



# Conceptual Principles for the Security Architect

By Sean M. Price – ISSA member, Northern Virginia, USA Chapter

Join the Discussion  
Connect

This article presents conceptually derived security principles along with those related to security architect objectives and security measurements that can aid security architecture management.

Principles capitalize on the experience of others and offer perspective in dynamic situations.

## Abstract

Managing security architecture dynamic elements is a difficult task. Frameworks are helpful, but gaps remain. Security principles can aid security architecture management. Conceptually derived guiding principles are presented along with those related to security architect objectives and security measurements.

A security architecture is comprised of many dynamic elements: the people, processes, and technology frequently change. Sometimes, those changes are unexpected. Creating a security architecture resilient

against these events is a difficult task. Yesterday's needs are out of step with current trends. Today's design might conflict with future implementations yet imagined. Tomorrow's deployment may not adequately consider emerging threats. Achieving an adequate security architecture seems illusive. How then can a security architecture be planned and managed?

Frameworks express methods to handle tasks such as security architecture management. COBIT, ITIL, and ISO 27000 are a few well-known frameworks. However, their implementations can be problematic. Knowledge gaps and interpretations introduce deviations from one professional to the next. Some situations not clearly handled by a framework may lead to a weakness. Trial and error result in lessons learned. Experience, in this regard, can be a harsh teacher. Avoiding the experience of difficult lessons is ideal. How then do we avoid harsh lessons and better manage a security architecture?

A lesson learned simply stated conveys wisdom. In time, wise phrases are recognized as truths. Formulating a truth into a directive implies a rule or standard. A group accepting a rule or standard as appropriate behavior gives rise to a recognized principle. Adherence to a principle is a wise means to avoid harsh lessons. Principles capitalize on the experience of others and offer perspective in dynamic situations. Security principles, therefore, provide important planning and management guidelines.

Creating a list of principles is a common activity for people within a community of interest. Group members often collaborate in a formal process to establish the list. A code of ethics identifying appropriate conduct is one type of internal use principle list. Both ISSA and (ISC)<sup>2</sup> have their own eth-

ics rules for their membership.<sup>1</sup> Principle lists are sometimes produced for external consumption. The National Institute of Standards and Technology (NIST), for instance, publishes security guidance for the United States federal government. One NIST publication identifies generally accepted security principles, *Generally Accepted Principles and Practices for Securing Information Technology Systems*,<sup>2</sup> and is aimed at security neophytes. In contrast, NIST's *Engineering Principles for Information Technology Security: A Baseline for Achieving Security*<sup>3</sup> targets seasoned security professionals. Both publications are useful to those external to the U.S. federal government.

The principles expressed in this article are taken from experience as opposed to a community of interest. In this regard, they are more conceptual instead of generally accepted rules or standards. Please note that the list provided here is in no way comprehensive. It is hoped, however, that a wise phrase or two will provide an alternate perspective on the reader's security architecture management activities.

The remainder of this article is divided into three sections, grouping security principles from general applicability to specific cases: *Guiding Principles* has applicability to any security endeavor; *Architect Objectives* focuses on the security architect role; finally, *Security Measurement* makes a few observations of security metrics.

## Guiding Principles

The principles in this section have broad applicability to all aspects of security.

### Principle 1 – Documentation is imperative

It takes a lot of time and effort to develop useful documentation. Many people do not enjoy this task. Unfortunately, a poorly documented security architecture has undesirable side effects: undocumented decisions are likely to be revisited, wasting time and energy; inadequate documentation can lead to confusion and poor decisions. It is also important to read existing documentation. Using the works of others can save time and may reveal aspects not previously known or considered. Provide enough detail in the documentation so that someone unfamiliar with the architecture could pickup where you leave off.

During a client-site security compliance review, I was tasked to take a cursory look at the local domain controller. Windows Task Manager was opened and discovered a web server running. I knew the web server was not specified in the security design, noted the discovery as a finding, and completed

my review. Upon return to the main office I contacted the primary security group, asking if the web server was ever authorized. They confirmed it was not. Then I contacted the IT group who informed me the web server was part of a remote management product suite. The product suite was previously approved by the security group. Knowledge of the web server functionality was neither widely known within the security group nor specified in the security documentation. The finding was eventually removed, but not in time to avoid the political fallout.

### Principle 2 – Anticipate unannounced changes

Things seldom stay the same. People come and go. New whiz-bang tools suddenly appear in the system, disrupting how things are done. Change is normal. Keep long-term flexible plans for the security architecture. Evaluate radical changes for their impact, making appropriate recommendations.

### Principle 3 – Plan alternatives for unexpected failures

Nothing is perfect. Vulnerabilities emerge at inconvenient times with unanticipated consequences. Play “what if” games for hypothetical situations. Sketch-out responses to these scenarios. Thinking through a scenario is good preparation when a related event occurs.

### Principle 4 – Integrity is the base of all security goals

Arguably, *Confidentiality* and *Availability* cannot be obtained without *Integrity*. Questionable integrity reduces assurance and increases risk. A breach of integrity enables circumvention of security. This applies equally to people, processes, and technology security controls.

## Architect Objectives

This section places emphasis on the roles of the security architect. However, the recommended principles are applicable to anyone with security architecture responsibilities.

A security architect is the person assigned the responsibility of creating and maintaining an enterprise information security structure.<sup>4</sup> The security architect identifies what needs to be protected, specifies how it should be done, and evaluates the effectiveness of the security architecture. To achieve these objectives the security architect must:

- Identify requirements
- Specify architecture
- Validate design
- Consider risk

Those objectives broadly cover what a security architect does. But, how is the security architecture created? Ideally, aca-

**Systems are always susceptible to flaws introduced by people.**

1 ISSA code of ethics. (2011). Retrieved July 2011, from <http://www.issa.org/page?p=17>; (ISC)2 code of ethics. (2011). Retrieved July 2011, from <https://www.isc2.org/ethics/default.aspx>.

2 M. D. Gaithersburg, M. Swanson, and B. Guttman, *Generally Accepted Principles and Practices for Securing Information Technology Systems* (NIST Special Publication 800-14). National Institute of Standards and Technology. (1996).

3 M. D. Gaithersburg, G., Stoneburner, C. Hayden, and A. Feringa, *Engineering Principles for Information Technology Security: A Baseline for Achieving Security, Revision A* (NIST Special Publication 800-27 Rev A). National Institute of Standards and Technology. (2004).

4 H. Tipton, ISSAP Introduction, *Official (ISC)² Guide to the ISSAP CBK*. CRC Press: Boca Raton, FL (2011).

demia would provide a solution to this problem. Such tasks have been accomplished before. Consider cryptographic aspects of security. We have a variety of security algorithms and protocols with formal proofs. Academia has mathematically shown the correctness and soundness of some. Unfortunately, it is difficult to mathematically prove many aspects of security management. For this reason, the rigor of the security profession is sometimes questioned.<sup>5</sup>

For the purpose of this article we will refer to an enterprise information security structure as the security architecture of a system. The collective physical and logical attributes of the people, processes, and technologies define a system. A security architecture should not be thought of as technology alone. While technology has its emphasis, people are the primary security component.

People, and the processes they implement, have the greatest affect on security. Systems are always susceptible to flaws introduced by people.

### Identify requirements

Security requirements specify the measures needed to protect a system and its data. Requirements can be broadly identified as formal or informal. Formal requirements are those that are documented by a recognized authority and must be followed, e.g., laws, regulations, industry standards, and local policies. In contrast, informal requirements represent guidance that is optional and which may or may not be documented. A best practice guide for a particular technology is an example of documented informal guidance. A manager's decision to implement an expert's verbal suggestion also represents an informal requirement.

#### Principle 5 – Formal requirements need interpretation

Documented requirements derived from laws and policies are often ambiguous and difficult to fully comprehend what exactly is required. This further clouds how they might be implemented. Documenting ambiguous requirement interpretations lays a foundation to derive a security architecture. A clearly defined requirement is easier to implement.

#### Principle 6 – Information requirements need formalization

Requirements without the support of policy are easily disregarded. Some managers will simply side step security best practices if they are not supported by policy. One way to avoid this scenario is to include the requirement in a System Security Policy – a formalized requirement is more likely to be enforced.

### Specify architecture

Security requirements identify *what* must be done to protect the system, but not *how*. We refer to security controls as the methods used to implement requirements. From this view point, security controls indicate how the requirement is achieved. Thoughtful conversion of requirements into a security architecture is a challenging process.

There are many attributes that make up a security architecture. Some of the most common attributes include:

- **Physical** – The locations of system components and users
- **Network** – Think Layer 3 and below of the OSI model
- **System** – Servers and workstations
- **Applications** – Commercial as well as in-house developed software
- **Components** – An inventory of all software and hardware items
- **Settings** – Configurations supporting operations and security
- **Connections** – Interfaces with external systems
- **Dependencies** – Attributes having security and operational reliance on another

Security architecture attributes are the puzzle pieces comprising the system. In most cases, manipulation of the pieces is possible. With that in mind, consider the following principles during architecture specification.

#### Principle 7 – Design is dependent on inventory

Security architecture depends on the composition of the system. An effective security architecture must consider all system components and how they fit together.

#### Principle 8 – Incomplete system knowledge implies incomplete security architecture

Designing a security architecture with incomplete knowledge of the system is frustrating. Imagine putting together a 1,000 piece jigsaw puzzle with 242 random pieces missing. The end result is something difficult to achieve and ultimately less than satisfying. The missing security architecture pieces could result in some ugly holes in the system.

#### Principle 9 – Understand how system components can be exploited

Knowing how something can be abused aides the development of countermeasures. Recurring news reports of SQL injection attacks illustrate this point.

#### Principle 10 – Difficult-to-use security mechanisms will be avoided

Transparent security is ideal, but not always possible. People are resistant to security for a variety of reasons. A security control should be as usable as it is useful. Simplicity in its use is critical.

**Requirements without the support of policy are easily disregarded.**

<sup>5</sup> D. B. Parker, *Fighting Computer Crime: A New Framework for Protecting Information*. Wiley (1998).

**OWASP** is a 501(c)(3) not-for-profit worldwide charitable organization focused on improving the security of application software. Our mission is to make application security visible, so that people and organizations can make informed decisions about true application security risks. Everyone is free to participate in OWASP and all of our materials are available under a free and open software license. Find us on the web at <https://www.owasp.org>

**Bzz.**



**You've heard about application security.  
Now make it visible.**



Register by September 9 with the code **ISSASAVE** for 15% off the OWASP AppSec USA software security conference September 22-23, 2011. [www.appsecusa.org](http://www.appsecusa.org)

### Principle 11 – Poorly described security architectures lead to weak implementations

Security weaknesses occur when it is unclear which requirements must be implemented. Furthermore, it is difficult to obtain a security control's desired effect when it is unclear how it should be implemented.

A security design review with the program manager and lead developer of a database application revealed an interesting problem. I asked if auditing was implemented. Both agreed. The developer stated the design requirements called for audit of database access. I asked the program manager if it was necessary to audit access to sensitive data. The manager replied emphatically that auditing was essential to detect misuse. I asked the developer if any record level auditing was implemented. The developer replied that only logon and logoff was audited. New detailed requirements were created and the application re-entered development.

### Validate design

Assurance in the security architecture is obtained through an examination of the system. This examination determines if the system reflects the requirements and the architecture. Testing within the examination process evaluates the following three objectives:<sup>6</sup>

1. *Security requirements are satisfied* – A test is performed to assess if a given security requirement is met by the system. In some way the people, processes, and technology-related controls must reflect the security requirement. Suppose a requirement stipulates passwords must expire in 90 days. Testing should include all aspects of the system to include network equipment, workstations, servers, databases, and

6 S. M. Price, Access Control Systems and Methodology, *Official (ISC)<sup>2</sup> Guide to the ISSAP CBK*. CRC Press: Boca Raton, FL (2011).



## ISSA International Conference – Design Your Future - Security 2020

October 20-21, 2011 • Baltimore, Maryland - USA • [www.issaconference.org](http://www.issaconference.org)



Stefano Zanero  
Director of ISSA  
International  
& Chair ISSA  
International  
Conference  
Committee

Dear colleagues and friends,

The work of the volunteers from our NOVA, Baltimore, and National Capital Chapters, along with excellent volunteers from the whole ISSA, has shaped this year's conference in a rare glimpse of security over the next 10 years. How will our jobs be in 2020? Let's discuss this together in Baltimore this fall!

You can find an agenda posted on [www.issaconference.org](http://www.issaconference.org). There's a lot of interesting stuff in there. I'm particularly looking forward to the session by former Brazil Chapter president Rodrigo Branco, one of the world's most renowned vulnerability researchers.

And you? Is there a particular session at the conference that you are interested in attending? One that is particularly relevant to your job function? Questions you have for a particular speaker? Let us make the most of the Conference, and begin to discuss here. Post your reply [HERE](#).

2020 is near, you know? :-) See you there! SZ



Eric  
Cowperthwaite  
CSO, Providence  
Heathcare & Co-  
Chair of the ISSA  
International  
Content  
Committee.

I can't wait either. The content committee, which I am co-chair of, has done some really good work to create the content, find speakers, get them engaged, and really make this a fantastic conference. It's hard for me to decide which session I am most looking forward to; they are all great.

But, if I have to pick just one, I would pick the keynote panel on Friday that will be moderated by Bob Bragdon. Bob is the Publisher of *CSO Magazine* for the last 10 years or so. He spends a ton of time talking to security practitioners around the world and has a great sense of where the industry is going, what the trends are, and what tomorrow is going to look like. On the panel will be Dave Estlick of Starbuck's, Roland Cloutier of ADP, Andy Ellis of Akamai, and Kris Herrin of Heartland Payment Systems. That's a group of CISO types who have been there and done that. And they are going to be talking about what you, the security practitioner, need to do to be ready for the future. Can't wait to see all of you there in October.

Posted at <http://connect.issa.org/message/4017>

Join the Discussion  
**Connect**

### Newly Added Sessions: Click the titles for session information

#### [Case Study: How FICO Leveraged Security Intelligence to Improve Customer Relations and the Bottom Line](#)

Nick Schilbe, Director of Solutions Architecture, WhiteHat Security & Glenn Leifheit, Senior Information Security Consultant, FICO (Fair Isaac Corporation)

#### [Death by a Thousand Facts: Criticizing the Technocratic Approach to Information Security Awareness](#)

Geordie Stewart, Managing Director, Risk Intelligence

#### [OWASP Top Ten Tools and Tactics](#)

Russ McRee, Manager, Incident Response, Microsoft Online Services Security & Compliance, Microsoft

applications. Each component providing accounts must have the capability to enforce the requirement.

2. *Security architecture is implemented as designed* – A resulting examination of the system’s management, operational, and technical attributes should reflect the designated security architecture. The security control should exist in every part of the system as specified by the architecture. Now suppose the 90-day password requirement applies only to ordinary user accounts used to access the network. The objective here is to identify gaps in the implementation. Testing seeks to assess if every system component supporting ordinary user accounts is configured to support the policy.
3. *The desired security functionality is obtained from the design* – Testing exercises the resiliency of a security control. The security control should serve its designated purpose and not easily fail. Considering our 90-day password requirement, the system should enforce password changes or account lockout. If an ordinary user can ignore or bypass the 90-day requirement, then the test fails.

Testing is a key security architecture management activity; but, it is a periodic event. System changes between tests can quickly invalidate the design. Assurance is affected by the time spans between design validations: longer time spans rapidly decrease assurance.

Automation of design validation is an ideal way to keep assurance at a higher level. However, there are many non-technical attributes which cannot be easily validated with automation. For example, audit log collection and review is a critical security activity. Validation of the actual review could be automated, but may not be practical due to its manual aspects. Furthermore, automated design validation tools would have a difficult time assessing the quality and completeness of an audit log review. While automation is important and desired, it will not solve all design validation challenges.

Consider the following principles when preparing or conducting a design validation.

#### **Principle 12 – Testing without regard to design is wasteful**

Obtain a thorough understanding of the requirements, security architecture, and system components before testing. Simply running a scanning tool will result in false positives and false negatives. Be mindful of what must be tested and the expected results before validation activities. Otherwise your time and the patients of other will be wasted.

On one occasion a security professional handed me a vulnerability scan with a number of findings. One of vulnerabilities noted was “MTA detected.” He affirmed that this particular issue needed to be corrected. To this day I still cannot understand why he considered a Message Transfer Agent on an email server to be a weakness.

#### **Principle 13 – System components integrated outside of the architecture design often introduce weaknesses**

New software or hardware not considered in the security architecture usually introduces security issues. The capabilities of a new component might unravel prior security efforts. Sometimes perceived risk increases when prior assumptions are invalidated. This problem may occur when change management allows the introduction of a component not integrated with the security architecture design. Ensure items are given due consideration for security before integration. Identify other changes that may need to be made to accommodate the new component.

**Assurance is affected by the time spans between design validations.**

#### **Principle 14 – An unmet requirement is a weakness**

Requirements not supported by the security architecture increase risk. A configuration conflicting with a security requirement implies a vulnerability. Mis-configurations identified by scanning tools often get management attention. Perhaps this is due to the voluminous reports produced. Or, maybe the output is just too intimidating to dispute. But, sometimes non-technical weaknesses are overlooked. It’s important to remember that shortcomings in operations and processes conducted by people affect the security architecture too.

#### **Principle 15 – Validation of requirements met implies compliance not security**

Requirements are seldom precise enough to ensure rigorous security. A thoughtful architecture regularly tested for its resiliency is a good starting point. Aggressive continuous monitoring of all controls leads to security nirvana.

#### **Principle 16 – Resilient designs fail**

If you build it, they will break it. So why try? Resilient designs provide the best bang for the buck. They provide operational efficiency with respect to availability. Examples of resilient designs include defense-in-depth, redundancy, and fail over. Non-resilient designs will fail more often and more spectacularly. The point here is to be prepared for failures. They will occur. Have a plan prepared to deal with this inevitability.

The security architecture’s resistance to failure helps achieve system integrity and availability. These two security goals are quite important. As previously mentioned, everything relies on integrity. But, a loss of availability can be quite noticeable and may directly impact the ability of the organization to function. Keep in mind that availability is usually the most important security goal for IT professionals. A loss of availability during validation testing is unwise.

#### **Principle 17 – Penetration testing alone does not validate design effectiveness**

A successful penetration test exploiting a human weak-

ness (e.g., spear-phishing) confirms what is blatantly obvious. Humans are the weakest link. Likewise, a failed penetration does not affirm the architecture is compliant or designed correctly. Penetration tests are best suited to identify unknown vulnerabilities, evaluate detection mechanisms, and test incident response methods. Use penetration testing to evaluate security architecture resiliency. The ad hoc nature of penetration testing encourages the evaluator to poke at the security architecture from different angles.

A few years ago I evaluated a Windows-based system relying on an automated tool that enforced policy by disabling accounts not used in the last 30 days. Passwords were required

to be changed every 90 days. The security managers were confident in the tool and their compliance standing. I ran a few scripts and dumped all accounts from Active Directory. After sorting the dump I provided the managers a list of active accounts with passwords older than 120 days. The security managers would have identified this issue themselves if they had tested the control instead of relying on what it reported. All controls should be tested from different angles to ensure they are working properly.

### Consider risk

Systems are imperfect and risk will always be present. Software vulnerabilities continue to plague systems. The growth of malware is thought to

exceed legitimate software.<sup>7</sup> The expansion of these problems outpaces security resource improvements. One might conclude the situation would give rise to an increased awareness of operational risk. We should hear a global call-to-arms to constrain risk. And yet, there is hardly a whimper. The security community wrings its hands about the problem, but effective progress has not emerged. Consider the following principles when communicating with those responsible for risk-based decisions.

### Principle 18 – Managers use risk assessments to justify cost avoidance

Sometimes managers have an overly confident view of threat likelihood or vulnerability exploitability. This exuberant optimism enables managers to justify a position of limited security spending. The problem is primarily a lack of awareness as opposed to incompetence. Managers with good estimates of protected data value and unauthorized disclosure costs will find it difficult to justify cost avoid-

ance. Identifying data value and exposure cost may not be easy, but it adds enormous value to the risk assessment. The bottom line (i.e., monetary value) is what managers' care about the most. And, it is the best way to help them evaluate risk.

On one occasion I was tasked to review the contingency plan for a client site located in New Orleans. The plan identified Gulf Port, Mississippi, as the backup site. I raised a concern about the backup site, but the managers apparently considered my analysis over zealous. Two months later hurricane Katrina devastated both cities, invalidating the contingency plan.

### Principle 19 – Risk accepted is not risk avoided

Low-risk issues tend to be readily accepted as do some moderate-risk items, given the costs and potential loss. This is a valid management decision given the correctness of the loss estimates. When this occurs there is the possibility the risk will be disregarded or forgotten as if it was never an issue. In truth, the associated weakness may eventually lead to a serious compromise. These orphaned problems might one day emerge with a vengeance. Therefore, the security architect should seek opportunities to mitigate these risks before an orphaned uprising embarrasses management.

## Security measurement

Not long ago security professionals fought for every penny. Metrics were looked upon as a way to justify a security budget. Presently, regulations and increasingly prevalent threats justify the need for security spending without a budget battle. The issue now is not if resources are necessary, but how much and where.

Interest in security metrics continues to grow. It is reasoned that metrics will improve security resource allocation.<sup>8</sup> Unfortunately, security measurement is not self evident. Determining *what* should be measured and *how* is a problem.<sup>9</sup> With this perspective consider the following security measurement principles.

### Principle 20 – Measurement for the sake of measurement is wasteful

Security data collected for measurement must potentially affect resource allocation. The overall purpose of the measurement should support requirements and program management. It might be interesting to track the number of malware detected by an antivirus solution, but is this useful information? A better measurement might be the number of workstations with installed, activated, and up-to-date antivirus solutions. If the measured information can be used to improve a process, reduce cost, track resources, or validate security, then it is a useful metric.

7 Symantec Internet Security Threat Report Trends for July-December 07 (2007). Retrieved July 2011, from [http://eval.symantec.com/mktginfo/enterprise/white\\_papers/b-whitepaper\\_exec\\_summary\\_internet\\_security\\_threat\\_report\\_xiii\\_04-2008.en-us.pdf](http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_exec_summary_internet_security_threat_report_xiii_04-2008.en-us.pdf).

8 W. K. Brothby, *Information Security Management Metrics*. CRC Press: Boca Raton, FL (2009).

9 A. Jaquith, *Security Metrics: Replacing fear, uncertainty, and doubt*. Pearson Education: Boston (2007).

**If the measured information can be used to improve a process, reduce cost, track resources, or validate security, then it is a useful metric.**

### Principle 21 – Measure discrete operational attributes impacting security architecture resources

Good metrics can be hard to define. Consider the following attributes when selecting what to measure:

- The measure must be quantifiable
- It should be something that can be manipulated by the organization
- An attribute's manipulation is associated with a cost center
- The metric is linked with an aspect of the security architecture

### Principle 22 – A measurement without a meaningful standard is of little use

An item measured must be compared against something known and quantifiable. Example standards include population totals (e.g., network devices), budgeted costs, required values (e.g., configuration setting), and frames of reference (e.g. help desk calls per hour). A standard is needed to evaluate performance, which is the primary purpose of a metric.

### Principle 23 – Management's valuation of a standard is directly proportional to their concern for the metric

Measurements must pinpoint management concerns. Three things managers want to avoid are surprise compromises, budget shortfalls, and findings by external auditors. These concerns and related questions they might raise include:

- **Exposure: How bad is it and what needs to be fixed?** This is a cry for help to manage risk. Providing management with regular measurements of weaknesses reduces surprise when a vulnerability is exploited. It also leverages management accountability by frequent reporting of the problems.
- **Performance: Did we do what was planned?** The concern here is whether the budgeted costs aligned with planned staff activity. This also helps identify areas that require a shift in resources. Performance tracking measurements may reveal aspects of a system insufficiently supported or those consuming disproportionate amounts of resources.
- **Compliance: Are we compliant?** Managers seldom believe auditors are “here to help.” Good metrics help managers identify non-compliant aspects of the security architecture. Non-compliant items might be low risk, but they can still embarrass management. Measuring items auditors are likely to find helps managers to avoid uncomfortable situations.

The concept of using metrics with security architecture is still maturing. But, performance evaluation is the hallmark of professionalism. If we wish to advance our profession beyond “folk art,”<sup>10</sup> then we must increase the rigor of our methods and recognized principles.

10 D. B. Parker, *Fighting Computer Crime*.

## Conclusion

The dynamic nature of systems and threats quickly overcomes static architecture designs. Long-term perspectives are needed to anticipate and counter these issues. Security architecture planning and management is enhanced when security principles are considered. The experience and wisdom of others can point out warnings or suggest best practices. Wise and truthful perspectives help the security architect determine the best way to manage an ever changing environment.

A security architect can improve his or her performance with the consistent application of security principles.

## About the Author

Sean M. Price, CISA, CISSP, is an independent security researcher and consultant living in northern Virginia. He specializes in designing and evaluating organizational information assurance programs and system security architectures. Research interests include insider threat, information flows, and applications of artificial intelligence to information assurance problems. Prior publications include contributions to the *Information Security Management Handbook*, *Official (ISC)<sup>2</sup> Guide to the CISSP CBK*, *IEEE Computer magazine*, as well as other journals and conferences. You can reach him at [sean.price@sentinel-consulting.com](mailto:sean.price@sentinel-consulting.com).



## ISSA Journal Editorial Calendar

SEPTEMBER – DUE 8/1/11

Trusted Platform Modules

OCTOBER – DUE 9/1/11

The Cloud / Virtualization

NOVEMBER – DUE 10/1/11

Risk Management:  
Making Theory Work in Business

DECEMBER – DUE 11/1/11

IT & Security Governance

For theme descriptions,  
visit <https://www.issa.org/page/?p=282>.

EDITOR@ISSA.ORG • WWW.ISSA.ORG