TOWARD SYSTEMICALLY SECURE IT ARCHITECTURES

Glenn Brunette, Client Solutions

Sun BluePrints™ OnLine — February 2006

Part No 819-5605-10
Revision 1.0, 2/2/06
Edition: February 2006

Please
Recycle

TABLE OF CONTENTS

Toward Systemically Secure IT Architectures

This Sun BluePrints™ OnLine article addresses the need for strong security guarantees in increasingly dynamic and flexible information technology (IT) environments. The Sun Systemic Security approach applies time-tested security principles, architectural patterns, and iterative refinement policies to weave security controls and assurances more systemically throughout an IT environment. Using a pattern-based 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 other business goals, such as increased agility, flexibility, efficiency, and availability. In fact, this approach can be used to help drive the adoption of new service and utility-based compute architectures.

This article contains the following sections:

Introduction

The convergence and availability of greater numbers of computers, mobile phones, PDAs, and other devices are fueling new opportunities and new styles of sharing, participation, and commerce. Traditional organizational and network boundaries continue to blur and fade as organizations find new ways of engaging their customers, partners, suppliers, and employees. Furthermore, the delivery of services is becoming more streamlined, as associations among components and data become more dynamic in response to “just in time” business decisions. Unprecedented levels of access and sharing are fast becoming the norm and helping to fuel what is being called “the Participation Age.”

Security risk accompanies all of the benefits that these opportunities offer—risk that cannot and must not be ignored. Attacks on IT resources can now be executed on a global basis, using the Internet or other communications networks, at a speed and on a scale previously unknown. News of identity theft, industrial espionage, and the ever-present insider threat is rapidly increasing. A steady stream of software patches flows as a result of poorly designed and written software, and trusting people across the world continue to be tricked by malware into divulging private information. While many of the common attack methods have largely not changed over the last ten years, their impact has been amplified as a result of a significantly increased number of potential targets, increased levels of dependence and connectivity among targets,

and heightened levels of attack automation, making the attacks easier to configure and execute on a global scale.

At the beginning of 2006, the IT landscape in many organizations is still riddled with limited or unenforced policies, incomplete and unevolving processes, unimplemented recommended practices, and ineffective or costly architectures. Further, 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 that they have a problem. While organizations generally recognize the need to improve, and indeed many have done so, there remains a struggle to effectively mature from a reactive to a more proactive stance toward information security, privacy, and compliance—especially in light of business goals to do more with less (budget, resources, and so on).

Organizations and industries (particularly those supporting Critical Infrastructure) have been placed under greater regulatory scrutiny, causing executives and IT professionals to rethink their information protection controls. Further, heightened consumer awareness, regulatory requirements requiring the public disclosure of security breaches, and a general focus on personal privacy and security protections are changing the way organizations do business. 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, marketplace tolerance for security and privacy failures is shrinking.

For these reasons, organizations must now view security as an essential quality that needs to be ever-present in their business and IT architectures and operations. Few organizations have the luxury of starting fresh with their IT landscape, where systemic security could be built in from the start. Most organizations need to adapt their existing legacy deployments, transforming them to support security and compliance more comprehensively. For some organizations, this might be as simple as a few minor adjustments to their overall IT security plan. For others, it might be more of an evolutionary process that will require a sustained commitment of time, money, resources, and organizational focus.

Regardless, to achieve such pervasive IT security, organizations must first understand how to balance risk, cost, and complexity in order to strengthen their environments without sacrificing flexibility, agility, or efficiency. This article describes how the Sun Systemic Security Program can help any organization improve the security and compliance of its IT environment through the careful use of architectural building blocks and patterns, reinforced with concrete suggestions for iterative refinement of both processes and controls.

Systemically Secure Architectures

Managing risk, cost, and complexity effectively requires achieving a careful balance across business, operational, and technical boundaries. Architectures must be sufficiently flexible to respond to ever-changing business opportunities, policies and regulatory pressures, and evolving threat profiles. They must also be sufficiently resilient, reliable, and predictable to support the most demanding, mission-critical environments. The Sun Systemic Security Program addresses the challenge of designing, implementing, and managing IT environments in which everything and everyone is securely connected to the network.

The Sun Systemic Security Program is an architectural approach for designing and implementing secure and compliant IT environments. We recognize that many organizations use products and solutions from multiple vendors in their IT environments. Rather than focusing on vendor-specific implementation details, the Sun Systemic Security Program centers on the use of architectural patterns with well-defined properties and interfaces that can be assembled and implemented in different ways, depending on the needs of an organization.

The Sun Systemic Security Program uses:

- architectural building blocks to build security into each step of the process
- time-tested security principles, applied at times in novel ways, to reap greater security rewards than would otherwise be possible, and
- iterative refinement methods to realize greater levels of integration, efficiency, and alignment with business goals.

We have learned that it is easier, less painful, and more cost effective to get security right the first time. Security must be built-in, not bolted-on. Unfortunately, security has not always been a key criterion in the design, implementation, and management of IT environments. As a result, organizations often need to adapt their current environments to repair deficiencies and support requirements for stronger security, privacy, and compliance.

This is precisely where the Sun Systemic Security Program can help.

Certainly, this transformation cannot be achieved in a single day. In fact, we recommend that organizations adopt the Sun Systemic Security Program in an iterative, incremental way. Often, a roadmap can be developed that guides an organization through the phases necessary for it to realize its goals. The result is continuous progress, made possible by stepping back and looking for ways to integrate security improvements into policies, training, processes, architectures, and services.

Sun Systemic Security, illustrated in Figure 1, is a comprehensive approach that enables organizations to architect, implement, and manage secure and compliant IT environments.

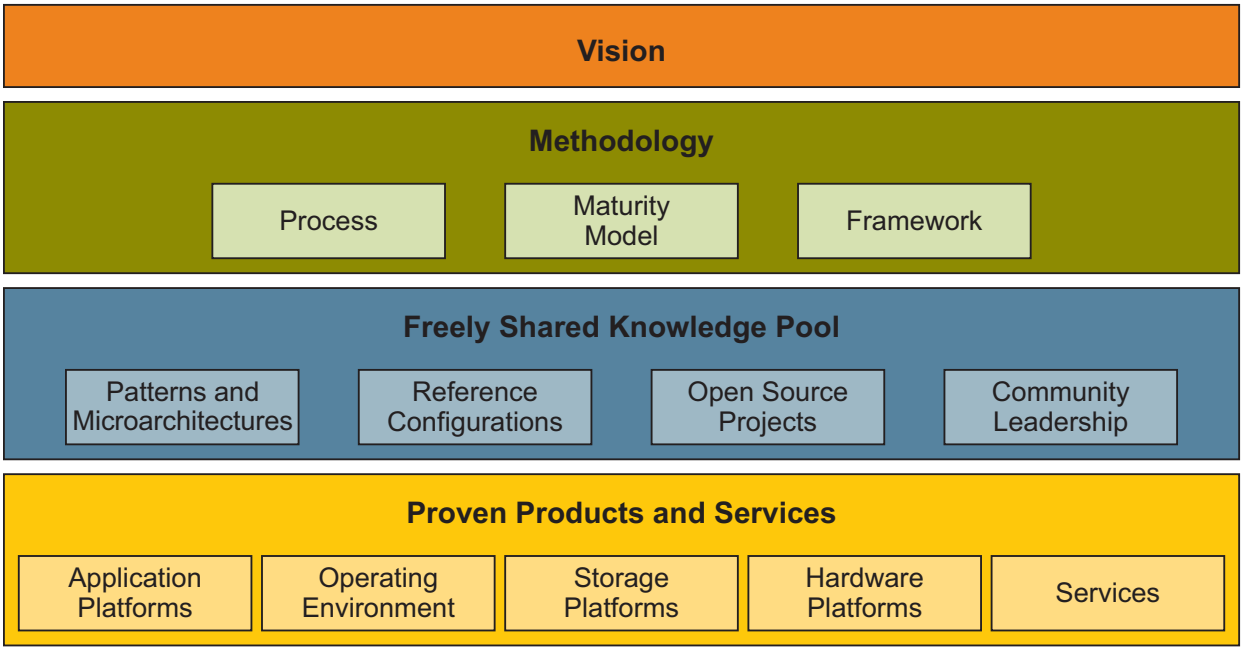

Figure 1. Sun Systemic Security Program

The approach focuses on four primary areas:

- Vision
- Methodology
- Freely Shared Knowledge Pool
- Products and Services

Vision

At the heart of Sun Systemic Security is a principle that security must not be treated as a product or service, but rather as a pervasive quality that is holistically and repeatedly applied throughout an organization's architecture and processes. This vision enables organizations to realize business value from the systemic integration of security into their existing IT architectures and practices. Organizations should not view security as some “thing,” but rather as one of the primary qualities that every architecture, service, product, or procedure must possess (in balance with other systemic qualities, such as reliability, availability, performance, cost, and so on).

Methodology

Fortifying the Vision, the Sun Systemic Security Program provides a practical method for helping organizations architect, implement, and manage IT architectures that are capable of satisfying their security and compliance requirements. For some organizations, only minor adjustments might be necessary, while for others, a more concentrated effort will be required. Similarly, some organizations know what they must do, while others are still searching for the first step. Regardless of an organization's situation, the Sun Systemic Security Program can help. The methodology comprises an architectural security Process, a security-focused Maturity Model, and a compliance-oriented Framework that

collectively are applied to develop and execute steps along an IT security and compliance improvement roadmap. This article describes how the Sun Systemic Security Maturity Model can be applied to help organizations iteratively and continuously improve the security of their IT environments.

Freely Shared Knowledge Pool
Sun Systemic Security includes a Freely Shared Knowledge Pool that collects security resources made available to organizations and customers around the world. A few examples include the Sun Security BluePrints articles, the Solaris™ Security Toolkit, and the OpenSolaris™ Operating System Security Community. In addition, this article describes a number of architectural security building blocks, including how they can be applied to help improve the security of modern IT architectures.

Products and Services
Regardless of the architecture to be developed, products and services will always be needed to make it work. Building on the accrued knowledge from protecting military, financial, and other information assets and critical infrastructure, Sun has continued to build security into all its products by design. As a result, the Sun Systemic Security Program includes Sun's diverse product, service, and training portfolio for those customers wanting to gain the benefits of using Sun technology in their environment. Recognizing that no single vendor can meet every customer's need, the Sun Systemic Security Program also includes IT and security products and services from members of the Sun Partner Advantage Program. Collectively, the products and services provide a strong foundation on which customers can rely.

While each of these areas is equally important, the focus of this article is on the use of architectural building blocks and patterns as a way of visualizing and constructing more secure IT environments. Further, this article describes the use of a maturity model as a vehicle and roadmap for IT architectural and process transformation. Additional information on the Sun Systemic Security Program can be found in “References” on page 32.

Architectural Security Principles

Each of the security building blocks and patterns described later in this article embody the following common architectural security principles.

- Self-Preservation
- Defense in Depth
- Least Privilege
- Compartmentalization
- Proportionality

Gaining a careful understanding of these principles helps clarify how each security building block and pattern embraces these security protections and how, when used in combination, those protections can be amplified and reinforced.

Self-Preservation

Self-preservation is defined as “protection of oneself from harm or destruction” and “the innate desire to stay alive” (source: http://www.dictionary.com). Applied to the topic of IT architecture, the principle of self-preservation dictates that an object must be configured, used, and managed in such a way that it protects itself from unauthorized external influence. Self-preservation, as applied to IT architecture, typically means that organizations should, at a minimum:

- Reduce the attack surface of objects, as well as the potential for undue exposure.
- Ensure, as best one can, that objects are in a known and trustworthy state, are free of vulnerabilities (for example, patched), and are configured appropriately for the environment in which they are used.
- Protect management and administrative interfaces from unauthorized access.
- Prefer the use of open and vetted protocols that implement strong authentication, confidentiality, and integrity protections.

Defense in Depth

Defense in depth mandates the use of multiple, independent, and mutually reinforcing security controls. Simply put, an IT architecture should strive to eliminate, where possible, single points of security failure. The number, placement, and type of security controls used will vary based on the threat profile of the architecture (and its published services), as well as organizational policies and preferences. Regardless of the actual controls or methods used to implement defense in depth throughout an architecture, the goal remains the same – namely, to defend an IT environment in the event that a single security control fails.

Note that defense in depth measures can be equally applied to a single architectural building block or to multiple ones. The principle of defense in depth, by its very nature, requires security to be integrated systemically throughout an IT architecture.

Least Privilege

The principle of least privilege states that “every program and every user of the system should operate using the least set of privileges necessary to complete the job.” More generally, when discussing IT architecture, this principle is applied to the export, use, and control of services and interfaces. Fundamentally, you should not offer what you do not want others to take. Only by establishing clear and unambiguous interfaces and privileges can it be possible to decide who may use them and under what conditions.

For example, the principle of least privilege can be applied in the following ways:

- Users may be given rights to access on certain systems, networks, and applications based on their organizational role.
- Applications may be started and run as unprivileged accounts with very limited access to the underlying operating system.
- Services running on an operating system may not be able to establish outbound network connections to other systems.
- Hosts residing on a given network may be restricted to communicate only with other hosts on the same network, and even then perhaps using only approved protocols.

Each of these examples shows how the principle of least privilege can be applied to different aspects of IT infrastructure and services. In each of these cases, interfaces and privileges were clearly defined, along with rules for how those interfaces and privileges could be used.

Compartmentalization

Compartmentalization is defined as the act of separating something into distinct parts, categories, or compartments (source: http://www.dictionary.com). Compartmentalization is a very useful approach for keeping separate (or isolated) unrelated interfaces, services, data sets, systems, networks, and user communities. It is reminiscent of the old adage, “a place for everything and everything in its place.” By viewing architecture in this way, it is possible to group and isolate objects in order to manage risk, including the potential for and impact of damage in the event that an object is compromised.

Just as with the principle of least privilege, compartmentalization can be applied across the typical IT environment in various ways, including:

- Isolate or group communities of services, networks, systems, and users.
- Provide a sandbox within which applications and services can be run.
- Enforce isolation between users, data, and objects operating in different security roles, zones, or risk profiles.

Proportionality

The principle of proportionality states that “information security controls must be proportionate to the risks of modification, denial of use, or the disclosure of information.” Put another way, proportionality means that the cost of protecting a given asset should not exceed its value. This is especially true when all of the various costs are considered: initial acquisition or purchase, customization and integration, ongoing support and maintenance, administration and troubleshooting, training and education, and so on. It is essential, therefore, that organizations work to achieve a balance between security and cost in how they architect, implement, and manage their environment.

Security does not need to be expensive to be effective. Many organizations have spent millions of dollars on “security solutions,” only to find that they do not offer the comprehensive protections that are required. Security can be dramatically improved, in many cases, simply by understanding and leveraging a few basic security principles, such as those discussed in this section. Integrating security systemically throughout an environment will offer organizations greater opportunities to not only manage their risk but also reduce their costs. For example, by bounding the selection and use of security controls (according to the principle of proportionality), organizations can better understand what assets are most critical and deserve the most protection. Similarly, it will also help organizations better select the baseline level of security that is required throughout their environment.

The degree to which these principles can be implemented will depend on the specifics of the product or service, the architecture in which it is being used, and any applicable business or technical requirements. Taken collectively and used carefully, the implementation of IT architectures based on these security principles will help drive higher levels of security and compliance throughout an organization.

Architectural Patterns and Building Blocks

Architecture patterns and building blocks are crucial components of the Sun Systemic Security Program. The Sun BluePrints article titled “Sun's Pattern-based Design Framework: The Service Delivery Network” (http://www.sun.com/blueprints/0905/819-4148.pdf) defines architectural patterns and building blocks and describes how they collectively fit into a pattern-based architectural design framework. Figure 2 illustrates the modular nature of building blocks and how they can be composed into patterns.

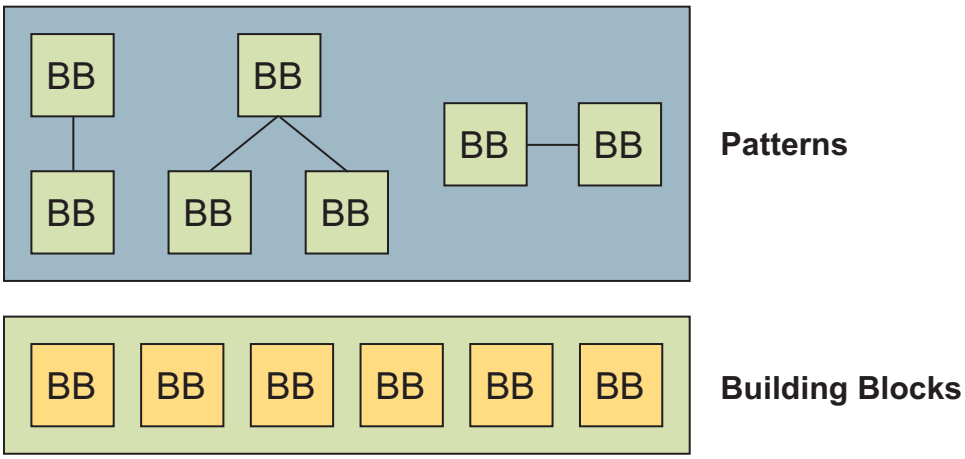

Figure 2. Building Blocks and Architecture Patterns

Building Blocks

Building blocks represent the smallest, irreducible components within a pattern or architecture. These building blocks are assembled to provide systemic functionality. As a general rule, building blocks should provide well-defined features, clean interfaces, and have limited interdependencies with other building blocks. This article describes building blocks that provide security protection or detection capabilities.

Patterns

A pattern is defined as a generalized, reusable design solution to a recurrent architecture design problem. Patterns are derived from the experience of solving the same or similar design problems over and over. Once defined, the generalized design solution, assembled using building blocks, can then be used heuristically and applied to customers with similar design challenges. In this way, patterns can be used to specify a “best fit” solution that is shaped by forces (such as design goals and constraints) that are present in each implementation project. Leveraging the lessons learned from each project, the patterns and building blocks are refined and improved (or new ones created) to better respond to future design challenges. This article describes patterns that can help organizations better manage their security risk and exposure throughout their IT environment.

Building Blocks and Patterns in the Sun Systemic Security Program

Sun Systemic Security is founded on the belief that each building block and pattern has intrinsic merit: that everything and everyone has a role to play in securing the overall IT environment. From this point of view, it becomes clear that the overall security readiness of an IT environment is derived from the synergy

of all its constituent components (building blocks, patterns, products, services, and so on) and not simply from the collection of “security” specific elements (firewalls, intrusion detection systems, virus scanners, and so on). Organizations that use secure building blocks and patterns when designing IT architectures can improve the security and compliance in the resulting solution architecture. Success is measured based on how well those building blocks and architecture patterns can be integrated within an organization's overall IT architecture. Greater rewards can be achieved for organizations that integrate multiple secure building blocks and architecture patterns in their designs, because they can be mutually reinforcing.

Security Patterns and Building Blocks

This section describes the following security patterns and building blocks and shows how they can be used individually and collectively in an IT environment.

- Secure Components
- Secure Execution Containers
- Secure Network Enclaves
- Consolidated, Shared Service Farms
  - Shared Infrastructure Services
  - Shared Application Services
- Secure Presentation Services
- Secure Desktop Services

The intent of this article is not to exhaustively discuss all possible building blocks and patterns, but rather to provide representative samples that are broadly beneficial and demonstrate the value and distinction of the Sun Systemic Security Program. An architectural focus on information security and compliance can also result in a more resilient and agile architecture, especially when a strong linkage between policy, process, and technology can be achieved.

Secure Components

All IT environments are composed of discrete IT elements that most often take the form of hardware (compute, network, and storage) platforms, firmware, operating systems, middleware, services, and applications. All too often, organizations find themselves vulnerable to attack at the level of individual components, because they have not been properly secured or maintained. Time and time again, such elements have been successfully attacked because they were not secured in accordance with industry recommended practices or even an organization's own security policies. If fact, most organization's best efforts are hampered by the variety and volume of IT elements that they must manage and protect against perpetually escalating threats.

This problem is exacerbated by the fact that, to deliver some measure of business value, many IT elements must often be integrated, layered, or combined, thereby increasing the likelihood of a vulnerability or exposure implicit in the resulting configuration. As a general rule, the more elements that need to be integrated together for something to work properly, the higher the likelihood that one or more elements (or perhaps the resulting aggregate) will contain a security vulnerability or exposure of some kind.

This article defines a Secure Component as:

> **a representation of a discrete or aggregate IT element that has been secured in accordance with industry recommended practices, and configured in accordance with an organization's security policies and goals.**

Examples of Secure Component building blocks include a Web server or an operating system. To be a Secure Component, however, the Web server or operating system must be installed and configured in accordance with the security principles described previously to reduce their overall attack surface, increase their security readiness, improve their survivability, and protect themselves from unauthorized access. These security goals must be achieved while each element still satisfies its required role and business function.

Note that an IT element need not be intrinsically secure to be leveraged as a Secure Component. Depending on the element, configuration adjustments might be needed, or additional elements added, to improve upon its default or “out of the box” security.

Figure 3 shows a few practical steps that organizations can take to transform an IT element (such as systems and network devices and services) into a Secure Component.

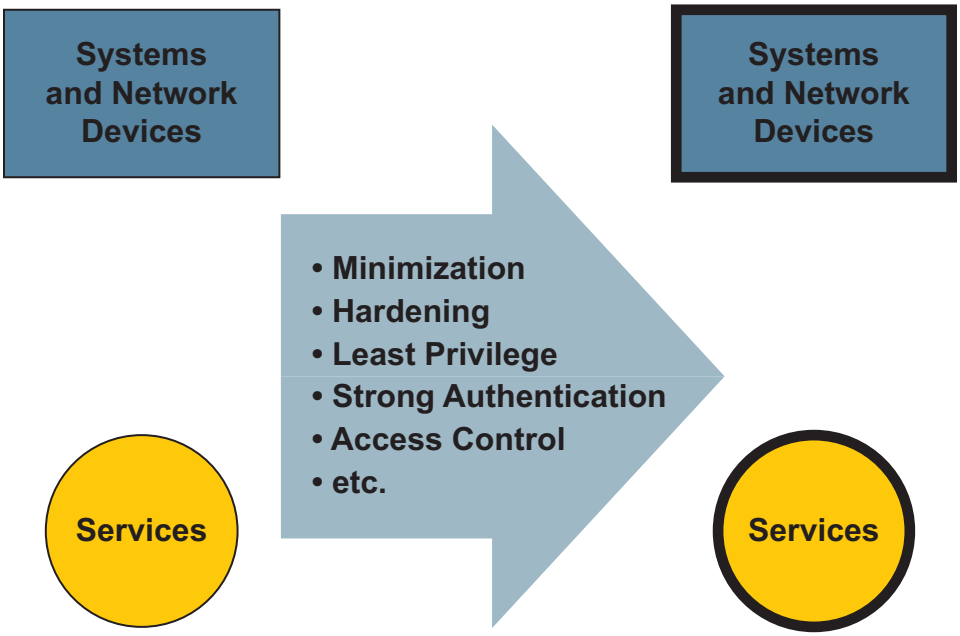

Figure 3. Making IT Elements Into Secure Components

As part of the Sun Systemic Security Knowledge Pool, Sun has collected a wealth of resources to help organizations improve the security of their IT elements. For example, since 1999, Sun has published over 60 articles and several books on a variety of security topics through Sun's BluePrints program. The program’s goal is to promote the research, development, and publishing of best practices texts and guides. Stemming from the Sun BluePrints work, the Solaris Security Toolkit offers Sun customers a freely

available and supported way to improve the security of their Solaris™ Operating System (Solaris OS) deployments.

To determine which security recommendations and steps are appropriate for any given IT element, refer to product documentation and to any supplemental publications provided by the vendor (such as Sun BluePrints), as well as content developed by industry groups (such as the Center for Internet Security) and government organizations (such as the U.S. National Institute for Standards and Technology). For example, Sun has been working with academia, industry, and government representatives for over three years on the development of consistent, supportable security configuration recommendations for the Solaris OS, which have been published through the Sun BluePrints program and also through the Center for Internet Security.

Drawing on the recommendations from sources such as these, along with your organization's own business and technical requirements, establish a standard security configuration for each IT element. Deploy each Secure Component based on the recommended configuration settings and other recommendations defined within your organization's standard security configuration. Thereafter, be sure to evaluate their deployed configurations periodically to verify whether they still comply with your standards. Remedy any detected deviations.

Regardless of the source or method used to create and monitor the actual IT elements, the practice of deploying and monitoring Secure Components is a necessary first step toward establishing a secure IT foundation.

Secure Execution Containers

Operating systems, command interpreters, and application environments provide a way for software instructions to be executed. The concept of execution containers is an architectural abstraction used to describe virtual compute resources.

This article defines a Secure Execution Container as:

> **a special class of Secure Component that provides a safe environment within which applications, jobs, or services can be run.**

Execution containers are frequently used within the context of operating systems: operating system instances (real or virtual) can themselves be run on physical, logical, or even virtual hardware platforms. Execution containers also can be environments in which applications, services, or other components are executed, such as Java 2 Enterprise Edition (J2EE) Containers.

Because Secure Execution Containers are also Secure Components, all of the attributes of Secure Components apply. As a result, a Secure Execution Container typically extends this foundation to:

- protect itself from unauthorized access or use by the services running within it
- protect any service running within the container from unauthorized external influence
- protect the IT environment outside of the container (should a running service be compromised)
- provide an audit log of events occurring within the container

These additional protections are necessary in order to minimize the damage resulting from an accident, misconfiguration, or successful malicious attack—on the container itself, or on any of the services running within it. Secure Execution Containers are charged with exposing (to their running services) only the interfaces that are specifically needed to support their successful operation and use. This is particularly crucial for more dynamic IT environments in which services are provisioned into, and executed within, Secure Execution Containers. Secure Execution Containers should also restrict, to the highest possible degree, the activities of the users and services running on the system based upon well-defined business and technical requirements.

The methods used to deploy a Secure Execution Container vary based on organizational requirements, product capabilities, and the threat profile for a given service or application. Some organizations or services might require physical separations, while others might employ virtualization at the electrical, logical, or resource level to achieve similar goals. Secure Execution Containers can be instantiated at the platform and OS layer using a variety of methods, such as:

- separate platforms to enforce physical separation
- separate dynamic system domains to provide electrical isolation
- separate Solaris™ Containers to offer logical separation
- chroot(2) Padded Cell to offer another form of logical isolation
- Solaris™ Resource Management to provide resource-level separation

Remember that patterns focus on the high-level requirements and do not get bogged down in the implementation details of a solution. A single pattern can therefore be instantiated in a variety of ways depending on customer requirements, preference, experience, training, budget, or other factors. With patterns, organizations have the freedom to develop repeatable and reusable architectures while not limiting themselves with respect to products or implementations.

Consider another example: a Solaris 10 Container into which a Web server is provisioned. Assume that the Web server is itself a Secure Component. In addition to the protections enforced by the Solaris 10 Containers' security model, the Web server can be configured to operate with significantly reduced privileges and restricted access to file system objects. Similarly, a host-based firewall could be added to restrict outbound communication from the Solaris Container. Taken collectively, these and other security controls, used proportionally and in accordance with an organization's security policy, can create a safe yet flexible environment within which the Web server can operate.

Figure 4 shows an existing service being deployed onto a system to create a single Secure Execution Container.

Figure 4. Secure Execution Containers

This convention is used to establish well-defined interfaces and rules surrounding what the individual Container can and cannot do, as well as to define which constraints may be placed on it. Although there is nothing inherent in the Secure Execution Container building block that precludes running multiple services within a container, organizations must assess and determine whether the expected rewards of running multiple services in a single container outweigh the potential risks. When deploying multiple services into a single Secure Execution Container, additional protections must be implemented that protect each service from the others running in the same container. It is critical that controls be implemented to ensure that the compromise of one service does not lead to the immediate or effective compromise of the remaining services running in the container. Further, resource controls should also be implemented that limit the exposure of services to resource exhaustion attacks.

Secure Network Enclaves

Within many IT environments, network access is relatively uncontrolled. Users who are connected to a network within an organization have an implicit authorization to attempt a connection to virtually any other system, device, or service. Compare this approach with the access model long adopted for accessing and communicating with entities outside of an organization (such as partners, suppliers, and even end users) in which barriers are consistently erected to restrict network, service, and data flows. What accounts for this difference in strategy?

The open internal architectural model persists in part from the commonly held belief that organizations do not need to protect themselves from their own employees, and that most security breaches originate from outside of the organization. This misconception is constantly challenged by experience. Organizations continue to be under growing pressure to comply with regulatory mandates that call for the protection of their internal systems, networks, and data. Further, conclusions drawn from the 2005 FBI and Computer Security Institute (“CSI”) Computer Crime and Security Survey indicate that “despite some variation from year to year, inside jobs occur about as often as outside jobs. The lesson here, though, surely is as simple as this: organizations have to anticipate attacks from all quarters.”

Secure Network Enclaves, illustrated in Figure 5, are used to help organizations better protect themselves from network-based attacks – regardless of their origin. Everyone is considered untrusted until proven otherwise. This article defines a Secure Network Enclave as:

> **a network compartment, containing one or more (physical or logical) user communities or services, that controls the flow of information through a set of well-defined interfaces exposed on behalf of its constituents.**

The exposed interfaces typically define what infrastructure or application-level services are provided or consumed by the entities within the enclave (and under what conditions). Grouping elements into secure network enclaves leverages well-defined interfaces and access policies, and is based on communities, organizational boundaries, service threat profiles, and so on.

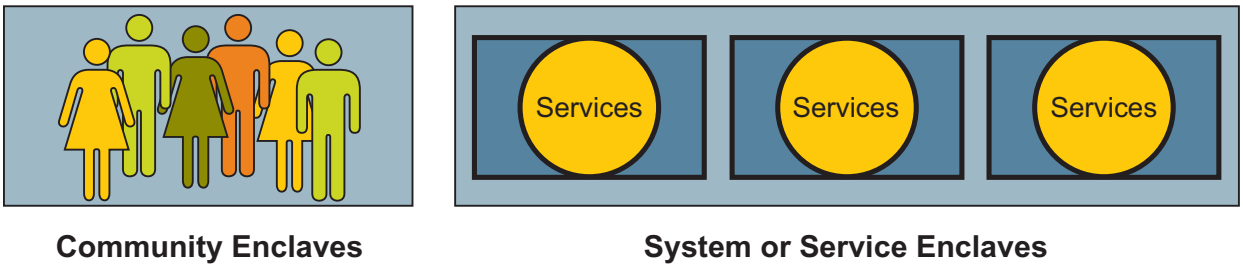

Figure 5. Secure Network Enclaves

For example, consider Figure 6, in which an enclave of Web servers exposes an HTTP service.

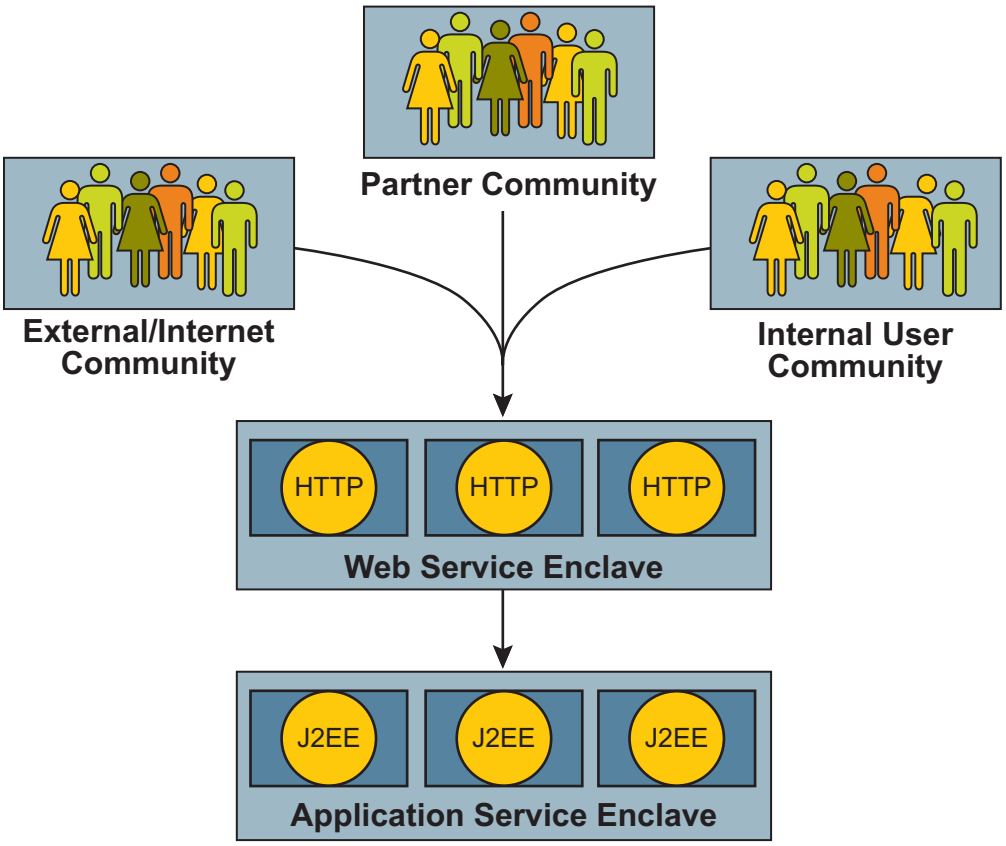

Figure 6. Secure Network Enclave for a Web Service

This Web service enclave will be the only avenue that allows HTTP requests to reach the Web servers and, even then, only when the requests originate from approved sources (such as other Secure Network Enclaves or external networks). Within the Web service enclave, communication between Web servers

could be limited or restricted altogether, subject to an organization's requirements. Enforcing isolation in this way helps contain damage should a single Web server be compromised. If the Web servers are providing only static content, they might not be permitted to initiate requests that leave the enclave boundary. On the other hand, they could just as easily be permitted to initiate requests to other enclaves that provide identity, access control, and application services.

The Secure Network Enclave building block is intended to be composable—flexible with respect to its contents, its definition, and its interaction with other network-based services. In this way, Secure Network Enclaves can be used as part of a defense in depth strategy by enforcing the principles of compartmentalization and least privilege at the network service level, containing security breaches and curtailing their spread through an enterprise.

The Secure Network Enclave approach forces organizations to look at their networks in a service-centric way. By understanding and documenting the relationships between communities of users and services, organizations can better understand and define how their services (such as those exposed through Secure Network Enclaves) are accessed and used throughout their organization. This is a necessary first step as an organization moves toward more flexible network models to realize greater levels of efficiency, reliability, and agility.

A Secure Network Enclave can contain one or more Secure Execution Containers, depending on the reliability, performance, and availability requirements imposed upon it. A fundamental design goal states that the enclave itself should not need to have its security configuration or interfaces modified to adjust its capacity, performance, or reliability characteristics. For example, if you need to add capacity to an enclave, you need only add to or upgrade the existing systems (CPUs, memory, disk, and so on) contained within the enclave, or to supplement them with entirely new systems that support your capacity requirements. To the external entities consuming the services provided by the enclave, the interface does not change – the capacity or performance simply improves. For example, consumers would still use the same IP addresses and port numbers to access the service, even though more systems may have been added to the enclave.

Just as with Secure Execution Containers, individual Secure Network Enclaves typically support only one service or user community at a time, although this convention is not a hard requirement. Organizational policies, requirements, and architectural threat profiles play a key role in shaping the actual implementation of enclaves and relationships between them.

Secure Network Enclaves have the following common characteristics:

- well-defined service interfaces (such as inbound, outbound, and management interfaces)
- virtualized point of inbound access (such as IP address and port pairs mapped to specific exposed services)
- default deny access policies
- intra-enclave segmentation (optional)

It should be noted that, so far, there has been no real discussion of traditional network security controls, such as firewalls, proxies, or intrusion detection systems. This is actually by design. Organizations are encouraged to adopt a more architectural, service-centric approach if they are to protect their IT

environments in a more systemic and effective way. That said, organizations should always carefully assess the risks and threats to their services and employ security controls where appropriate in order to manage their risk to an acceptable level.

Consolidated, Shared Service Farms

Consolidated, shared service farms are a natural progression from the concept of Secure Network Enclaves. When an enclave exposes only a single or related set of services that are consumed by other enclaves, it can be thought of as a shared service. By leveraging and connecting groups of Secure Network Enclaves, 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” (using the exposed interfaces) between relevant enclaves.

The shared services approach offers a number of relevant security benefits, particularly the use of compartmentalization, least privilege, and defense in depth inherent in their design. Shared service farms benefit from the structures of which they are composed—such as Secure Components, Secure Execution Containers, and Secure Network Enclaves. A shared services model actually simplifies the implementation of these building blocks because individual shared services generally expose an inherently limited set of inbound, outbound, and management interfaces.

Furthermore, by leveraging common components and standardized 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 (resulting in increased change and configuration control challenges). Therefore, instead of needing to scour an entire network for all instances of a given service so that a critical fix can be applied, organizations are able to focus their efforts on just a single or a small set of shared service farms that provide a given service. In this way, consolidated, shared service farms effectively help reduce the attack surface of an environment at the network level.

Reducing component-level diversity makes shared services typically easier and less costly 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. Organizations can therefore focus their remediation efforts in a more fine-grained way to more quickly and easily eliminate the problem. The shared services approach 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.

Finally, shared services can also support the adoption of more secure products. By providing a service-centric interface, individual products may be substituted with versions that offer greater security assurances or capabilities. This assumes, of course, that the new product will expose its service using the same (or a similar) set of protocols and interfaces as the one being replaced. If this is not the case, adapters might need to be implemented, or adjustments to other services running in other enclaves might be necessary. Therefore, to help in bullet-proofing an organization's shared services environment, it is important to leverage open standards wherever possible.

Shared Infrastructure Services

Traditionally, core infrastructure services have existed on networks throughout an enterprise. There is often little or no access control that restricts which systems or networks are permitted to access or use them. As a result, an exploited security vulnerability in just one of these services can leave an entire organization vulnerable. During a computer forensics analysis, an investigator could ask, “Why was the Web server permitted to directly access the database server?” This begs the larger question, “Do you know which systems, networks, and services on your network today are authorized to interact with one another at some level?” This very simple question has an answer that is often quite complex, due in part to the way in which large networks were created and have evolved over time.

This article defines a Shared Infrastructure Service as:

> **an architectural pattern representing any core network service that can be leveraged and shared (logically or physically) across an enterprise.**

For example, Shared Infrastructure Service farms could easily support individual services, such as DNS, FTP, HTTP, LDAP, SMTP, or even NTP. By grouping together similar or related services, organizations can better define and control how those services are consumed. Note that infrastructure services may be shared and synchronized across an enterprise using a variety of methods, including—but not limited to—local or geographic load balancing, content and/or state replication, master/slave configurations, or other service-specific capabilities.

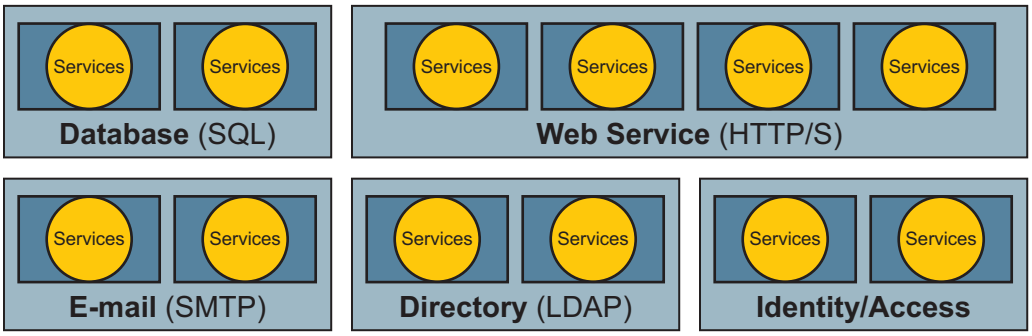

Figure 7. Shared Infrastructure Services

Certainly, some infrastructure services might be more widely used across an enterprise (such as DNS). In this case, the inbound rules controlling access to the service could be fairly loose. The outbound and management rules could be fairly tight given the nature of the DNS service. For example, it is likely that a DNS service will establish outbound connections to only a limited set of systems (for example, for DNS zone transfers).

On the other hand, certain infrastructure services are used only by other infrastructure or application-level services and are never directly accessed by end users. In such cases, it might make sense to restrict all end user access. Remember—do not offer services that you do not want others to access and use.

Shared Application Services

Over time in many organizations, disparate lines of business have funded silos of applications and services that have been developed and deployed with limited or no way of integrating them. Where integration attempts have been made, the result was often brittle because applications were tightly coupled to one another, resulting in inflexible solutions that were costly and time-consuming to build and maintain. Current software development models have moved away from this approach, favoring instead a more loosely coupled, services-driven model that improves service flexibility and allows applications to more easily evolve with business processes.

Unfortunately, whether non-integrated or tightly-coupled, most of today's applications rely on their internal identity, authentication, access control, and auditing functions, resulting in silos of security-relevant configuration and data that need to be managed. The greater the number of silos, the greater the cost. When asked, “What services are available to a given employee?”, the answer is often found only after some concerted effort to verify access within each of the applications. Similarly, if asked, “Can we easily give greater levels of access to an employee who has taken on a new role?”, the answer might essentially be “no” unless changes are made to one or more applications or services.

Siloed security functions need to be individually configured, patched, and managed. They also cause consistency problems for IT environments that need to standardize security in such areas as authentication mechanisms, authorization models, and encryption algorithms. These silos can create unwanted variation in security configurations and management methods that create inefficiencies and provide fuel for compliance auditors.

This article defines a Shared Application Service as:

> **an architectural pattern that describes a modular, composable service, component, or connector that can be linked with other application services to form higher-order IT and business applications.**

A Shared Application Service pattern can be used to implement security services used throughout an enterprise, such as:

- Identity
- Authentication
- Authorization
- Auditing

Figure 8 shows shared security services and centralized security services that could be adopted by an organization and used across all of their business services and applications.

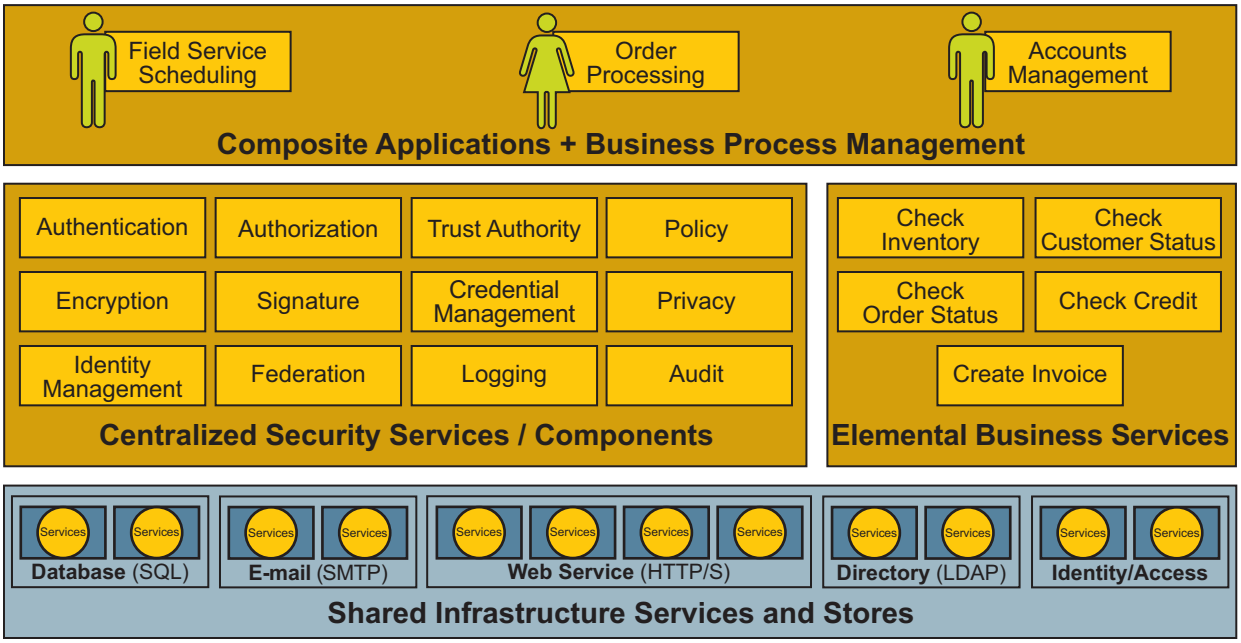

Figure 8. Shared Application Services

In this approach, application security qualities are refactored into a common set of reusable security services that are made available across the enterprise. As a result, each of these application services can share one or more security services that can be carefully examined, updated, and enhanced without requiring existing business services to be modified. Well-designed Secure Application Services can implement centralized, policy-based security service access while also supporting variation where needed. For example, a single authentication service could support passwords, tokens, certificates, and so on, and use the correct authentication scheme based on a policy defined for a given application, role, or service request.

Shared Application Services, particularly those supporting security services, provide a number of security advantages over their siloed counterparts:

- Their inherent modularity means that a version can be quickly and easily replaced (for example, product updates, new vendor implementations, and so on) as long as the exposed interfaces remain unchanged.
- Their small and modular nature allows organizations to concentrate security efforts on ensuring that the service performs as expected and complies with their policies and requirements.
- Their composable nature means that greater numbers of applications can more quickly and easily include stronger security protections without adding undue burden on the application developer. This can be accomplished by sharing with the application developers a set of secure components or proxies that can in turn invoke security services as needed. In this way, the application developers are freed to focus on core business logic, while the security developers can focus on providing high-quality security services that can be used across all of the organization's business services and applications.

Enterprise Grid Architectures

Enterprise grid architectures have recently emerged as a promising new way to manage traditional data center environments, building upon the experience and lessons learned from their high-performance and technical compute grid counterparts. From a security perspective, most of the traditional data center components and inter-relationships remain the same in an enterprise grid architecture, but there are differences in the way in which components are provisioned, personalized, and managed. Enterprise grid architectures rely on a shared 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 even entire data centers.

Sun Systemic Security aligns with—and is reinforced by—the consolidation, standardization, and automation that is required to support such environments. For example, shared management services can be used to:

- enforce consistent policies and processes across the entire enterprise grid
- reduce deployment and management time, cost, and complexity
- improve the repeatability of process and consistency of configurations across sets of deployed IT elements (whether they are Secure Components, Secure Execution Containers, Secure Network Enclaves, or the shared services that are built upon them)
- detect and respond more quickly and effectively to audit failures, security vulnerabilities, and breaches

The concepts and technology behind enterprise grid architectures are still maturing. Regardless of the actual protocols, technologies, or implementations used, the inherent value of a shared management capability has been already demonstrated in many organizations. Note that enterprise grid architectures have their own class of security risks and concerns. Industry groups (such as the Enterprise Grid Alliance) have worked to monitor, document, and raise awareness about such security issues. Still, just as with any other architecture or technology, each organization must carefully assess and manage risk in the most appropriate manner for its environment.

Secure Presentation Services

IT environments have grown increasingly more difficult to manage and secure. This is, in part, a result of the sheer number of services that organizations must provide to their employees, partners, suppliers, and customers, along with the variety of ways in which those services can be accessed, controlled, and audited. There exists a problem of scale that impacts efficiency, and cost, as well as security and compliance.

The Secure Presentation Services pattern can be used to better manage the challenge of presenting services to users in a consistent, efficient, and secure manner. This paper defines a Secure Presentation Services pattern as:

> **a trusted mediator that brokers requests and manages communication between the application services provided by an organization and the user communities that are permitted to consume those services.**

Figure 9 shows a Secure Presentation Services layer (such as portals or proxies) that handles communication between user communities and network services. Secure Presentation Services include a wide array of capabilities and potential strategies for implementation. Web portal and proxy-based solutions are example implementations of this pattern, but the context of Secure Presentation Services extends beyond Web-enabled services and applications. For example, in a truly services-oriented world, even desktop access could be viewed, offered, and controlled as a service. Consequently, the Secure Presentation Services pattern could be used in the form of desktop utilities that provide communities of users with access to applications granted strictly based on need.

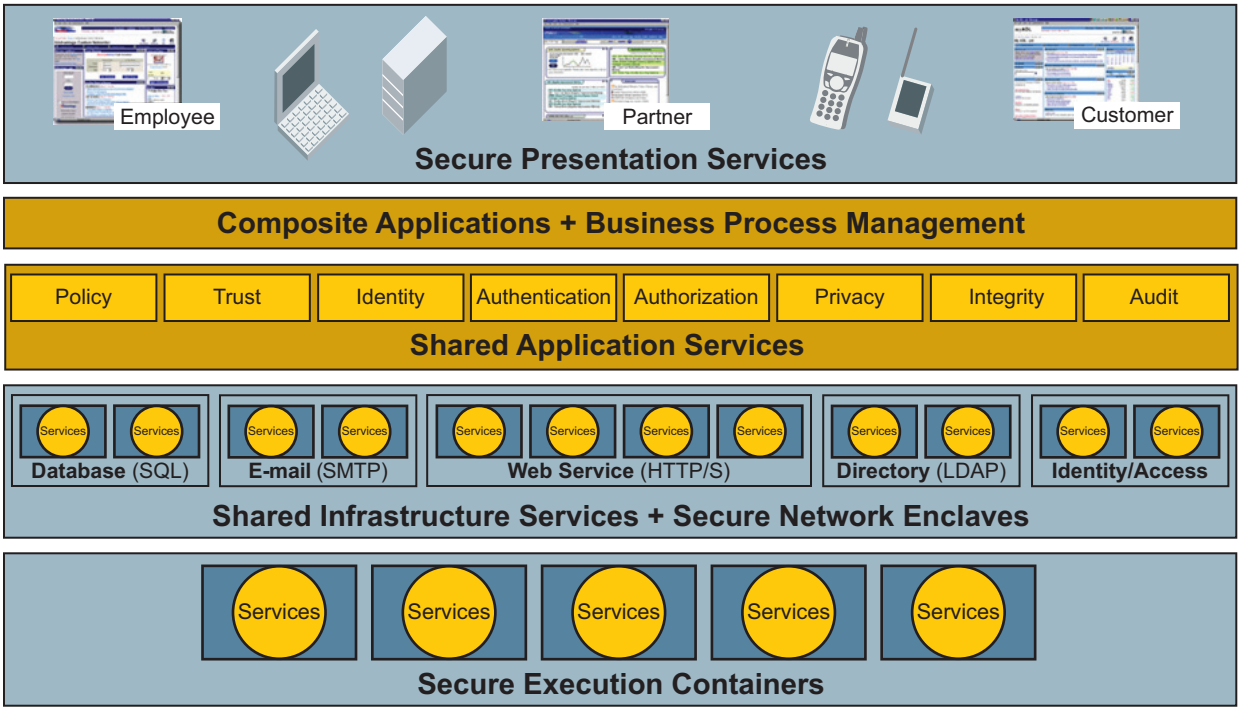

Figure 9. Secure Presentation Services

The Secure Presentation Services pattern provides the following important security characteristics that serve to reinforce the security and compliance of the overall IT architecture by controlling user access to services.

A Secure Presentation Service establishes a barrier between user communities (and their respective Secure Network Enclaves) and the enclaves that provide access to applications and services. In this way, a Secure Presentation Service acts as a trusted mediator or proxy that handles service requests on behalf of users.

A Secure Presentation Service can act as an aggregation point that allows an organization to expose its services through a single (or small set of) controlled interfaces. This is an important architectural strategy because it limits the actual number of interfaces and ports that need to be exposed to communities of users from a data center. For example, this means that infrastructure attacks originating from within those

user communities (whether internal or external) cannot reach any systems, networks, or services except those that are explicitly exposed through the Secure Presentation Service pattern.

A Secure Presentation Service can detect and leverage additional information about the actual physical device used to access a service. Access decisions can then leverage this information to better restrict sensitive operations or transactions. For example, a user might be allowed to access a sensitive service or function from a trustworthy device (such as a dedicated system in a secured area). However, the same user could be denied that access if the attempt came from some other source (such as a PDA or mobile phone).

Secure Presentation Services provide a consistent and centralized interface for users who want to access services offered by an organization. By leveraging a unified identity and access management service, organizations are also in the best position to ensure that users can access only those services to which they have been explicitly entitled. For example, managers might have necessarily different or greater levels of access than other members of their teams. Similarly, should a user no longer need access to a service, the Secure Presentation Service provides the centralized inspection point from which access can be universally revoked. Note that, in support of compliance efforts, Secure Presentation Services are also able to inspect and audit service access and user activities in accordance with an organization's policies.

Secure Desktop Services

Traditional thick-client desktops are a costly solution for providing ubiquitous access to services. They are also a source of many well-documented security risks, including software piracy, data theft and loss, and malware infection and propagation. Despite these challenges, organizations continue to invest (often quite extensively) in reinforcing the security of their thick-client deployments through the purchase of various bolt-on security and management packages.

The use of thick-client technology amplifies the security challenges facing organizations today. The sheer number of deployed systems often makes it difficult and costly to ensure that they are operating in a consistent, compliant, 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 might not be safeguarded to the level required by an organization's security policies. Finally, thick clients have an intrinsic value, making them valuable targets for theft and illicit resale of hardware and software. Once stolen, the information stored locally on the thick client can be accessed, used, or sold, with the potential for causing damages far beyond the intrinsic value of the stolen machine.

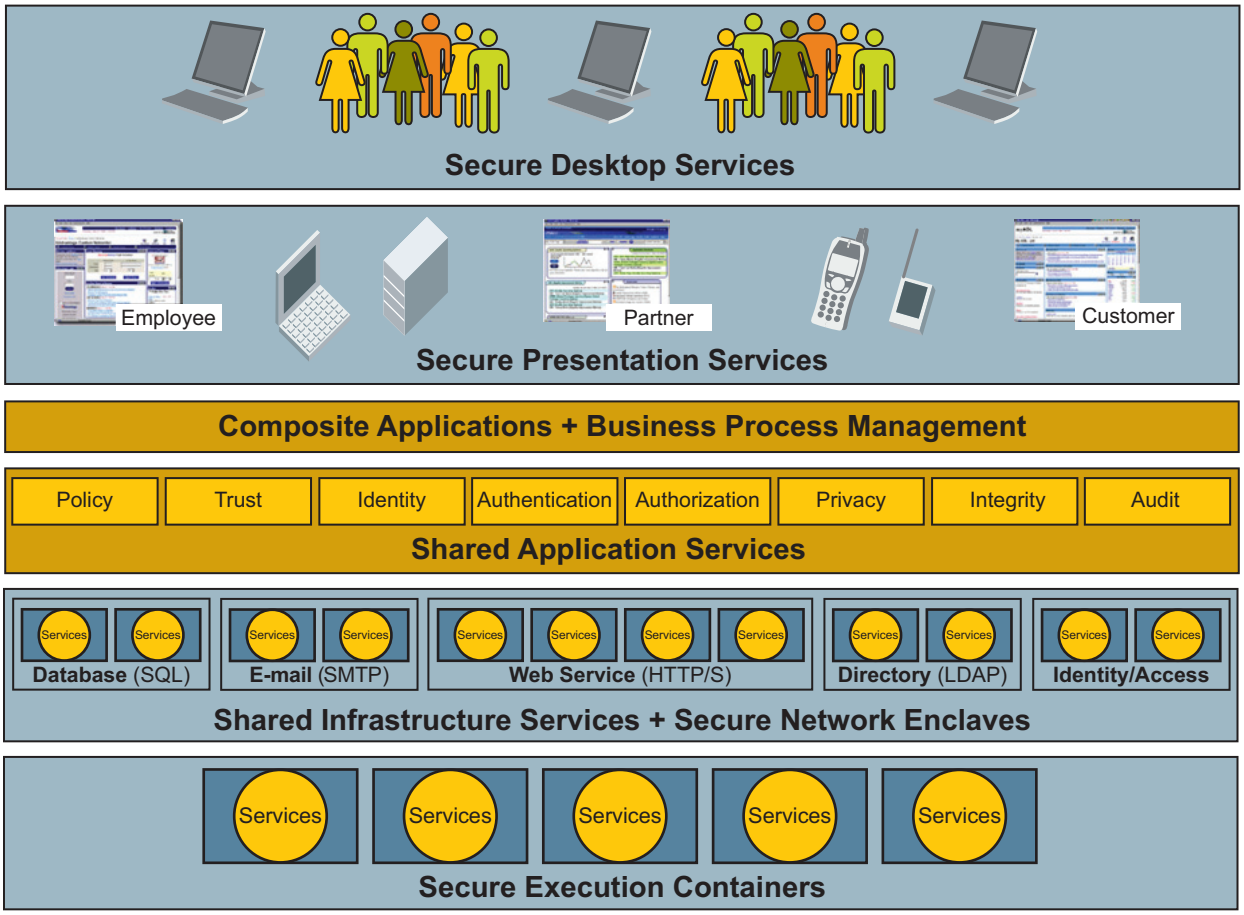

Figure 10. Secure Desktop Services

The Secure Desktop Services pattern addresses many of these security issues through the effective creation of desktop utility environments in which small, stateless networked devices replace traditional thick clients as the desktop. These devices have no local configuration, storage, or state, and they must be used in conjunction with a server environment (a desktop utility). With secure desktops, there is no longer a need to deploy security controls on each and every desktop, and there are fewer audit logs to collect and analyze, because configuration and policy enforcement are centrally managed. Consequently, the use of thin-client devices can help reduce the overall administrative burden and budget, allowing people and resources to be directed toward more strategic and proactive initiatives.

Thin-client devices used in support of a Secure Desktop Services strategy have a much lower intrinsic value and are therefore a less interesting target for thieves for the following reasons:

- They cannot be used in a standalone capacity outside of an existing thin-client environment.
- They do not retain any state, configuration, or data that can be copied, ransomed, or deleted if a unit is stolen.
- Finally, because thin-clients are effectively networked display terminals, they can leverage older, slower processors, smaller memory packages, and even operate without hard drives – all without adversely

impacting the end-user experience. This means that even the parts that make up the thin client would not command high resale value should a thin-client be stripped and sold for parts.

Secure Desktop Services help simplify security management by providing a single control point for accessing, delegating, and auditing access to services and data, whether through Secure Presentation Services or via more traditional means. The Secure Desktop Services architectural model enables organizations to more rapidly patch or update configurations and software in response to security alerts. For example, a security patch applied to one system could correct a security flaw that impacts hundreds of users, rather than distributing the fix to hundreds of desktops individually. Even if the fix were distributed across many machines, patch installation is not even guaranteed to succeed (for example, suppose the desktop was being rebooted or was shut down at the time the patch installation was attempted).

Secure Desktop Services can enable the creation of “Switch Gear Offices” (see Figure 11). In a nutshell, “Switch Gear Offices” represent an IT architectural model in which only network elements (for example, switches and routers) and thin-clients exist in office environments (including remote field offices). All of the compute and storage resources, including all of the software applications and services used on the desktop, are provided from within a data center environment over encrypted, highly available network channels to the thin-clients in the office. Using this model, organizations can effectively eliminate the need for servers to reside in non-data center environments. A “Switch Gear Office” is an example of a desktop utility service that can be offered to offices across the enterprise, allowing organizations to retain tighter control over how their servers and information are being accessed and protected, and how those protections are being validated.

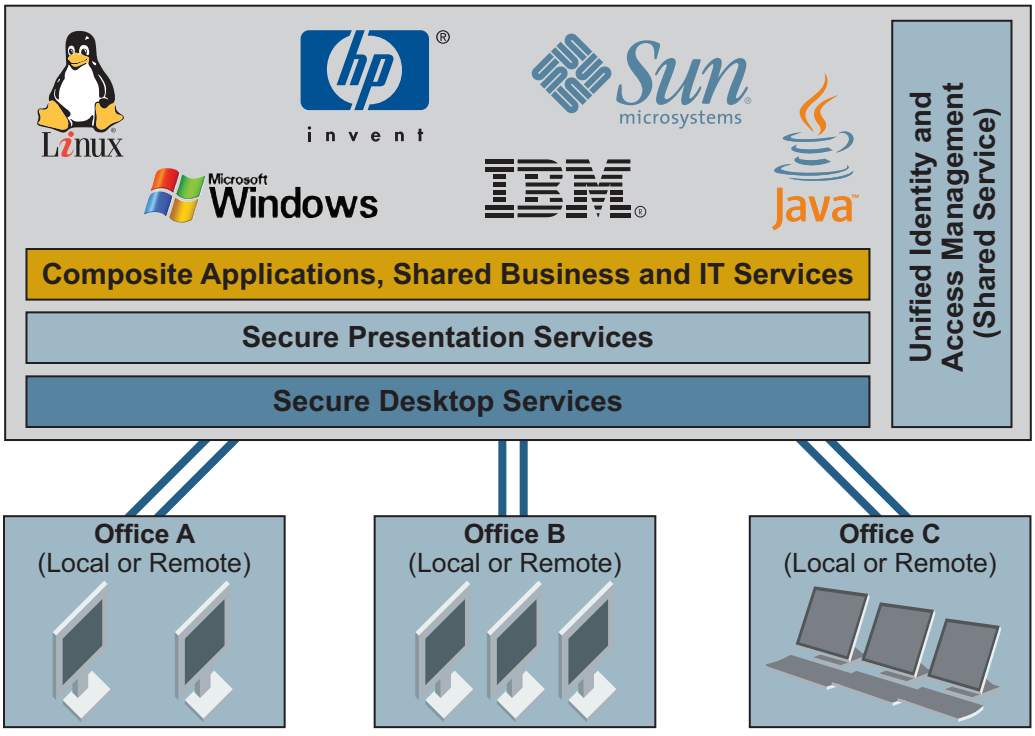

Figure 11. Switch Gear Offices

When core business and IT services are centrally offered from within their data centers, organizations are free to eliminate server and storage farms in their remote offices and locations. Using the Secure Desktop Services pattern, in combination with the Secure Presentation Services pattern, organizations can provide reliable, high-speed, and secure access to core business and IT services without a local server or storage footprint. Services and data can be centrally protected, audited, and archived within their data center environment. In this way, using the Secure Desktop Services pattern can significantly enhance an organization's efforts to improve security, manage risk, and maintain compliance.

Transformation Phases

This article has described some typical architectural patterns and building blocks that can be used to construct more secure and compliant architectures. To fully gain their potential benefits, organizations must marry these architectural patterns and building blocks, along with their existing policies and processes, to a culture of continuous improvement made accessible to them using a process of iterative refinement.

For Sun Systemic Security to be effective, it must be able to be applied to any organizations and virtually any situation. If Sun's approach required that an organization completely redesign their systems, networks, and services, it simply would not be adopted because the organizational cost and burden would be too great. Using an iterative refinement process, however, organizations can iteratively improve the security and compliance state of their environment at their own pace. Iterative refinement seeks first to create a close approximation of a solution and, once achieved, gradually improve upon that solution over successive iterations. This concept can be applied in both a horizontal and vertical manner. That is, iterative refinement could be applied “horizontally” to a single project or department. As progress is made from that experience, the lessons learned are used to gradually improve other projects or departments. Applied “vertically,” iterative refinement could be used to develop an initial baseline security configuration for a service that, over time, is improved as an organization becomes more comfortable with stronger security settings, and when the ramifications of the additional security changes are better understood.

Through the use of such improvement methods, organizations can better manage the complexity and cost of changes while constructing more agile, flexible, and compliant architectures that are capable of meeting their business goals. By progressing through the transformational phases described later in this section, organizations may find that they are better able to:

- react more quickly and effectively to security events and emergencies
- more easily detect non-compliant configurations and services
- reduce the number of exposed defects associated with their security configurations
- improve the focus and effectiveness of existing IT and security teams through a continuously improving and proactive approach to security and IT management

Few organizations have the luxury of starting fresh with their IT landscape, where systemic security could be built in from the start. Most organizations need to adapt their existing, legacy deployments and transform them to support security and compliance more systemically. For some organizations, this might be as simple as a few minor adjustments to their overall IT security plan. For others, it might be more of an

evolutionary process that will require a sustained commitment of time, money, resources, and organizational focus.

Building upon Sun's extensive experience in IT Service Management and its leadership in Operations Management, we have identified a series of 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. Figure 12 shows these transformational phases (or processes):

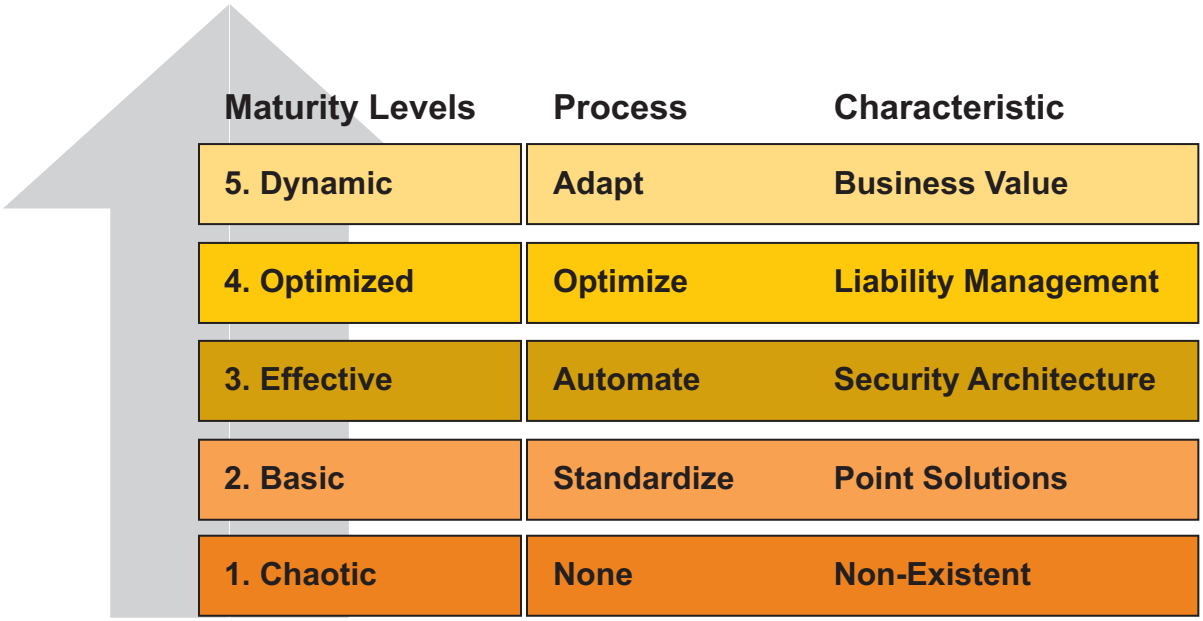

Figure 12. Maturity Model

Organizations cannot rush through each of these levels—they must mature gradually through them. While not every organization will aspire to the highest levels of optimization or adaptation, there is always room for improvement. Different aspects of IT security can and often do exist in different phases at the same time. For example, an organization's system and network security organizations, which have a long and robust tradition, may be further along than those supporting newer technologies, such as Web services. Organizations should also make informed risk management decisions that limit how far they want to progress in a given area. For some organizations, the trade-off between cost, risk, and value might mean that they choose to focus their goals at one of the lower levels. This is both normal and acceptable.

Sun Systemic Security is much more than simply meeting some number of items on a checklist. It defines and documents a customizable and flexible model that adapts to organizational policies and needs, with the goal of transforming an existing IT architecture into a more secure, compliant, agile, and optimized environment that is capable of delivering sustained compliance and business value.

Phase 1: Standardize

Although secure components and other building blocks may have been used initially, rigorous maintenance is not easy, particularly for large IT environments. All too often, organizations suffer from the lack of structured IT governance and configuration and change control, which results in endless variation throughout an IT environment. Over time, existing practices often become outdated or less scalable, configurations become less secure and consistent, and administration becomes more resource intensive and error-prone as the result of unchecked variation across the 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 or problematic, creating opportunities for security vulnerabilities and exposures. By standardizing to reduce IT diversity to an acceptable level, organizations are better able to more effectively improve security and maintain compliance.

Standardization is defined as “to reduce to or compare with a standard” and “to bring into conformity with a standard” (source: http://www.dictionary.com). This first transformation phase consists of consolidating and standardizing existing IT architectures, services, protocols, and configurations. Leveraging standardized configurations, for example, provides greater assurance about the interfaces and capabilities that individual IT elements will possess. Without a reduction in variation, organizations are not able to respond as quickly or thoroughly as they otherwise could. For example, if each operating system instance 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 might be at risk due to a recently announced vulnerability, or to restore services in the event of a disaster. This is clearly an area in which security intersects with disaster preparedness and business continuity.

The development of internal standards and reference configurations applies to all IT components: hardware platforms, operating system software, middleware, applications, and so on. At the application level, the selection of appropriate security standards is complicated by the sheer number of applications that are available or are in development (such as authentication mechanisms, policy languages, encryption algorithms and strengths, Web services standards, and so on). It is therefore critical that organizations undertake a careful evaluation of existing standards to identify both the security standards and reference configurations that are best for the enterprise. This does not mean applying a single standard across an entire organization. Instead, there should exist a small and manageable number of configurations that collectively support an organization's security goals. While not every configuration may be able to take advantage of such standards (perhaps due to conflicting requirements), exceptions should be just that: exceptions and not the rule. Further, exceptions must be managed, and deviation from standards must be justified and kept to a minimum. This is necessary to prevent a tendency toward runaway chaos in which continuous deviation creates inertia, which accelerates and branches out over time.

Standardization also involves developing and documenting security policies, standards, and processes where they are lacking, and educating people on their existence and effective use. Written checklists and recommendations help ensure that all IT elements are configured in accordance with company standards and/or vendor- and industry-accepted recommendations. Recommended interfaces, services, protocols,

and 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.

These standards need to exist and be used for any products and services that are deployed in the IT environment. Further, the definition and use of such standards needs to be enforced through a strong IT governance process. Standards must be kept up to date as installations and products evolve over time. For example, a security checklist for an older version of an operating system might be incomplete and fail to address key security issues introduced in a newer version of the product. Documented standards provide a baseline for verifying whether systems are configured in accordance with organizational policy. Documented standards are also the prerequisite for the subsequent phase of automation. Bypassing standardization to more rapidly adopt automated processes will only allow an organization to spread chaos through its environment more quickly.

Recognizing that, ultimately, it is the people—the lifeblood of an organization—who will determine the success of these efforts, it is critical to ensure awareness of, understanding of, and compliance with these standards. Organizational, cultural, personal, and training barriers must be assessed and addressed in order to expedite acceptance of these new standards and to garner the support of evangelists and stakeholders alike.

Phase 2: Automate

Once organizations have reduced diversity in their environments and established well-defined capabilities, configurations, and interfaces for their IT components, higher levels of automation become more realistic. Automation is defined as “the automatic operation or control of equipment, a process, or a system” and “the condition of being automatically controlled or operated” (source: http://www.dictionary.com).

Automation allows organizations to manage IT environments that might be fundamentally more complex than they could otherwise manage. This is because automated processes can often provide an insulating buffer between administrators and the inherent complexity of the environment that they manage.

Automation has traditionally been applied to IT component provisioning and to a subset of typical operations activities (such as data backup, user provisioning, and so on). Automation can be used to enforce consistent installation, upgrade, configuration, and patching of IT elements at the infrastructure and application levels. Further, automation can even be applied to the software development lifecycle: security components could automatically be installed into applications to help secure them in a consistent and standard manner. In addition to simply making changes to IT elements, automation can also use the policies and standards that have been developed to assess deployed components, detect violations or deviations, and potentially correct non-compliant configurations.

Automation can often be used throughout the IT lifecycle to apply, undo, or assess security changes throughout an IT environment in accordance with well-defined profiles or policies. By viewing IT security through this perspective, new opportunities emerge that allow organizations to use automation in increasingly more sophisticated and comprehensive ways (starting small and using a building block approach) in order to continuously improve security and redirect time, money, and other resources in a more productive and ideally strategic manner.

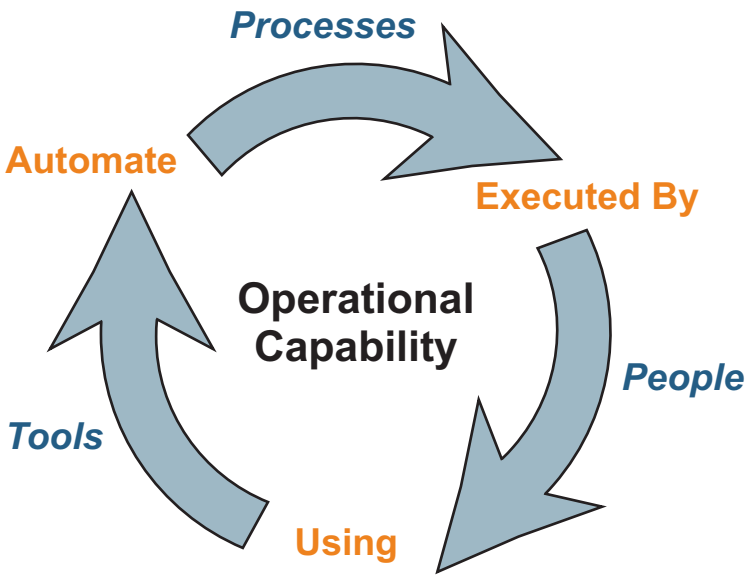

Figure 13. Operational Capability

Automation is also an excellent opportunity to capture knowledge and refine processes. 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 and improving upon this knowledge using automated processes, organizations can begin to move from a culture of heroes to one that delivers a more consistent, repeatable, and measurable IT experience.

Phase 3: Optimize

Optimization is defined as “the procedure or procedures used to make a system or design as effective or functional as possible” (source: http://www.dictionary.com). Organizations looking to optimize security in their IT environments must have already passed through the standardization and automation phases. Optimizing organizations leverage iterative refinement to make additional improvements.

This can take the form of expanding the scope of effort to include new divisions, projects, data centers, or services. As noted previously, processes like standardization and automation are not trivially accomplished, especially at an enterprise scale. As a result, organizations will typically look for early adoption projects for the purpose of verifying and refining the concepts and process before attempting a larger scale effort. Further, organizations are not static: whatever worked yesterday may not necessarily be effective tomorrow. It is therefore critical that organizations maintain their standardized policies and configurations and automated processes, and also look for new ways to improve upon them.

Optimization is about taking stock in the organization's efforts, learning from its mistakes, and building upon its successes to realize sustained effectiveness and continuous improvement. Failures, root cause analysis, and lessons learned are used as a key part of this phase to improve existing work and to prevent the same failures from recurring. Recognizing the strong ties between security and IT service management, organizations will begin to strengthen the linkages between security and areas such as change control, configuration management, and so on.

Organizations that reach the optimization phase are actively learning from their mistakes and perhaps the mistakes of their partners, suppliers, and even competitors. They will spend less time fighting fires and more time adding value to the business. Similarly, such organizations are typically able to quickly and easily demonstrate their level of compliance with internal policies and external regulations. Liability management will be in a state of equilibrium, allowing organizations the time to refine existing practices and to focus their efforts on increasing business value in addition to strategically aligning with business goals and future directions.

Phase 4: Adapt

In an effort to reduce costs and improve responsiveness and efficiency, some organizations will move toward more adaptive IT environments. Adaptation is defined as “modification of an organism or its parts that makes it more fit for existence under the conditions of its environment” and “something, such as a device or mechanism, that is changed or changes so as to become suitable to a new or special application or situation” (source: http://www.dictionary.com).

Adaptive IT environments are those that are more readily able to evolve and respond to higher level technical or business instructions or changes. Enterprise Grid Architectures, discussed previously, represent one type of adaptive enterprise in which data center assets are managed and controlled in a unified and systematic way.

Some variants of adaptive architectures are also able to adjust their own configurations, for example, in response to technical or business rules or even external conditions. Consider the case of a new service provisioned into a Secure Execution Container running within a Secure Network Enclave. In this example, it might be necessary, depending on the actual implementation, for platform, operating system, and network security changes to be made to each of these components (and perhaps others) in order to make this new service available and ready for use. The standardized configurations, well-defined interfaces, and automated processes (developed in the earlier maturity levels) make this kind of task much easier to complete than would have otherwise been possible.

Fundamentally, adaptive computing is a natural progression for environments that have reached higher levels of optimization. Adaptive environments allow organizations to be more responsive to changes in technical conditions (such as component failures, resource exhaustion, security violation, and so on), as well as those impacting the business (such as new service introductions, compliance verification, and so on). This level of adaptability is contingent, however, upon well-defined and implemented standards and consistently implemented processes.

Clearly, adaptive computing is not for everyone. Organizations that cannot develop and enforce standards will find it difficult moving to adaptive architectures. Similarly, those with poor change and configuration management skills could end up doing more harm to their IT environment than good, especially if configuration changes are made outside of the adaptive framework.

For this reason, organizations should always carefully assess which level of the maturity scale is most appropriate for them, keeping in mind that different parts of the organization may be at different levels of maturity at the same time. This can be for many reasons, including attrition, new acquisitions and mergers,

or restructuring, as well as awareness, training, and resource availability. For example, if a team does not have security training, resources, or support, it will likely be difficult for them to realize any of the higher levels of security maturity. Further, a development organization may not require the high levels of automation and optimization that would otherwise be required for services that are put into a production environment. Finally, new technologies are even a factor. As noted above, an organization that might be strong in systems and network security practices might be relatively weak in the newer area of Web services security, including how it integrates into their existing IT landscape.

Regardless of where an organization is at a given point in time, it is nonetheless critical for organizations to have these kinds of discussions in order to better understand where they are today and where they want to be. This analysis can lead to constructive discussions about the future of their architecture and services, as well as a roadmap leading them toward their goals.

Conclusion

Sun Systemic Security is a comprehensive, architecturally focused approach for securing new and existing IT environments. Key to this approach is the use of architectural patterns, building blocks, and continuous improvement methods. This paper described the following architectural patterns and building blocks:

- Secure Components
- Secure Execution Containers
- Secure Network Enclaves,
- Shared Infrastructure Services
- Shared Application Services
- Secure Presentation Services
- Secure Desktop Services

Individually, these patterns and building blocks support important IT security characteristics. Assembled together, however, they serve to form the foundation of a flexible, resilient, and compliant services-oriented architecture. Architecture alone is not enough, however. Organizations must enable security and compliance through a culture of continuous improvement involving the following key transformational phases:

- Phase 1: Standardize
- Phase 2: Automate
- Phase 3: Optimize
- Phase 4: Adapt

The Sun Systemic Security Program uses these transformational phases to enable organizations to continuously improve at their own pace, realizing value each step of the way. It leverages these transformational phases, along with its compliance-oriented methodology, knowledge pool, and comprehensive set of Sun and partner products and services. Collectively, they can be applied to help organizations reach their security and compliance goals while still supporting (and accelerating) their business and IT objectives—all without breaking the bank.

References

For more information on the Sun Systemic Security Program, including related products, offerings, and resources, visit http://www.sun.com/security. If you are interested in learning how to apply Sun Systemic Security to your environment, consider hosting a Sun Systemic Security workshop. For more information, contact your local Sun sales representative.

About the Author

Glenn Brunette is a Sun Distinguished Engineer with nearly 15 years' experience in information security. Glenn currently works in Sun's Client Solutions CTO as the Director and Chief Architect of the CSO Security Office. In this role, Glenn is responsible for global security strategy and architecture, security-focused collaboration and knowledge sharing, as well as improving the quality and security of products and services delivered to Sun's customers.

Glenn is the driving force behind Sun's Systemic Security approach and is also an OpenSolaris Operating System Security Community, the co-founder of the Solaris Security Toolkit software, and a frequent author, contributor, and speaker at both Sun and industry events. Externally, Glenn has served as the Vice-Chair of the Enterprise Grid Alliance Grid Security Working Group and Working Group Champion for the National Cyber Security Partnership's Technical Standards and Common Criteria Task Force. Finally, Glenn is an active contributor to the Center for Internet Security's Unix Benchmark team. Glenn is a Certified Information Systems Security Professional (CISSP) and has been trained in the National Security Agency's INFOSEC Assessment Methodology (IAM).

Acknowledgements

The author would like to thank the following people for their inspiration, technical feedback, and overall support in the development of this article: Jim Baty, Jason Carolan, Bruce Gossard, Eric Ivory-Chambers, David Jones, Barbara Kay, Mikael Lofstrand, Eve Maler, Danny Smith, Hal Stern, and David Walker. In addition, the author would especially like to thank Rafat Alvi, Christoph Schuba, and Joel Weise for their significant architectural contributions to the overall Sun Systemic Security Program and to this article.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject at `http://docs.sun.com/`.

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at:
`http://www.sun.com/blueprints/online.html`