Reducing Application Cost and Risk through Centralized SOA Security

by Mamoon Yunus, CEO of Crosscheck Networks

Abstract: This article compares centralized and decentralized application security models. It focuses on technical costs and organizational considerations while comparing these models. The analysis shows that centralized management of security policies has significant advantages over decentralized application security deployments including cost reduction, better risk mitigation and greater freedom for application developers to focus on creating business value.

Introduction

Now, more than ever before, the global business environment expects greater customer service, demands deeper value chain integration and drives fiercer competition while requiring corporations to perform efficiently with diminishing resources. IT departments are in the midst of this global storm and are now pushed to deliver applications rapidly while minimizing costs. Fortunately, with the maturity of agile development, SOA and related standards, and cloud computing, the foundations are available for building resilient, nimble and cost effective IT infrastructure that is responsive to business needs.

Modern applications that meet current business needs consume information from multiple sources, internal and external. Composite application, Rich Internet Application, service APIs, virtualization, and cloud services provide extensive integration of data for real-time information access. This drive to open up business applications for integration comes at a cost: application security. As companies move towards opening up systems for greater information access they expose systems to broader security risks, including sensitive data leak, unauthorized information access and an increasing vulnerability attack surface area. In this article, we will contrast centralized and decentralized security models and explore how corporations are using centralized application security for cost-effective, consistent, and manageable security.

Overview of Security Models

Application security is deployed within corporations in centralized (hub-spoke), decentralized (point-to-point) or hybrid models. The early stages of an IT build-out heavily focus on developing functionality and processes that meet business needs. Little focus is directed at putting in place a shared framework or infrastructure. The first few application development projects typically face immense time and resource pressures. Only minimum application security, such as enabling HTTP Basic Authentication for identity and SSL for protocol encryption, are coded which typically occurs at the very end of the project.

Figure 1 shows a decentralized, point-to-point integration model where applications 1-to-m consume services from service producers 1-to-n. The service producers can be internal services, external partner services or cloud-based SaaS services. As the number of applications and services that they consume increase, the number of paths that require application security policies grow by the product of application “m” and the number of services consumed “n”.

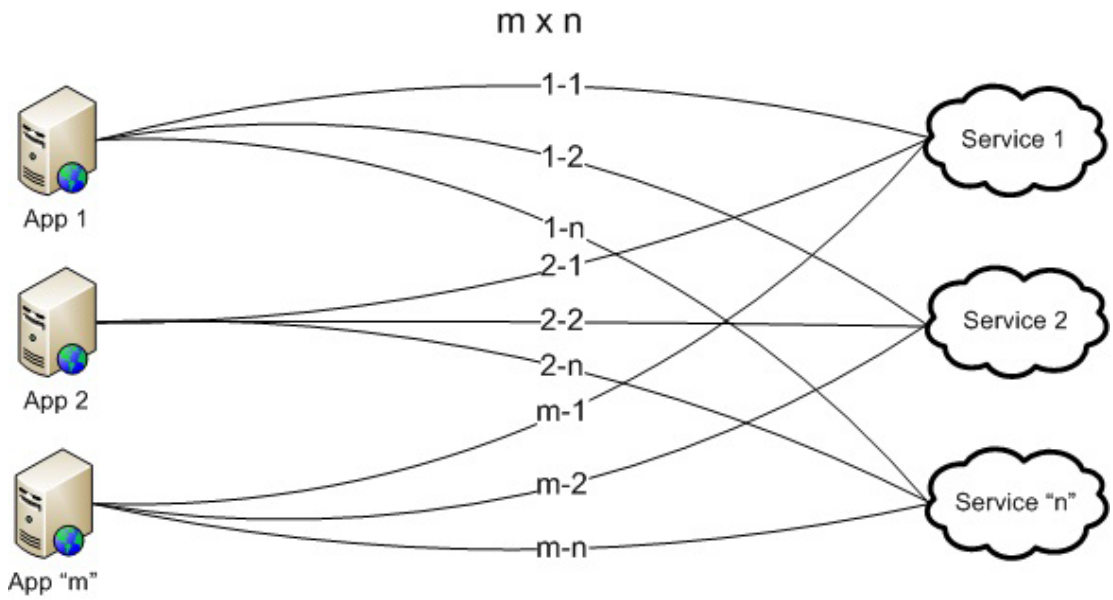

*Figure 1 – m * n Paths Require Security Policies in a Decentralized Model.*

Over time, as multiple application development projects pass through their lifecycle, corporations recognize and implement shared re-usable services. Figure 2 shows a centralized model that utilizes a gateway for application security policies. Regardless of the number of security policies required for integration, the centralized model serves to reduce the number of integration paths required to m + n. The security policy problem is reduced from a square to a sum of integration nodes problem simply by deploying a gateway for centralizing security policies.

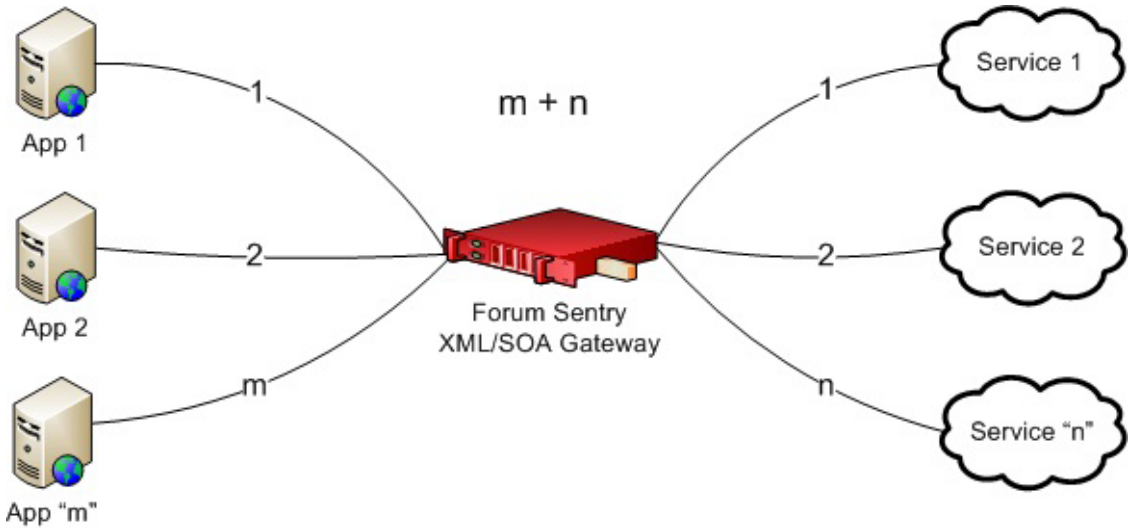

Figure 2 – m + n Paths Require Security Policies in a Centralized Policy Model.

Application security services are ideal candidates for reuse in a centralized model. Instead of coding m * n application security policies, a centralized model can remove the coding burden of these policies to a centralized gateway and make these policies reusable. Typical security services include identity, privacy and integrity, auditing, and information protection functions that are necessary for safely running applications.

To understand the impact of application security policies on centralized and decentralized models, let us consider that only identity policies with “Services 1 - n” in Figure 1 and Figure 2 require unique identity credential formats, such as:

- Service 1: HTTP Basic Authentication
- Service 2: SAML Tokens
- Service n: Adhoc token type

In the decentralized scenario, as shown in Figure 1, each application has to code “n” different token type interactions as expected by Services 1 - n. As more applications are required to interact with Services 1 - n, the different token types listed above are repeatedly coded in each new application. In contrast, with a centralized gateway, the interaction between the gateway and the producer Services 1 - n are configured once. Beyond this configuration, the applications can be provided a uniform identity token requirement like with the use of HTTP Basic Authentication tokens, for example. This removes the burden of coding different token types for each application and provides the gateway as a single interaction point for applications.

Extending beyond just identity, once threat mitigation, privacy, integrity and auditing policies are mandated, the advantage of a centralized vs. a decentralized model becomes even more evident. In addition to the advantages described so far, a centralized model that uses gateways typically is a code-free configuration environment for setting application security policies compared to a decentralized, point-to-point setup that requires coding security policies.

Although Figure 2 shows a physical device for enabling centralized application security, a logical model for centralized application security can also be deployed. Should an enterprise choose a logical deployment model with a combination of software and hardware components, they have to take a few crucial steps including adhering to standards closely, building a framework for governance that can scale while adding new systems and trading partners, and staying up to date with emerging threats. In practice, building a logical variant is a challenging proposition for large B2B deployments. The remainder of this article concentrates on physical centralization.

Technology Costs

In this section, we will model the costs associated with application security policies, categorized by policies used most frequently, to ones that are used in more complex and sophisticated deployments. As shown in Table 1, policies are sorted in three categories: Beginner, Intermediate and Advanced. For further quantitative comparison, a scale of 1-10 is used for the difficulty level of coding a policy, with level 1 being the easiest and level 10 being the hardest. The same scale is used for configuration. The Coding/Configuration ratio shows relative difficulty between coding and configuration for each of the policies. This ratio can also be used to compare the units of work required for each policy. For example, coding WS-SAML tokens may take 10 days whereas configuring this policy may require 2 days.

Security Policy	Coding: Config Ratio	Ratio Explanation
Beginner Level	**8:3**	
Simple Authentication	2:1	Easy to code vs. Trivial to configure
Authorization	2:1	Easy to code vs. Trivial to configure
SSL	4:2	Moderate to code vs. Easy to configure
Intermediate Level	**30:9**	
Threat - DoS/Malware	8:2	Hard to code vs. Easy to configure
Threat - Data Leak	8:2	Hard to code vs. Easy to configure
Data Transformation	6:2	Slightly Hard to code vs. Easy to Configure
Advanced Level	**72:17**	
PKI Management	8:2	Hard to code vs. Easy to configure
Content Security	9:2	Very hard to code vs. Easy to configure
WS-SAML Tokens	10:2	Extremely Hard to code vs. Easy to configure
Enrichment	5:1	Moderate to code vs. Trivial to configure
Reliability Management	10:1	Extremely Hard to code vs. Trivial to configure

Table 1 – Coding vs. Configuration Ratios for Increasing Level of Application Security Difficulty.

As deployments move from Beginner to Advanced configurations, the number of days required for the policies also increases. For the advanced level, that includes the previous two levels, the total number of units required for coding is 72 days, whereas the units required for configuration is 17 days.

The difficulty levels shown in Table 1 are estimates and provide a framework for organization when embarking on application integration efforts. The categorization into Beginner, Intermediate, and Advance sections is based on the author's personal experience in enabling application security across the variety of industries. Each organization has its own rules, compliance and technology skill sets that will result in different ratios for each policy. Fundamentally, corporations have to consider the following criteria in planning for their SOA application security:

- How many services "n" do they expect to consume?
- How many applications "m" will consume the services?
- Should corporations follow a centralized application security model or a decentralized one? This results in (m + n) vs. (m * n) integration paths.
- What levels of security policies – Beginner, Intermediate or Advanced – are mandated by the corporation and industry regulations?

- Should they build the application security policies or should they buy products that enable them to configure the policies without writing code?

Each decision impacts the effort in deploying, enforcing and managing application security. In Figure 3, we evaluate the Work Units for the Number of Systems deployed for centralized and decentralized environments. The Number of Systems simply represents the nodes “m” and “n” within the environment. For decentralized application security, the number of paths, as shown in Figure 1, is (m * n) and for centralized architecture, the number of paths is (m + n). For simplification, we consider m = n. Therefore, as the Number of Systems increases, the decentralized model increases by a factor of (Number of Systems)2 and the centralized model increases by a factor of 2(Number of Systems). Additionally, we layer coding vs. configuration ratios of 8:3 for the Beginner Policies (BP) as illustrated in Table 1. For simplicity, we only evaluate the following two scenarios:

- Decentralized-Coding: (Number of Systems)2 * 8
- Centralized-Configuration: 2(Number of Systems) * 3

The effect of increasing Number of Systems that require application security policies is shown in Figure 3.

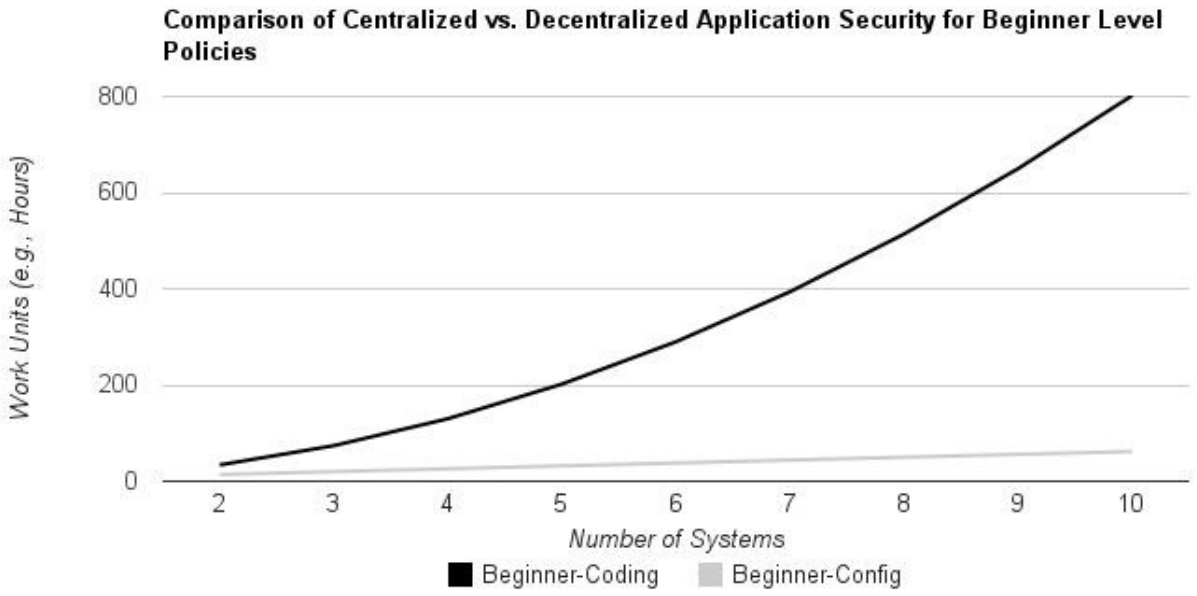

Figure 3 – Normalized Work Units for Beginner Policies with Increasing Number of Integrations.

In the graph shown in Figure 3, the top line represents the Work Units for the decentralized coding model whereas the bottom line represents the Work Units for the centralized configuration model. With m =10 and n =10 (or (Number of Systems)2 = 100), the decentralized model for Beginner Policies results in 800 Work Units. Similarly, for m =10 and n =10, (or 2(Number of Systems) = 20), the centralized model results in 60 Work Units. For a 10 * 10 integration environment, even Beginner Policies can have over a 1000% increase in work effort between centralized and decentralized application security models.

In building this model, we have equated difficulty with Work Units. Moving from Working Units to actual costs

requires further analysis that should be undertaken by a corporation based on their business practices. While building such frameworks for assessing coding vs. configuration options for application security policies, an enterprise should consider the following factors that impact cost estimates:

- Initial cost of acquiring a gateway with ongoing support costs
- Cost of changing artifacts for existing security policies. For example, the X.509 certificate may expire and impact SSL, SAML and content security policies
- Performance consideration with increasing traffic that requires additional hardware vs. scaling with dedicated gateway using hardware accelerators
- Complexity of managing multiple policies coded in a single application container vs. configuration in a gateway
- Costs of skills required to code vs. configure systems

Organizations that require only a few (2-4) systems to integrate with the Beginner Level application security policies tend to remain in a decentralized coding mode. As the number of systems increase, enforcing and managing even Beginner Level policies becomes a significant endeavor within a decentralized environment. Eventually, enterprises trend towards greater efficiencies through better internal and trading partner integration. With an increase in the number of systems and a move towards Advanced Level policies, the move towards a centralized model becomes inevitable.

Organizational Impact

Centralized vs. decentralized application security models have a significant impact on a company's IT team organization, roles and responsibilities, and on-going application support.

In a decentralized model, team organization is ad-hoc. It may consist of project managers, technical architects and application developers. Seldom are application security professionals included in a project development timeline. Usually, application developers are asked to code security policies in the application itself with basic security coverage. The security-application developers may be contractors associated with the project only for the duration of the project. At the completion of the project, the support and maintenance of the security policies are generally lumped within the general support contract, which is more focused on ensuring that the application continues to run smoothly rather than on preemptive mitigation of emerging security threats.

On the other hand, a centralized application security comprised of highly skilled application security personnel is solely focused on mitigating risk across multiple applications within the enterprise. The organizational roles and responsibilities in the centralized model are clearly defined unlike the ad-hoc roles in the decentralized model. In case of a security breach, in the centralized model, there is ownership and a rapid remediation path. As applications are extended, new security policies can be easily configured. This makes on-going support and maintenance in a centralized model more effective both from a cost and risk mitigation standpoint.

A CIO should fund a centralized security application team and build out a security Software-as-a Service (SaaS) model within his or her data center. The SaaS model provides significant reuse and policy standardization. In the centralized model, security is treated as a first class citizen with application security teams responsible for keeping security policies updated against latest attack vectors. Corporate risk governance rules can be centrally maintained and enforced. The cost of centralized SaaS application security can readily be tracked and shared amongst applications while avoiding multiple and non-uniform decentralized application security deployments.

Conclusion

A centralized application security model has significant advantages. It reduces corporate risk in a consistent and measurable way while keeping expenses low. One of the major benefits of a centralized application security model is that it provides a centralized team, much like human resource or legal departments, within an enterprise that have shared and dedicated roles and responsibilities. This application security team has full accountability and clear goals for ensuring that projects are successful without compromising security.

A majority of application integration projects suffer from myopia where the successful implementation of the application function is all that matters. At a project level, there is minimal consideration for reusable policies that protect information. Without significant oversight and involvement from centralized application security architects with a broader corporate vision, a corporation's application infrastructure can eventually become unmanageable and vulnerable.

In modern, agile-like application development environments, functionality is built and released within days. The expectations of building and releasing valuable functionality are increasing dramatically. This pace of development coupled with ever-changing development teams that consist of contractors and full time employees, results in a higher risk exposure that can only be mitigated by centralized security because it has dedicated full-time employees and security policies that are decoupled from the application development process. Separation of application code from security policies through centralized application security is paramount. It is the only way of building an agile, secure and scalable application infrastructure. With ever-increasing cyber threats and an expanding vulnerability surface area, the centralized security model is the only prudent choice for protecting business-critical applications.

Mamoon Yunus

Mamoon is an industry-honored CEO and visionary in Web Services- and SOA-based technologies. As the founder of Forum Systems, he pioneered Web Services Security Gateways and Firewalls. Mamoon has spearheaded Forum's direction and strategy for six generations of award-winning Web Services Security products. Prior to Forum Systems, he was a Global Systems Engineer for webMethods (NASD: WEBM) where he developed XML-based business integration and architecture plans for Global 2000 companies such as GE, Pepsi, Siemens, and Mass Mutual. Mamoon has held various high-level executive positions at Informix (acquired by IBM) and Cambridge Technology Group. He holds two Graduate Degrees in Engineering from MIT and a BSME from Georgia Institute of Technology. InfoWorld recognized Mamoon as one of 4 "Up and coming CTOs to watch in 2004." He is a sought after speaker at industry conferences such as RSA, Gartner, Web Services Edge, CSI, Network Interop, and Microsoft TechEd. Mamoon has the distinction of showcasing Forum Systems' entrepreneurial leadership as a case study at the MIT Sloan School of Management. He has also been featured on CNBC as Terry Bradshaw's "Pick of the Week."

Contributions

- Reducing Application Cost and Risk through Centralized SOA Security
- Fundamentals of SOA Security Testing
- Watch Your SOA Blind Spots: A Checklist for Testing Web Services