Toward Systemically Secure IT Architectures*

Glenn M. Brunette, Jr. and Christoph L. Schuba
Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 - USA
{Glenn.Brunette,Christoph.Schuba}@Sun.COM

Abstract

This paper addresses the need for strong security guarantees in increasingly dynamic and flexible information technology (IT) environments. It applies time-tested security principles, architectural patterns, and continuous improvement constructs to weave security controls and assurances more systemically into an IT environment. Using a building block approach and a focus on iterative refinement, organizations can transform their existing legacy deployments into resilient architectures that meet not only their security, privacy, and compliance needs, but also satisfy business goals such as increased agility, flexibility, efficiency, and availability.

1 Introduction

We are moving toward a future with more and more computers, cell phones, PDAs, and other devices connected to one another over a digital heartbeat. Traditional organizational and network borders continue to blur as organizations allow greater levels of access to their customers, partners, and suppliers. Furthermore, the delivery of services is becoming more streamlined as associations between components and data become more dynamic in response to *just in time* business decisions.

We must never forget, however, that with all of the benefits that these opportunities offer, also comes risk. We have all witnessed the speed at which attacks can now be executed on a global basis using the Internet or other communications networks. We have also been inundated by news of identity theft, industrial espionage, and the ever present insider threat. Worse still, this trend does not seem to be improving.

As of mid 2005, the IT landscape is riddled with limited or unenforced policies, incomplete and unevolving processes, unimplemented recommended practices, and ineffective architectures. Many IT environments suffer from unmanaged or worse yet unidentified risk. Often, these organizations are operating with serious security, privacy, and compliance exposures - perhaps without realizing they have a problem.

Consumers are growing more concerned with security and privacy issues and how they impact their lives. Organizations and industries (particularly those supporting critical infrastructure) have been placed under greater regulatory scrutiny. Recent regulations and well-published security failures have pushed security, privacy, and accountability out of the data center and into the boardroom, making it an executive issue. Security failures can adversely impact consumer trust and retention, shareholder confidence, competitive advantage, and market perception. Simply put, market tolerance for security and privacy failures is shrinking.

For these reasons, it is critical that organizations view security as an essential quality that must be ever present in their business and IT architectures and operations. To achieve such a state, however, organizations must first understand how to manage risk, cost, and complexity to strengthen their environments without constricting flexibility, agility, or efficiency.

1.1 Systemically Secure Architectures

Managing risk, cost, and complexity effectively requires that a careful balance be achieved across business and technical boundaries. Architectures must be flexible to respond to ever-changing business opportunities, policy and regulatory pressures, and evolving threat profiles. The systemically secure architectures concept addresses the problem of architecting, implementing, and managing IT environments where everything and everyone is *securely* connected to the network.

*This paper is a summary of the keynote presentation entitled *Systemically Secure Architectures* held on June 14, 2005 at WET ICE 2005 at Linköping Universitet in Linköping, Sweden, by Christoph Schuba of Sun Microsystems, Inc.

There are many ways to create systemically secure architectures. We recognize that one size does not fit all. Our approach centers around the use of architectural building blocks with well-defined properties and interfaces that can be assembled in a variety of ways depending on the needs of an organization. Fundamental to this approach is the use of (1) architectural patterns to build security into each step of the process, (2) time-tested security principles applied at times in unconventional ways to reap greater security rewards than would otherwise be possible, and (3) continuous improvement methods to realize greater levels of integration, efficiency, and alignment with business goals.

It is well-recognized that it is easier, less painful, and more cost effective to *do security right the first time.* Unfortunately, security has not always been a key criteria in the design, implementation, and management of IT environments. As a result, organizations often need to adapt their current environments to support stronger security, privacy, and compliance efforts as they emerge. The complexities and challenges facing existing deployments can be daunting and can leave organizations frozen as they search for where to begin.

By taking an architectural, comprehensive building block approach to security aligned with time-tested security principles and continuous improvement methods, this complexity can be managed. Organizations can transform their existing IT infrastructure into a secure, compliant, agile, and optimized environment that delivers not only compliance, but also sustained business value. While this transformation cannot be achieved in a single day, a roadmap can be developed and continuous progress made a reality by stepping back and looking at the ways that security can be systemically integrated into policies, processes, architectures, and services.

1.2 Building Blocks and Transformation Phases

The systemically secure architectures approach centers around the use of modular, standardized, and composable building blocks aligned with an automated, repeatable, and auditable process. This approach allows for the capture, reuse, and refinement of knowledge about IT infrastructure and applications as well as the inter-relationships between building blocks and their underlying components. Furthermore, it enables a better understanding of which patterns may be appropriate for a given situation given a set of requirements, dependencies, and constraints.

Beyond the patterns themselves, we have worked to develop continuous improvement methods enabling organizations to advance through several transformational phases each of which can result in greater levels of consistency, efficiency, and of course security and compliance. As organizations continue down their path of improvement, they will find themselves growing more able to focus their efforts on their strategic goals and optimization rather than on continuously fighting fires.

In the following sections, we present several of the common architectural building blocks that can be used and combined by organizations to achieve a stronger security posture, followed by a discussion of several transformation phases through which organizations can progress on their way to realizing systemically secure architectures.

2 A Building Block Approach

There is no one set of building blocks that will work for everyone. Every organization has its own set of policies, priorities, and business and security goals. There are common traits that can be found between organizations, however. This observation is reflected in some of the common patterns that have been developed and are discussed below. Organizations should consider, assemble, and combine only those building blocks that are appropriate for them.

Each building block has some intrinsic merit. By implementing a building block, an organization should be able to realize some security or compliance benefit. The degree to which a benefit can be realized is based on how systematically it can be applied to and integrated with the overall IT architecture. Greater rewards can be achieved for organizations leveraging and integrating multiple building blocks because of their ability to reinforce one another.

We have selected several architectural building blocks that will serve to better illustrate this concept. It is not our intent to exhaustively discuss this subject matter, rather, to provide representative samples that are indicative of our systemically secure architecture approach. The building blocks that are discussed in more detail below include secure components, secure execution containers, secure network enclaves, consolidated, shared service models, enterprise grid architectures, portal-based architectures, and thinner client architectures.

2.1 Secure Components

All IT environments are comprised of discrete components. These components can take the form of hardware platforms, firmware, operating systems, middleware, or applications. All too often, organizations find themselves vulnerable to attack, at this most basic level, because individual components have not been properly secured. Time and time again, components have been exploited because the components were not secured in accordance with industry recommendations or an organization's own security policies.

This problem is compounded by the fact that to deliver some measure of business value often many components

must be layered or combined, thereby increasing the likelihood of a vulnerability or exposure being present in the resulting configuration. Individual components must be secured regardless of the their type and similarly any aggregation of components must also be secured. While quality may be present trivially by combining secure components, in general additional steps must be taken to secure the aggregate ([5]).

While the actual steps taken to secure individual components will vary by component type and vendor, often there are a number of ways in which a component can be tuned for greater security. Some of the more common ways include minimization, hardening, role-based access control, configuration and integrity assessment, and auditing.

Security recommendations for specific products are often documented in vendor, industry group (e.g., The Center for Internet Security (CIS)), or government organizations (e.g., National Institute of Standards and Technology (NIST)) publications. Regardless of the source or method used, securing individual components is the first step toward establishing a secure IT foundation.

2.2. Secure Execution Containers

Building upon the foundation provided by individual components that have been secured, we look at ways in which services and applications can be compartmentalized to protect them from accidental or malicious attack and to contain the damage that can result from a service being successfully exploited. As a core building block, secure execution containers refer to the ability of an application or service to run in its own isolated compartment with exactly the privileges and capabilities that it needs to perform its business function ([8]).

Usually, only one service or application is placed within a single secure execution container. This convention is used to establish well-defined rules about what the individual container can and cannot do and to define what constraints may be placed on the container. There is nothing inherent in the concept of a secure execution container that would preclude more than one service being installed within a container, however. Organizations must do their own risk assessment to understand if the expected reward outweighs the potential risk.

This building block attempts to apply security principles such as self-preservation, defense in depth, least privilege, and compartmentalization to the domain of service delivery. The actual method used to construct a secure execution container varies based on organizational requirements, product capabilities, and of course the threat profile for a given service or application. Some services may require physical separation whereas others may employ electrical, logical, or resource-level separation to achieve similar goals albeit with different degrees of assurance.

For example, secure execution containers can be constructed at the platform and operating systems layer using the Solaris Operating System using a variety of methods such as employing individual systems (physical), Sun dynamic system domains (electrical), Solaris 10 containers (logical), virtual machine technology (logical), and chroot environments (logical). While this example focused on operating systems, a similar approach can be applied to the application domain as well.

2.3. Secure Network Enclaves

A progression from the use of secure execution containers are secure network enclaves. These building blocks are used to compartmentalize and control access to communities of users and services at the network level. They act as network guards that only permit communication over approved protocols between approved parties. Just as with secure execution containers, individual enclaves only support a single service or user community although this choice is a matter of convention and is not a requirement. Secure execution containers may consist of a single or multiple execution containers depending on the reliability, performance, and availability requirements for a given service. A fundamental design goal is that the enclave itself should not need to be modified to adjust capacity or reliability characteristics.

Secure network enclaves can be used to group communities of users such as those residing in the same physical office location or in logical departments (e.g., human resources or engineering) that may span multiple physical locations. They can also be used to compartmentalize access to specific infrastructure or application-level services. In this capacity they can be used as part of a defense in depth strategy to contain breaches of security and to curtail the spread of malware through an enterprise.

We apply the principle of least privilege to secure network enclaves by restricting access to the enclave based on the protocols or services that it is permitted to import or export and by the user or service communities with whom the enclave is permitted to communicate. This approach represents a service-centric view for organizing and securing access to network services.

For example, users in an engineering community may not be able to directly access finance services (nor the containers in which those services reside) except to access a paycheck application, access to which has been granted to that community. Everyone is untrusted until proven otherwise through a process of identification and authorization. Once authenticated, users and services may then only access services to which they have been authorized. Today, this approach represents a paradigm shift from how

networks are protected because this fine-grained service-centric access control model replaces the much cruder model that was based on the distinction between external and internal access.

Secure network enclaves often have the following capabilities: well-defined service interfaces (inbound, outbound, and management), default deny access policies, virtualized point of access (e.g., IP address), intra-enclave compartmentalization, and load balancing or clustering support. Not all of these components need be used in every enclave instance. As with other security controls and patterns, organizations must leverage the proper controls that manage risk, cost and complexity. So far, there has been no real discussion of traditional network security controls such as firewalls, proxies or intrusion detection systems. This omission is by design. It is not our intent to imply that these controls are not appropriate, but rather that there are other architectural approaches that can be used to help secure modern networks. Particularly with the convergence of security and networking technologies, new options become possible. That said, organizations should always carefully assess the risks and threats to their services and employ additional controls where appropriate to manage their risk to an acceptable level.

2.4. Consolidated, Shared Service Models

The concept of secure network enclaves can be extended to implement consolidated, shared service models for both infrastructure and application services. By leveraging and connecting groups of secure network enclaves based on business and technical requirements, organizations can more effectively deliver services on demand to only those who need them. Provisioning or using a new service then becomes a matter of establishing a connection between relevant enclaves. Services themselves are supported by a strong, layered set of security protections as they have been built upon the foundation provided by the earlier building blocks.

Example infrastructure shared service farms could support services such as DNS, HTTP, FTP or LDAP. In a similar fashion, application-level shared services could be modular web-services components or connectors integrated within a service-oriented architecture. The secure network enclaves provide the underlying infrastructure to deliver these shared components that are then composed and aggregated at the application layer to construct business services. Individual web service components could provide critical security services such as identification, authentication, authorization, and auditing and can be aligned and integrated with their infrastructure counterparts.

The shared services approach has a number of security benefits that are relevant for systemically secure architectures. Chief among them is the application of compartmentalization, least privilege, and defense in depth in their inherent design. Simply put, shared service farms benefit from the structures from which they are composed. Furthermore, by leveraging common components and configurations, shared services can be more easily secured, because each of the shared service components is grouped (physically or logically) with its peers rather than being scattered across an enterprise where change and configuration control issues become more challenging. Said another way, rather than having to scour an entire network for all services of a given type so that a critical fix can be applied, organizations become able to focus their efforts on just a single or small set of shared service farms (that provides a given service).

Reducing component level diversity yields less variation, making shared services typically easier to secure, maintain, and monitor. Should a vulnerability be found within a shared service, it can be more easily eradicated because its configurations and interfaces are well-defined and understood. This approach allows organizations to focus their remediation efforts in a more fine-grained way to more quickly and easily eliminate the problem. Lastly, the shared service construct is useful for detecting non-compliance as each of the components used to support a shared service should generally be configured in a consistent manner. Any deviation from the expected result could trigger an alarm.

2.5. Enterprise Grid Architectures

The enterprise grid architecture is a recent approach to managing traditional data center environments. From a security perspective, most of the components and interrelationships remain the same, but the way in which components are provisioned, personalized, and managed changes in an enterprise grid architecture. Such architectures are predicated on the use of a common management framework that is used to help reduce the time, cost, and complexity associated with more traditional forms of building out and managing systems, networks, services, and entire data centers.

Systemically secure architectures benefit from and are reinforced by such centralized management capabilities. They can be used to enforce consistent policies across grid elements, to reduce deployment time and complexity, and to improve repeatability of process and consistency of configurations across sets of deployed elements whether they are secure components, secure execution containers, secure network enclaves, or the shared services that are built upon them. Enterprise grid architectures can be employed to not only improve consistency, awareness, and security, but they can also be used to automate the creation, assembly, and management of systemically secure architecture building blocks.

Centralized management frameworks can support the assessment or validation of components, containers, enclaves, and services allowing organizations to detect and respond more effectively to breaches of security or process (e.g., configuration or change control). More concepts and products able to construct and maintain systemically secure architectures are expected to be developed and standardized as the technology behind enterprise grid architectures matures. However, enterprise grid architectures themselves are not without their own class of security risks and concerns. In mid 2005, organizations such as the Enterprise Grid Alliance are working to identify and remedy those security issues specific to enterprise grid technologies.

2.6. Portal-based Architectures

The portal-based architecture building block serves a number of uses in a systemically secure architecture. Firstly, it provides a barrier between user communities (and their respective network enclaves) and the network enclaves that are used to offer specific services. In this regard, a portal is acting as a proxy that requests services on behalf of a user.

Secondly, portal based architectures provide an aggregation point that allows an organization to focus its access requests on a single (or small set) of locations rather than opening up access from anywhere on the network. Some portal technologies can also detect the type of device used to access the portal thereby providing an additional criteria to be used for access control decisions. For example, a user may be allowed access to sensitive information or functions when accessed from a more trustworthy source such as a dedicated system connected within the organization as opposed to from a PDA connected via a third-party service provider.

Lastly, portal-based architectures can provide a consistent and centralized interface for users wishing to access services offered by an organization. Portals are therefore in the position to leverage a unified identity and access management service to grant access to those services to which a user has been entitled. Portals can inspect and audit user activities. In this capacity, traditional portal-based architectures may be augmented with additional security capabilities such as application proxies or XML firewalls. Should a user no longer need access to a service, a portal provides a centralized inspection point where access can be revoked. While a portal can perform initial authentication for users, some applications may require that a user re-authenticate using the same or different mechanisms to grant access to specific services or functions.

2.7. Thinner Client Architectures

Traditional fat-client desktops are not only a costly solution to providing ubiquitous access to services, they are also a source of many well-documented security problems including software piracy, data theft and loss, and malware infection and propagation. Despite all of these challenges, organizations continue to invest (often quite heavily) in reinforcing the security of their fat client deployments through the purchase of various bolt-on packages.

The use of fat client technology amplifies the security challenges facing organizations today for a number of reasons. Primarily the sheer number of deployed systems often makes it difficult and costly to ensure that they are operating in a consistent and safe manner. Furthermore, organizations often lack sufficient control over what software is installed on those platforms by end users either intentionally or otherwise. Similarly, data is often copied to or cached on desktop platforms where it may not be safeguarded to the level required by an organization's policy. Finally, fat clients have an intrinsic value making them valuable targets for theft. Once stolen the information stored locally on the fat client can be accessed, used, or sold, with the potential for causing damages far beyond the intrinsic value of the stolen fat clients.

Thinner client architectures solve many of these security issues through the effective creation of desktop utility environments where small, stateless networked devices are used to replace traditional fat clients as the desktop. These devices have no local configuration, storage, or state and must be used in conjunction with a server environment. As a result, there is no longer a need to deploy security controls on each and every desktop because that is accomplished centrally through the desktop utility. Thinner client architectures have a much lower intrinsic value and are therefore a less interesting target for thieves.

Thinner client architectures also have the benefit that they help simplify the security problem by providing a single control point for accessing and delegating access to services and data whether through portal-based architectures or through more traditional means. The level of centralization also means that configurations and software can be more rapidly updated or patched in response to security alerts. For example, a security patch could be applied to one system to correct a security flaw impacting hundreds of users rather than having to distribute and implement the fix to hundreds of desktops individually.

3 Transformation Phases

So far, we have discussed several typical building blocks that can be used in the construction of systemically secure architectures. These technological building blocks alone are

necessary though insufficient for those organizations wishing to reap the benefits of systemically secure architectures. To fully realize their true potential, organizations must jpin these architectural patterns and building blocks and their existing policies to a process of contiunuous improvement and iterative refinement.

Through the use of such improvement methods, organizations will be able to better manage complexity and cost while constructing a more agile, flexible, and compliant architecture capable of meeting their business goals. By progressing through the transformational phases listed below, organizations may find that they are better able to react more quickly and effectively to security emergencies, reduce the number of defects associated with security configurations, and improve the focus and effectiveness of existing security teams through a continuously improving and proactive approach to security.

Few organizations have the luxury of starting fresh with their IT landscape where security can be considered from the start. Rather, most organizations need to adapt their existing, legacy deployments to transform them to systemically secure architectures. For some organizations, this process may be as simple as a few minor adjustments to their overall IT security plan. For others, it may be more of an evolutionary process that will take a sustained commitment of time, money, resources, and organizational focus.

We have identified several transformational phases through which organizations progress in order to improve upon their level of architectural and operational maturity as well as to better integrate and align IT security with their business goals. The transformational phases that are discussed in more detail below include consolidation, standardization, automation, and optimization.

Organizations cannot rush through each of these phases, but instead, they must mature gradually through the various phases. Different aspects of IT security can and often do exist in different phases at the same time. Organizations may make informed risk management decisions that limit how far they want to progress in a given area. These are both normal and acceptable.

The systemically secure architectures concept is not about simply meeting some number of items on a check list. Rather, it is a flexible model that adapts to organizational policies and needs with the goal of transforming an existing IT architecture into a secure, compliant, agile, and optimized environment capable of delivering not only compliance but also sustained business value.

3.1 Consolidation

Organizations focused on the transformational phase of consolidation are focused on managing variance in their environment. All too often organizations suffer from the lack of structured configuration and change control which results in endless variation through an IT environment. Note that we are not talking about creating homogeneous environments; instead, we are talking about managing IT diversity to an acceptable level that can be more easily managed and secured.

This first phase consists of consolidating existing deployments into a manageable set of systems, software, and configurations. Over time, existing practices often become less scalable, configurations become less secure and consistent, and administration become more resource intensive as a result of unchecked variation in an environment. For example, differences between configurations may result in unmitigated vulnerabilities, inconsistent interfaces, or even support and troubleshooting problems. The greater the diversity, the higher the probability that something will be missed along the way, creating security vulnerabilities and exposures. By reducing this level of diversity to an acceptable level, organizations can be better prepared to move to the standardization phase discussed next.

During the consolidation phase, organizations commence by creating an inventory of which types of computing systems, operating systems, application software packages, or configurations exist and by consolidating them (where possible) into a set that has just the right mix of diversity. We will not go into details as to what constitutes the *right* set of systems and configurations here because it depends too much on individual circumstances. Suffice it to say, keeping diversity at a minimum has security, management and cost advantages, while increasing diversity can help mitigating various types of risks, such as those caused by dependencies on vendors, staff, or service providers.

3.2 Standardization

This second phase is called standardization and goes hand in hand with the first phase of consolidation. Leveraging standardized configurations provides assurance of which interfaces and capabilities individual IT elements possess. Once these attributes are well understood, greater levels of automation and optimization can be implemented to further improve security, consistency and efficiency. Without such standardization, organizations will not be able to respond as quickly or completely as they otherwise could. For example, if each system in an organization were configured differently, it would be very difficult to move services between systems in a timely manner in response to changing business conditions, to determine which systems may be at risk due to a recently announced vulnerability or to recover disabled platforms should disaster strike.

The development of internal standards and reference configurations applies to middleware and applications as much as it does to hardware platforms and operating sys-

tem software. We are not necessarily advocating a single standard that should apply across an entire organization, but rather there should exist a small and manageable number of configurations that collectively support an organization's goals. While not every element may be able to take advantage of such standards, exceptions should be just that: exceptions and not the rule when considering an organization as a whole.

The phase of standardization consists of developing and documenting security policies, standards and processes if they do not already exist. Written checklists and recommendation guides help ensure that all IT elements are configured in accordance with company standards and/or vendor and industry accepted recommendations. Recommended interfaces, services, protocols or options used by applications and services must also be documented, where appropriate, in order to provide critical IT security guidance to system administrators and application developers alike. As organizations move toward systemically secure architectures, it is critical that such interfaces and standards be used as the IT architecture and services evolve to maximize security and efficiency.

These standards need to exist for all products and services that are used and need to be enforced through a strong IT governance process. Further, they need to be kept up to date as installations change over time. For example, a security checklist for an older version of an operating system may be incomplete and fail to address key security issues introduced in a newer version of the product. Documented standards provide a baseline against which to verify that systems are configured in accordance with organizational policy and are the prerequisite for the subsequent phase of automation.

3.3 Automation

Once organizations have consolidated diversity in their environments and established well-defined capabilities and interfaces for their IT components, higher levels of automation become more possible. Automation allows organizations to manage IT environments that may fundamentally be more complex than they could otherwise manage because it can provide a buffer between administrators and the inherent complexity of the environment.

Automation has traditionally been applied to IT component provisioning and to a subset of typical operations activities such as data backup. Automation can be used to enforce consistent installation, upgrade, configuration, and patching of IT elements whether at the infrastructure or application level. In addition to simply making changes to IT elements, automation can also leverage the policies and standards that have been developed to assess deployed components to detect deviations and to spot and potentially correct non-compliant configurations.

In fact, by examining the entire life cycle of security, automation can be used to apply security changes throughout an IT environment in accordance with specified profiles or policies. Automation can provide the ability to undo changes previously made and assess deployed configurations to detect if they are in compliance with the profile against which they had been secured in the first place. By viewing IT security through this perspective, new opportunities emerge, allowing organizations to leverage automation in increasingly more sophisticated and comprehensive ways (starting small and using a building block approach) in order to recapture time, money, and resources that can most certainly be spent in a more productive and ideally strategic manner.

Automation is also an excellent opportunity to capture process and knowledge. All too often organizational memory is captured only in the minds of its employees. As employees transition to new roles, retire, or find new opportunities, knowledge and history is invariably lost. By codifying this knowledge into automated process, organizations can begin to move from a culture of heroes to one that delivers a more consistent, repeatable, and measurable experience.

3.4 Optimization

The final phase is optimization. This phase takes advantage of the phases that have come before in order to look for greater levels of security and efficiency. Organizations are not static and whatever worked yesterday may not necessarily work tomorrow. It is critical therefore that organizations not only maintain their consolidated and standardized configurations and automated processes but that they also look for ways in which they can be improved. Failures, root cause analysis, and lessons learned are leveraged as part of this phase to improve existing work and to prevent the same failures from occurring again. Organizations at this phase are proactive with respect to security and are actively learning from their mistakes.

Organizations who reach the optimization phase will often find themselves in compliance with external regulations and internal policies. Liability management will be in a state of equilibrium allowing organizations to refine existing practices and focus their efforts on increasing business value in addition to strategic alignment with business goals and future directions.

4 Conclusions

This paper has introduced the concept of systemically secure architectures, a comprehensive, architecturally focused approach to securing IT environments. Using ar-

chitectural patterns and building blocks founded on time-tested security principles, organizations can more readily construct architectures that can meet their security, privacy and compliance needs. Several of the more common building blocks that were discussed in more detail include secure components, secure execution containers, secure network enclaves, consolidated, shared service models, enterprise grid architectures, portal-based architectures, and thinner client architectures.

Constructing secure IT architectures is relatively easy compared to the challenges of keeping them secure. Recognizing this fact and knowing that most organizations will need to adapt their existing, legacy deployments, the paper identifies the four transformational phases of consolidation, standardization, automation, and optimization through which organizations must progress in order to realize the the security, agility, and efficiency benefits afforded by the systemically secure architecture approach. Through each of these transformational phases, organizations can work to improve upon their level of architectural and operational maturity, as it pertains to IT security.

Together, these elements can be leveraged, adapted, and combined to help organizations construct and maintain secure IT architectures that are able to satisfy their security, privacy, and compliance needs while still supporting (and potentially accelerating) other business and IT objectives.

Acknowledgments

We are grateful to acknowledge the support of Hal Stern, Jim Baty, David Jones, Barbara Kay, and Bruce Gossard in preparation of this manuscript and the invitation by the general chair of WET ICE 2005, Nahid Shahmehri, to deliver its content as a keynote presentation.

References

[1] G. M. Brunette Jr. Adaptive Security for Dynamic Computing Environments. In *Proceedings of the SUPerG Conference*. Sun Microsystems, Inc., Apr. 2004.

[2] E. Bryant, J. Early, R. Gopalakrishna, G. Roth, E. H. Spafford, K. Watson, P. Williams, and S. Yost. Poly^2 Paradigm: A Secure Network Service Architecture. In *Proceedings of the* 19^{th} *Annual Computer Security Applications Conference*, pages 342–351, Las Vegas, Nevada, Dec. 2003. IEEE Computer Society.

[3] Z. Liang, V. N. Venkatakrishnan, and R. Sekar. Isolated Program Execution: An Application Transparent Approach for Executing Untrusted Programs. In *Proceedings of the* 19^{th} *Annual Computer Security Applications Conference*, pages 182–191, Las Vegas, Nevada, Dec. 2003. IEEE Computer Society.

[4] P. G. Neumann. Achieving Principled Assuredly Trustworthy Composable Systems and Networks. In *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03)*, Washington, D.C., May 2003. Defense Advanced Research Project Agency.

[5] P. G. Neumann. Principled Assuredly Trustworthy Composable Architectures. Final Report Contract number N66001-01-C-8040, Defense Advanced Research Project Agency, Dec. 2004.

[6] A. Nordergraaf and G. M. Brunette Jr. *Securing Systems with the Solaris Security Toolkit*. Sun Microsystems, Inc. BluePrint Publications, Apr. 2003.

[7] A. Nordergraaf and K. Watson. *Solaris Operating Environment Security*. Sun Microsystems, Inc. BluePrint Publications, June 2002.

[8] D. Price and A. Tucker. Solaris Zones: Operating System Support for Consolidating Commercial Workloads. In *Proceedings of the* 18^{th} *Usenix LISA Conference*, pages 243–256, Atlanta, Georgia, Nov. 2004. USENIX.

[9] S. Probst, W. Essmayr, and E. Weippl. Reusable Components for Developing Security-Aware Applications. In *Proceedings of the* 18^{th} *Annual Computer Security Applications Conference*, pages 239–248, Las Vegas, Nevada, Dec. 2002. IEEE Computer Society.