the vehicle under abnormal conditions. Clear parallels can be drawn to security. The random and potentially unsafe actions of the driver compound the entropy of driving conditions, maintenance disposition of the vehicle, and electronic logic in many cars that supplement driver behavior; there are also the compounded actions of the driver. Potentially every safety design within the vehicle can be violated if the driver pushes the vehicle to 120MPH and buries the steering wheel to one side just in front of a telephone pole.

The potential randomness of not only integral safety factors but also of external events drives the motor industry to consider safety as the controllability of the process by the system for a given breach of safety protocols, as per the Motor Industry Software Reliability Associations Development Guidelines for Vehicle Based Software [4].

Controllability Category	Acceptable Fail Rate (successful attack)	Integrity Level
Uncontrollable	Extremely Improbable	4
Difficult to Control	Very Remote	3
Debilitating	Remote	2
Distracting	Unlikely	1
Nuisance Only	Reasonably Probable	0

Table 4 – Motor Industry Guidelines

Given that SIL's are essentially a mechanism to understand, plan, and design to accommodate failures, a similar model can be extracted for security. Security levels can be defined in context of understanding the controllability of the process, where a higher level demonstrates that an insecure and/or unsafe condition can result in dangerous loss of control of a process. Achieving a security level requires assigning probability of failure of protection mechanisms within the context of the device in question and its immediate connection points, such as I/O, HMI, etc., and then demonstrating that the component, through device testing, can achieve a level of resilience against such failures to achieve the "desired" failure rate.

This does require that SALs take a "product oriented" view in complement with the safety "process oriented" view. Further, a deeper understanding of the potential device faults is required. Where safety can focus primarily on hardware faults and additionally on some level of software faults, security levels must focus heavily on both hardware and software, but also on network communication faults. As such, device testing must not only test logical and electrical conditions that are out of spec, but also communications and protocol violations through fuzzing, exploitation techniques, and network resilience testing.

10.1 Additional Requirements

SAL will require additional considerations when evaluating devices for security resiliency and protection against violating security controls. Secure coding practices, network protocol analysis and testing, and security vulnerability and penetration testing must be added to the basic complement of similar tests such as voltage spikes and sags, electromagnetic interference, and other installation and environmental factors.

- **Secure Coding Practices:** Following good design patterns, unit testing, system testing, avoiding deprecated libraries and objects, source code standards, and peer review.
- **Network Protocol Analysis and Testing:** Communications protocols must be inspected and tested for conditions such as SYN flood attacks, exploitable conditions in reliability and data transmissions, device network communication tolerances to avoid Denial of Service, etc. Testing must include bounds testing, flood/storm testing, protocol fuzzing, etc.
- **Security Vulnerability and Penetration Testing:** Utilizing known security flaws or flaws discovered in earlier testing to conduct penetration testing to prove exploitable conditions and to determine mitigating controls.

11 Summary

Many are quick to look at security through safety lenses in industrial automation and controls due to the obvious connections between security compromises and the potential for safety issues. Since safety is largely driven by industry standards and regulation extrapolated from the SIL level concept, many of these same people are quick to try and find a quantitative means to understand security and to mitigate in the same manner, through a process of targeted risk reduction based upon the individual resiliencies to fault of the system components. While the SIL model does supply several interesting options to security analysis, the model does fall short in several areas. Safety focuses more heavily on the likelihood of device faults under normal operating conditions, where normal is considered to be all system components available and functioning in their as-designed state, regardless of a given system failure.

Security presents the challenges of being able to violate such design constraints, and also requires additional emphasis on software and network communications integrity. These additional requirements and associated potential security risks require that the SIL model be extended in its scope to incorporate these elements. While safety can operate in a more functional analysis, security requires much more of a component focus to fully understand the resiliency to network based attacks and exploitable software or hardware conditions within the components.

This paper proposes that the SIL model is a suitable model, but the addition of the above suggests that an independent Security Assurance Level (SAL) is required as well. Such a model must look at the resiliency of the component against compromising security controls or designed function. These measurements provide an indication as to whether or not the component is suitable to operate in a given environment. Similar to safety, security should assess whether or not the component is suitable. Rather than look at safe versus unsafe, the model utilized in the motor industry that considers the controllability of the vehicle seems more appropriate. A given process can be considered in terms of controllability of the process in an abnormal state, and rated suitably against the potential consequences or damages.

It is the intent of the authors above to highlight where many of the SIL level concepts are appropriate, and where additional considerations are required, and to propose just such requirements. This document will provide a foundation for continuing research and development in the coming months. Driving towards a successful Security Assurance Level model ultimately benefits industry, and as such continuing work should support international standards and regulatory efforts.

About the Author – Bryan Singer is Vice President of Professional Services with Wurldtech Security Technologies (http://www.wurldtech.com) and Co-chairman of ISA-99 Industrial Automation and Control Systems Security standard. He began his professional career with the US Military focusing on issues such as physical, systems, network security and force protection. Since that time, he has worked in software development in over 25 professional coding languages, worked in UNIX and mainframe systems, supported large scale ERP, MES, LIMS, and SPC implementations, and has spent significant time in cyber security projects focusing on risk analysis, vulnerability testing, penetration testing, risk mitigation strategies, and enterprise architecture and design including technical and policy based countermeasures and remediation strategies. Mr. Singer holds the CISSP and CISM certifications, and runs a personal blog site at http://www.cipiq.com and is an active contributor to http://www.wurldtech.com/blog.

Dr. Nate Kube is co-founder and CTO of Wurldtech Security Technologies (http://www.wurldtech.com). He brings more than 10 years of extensive experience in network security with the proven talent and ability to quickly convert technical ideas to marketable products. As CTO, Kube works closely with Wurldtech's Research, Engineering and Marketing teams to assure a successful implementation of advanced security technologies into SCADA and process automation environments. Dr. Kube worked as a technical advisor to the Group for Advanced Information Technology (GAIT) at the British Columbia Institute of Technology where he designed a number of security technologies for the Industrial Automation industry. He holds a bachelor's degree in mathematics, as well as a doctoral degree in computer science. Dr. Kube has been the recipient of the prestigious: Riddle Memorial Scholarship for Mathematics, National Research Council Graduate Scholarship, and accepted awards from the British Columbia Innovation Council.

References

[1] P. Lindsay and J. McDermid: A Systematic Approach to Software Safety Integrity Levels. Proceedings of the 16th International Conference on Computer Safety, Reliability and Security, York, 7-10 September 1997. Springer Verlag, London, 1997.

[2] F. Redmill: Understanding the Use, Misuse and Abuse of Safety Integrity Levels, Proceedings of the Eighth Safety-critical Systems Symposium, Southampton, UK. Springer, 8-10 February 2000.

[3] IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems. International Electrotechnical Commission, Geneva, 1999.

[4] The Motor Industry Software Reliability Association: Development Guidelines for Vehicle Based Software. The Motor Industry Research Association, UK, 1994.

[5] ANSI/ISA-SP-84.01: Application of Safety Instrumented Systems for the Process Industries, Instrument Society of America Standards and Practices, 1996.