Measuring and Comparing Security in Different Control System Applications

Jason Holcomb, Dale Peterson
Digital Bond, Inc.
Sunrise, Florida
{holcomb, peterson}@digitalbond.com

Abstract: Bandolier Security Audit Files[1] provide a means for control system owners to determine if all of the operating system and application security parameters are in their optimal, most secure configuration. However, the best possible security configuration for one application may be dramatically more or less secure than the best possible configuration for another application. How does an asset owner considering a purchase of a new DCS or SCADA system evaluate the security of systems from different vendors? This paper offers one possible approach.

In this paper, the authors take the Bandolier Security Audit File data and security controls from standard and guideline documents, such as ISA 99 and NIST SP800-53, to identify desired security configuration categories and elements in these categories. These categories and elements are then used in proposed metrics to put a number on the security configuration for comparison purposes. With the detailed application information available in Bandolier, we are able to quantify a range for a security configuration setting rather than a simple yes/no measure typically found in a standard or guideline documents.

Keywords: Bandolier, Security Metrics, Secure Configuration

1 Introduction

Bandolier is a Digital Bond research project that allows vendors and asset owners to use the Nessus Vulnerability Scanner to audit control system components against an optimal security configuration. A team of Digital Bond researchers and control system vendor application and security experts identify all potential security settings in the operating system, control system application and supporting applications, such as web servers or databases. Then a Digital Bond and vendor team of experts determine the optimal setting for each security parameter.

Optimal, in this case, refers to the best possible security configuration for a particular server or workstation based on application requirements. Suppose the application requires a service that has a level of risk and security hardening guides from NIST,

[1] Bandolier is the name of a Digital Bond research project funded by a contract from the US Department of Energy.

Microsoft, CIS and others recommend disabling the service during a security hardening process. Since the control system application requires the service to operate as designed, the Bandolier Security Audit File allows the service to run and still pass the security audit. Traditional IT security configuration guidance often causes control system application specific issues, including failure to work properly, and thus needs to be customized by the Bandolier process to be used in control system environments.

The Bandolier project does not include a mechanism to compare the security of one vendor's system to another vendor's similar system. Each Bandolier Security Audit File represents the optimal security configuration discussed above – the best possible security configuration for servers and workstations running a particular control system application. This is appropriate for cases where the control system has already been deployed or already has been purchased. In these cases the asset owner cannot change the underlying design and features of the SCADA or DCS, but they can insure they have deployed the system with the highest possible security posture.

The Bandolier project, at the time of this writing, has now produced over twenty Bandolier Security audit files for unique application components from eight vendors. It has become evident that the optimal security configuration truly does vary from one application to the next. One vendor's HMI is more secure than another's. One vendor's historian on a DMZ is more secure than another's. The question the Bandolier team has grabbled with is how the detailed information on security parameters from Bandolier audits can contribute to a quantitative measure of security that can be used to compare security between similar systems from different vendors. This knowledge would be a welcome addition to be considered in a purchasing decision.

This paper attempts to answer that question based on a combination of the Bandolier project results and additional security standards. The goal is to create a formula that takes technical security configuration input and produces a numeric value to represent the security level for a particular SCADA or DCS application.

Security metric work has been presented at previous S4 events. Leversage and Byres presented a method to calculate Mean Time to Compromise [1], but it focused on vulnerabilities and exploits. McQueen et al focused on security ideals and metrics [2] that is the work most closely related to this paper. However, only a portion of this dealt with configuration issues and in much less detail as a high level metric approach was pursued. Holstein took a detailed statistical approach that calculated a metric [3] similar to this paper, but he focused on attacks to calculate the statistic. The technical approach in this paper, based on security controls and configuration, is different from previous work that generally focuses on risk, threat, and impact.

1.1 Scope

Standards and metrics research often takes a broad approach, looking at very encompassing high-level issues. Bandolier and this paper, on the other hand, take a more bottom-up, technical approach. The approach involves looking at specific server, workstation and application security configuration settings. This includes ports and services, permissions, authentication, authorization, logging and other factors. As an example of the difference, higher-level standards may address ports and services by

simply stating that they should be kept to a minimum based on the principle of least privilege, the Bandolier approach actually addresses specific ports and services and makes a security judgment.

The authors do not argue that a bottom-up or high level / top-down approach is superior. A set of components that rated highly in a Bandolier / bottom-up approach could be combined in an architecture or other manner that would be highly insecure. However, the top-down approach has limitations that detail is not considered. Vivid examples of this are occurring in NERC CIP [4] compliance efforts where high level security controls are being met while sometimes actually decreasing security and reliability.

2 Process

We begin this effort with a simple premise: two applications can be compliant with a high level standard or guideline document but still have vastly different levels of technical security risk. This was clear to the authors in reviewing the Bandolier Security Audit Files in some important security categories that affected the attack surface in general and for users in various roles.

The Bandolier team could look at any two applications in which Bandolier Security Audit Files were being created and quickly discern if they had an approximate same level of security in configuration or if one application was significantly more secure than another. But this determination was made through experience and an understanding of important security factors. It did not have the rigor of a process nor did it attempt to answer how more or less secure quantitatively.

An important caveat to this process is that it does not address code quality. This could vary a great deal based on a vendor's security development life cycle, and the development teams' security talent and awareness level. So a control system deemed more secure by some quantitative score from the methodology in this paper could still be rife with overflows and other easily identified and exploited vulnerabilities.

The first step in the process was determining what to measure. This considered both what we would like to measure, and importantly what we could measure that would have a range of values rather than a simple yes/no or 1/0. To start an example that will be used throughout this paper, a requirement that a control system component have a list of required TCP and UDP ports and related services, and the component only have those ports/services open does not meet our requirement for a useful component for the metric because the result would be yes or no. However, related valid, measurable questions include: How many ports are open? Are they TCP or UDP ports? Are they well known or frequently attacked ports? Using questions like this, the authors take general requirements from a standard and convert them to a metric. This is possible because of the information available in the Bandolier Security Audit Files.

The authors looked at both the standard and guideline documents as well as the individual Bandolier Security Audit checks to identify potential metrics with the following approach:

1. Extract items that are oriented toward server and workstation security configuration from the standard and guideline documents, namely NIST SP800-53 [5] and the ISA-99 Foundational Requirements [6]
2. Normalize them across the two documents
3. Determine where Bandolier Security Audit Checks fit into the extracted items
4. Determine if there are additional categories from Bandolier that are not covered by NIST and ISA documents
5. Evaluate if the extracted items from steps 3 and 4 support more than a yes / no rating for each of the categories

Step 1 involved a detailed review of the NIST SP800-53 controls to determine which controls are applicable to a technical measure of application security at the server and workstation level. The first pass of review included all the controls within the following control families under the technical category:

- Access Control (AC)
- Audit and Accountability (AU)
- Identification and Authentication (IA)
- System and Communication Protection (SC)

These controls were compared to the Bandolier audit files to determine if what was contained in the files could be correlated with a control. The comparison revealed that a key configuration item, ports and services, did not correlate to any of these controls. To have something to map that part of the Bandolier configuration guidance required looking outside the technical categories into the operational categories. After doing this, a Configuration Management (CM) family was added to the Step 1 review process. The CM family brought the total number of NIST SP800-53 controls that potentially affected server and workstation level security to 69.

The research team then evaluated the 69 controls on an individual basis. Some of the controls address non-technical requirements such as policy and documentation. Those were eliminated to get to the truly technically oriented controls that could contribute to the metric sought by the team and discussed in this paper. That left 42 technical controls.

Step 2 of the plan called for normalization of the standards used. Since the ISA-99 Foundational Requirements already correlate to the NIST SP800-53 controls, there was little work involved. Annex A from the ISA-99 Foundational Requirements document actually maps the requirements specifically to NIST SP800-53.

Step 3 took the Bandolier Security Audit Checks and mapped them to the 42 controls identified in Steps 1 and 2. This process was somewhat iterative as the Bandolier Security Audit Files were used through each analysis cycle as a sanity check to help verify if all the relevant controls were included. The authors evaluated how much coverage of the 42 controls was evident in the Bandolier Security Audit Files. In other words, how many of the 42 controls do the existing Bandolier Security Audit Checks address? The

result was less than 10 of the controls. This further validated the need to pull metrics from both Bandolier and the ISA/NIST controls. Reasons for this narrow coverage are varied. One primary factor, though, is that the Bandolier checks depend on technical automation to evaluate a security configuration setting. If that automation mechanism is not available for a particular setting, then it is not measured.

The purpose of Step 4 was to determine if there are additional categories that exist within the Bandolier Security Audit Checks that do not map to a NIST SP800-53 control. At this step, the team decided to adopt the control family names as general categories and add any additional items not covered by a specific control under the category that fit best. For example file system permissions were not specifically addressed by a control, but they are a common security configuration setting inspected by the Bandolier Security Audit Files. In this case, the team assigned files system permissions to the AC family. It is closely tied to some of the controls in the AC family though it is not addressed specifically by any of them.

The end goal of a numeric metric suggests that, where possible, configuration is measured in a numeric fashion. Many of the controls are a simple yes/no question; "Is measure X implemented?" The team sought to identify where the Bandolier files could help provide meaningful input that is not Boolean in nature. One area where this has a significant potential impact is in the attack surface provided by open ports and services. The more open ports, the greater the likelihood an attacker can exploit a security weakness. This led to several non-Boolean factors that can affect security level.

3 Metric Development

To test the weights and formulas, the authors used a subset of the categories and controls defined in the evaluation process. Specifically, we use the following two categories as prototypes: *Ports and Services* and *File System and Service Permissions*. The categories map into NIST/ISA families – in this case, Configuration Management (CM) and Access Control (AC), respectively. Within each category are specific metrics that, where possible, measure something beyond a yes/no response.

3.1 Identification

The authors identified several metrics within the Ports and Services category. From Bandolier and other control system security experience, it was evident that control system applications handle ports and services in different ways. Many use both TCP and UDP for different components. Some rely on fixed ports for specific purposes while others require broad, often dynamic, port ranges that number in the dozens or even hundreds. This can make host level security difficult but is particularly problematic with communication that spans zones with different security levels separated by a firewall. For this reason, the authors wanted metrics that captured not just the number of ports, but the number that are required for communication across security zones in a typical configuration. Communication with a historian in a DMZ would be an example of cross security zone communication.

Consideration was also given to whether the application requires a "push" or "pull" scenario from lesser to more secure zones. Depending on the type of firewall and communication, this determines whether rules are required for inbound, outbound, or both directions and creates varying exposure levels.

For well-known or common ports, a good security practice is to change the port number so an attacker or malware cannot easily discover it. This is especially true for ports and services with a known history of problems such as web servers and database servers.

All of these factors have to do with minimizing the attack service of the system. They led to the following metrics:

- Number of open TCP ports required beyond a hardened OS configuration
- Number of open UDP ports required beyond a hardened OS configuration
- Percentage of well-known ports that can be changed
- Number of ports required from less secure to more secure zone (inbound)
- Number of ports required from more secure to less secure zone (outbound)

The second category used for prototyping is File System and Service Permissions. Again, the Bandolier project was able to help show disparity in how vendors choose to address the granting of user permissions to files, directories and services. Within the control system applications, the vendors often take great care to ensure that users only have rights to the necessary control functions. This includes role-based authorization and other mechanisms such as Area of Responsibility (AOR). The same type of least privilege, however, rarely extends to the underlying operating systems. Among the worst-case scenarios is where any user on the operating system has full access to the control system applications and directories. This often exposes configuration files that could allow a user to change the way the application functions or change application-level permissions.

For those vendors that do address permissions issues at the operating system level, there are a number of different approaches. The most granular, and arguably the best from a security perspective, are those that correlate operating system security groups to application roles. Examples include "Operators", "Engineers", "SCADA Administrators", and "IT Administrators". This is especially relevant when the operating system account use is used for authentication to the application, a common practice. Some applications take this a step further by assigning application authorization based on operating system group membership. Other schemes used for security groups revolve around the actual permissions they are granted – for example, "Read Only" or "Full Control". On the lower end of the scale are applications that simply differentiate between privileged users and everyone else, usually based on built-in operating system security groups.

At the file and directory level, this category does not concern malware and attackers as much as internal access control. For example, operators are often restricted from accessing operating system functions and IT personnel generally do not have access to the control system applications. Inadequate file system permissions can render these

controls ineffective. Where services are concerned, however, permissions can be a risk for exploitation by an attacker or malware. An attacker that compromises a service often then gains the privileges of the service owner and access to the files used by the service. The controls for this differ slightly between Windows and Unix operating systems. In both cases, it is advisable to run services with the minimal set of privileges needed.

Based on the discussion above, the authors identified the following metrics from the File System and Service category:

- Percentage of application files and directories for which permissions are explicitly defined according to least privilege
- Security groups defined for assigning permissions
- Percentage of server files separated from client files
- Services owned by privileged accounts

The authors also assigned ID's to the metrics based on the control family, category and metric number. For example, the first Ports and Services metric is in the Configuration Management family so its full ID becomes CM-1-M1. Subsequent categories in the CM family would then be CM-2-M1, and so on.

3.2 Scoring

Beyond identification of the metrics for the prototype categories, the authors also had to define scoring scales. Some are simple numbers or percentages, while others need to have ranges assigned to specific scores. Each metric awards a maximum of 100 points and a minimum of 0 points. A higher number equates to better security. Later, the metrics are assigned weights across the categories according to impact to generate a category score. The sections below describe the scoring strategy for each of the prototype metrics.

3.2.1 **CM-1-M1**: Number of Open TCP Ports Required

The scoring for this metric uses a scale. The authors based the ranges on experience from Bandolier and other observation. Of course, a functioning control system must have some open ports to operate. The ports that count in this case are only those that exist beyond a hardened OS configuration.

- 0-1 Ports = 100 Points
- 2-10 Ports = 75 Points
- 11-25 Ports = 50 Points
- 26-50 Ports = 25 Points
- 51+ Ports = 0 Points

3.2.2 **CM-1-M2**: Number of Open UDP Ports Required

TCP is preferable to UDP from a security perspective, especially when the communication must pass through a security perimeter. Therefore the scoring for open ports was more restrictive.

- 0 Ports = 100 Points
- 1-2 Ports = 75 Points
- 3-5 Ports = 50 Points
- 5+ Ports = 0 Points

3.2.3 **CM-1-M3**: Percentage of Well-known Ports That Can Be Changed

Of the well-known ports used by the application, this metric captures the percentage that is changeable by the asset owner in a vendor-supported configuration. A simple example is an application that uses an Oracle database and an Apache web server. If the vendor supports changing the Apache port but not the Oracle port, then the score for this metric is 50.

3.2.4 **CM-1-M4**: Number of Inbound Ports

This metric addresses inbound communication from a less secure zone, such as the corporate network or a DMZ, to a more secure zone, such as the control center zone.

- 0 Ports = 100 Points
- 1 Port = 25 Points
- 2 Ports = 10 Points
- 3+ Ports = 0 Points

3.2.5 **CM-1-M5**: Number of Outbound Ports

This metric addresses outbound communication from a more secure zone to a less secure zone. In this metric we recognized the reality that there are significant operational and economic reasons why control system information should be provided to the corporate network. A perfect score was given to one outbound port, rather than zero outbound ports.

- 1 Port = 100 Points
- 2 - 5 Ports = 50 Points
- 6 - 10 Ports = 25 Points
- 11+ Ports = 0 Points

3.2.6 **AC-1-M1**: Percentage of Application Files and Directories for which Permissions are Explicitly Defined According to Least Privilege

This metric is a simple percentage based on whether the vendor has defined what permissions should exist for the application files and directories. Permissions are often assigned at higher levels and then inherited by lower level files and directories. This is acceptable as long as the vendor has explicitly defined which key files and directories should have additional or more granular permissions.

3.2.7 **AC-1-M2**: Security Groups Defined for Assigning Permissions

This metric captures the granularity available for assigning permissions based on operating system security groups. It awards points based on this scale:

- Groups based on application-level roles = 100 Points
- Groups based on read / write / execute functions = 75 Points
- Groups based on simple privileges (Admin / User) = 50 Points
- A read-only group is defined = 25 Points
- No groups are defined = 0 Points

This scale is cascading, meaning that the higher points categories will likely include elements from the lower level categories.

3.2.8 **AC-1-M3**: Percentage of Server Files Separated from Client Files

Assigning permissions can be much easier if it is done at the directory level. With the way applications group files, it can sometimes be difficult – especially when both interactive users and service accounts need access to a common directory. This metric attempts to capture a simple percentage based on whether files are divided between those used by a client and servers.

3.2.9 **AC-1-M4**: Services Owned by Privileged Accounts

The authors discuss this metric at more length in the metric identification section of this paper. Though Windows and Unix systems treat this issue differently, this metric attempts to capture its extent based on number of services running with privileged accounts.

- All services run with restricted accounts or use chroot = 100 Points
- 1 service runs with a privileged account = 50 Points
- 2 services run with a privileged account = 25 Points
- 3+ services run with a privileged account = 0 Points

3.3 Weighting

All security metrics, even the small sample in this prototype, have differing degrees of impact. The authors chose to weight the metrics in each category based on a total available weight of 1. Higher weights have the effect of placing more emphasis on a particular metric within that category. In the ports and services example, the metric with the highest weight (0.5) is the number of ports allowed inbound between security zones. The ability to change well-known ports, however, is less critical and receives a lower weight (0.1).

The tables below show the scoring and weights for the prototype metrics. The three applications use mocked values to show the highest possible, lowest possible and in-between scores. The table below shows the final set of weights for the prototype metrics.

ID	Metric	Weight
CM-1-M1	Number of open TCP ports required beyond a hardened OS configuration	0.1
CM-1-M2	Number of open UDP ports required beyond a hardened OS configuration	0.2
CM-1-M3	Percentage of well-known ports that can be changed	0.1
CM-1-M4	Number of ports required from less secure to more secure zone (inbound)?	0.5
CM-1-M5	Number of ports required from more secure to less secure zone (outbound)?	0.1
AC-1-M1	Percentage of application files and directories for which permissions are explicitly defined according to least privilege	0.3
AC-1-M2	Security groups defined for assigning permissions	0.2
AC-1-M3	Percentage of server files separated from client files	0.1
AC-1-M4	Services owned by privileged accounts	0.4

Figure 1 – Weights for Two Prototype Metric Categories

4 Case Study

For the case study, the authors chose three applications from the Bandolier project. In the interest of anonymity, the case study refers to them simply as App 1, App 2, and App 3. An asset owner considering purchase of one of the applications would request this data from the respective vendors. In this case, the authors derived or inferred the data from the Bandolier development process. The table below shows the raw data used for populating the metrics.

ID	Metric	App 1	App 2	App3
CM-1-M1	Number of open TCP ports required beyond a hardened OS configuration	108	12	34
CM-1-M2	Number of open UDP ports required beyond a hardened OS configuration	5	0	38
CM-1-M3	Percentage of well-known ports that can be changed	50%	100%	0%
CM-1-M4	Number of ports required from less secure to more secure zone (inbound)?	0	1	0
CM-1-M5	Number of ports required from more secure to less secure zone (outbound)?	1	2	1
AC-1-M1	Percentage of application files and directories for which permissions are explicitly defined according to least privilege	100%	100%	0%
AC-1-M2	Security groups defined for assigning permissions	100	50	0
AC-1-M3	Percentage of server files separated from client files	100%	100%	100%
AC-1-M4	Services owned by privileged accounts	86%	100	0

Figure 2 – Case Study Raw Data for Two Categories

The authors used this data set to populate and test the metrics in a scenario that is very close to a real-world example. In a simple diagram, the architecture of the three applications looks very similar. Figure 2, however, helps highlight how differently the applications work at a technical level, even before scoring and weighting. The number of ports varies greatly as do some other factors such as the ability to change well-known ports and services, security groups, and services running with privileged accounts. Figure 3 shows how the applications score in the Ports and Services category.

CONFIGURATION MANAGEMENT (CM)								
CM - 1 - Ports and Services			**App 1**		**App 2**		**App 3**	
ID	*Metric*	*Weight*	*Raw*	*Weighted*	*Raw*	*Weighted*	*Raw*	*Weighted*
M1	Number of open TCP ports required beyond a hardened OS configuration (0-1) = 100 (2-10) = 75 (11-25) = 50 (26-50) = 25 (51+) = 0	0.1	0	*0*	50	*5*	25	*2.5*
M2	Number of open UDP ports required beyond a hardened OS configuration (0) = 100 (1-2) = 75 (3-5) = 50 (5+) = 0	0.2	0	*0*	100	*20*	0	*0*
M3	Percentage of well-known ports that can be changed (0-100)	0.1	50	*5*	100	*10*	0	*0*
M4	Number of ports required from less secure to more secure zone (inbound)? (0) = 100 (1) = 25 (2) = 25 (3+) = 0	0.5	100	*50*	25	*12.5*	100	*50*
M5	Number of ports required from more secure to less secure zone (outbound)? (1) = 100 (2-5) = 50 (6-10) = 25 (11+) = 0	0.1	100	*10*	50	*5*	100	*10*
		Score	250	**65**	325	**52.5**	225	**62.5**

Figure 3 – Case Study Scoring for Configuration Management

App 2 is the only application with inbound access, a DMZ web server that requires inbound access, and scores the lowest even though it has a much smaller number of ports that the others. This reflects the heavy penalty given by the weight of the inbound access metric. In a real-world scenario, the organization would have to decide if the DMZ web server was a business requirement. If so, the other applications would need to adapt to that model and may score differently. Incidentally, if the inbound access is not included, the ports and services score for App 2 becomes 90 – a clear leader in this category.

Figure 4 shows the scores for the File System and Service Permissions category. The results again show a disparity between the applications. App 3 has no permissions defined for the application files and directories. The heaviest weighting penalty, though, comes from the fact that its services run as privileged accounts.

ACCESS CONTROL (AC)								
AC - 1 - File System and Service Permission			**App 1**		**App 2**		**App 3**	
ID	*Metric*	*Weight*	*Raw*	*Weighted*	*Raw*	*Weighted*	*Raw*	*Weighted*
M1	Percentage of application files and directories for which permissions are explicitly defined according to least privilege (0-100)	0.3	100	*30*	100	*30*	0	*0*
M2	Security groups defined for assigning permissions (App roles / AORs) = 100 (Read / Write / Execute) = 75 (Admin / User) = 50 (Read Only group defined) = 25 (None) = 0	0.2	100	*20*	50	*10*	0	*0*
M3	Percentage of server files separated from client files: (0-100)	0.1	100	*10*	100	*10*	100	*10*
M4	Services owned by privileged accounts (All restricted/chroot) = 100 (1 service) = 50 (2 services) = 25 (3+ services) = 0	0.4	86	*34.4*	100	*40*	0	*0*
		Score	386	**94.4**	350	**90**	100	**10**

Figure 4 – Case Study Scoring for Access Control

5 Results

The case study proves that the metrics and weights reflect the desired result for the prototype categories. The next step is then to assign a weight to each category and determine a composite score for the applications. The authors performed this step as a demonstration of the final, but the results are likely skewed since only two prototype categories are used.

		App 1		**App 2**		**App 3**	
Metric Category	*Weight*	*Raw*	*Weighted*	*Raw*	*Weighted*	*Raw*	*Weighted*
CM-1 Ports and Services	0.8	65	52	52.5	42	62.5	50
AC-1 File System and Service Permissions	0.2	94.4	18.88	90	18	10	2
			70.88		**60**		**52**

Figure 5 – Case Study Weighted Score from Two Categories

The authors heavily weight the ports and services category because of its security impact. File System and Service Permissions, though important, receive a much lower weight. The weighting would be more dispersed with more categories being included, and no category would comprise 80% of the score. However, a 4 to 1 ratio of weighting still may be appropriate between two categories.

App 1 is a clear winner and demonstrates that, even with this limited set of categories, an asset owner can assign a numeric value to the security of a system under purchase consideration.

6 Conclusion and Future Work

The goal of this paper was to determine if the detailed security parameter information learned from Bandolier could contribute to a quantitative measure of security that an organization can use to compare the security of similar control systems. The authors used other standards to help classify and organize the metrics. The case study proves that this is possible, but also shows that there is more work to be done to identify, score, and weight metrics that tie into general technical controls categories. The next step is to develop more metrics based on other security configuration settings and information.

Even in this limited case study it is clear that the weighting will dramatically affect the outcome. While a single number representing security may be the goal for ease of use, an asset owner may want to look at the components in very high and low scoring categories to determine if a small number of settings are affecting the score. And if this is the case, are those small number of settings important to the asset owner?

The Bandolier project had the unintended consequence of the vendors using the audit files internally for a variety of testing. Perhaps a full set of metrics like those proposed in the paper would have the effect of influencing vendors' technical design decisions, thus increasing security levels for new product releases.

Since the most likely user of these metrics is an asset owner making a purchase decision, future work could also include mapping to or integration with the DHS Cyber Security Procurement Language for Control Systems.

About the Author – Jason Holcomb is a Senior Security Researcher at Digital Bond. He is the lead of the Dept of Energy funded Bandolier project, and a team lead for asset owner security assessments. Jason has more than ten years of electric, natural gas and water utility experience with a background in network engineering and information security.

Dale Peterson is the founder of Digital Bond and has been involved in computer security for more than twenty years. He began his career as a cryptanalyst at the US National Security Agency. For the last ten years Dale has focused on control system security and has led numerous research projects, consulting engagements as well as writing and speaking on the subject.

References:

[1] Leversage and Byres, *A Methodology to Calculate Mean Time to Compromise*, Proceedings of the SCADA Security Scientific Symposium, Digital Bond Press 2007

[2] McQueen, Boyer, McBride, Farrar, and Tudor, *Measureable Control System Security through Ideal Driven Technical Metrics*, Proceedings of the SCADA Security Scientific Symposium, Digital Bond Press 2008

[3] Holstein, Dennis, *Security Metrics of Cyber Security Assurance*, Proceedings of the SCADA Security Scientific Symposium, Digital Bond Press 2009

[4] NERC Reliability Standards – Critical Infrastructure Protection (CIP) 002 – 009, http://www.nerc.com/page.php?cid=2%7C20, 2009.

[5] NIST (National Institute of Standards and Technology) SP (Special Publication) 800-53, "Recommended Security Controls for Federal Information Systems", Revision 2, Dec 2007.

[6] ISA-99.01.03, "Security for Industrial Automation and Control Systems: System Security Requirements and Security Assurance Levels", International Society of Automation, Jan 2009.