PANORAMA DA **INTEROPERABILIDADE** NO BRASIL

MINISTÉRIO DO PLANEJAMENTO ORÇAMENTO E GESTÃO
Secretaria de Logística e Tecnologia da Informação

PANORAMA DA INTEROPERABILIDADE NO BRASIL

Organizadoras:

Cláudia do Socorro Ferreira Mesquita
Nazaré Lopes Bretas

Brasília, DF
2010

Elaboração: MINISTÉRIO DO PLANEJAMENTO, ORÇAMENTO E GESTÃO
Tiragem: 1000 exemplares
1ª edição: Ano 2010
Disponível também em: www.eping.e.gov.br

Brasil. Ministério do Planejamento, Orçamento e Gestão. Secretaria de Logística e Tecnologia da Informação.
Panorama da interoperabilidade no Brasil / Ministério do Planejamento, Orçamento e Gestão, Secretaria de Logística e Tecnologia da Informação. Org. Cláudia S. F. Mesquita e Nazaré L. Bretas. - Brasília : MP/SLTI, 2010.

251 p.: il. color.

ISBN 978-85-89199-07-0

1. Interoperabilidade - Serviço Público. 2. Informática – Arquitetura e-PING 3. Software Público. I. Título. II. Mesquita, Cláudia do Socorro Ferreira. III. Bretas, Nazaré Lopes.

CDU 316.776:35

Títulos para indexação:
Em Inglês: Overview of Interoperability in Brazil
Em Espanhol: Panorama de la Interoperabilidad en Brasil

Presidente da República
Luiz Inácio Lula da Silva

Ministro do Ministério do Planejamento Orçamento e Gestão
Paulo Bernardo Silva

Secretaria de Logística e Tecnologia da informação – SLTI
Loreni F. Foresti – Secretária Substituta

Chefe de Gabinete
Maria Lúcia de Carvalho Porto

Departamento de Gestão Estratégica da Informação – DGEI
Clesito Cezar Arcoverde Fechine

Departamento de Integração de Sistemas de Informação - DSI
Nazaré Lopes Bretas

Departamento de Serviços de Rede – DSR
Antonio Carlos Alff

Departamento de Governo Eletrônico – DGE
João Batista Ferri de Oliveira

Departamento de Logística e Serviços Gerais – DLSG
Januário Flores

Departamento Setorial de Tecnologia da Informação – DSTI
Fernando Antônio Braga de Siqueira Júnior

Colaboradores

Marcelo Martins Villar
Marcus Borges de Souza

Revisores Técnicos

Alex Pires Bacelar
Cláudia do Socorro Ferreira Mesquita
Corinto Meffe
Danielle Eulália Lelis dos Santos
Dayse Vianna
Fábio Gomes Barros
Fernando Almeida Barbalho
Flávio Soares Corrêa da Silva
Hime Aguiar e Oliveira Junior
Jose Ney de Oliveira Lima
Marcello Alexandre Kill
Marcos Antonio André da Rocha
Paulo Roberto da Silva Pinto
Raul Coelho Soares
Renan Mendes Gaya Lopes dos Santos
Sérgio Augusto Santos de Moraes
Xênia Soares Bezerra
Yuri Fontes de Oliveira

Sumário

CAMINHOS PARA INTEROPERABILIADADE

Andrea Lazzarini Secretaria de Recursos Humanos (SRH) – Ministério do Planejamento, Orçamento e Gestão (MP) – andrea.lazzarini@planejamento.gov.br

Evandro Oliveira SRH – MP – evandro.oliveira@planejamento.gov.br

Fatores Críticos de Segurança em Web Services

A tecnologia denominada, genericamente, Web Service tem avançado como solução tecnológica para diferentes tipos de dificuldades no fornecimento de informações atualizadas e sincronizadas através da Internet. Vários órgãos, entidades, empresas e provedores de informação implementam esta tecnologia de maneira a atender alguns requisitos funcionais, mas sem observar requisitos não funcionais de segurança, essenciais a este tipo de serviço. Neste artigo, faz-se necessário, devido ao contexto de interoperabilidade e aplicação do mesmo, um nivelamento prévio nos conceitos, procedimentos, tecnologias adjacentes e características de Web Services. Posteriormente, detalharemos e aprofundaremos alguns dos requisitos de segurança aplicáveis e exigíveis para a tecnologia. Pretende-se, a partir destes conceitos e propriedades, explicitar as formas e opções de interação entre o fornecedor e o requisitante da informação, com destaque aos fatores críticos associados à segurança dos processos e do serviço. Poderemos, assim, qualificar e dimensionar como serão utilizadas as ferramentas e tecnologias, colocadas à disposição de gestores e depositários das informações, de maneira segura e confiável.

1. INTRODUÇÃO

Embora o passado recente de eventos e vulnerabilidades de serviços colocados à disposição dos usuários finais esteja repleto de exemplos de ataques devastadores a servidores *Web*, é crescente a demanda por serviços com acesso real a dados e informações dos bancos de dados das organizações.

As técnicas e procedimentos básicos para segurança da informação e de dados publicadas na internet, ainda que por redes protegidas, deveriam evoluir com a utilização de *Web Services,* mas a incapacidade de assimilação de parcela significativa dos projetistas desses serviços eleva a vulnerabilidade dessas inovações tecnológicas.

Muitos processos e procedimentos têm sido chamados de *Web Services,* inadequadamente associados às especificações tecnológicas mínimas do serviço. Para efeito desta abordagem, utilizaremos algumas definições bastante comuns e aplicadas a serviços de governo e disponibilidade de informação, sem prejuízo de outras definições.

Faz-se necessário o nivelamento de conceitos antes de iniciar a apresentação do viés segurança de *Web Services.* Esses serviços estão sendo implementados em diversas organizações e no setor público de forma diferenciada dos preceitos e conceitos basilares, o que provoca dificuldades de interoperabilidade e de comunicação e joga por terra os princípios fundamentais da tecnologia.

Um *Web Service* é um *software* projetado para suportar sistemas heterogêneos que se interoperam em uma rede ou através de diferentes redes. O serviço tem uma interface descrita em um formato processável por máquina, denominada WSDL.

Outros sistemas interagem com o *Web Service* de uma maneira descrita previamente, utilizando mensagens SOAP, tipicamente transmitidas pelo protocolo HTTP, com uma serialização na linguagem XML, em conjunto com outras normas aplicáveis a serviços *Web*.

Considerando que *Web Services* é o conjunto de tecnologias disponíveis que mais se adapta às necessidades de interoperabilidade requeridas pela diversidade de sistemas heterogêneos, dadas suas características de permitir conexão de dados desses sistemas e entre arquiteturas diferentes, de maneira independente das suas implementações, é mais do que um fator crítico de sucesso para essas integrações o equilíbrio dos mecanismos de segurança.

O crescimento das diferentes tecnologias, em especial das proprietárias, que buscam se consagrar como padrões *de facto,* criou ambientes, até dentro de uma mesma organização, com modelos e tecnologias incapazes de trocar informações diretamente e de maneira sincronizada. Este quadro tem a criticidade elevada quando os mecanismos de segurança de cada um desses ambientes implementam camadas diferentes, com protocolos variados e de interoperabilidade complexa ou não factível.

Este contexto exige que as especificações de artefatos de segurança apresentem um modelo confiável, adaptável e interoperável, proporcionando interconexão e sessões com patamares mínimos de segurança.

Em alguns casos, a segurança deve se valer de um terceiro confiável, pois os envolvidos – requisitante e fornecedor do serviço – terão processos, protocolos e ambientes tão distintos que a segurança não poderá ser realizada por nenhum dos dois lados, separadamente, e a execução nos dois ambientes provoca *overhead* prejudicial ao processo.

2. *WEB SERVICES* E INTEROPERABILIDADE

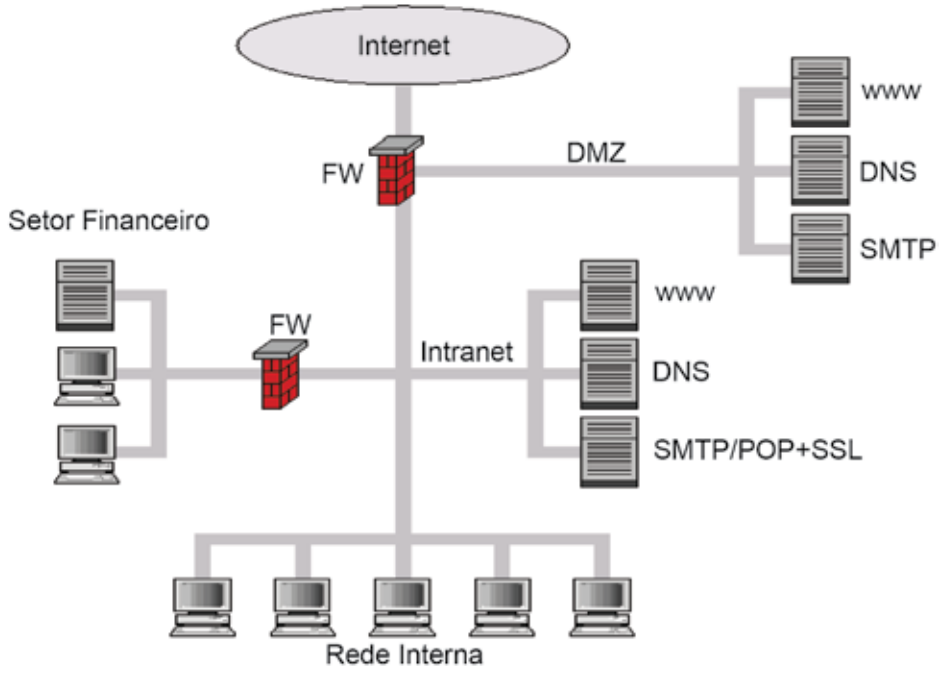
2.1 *Web Service* Tradicional

Um usuário utiliza-se de aplicações em diversas linguagens como HTML, PHP, PERL, ASP etc., para obter informações pré-formatadas e padronizadas de um servidor.

Serviços via Internet geralmente apresentam formas de atualização e consulta ditas *on-line*, mas feitas através de serviços separados das bases de dados reais. Os mecanismos de segurança existentes possuem limitações sobre o perímetro que atuam e utilizam artifícios para simulação de processos *on-line real-time*.

A figura 1 apresenta um exemplo de serviços tradicionais na Internet, disponibilizado por um ISP, e que implementa modelos considerados seguros na proteção do acesso à informação e a serviços primários.

Figura 1. Modelo Tradicional de Serviços de Internet



Observamos que as informações e dados de um setor financeiro, fiscal e tributário de um contribuinte ou informações pessoais de um funcionário, colocados numa rede interna, são protegidos por diversas barreiras de isolamento do *firewalls* e esses dados, preparados

ou transformados, são disponibilizados em uma DMZ[1]. Esta sub-rede, com mecanismos e protocolos diretamente associados ao TCP-IP, permite o provimento de serviços diretos aos usuários através de uma rede de característica não segura.

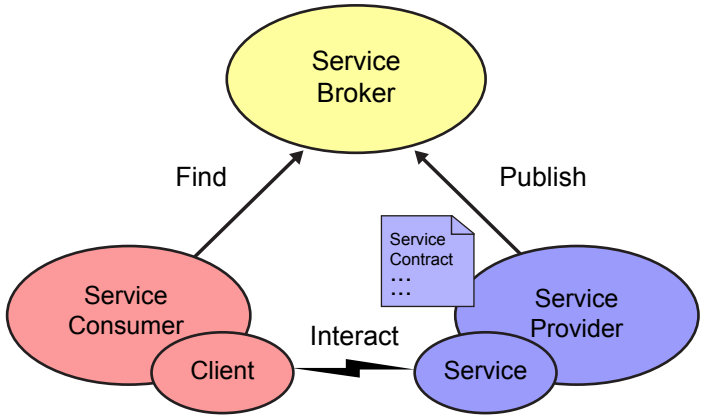
Os bancos de dados reais e internos, com proteção das barreiras mostradas, em muitas organizações, são adaptados e transferidos para uma DMZ e lá disponibilizados. Aplicações próprias e internas cuidam das atualizações necessárias à manutenção de bases consolidadas e reais.

A evolução das aplicações distribuídas entre redes locais interligadas provocou exigências de protocolos de comunicação e armazenamento de dados, com níveis de segurança lógica superiores aos requisitos de segurança baseados nos aspectos de acesso físico.

Dessa forma, as novas tecnologias devem prover segurança nas diversas camadas de acesso físico, nas camadas de acesso lógico e nos dispositivos de transmissão e armazenamento de dados.

2.2 Arquitetura Orientada a Serviços (SOA)

Figura 2. Arquitetura Orientada a Serviços (SOA)



Numa arquitetura SOA, a principal preocupação é o provimento de dados e informações, associados a serviços diferenciados de maneira estruturada do ponto de vista do usuário final.

A figura 2 demonstra um *Service Provider*[2] que publica um Contrato de Serviço representando os serviços, dados e informações colocados à disposição dos interessados. Um cliente ou consumidor de serviços procura os serviços e recursos disponíveis para fazer sua requisição e receber suas respostas.

1. DeMilitary Zone – Zona Desmilitarizada ou área em que as informações possam ser disponibilizadas à internet sem possibilidade de intervenção e acesso externo ao dado primário.
2. Provedor de Serviços da Internet para usuários registrados e não registrados.

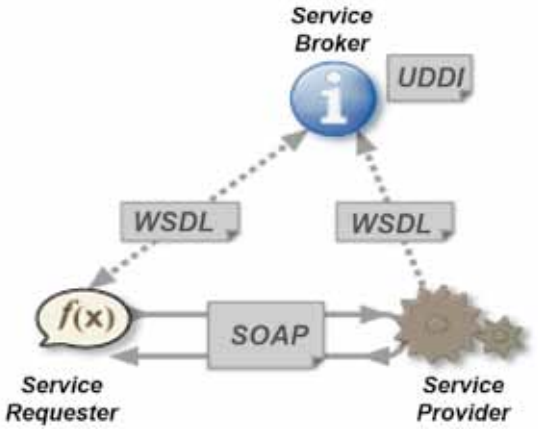
Esta arquitetura tem como fundamento a disponibilização das funcionalidades implementadas pelas aplicações na forma de serviços. Estes serviços, normalmente, são conectados através de um barramento de serviços[3], que disponibiliza interfaces ou contratos, acessíveis através dos chamados *Web Services* ou de outra forma de comunicação entre aplicações. A SOA se baseia nos princípios da computação distribuída e utiliza a simplificação do modelo *request/reply* para estabelecer a comunicação entre os sistemas clientes e os sistemas que implementam os serviços, não se importando com as tecnologias pelas porções cliente e servidor e seus respectivos ambientes de implementação.

A SOA deve utilizar conjuntos de boas práticas e criar um processo de facilitação para as atividades de encontrar, definir e gerenciar os serviços disponibilizados.

Adicionalmente, SOA se insere na reorganização dos métodos e processos em departamentos de tecnologia da informação, propiciando melhor relacionamento entre as áreas que dão suporte tecnológico às organizações e aos setores responsáveis pelo negócio. A melhoria verificável é possível graças à maior agilidade na implementação de novos serviços e à reutilização dos objetos funcionais e ativos existentes.

2.3 *Web Services* - SOA

Figura 3. *Web Services* (SOA)



A figura 3 representa um *Web Service* aderente a uma arquitetura SOA. Através do protocolo SOAP, um usuário faz uma requisição [*f(x)*] e recebe a resposta de um provedor de serviços, previamente acordados através de um WSDL, que deve ter sido publicado, previamente, num catálogo de serviços através de uma UDDI.

2.4 Interoperabilidade

Os processos descritos nos itens 2.1, 2.2 e 2.3 permitem que as tecnologias utilizadas no requisitante de serviço e nos provedores de serviço possam ser diferentes, desde que, para integração, sejam utilizadas interfaces que as tornem interoperáveis.

3. Do inglês Enterprise Service Bus (ESB).

Num modelo proprietário ou não interoperável, os requisitos de comunicação e de segurança são providos pela própria ferramenta que fornece dados e informações. Os requisitantes devem se submeter às regras do provedor.

Web Services que utilizam padrões abertos para comunicação, permitem maior independência, se comparados aos modelos proprietários. A interoperabilidade de um *Web Service,* em linhas gerais, é feita com artefatos que estabelecem uma padronização a ser implementada e aceita pelas partes fornecedoras e requisitantes.

Em determinado momento, as funções de requisitante e fornecedor são invertidas. Constituem-se exemplos essenciais da adoção da tecnologia de *Web Service* interoperável os seguintes itens:

a) Mensagens XML;

b) Interface WSDL;

c) Protocolo SOAP;

d) Catálogo UDDI;

e) WS-Security;

f) WS-Trust.

Estas formas de mensagens, protocolos, interfaces e catálogos não são exclusivas e nem únicas, mas são, atualmente, as mais utilizadas e cujo domínio é essencial para implementar e utilizar adequadamente *Web Services* com níveis de segurança mínimos e satisfatórios.

2.4.1 Mensagens XML

XML é uma linguagem de marcação, recomendada pelo W3C, que produz mensagens a partir de infraestrutura única para diversas linguagens. Permite a comunicação de diferentes linguagens, independentemente daquelas utilizadas pelo requisitante ou fornecedor de dados e informações.

2.4.2 Interface WSDL

WSDL é uma linguagem baseada em XML que funciona como um contrato do serviço a ser publicado e disponibilizado, com descrição de tarefas, operações e métodos, obrigatórios ou acessórios, destinados a padronizar as trocas entre requisitantes e provedores do serviço.

2.4.3 Protocolo SOAP

SOAP, um protocolo de troca de informações, utiliza tecnologias que permitem a construção de mensagens que podem trafegar entre protocolos diferenciados. SOAP tem como componentes: um envelope das mensagens; regras de codificação; convenção para RPC; ligação com protocolos subjacentes. Adicionalmente, possui mecanismos que possibilitam:

a definição de Unidade de Comunicação; o estabelecimento de Parâmetros de Tratamento de Erros; as extensões que permitem evoluções e trocas; e a representação de Tipos de Dados em XML para possibilitar troca de mensagens SOAP e HTTP.

2.4.4 Catálogo UDDI

UDDI é um protocolo que especifica um método para publicar e pesquisar diretórios de serviços, como um catálogo de páginas amarelas, atualizável conforme regras estabelecidas em uma arquitetura SOA. Uma UDDI gerencia informações, implementações e metadados dos serviços oferecidos. A partir do catálogo publicado, usuários interessados nos dados e serviços selecionam e obtêm o que foi requisitado. Uma especificação UDDI contempla as APIs SOAP para publicar e obter informações; os esquemas XML do modelo de dados e do formato das mensagens SOAP; as definições do WSDL e as definições do próprio registro UDDI. O uso do protocolo UDDI possibilita o reuso de partes do catálogo e a categorização com adoção de conceitos de herança e hierarquia.

2.4.5 WS-Security [OASIS 2005]

Especificação que descreve aperfeiçoamentos para mensagens SOAP com intuito de proporcionar qualidade de proteção das mensagens através da verificação de integridade, inclusão de atributos de confidencialidade e sincronismo de autenticação única de mensagem.

WS-Security também fornece, de maneira geral, mas extensível, mecanismo para associar *tokens* de segurança de mensagens, através da especificação WS-Trust.

Além disso, o WS-Security descreve como codificar *tokens* de segurança, certificados X.509 e tíquetes Kerberos, bem como a forma de incluir proteção às chaves criptografadas.

2.4.6 WS-Trust [OASIS 2007]

É a especificação de mecanismos básicos para troca de informações de maneira segura a partir de requisitos do WS-Security. Quando duas partes precisam se comunicar, sejam elas requisitante ou fornecedor da informação, faz-se necessária a definição de primitivas adicionais e extensões para a troca de um *token*, ou outro mecanismo de segurança, para que as credenciais de segurança entre as partes possam ser trocadas num ambiente seguro, de maneira direta ou indireta. Esta especificação define extensões dos métodos para a emissão, renovação e validação de *tokens* de segurança a serem trocados.

A partir do uso dessas extensões, os aplicativos podem iniciar comunicação segura projetada para trabalhar com o quadro geral de serviços da *Web*, incluindo descrições de serviço WSDL, UDDI e SOAP.

2.5 Segurança em Serviços via *Web*

Os níveis de segurança dos serviços de internet são dependentes de ferramentas e protocolos adicionais a esses serviços. Numa estrutura simples de aplicação *Web* disponibilizada através de HTML ou de outra linguagem, conforme apresentado no item 2.1, é exigida a adição de protocolos e serviços adicionais como HTTPS[1].

É fundamental, a fim de evitar ataques ao serviço e acesso indevido às informações, a elevação de níveis de segurança à medida que os dados reais ficam mais próximos dos requisitantes, condição ainda mais essencial com o aumento da possibilidade de requisitantes diferentes terem acesso aos dados e informações. Esses cuidados devem ser tomados através de adoção de medidas, controles e tecnologias preventivas em contraposição às iniciativas reativas aos problemas de segurança.

Algumas características de *Web Services* incrementam suas vulnerabilidades. A extensibilidade é um exemplo que pode comprometer a segurança pelo uso de um benefício ou vantagem do serviço. Sob outros aspectos, a interrupção do serviço, tanto acidental quanto intencional, devido a ataques de negação de serviço, retrata a relação entre fornecer mais serviços a mais requisitantes e aumentar os riscos de paralisação ou queda no nível de serviço contratado.

Se *Web Services* elevam o nível de riscos, reputações podem ser prejudicadas e investimentos perdidos. Medidas de segurança devem prevenir servidores de ataques, falhas e acidentes, visando ao acesso confiável e à maior disponibilidade das informações.

Numa conceituação geral, um servidor seguro e confiável é um servidor espelhado ou que possui cópias de segurança que podem, no caso de falha do *hardware* ou do *software*, ser reconstruído rapidamente e retomar o nível de serviço.

Nesse contexto, os maiores problemas de segurança em ambientes *Web Service* são:

1.Proteger o servidor de dados: garantir que o servidor continuará sua operação, que os dados/informações não serão modificados ou acessados sem autorização;

2.Proteger o servidor de aplicações: garantir que o servidor não apresentará falhas de serviço, seja por capacidade ou disponibilidade;

3.Proteger informações que transitam entre servidor e cliente: garantir que as mensagens sejam passadas de maneira confiável, íntegra e que a informação seja fornecida somente para aqueles autorizados a obtê-la.

Além das ferramentas e técnicas de segurança tradicionais que podem ser adotadas para servidores, como restrição de acesso, *backups* e localização em ambientes com proteção física e lógica, deve-se garantir que as pessoas que irão acessar o façam através de meios de comunicação seguros.

Nesse contexto, os *firewalls* são configurados para que todas as conexões externas a uma rede interna passem por poucos locais, controlados e bem monitorados. O posicionamento

desses anteparos na rede, em relação ao servidor da aplicação e dados, é estratégico e deve ser analisado detalhadamente.

A vantagem de colocar o servidor protegido por um ou mais *firewalls* é que essas barreiras bloquearão o acesso de outras aplicações a outros serviços da rede na qual está o *Web Service,* pois, normalmente, um servidor oferecerá somente serviços TCP/IP para responder às requisições dos clientes: HTTP na porta 80, e HTTPS[4] na porta 443 etc.

Vantagens e desvantagens, custos e benefícios são diferentes para cada *Web Service,* dependendo do posicionamento do mesmo em relação a um ou mais *firewalls* e nas redes internas. Transações que utilizem apenas a porta 80 do protocolo/serviço HTTP sobre TCP/IP apropriam-se de um menor nível de segurança quando comparadas ao HTTPS.

Escolhas inapropriadas de configurações e posicionamento do servidor destinado a responder requisições de um *Web Service* podem comprometer toda a estrutura do serviço em termos de segurança de nível do serviço. Antes de projetar e implementar mecanismos de segurança para *Web Services*, o modelo de funcionamento deve ser avaliado e os requisitos de segurança serão reforçados ou priorizados em função da melhor configuração de oferta dos serviços. A adoção de diversas ferramentas e diferentes tecnologias de segurança não significa melhoria no nível de segurança. Cada serviço determina como cada ferramenta e tecnologia deve ser customizada para elevar esses níveis de segurança individualizados por serviço e o nível geral de segurança do ambiente.

3. REQUISITOS DE SEGURANÇA EM *WEB SERVICES*

Com a padronização dos protocolos para *Web Services*, as necessidades de segurança passam a ser exigidas em camadas ou subserviços que devem estar presentes num mapa de risco, discriminado serviço a serviço, a fim de demonstrar níveis diferentes de investimento, absorção de risco e resultados previstos.

Web Services, distribuídos em várias plataformas e domínios, devem ser baseados na análise das ameaças existentes, as previsíveis e as possíveis, para cada ponto requisitante do serviço. Como o provedor não tem controle da rede ou estação solicitante do serviço, a prevenção, através de análise completa do serviço fim-a-fim, deve ser mandatória e documentada.

Web Services devem ser compatíveis com a política de serviços e segurança da informação da organização provedora de dados e informações, desde a fonte do dado até a entrega do mesmo ao requisitante. Assim, é necessária a construção de um *framework*[5] aderente ao modelo e políticas de segurança publicadas e avaliadas para cada um dos *Web Services* colocados à disposição.

São fatores críticos de sucesso para serviços via *Web* a compreensão, análise, detalhamento e indicação dos processos essenciais para diminuição das vulnerabilidades e defesa dos requisitos de segurança mínimos.

4. Diferencia-se do HTTP, basicamente, por utilizar protocolos de segurança como SSL e TLS.
5. Ver Glossário - *Framework*.

3.1 Infraestrutura Básica de Funcionamento de *Web Services*

Os *Web Services* prescindem de algumas tecnologias ou mecanismos lógicos apropriados para que possam atender aos requisitos de segurança exigidos. A adoção da arquitetura SOA tem contemplado as áreas de Redes de Computadores, Engenharia de *Software* e Arquitetura de Sistemas. A proposta da Engenharia de Sistemas Orientados a Serviços[6] (SOSE), segundo Tsai [12], é um novo paradigma que envolve computação orientada a objetos, baseada em componentes, com os desenvolvedores segmentados em três entidades colaborativas, denominadas Construtores de Aplicação, Publicadores de Serviços e Desenvolvedores de Serviços. Esta separação facilita a compreensão da divisão das camadas, necessária à mudança e alteração dos modelos de desenvolvimento e implementação.

A arquitetura SOA sugere que as funcionalidades de um *software*, associadas àquelas atribuídas a *middleware* e *hardware*, podem ser disponibilizadas como serviços. É nessa lógica que os *Web Services* avançam na integração de novas aplicações e no aproveitamento de aplicações e dados existentes, constituindo aplicações e serviços compostos.

3.1.1 Aplicações Compostas

As Aplicações Compostas propõem a integração de novas aplicações, baseadas em objetos e modelos de dados já existentes. Essas aplicações devem ser projetadas para que possam ser compostas, sincronizadas e concatenadas com outras, criando, assim, novas aplicações que estariam definidas, segundo Keyser [6], em três camadas, claramente separadas, determinando e delimitando a composição do serviço. São componentes desta proposição:

- Interface do Usuário
- Composições de Serviços
- Composições de Dados/Informação

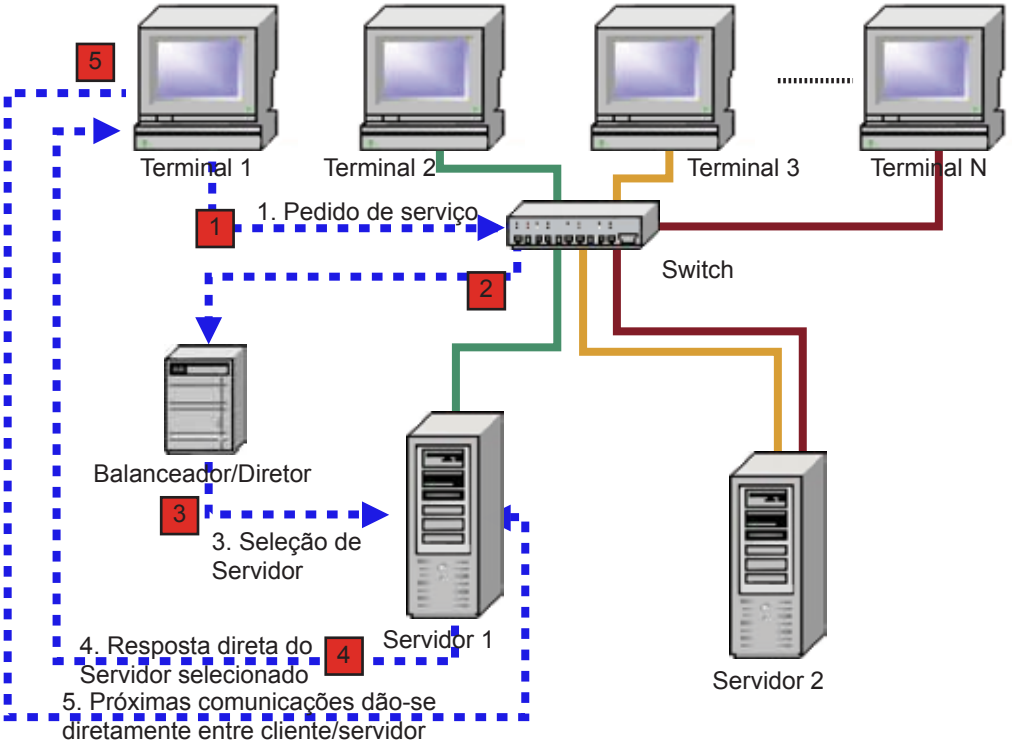
Keyser [6] também descreve uma série de características que os sistemas devem ter para serem usados em composições determinantes para a segurança do serviço:

- Identidade do Serviço
- Sensibilidade ao Contexto
- Infraestrutura de eventos[7]
- Repositórios e mecanismos de descobrimento
- Interface do Usuário
- *Frameworks* de modelagem

6. Do inglês, Service Oriented System Engineering.
7. *Web Services* apresentam severas limitações neste componente.

Montado este quadro, temos uma estrutura de fornecimento de serviço que deve ser protegida nas suas três principais camadas, conforme figura a seguir.

Figura 4. Conexão - Requisição - Resposta



O modelo apresentado na figura 4 é genérico e não prevê nenhuma proteção. É como se a mudança da tecnologia fosse implementada para melhoria do processo, rapidez de processamento, distribuição do processamento, e sem a preocupação de que os terminais sejam algum ponto de rede confiável ou não confiável.

A colocação de "Servidor 1" e "Servidor 2" com acesso a dados reais e base de dados única e disponibilizando serviços para redes não confiáveis ou para toda a internet, como acontece com *Web Services*, exige a adição de elementos, entre eles componentes e/ou camadas, que elevem a segurança e promovam a eliminação de algumas vulnerabilidades conhecidas e cuja implementação é condição básica ao fornecimento da informação com segurança e confiabilidade.

3.2 Segurança em *Web Services*

Um das maiores dificuldades em elevar o nível de segurança em *Web Service*s, em patamares superiores ao modelo tradicional de servidores localizados em uma DMZ com acesso *off-line* dos usuários finais às bases de dados reais, reside na constatação de que a maioria dos sistemas e aplicações desenvolvidos preocupam-se com seus requisitos funcionais e relegam os requisitos não funcionais a um segundo plano, de maneira que, no momento da implantação, os ajustes feitos, via de regra, deixam de lado exigências primárias de segurança.

A seguir, três serviços-conceito que envolvem *hardware* e *software* especializados e elevam a segurança de servidores de aplicação e de dados e que, se apropriadamente utilizados, diminuem os riscos dos *Web Services*.

3.2.1 *Load Balance* (Balanceamento de Carga)

Todo conjunto de *hardware*/*middleware*/*software* provê disponibilidade de serviços com limites de capacidade e utilização, especialmente em aplicações com fila intensa. O mesmo serviço tem que ser dividido entre vários provedores de serviço, sob pena de se tornar congestionado. Se normalmente isso é previsível, um ataque por negação de serviço funciona como uma carga excepcional, além do previsto, acarretando falha de segurança.

A solução tecnológica de *Load balance* permite especialização e separação em pequenos grupos de fornecedores de informação e serviços sobre os quais se faz um balanceamento de carga. A utilização das CPUs, dos dados armazenados, ou de toda uma rede, introduz o conceito de clusterização ou *Server Farm*[8], já que o balanceamento será feito para vários servidores. Em uma rede de servidores, o balanceamento é uma técnica para distribuir a carga de trabalho uniformemente entre dois ou mais computadores, entre enlaces de rede, entre CPUs, entre dispositivos de armazenamento ou outros recursos computacionais, objetivando otimizar a utilização de recursos, maximizar o desempenho, minimizar o tempo de resposta e evitar sobrecarga.

Utilizando múltiplos componentes com o balanceamento de carga, em vez de um único componente, pode-se aumentar a confiabilidade através da redundância. *Load balance* não elimina diversas vulnerabilidades se implementada isoladamente. Entretanto, se associada com recursos de *honeynets*[9], possui a capacidade de elevar a eficácia dos *Web Services*. Permite ao gestor do *Web Service* administrar a sua própria segurança, estabilidade, disponibilidade, capacidade de resposta e itens relacionados à performance do serviço. A escalabilidade e flexibilidade das Gerências de Capacidade, Falhas e Configuração ficam facilitadas com a adoção deste tipo de mecanismo.

3.2.2 Certificação Digital

Considerando o uso de Certificados Digitais com a apropriada utilização de uma Infraestrutura de Chaves Públicas, várias das vulnerabilidades são, verdadeiramente, eliminadas.

O uso de Certificados Digitais de servidores e usuários provê vários requisitos para atendimento à elevação da segurança em *Web Services*.

Os Certificados Digitais podem identificar e autorizar as operações e requisições dependendo do usuário que faz as solicitações. Dessa forma, a troca de mensagens entre duas URLs abrigadas em servidores de redes diferentes atende ao princípio da Autenticação, em que somente aqueles servidores reconhecidos poderão fazer requisições.

8. *Server Farm* é um cluster de servidores de rede. Um grupo de servidores gerenciados como entidade única e que compartilha a mesma forma de conexão física e armazenamento de dados.
9. Ver item 3.2.3

Os usuários de *Web Services* são corretamente e univocamente identificados, de maneira que todos os procedimentos realizados por um usuário e/ou servidor fiquem armazenados numa trilha, com registros de todas as ações, horários e responsáveis disponíveis para os processos de auditoria e fiscalização.

3.2.3 *Honeypot/Honeynet*

Segundo Chave, Hoepers e Stending-Jessen [1], *Honeynets* e *Honeypots* são recursos computacionais, especialmente destinados a sofrerem ataques ou serem comprometidos no lugar dos servidores reais de provedores de serviços, aplicações ou dados.

Honeynets e *Honeypots* podem ser classificados como de baixa ou alta interatividade. Conceitualmente, os *honeypots* são considerados de baixa interatividade por somente tratarem de desviar os ataques e proteger os ativos principais com a simulação de um ambiente real. *Honeynets* ou *Honeypots* especializados são considerados de alta interatividade por possuírem elementos que, além de protegerem os dados reais, tratam de pesquisar e obter informações dos atacantes para projetar formas mais completas e avançadas de proteção.

Nas *Honeynets* existem mecanismos de contenção robustos, com múltiplos níveis de controle, com subsistemas para captura e coleta de dados e consequente geração de alertas de maneira mais rápida e eficaz do que em *Honeypots* com baixa interatividade.

A utilização de *Honeynets* ou *Honeypots* eleva o nível de segurança dos *Web Services* contra diversos tipos de ataques normalmente originados por computadores e redes não confiáveis ou hostis, evitando ou diminuindo ataques conhecidos como DDoS, DoS, DNS e Spoofing.

3.3 Interoperabilidade de *Web Services* com segurança

A adoção dos três serviços-conceitos descritos nos itens anteriores cria uma base de segurança mínima essencial para cada *Web Service* implementado num ISP.

A partir dessa base, devem-se avaliar as características e funcionalidades de cada *Web Service* e seus componentes, que deverão estar representados ou reproduzidos nos itens de segurança necessários ao *Web Service*.

Assim como em qualquer serviço colocado à disposição via *Web*, a proteção deve ser implementada a partir das possibilidades de ataque e vulnerabilidades dos recursos tecnológicos utilizados. Nesse sentido, a proteção deve ser pensada e aplicada a dois eixos de ataque: intencionais e não intencionais.

Normalmente, um possível atacante intencional gasta mais de 80% de seu tempo buscando vulnerabilidades. No caso de ataques não intencionais, a fragilidade é explorada por erros de projeto ou diferenças entre conceituação, concepção e implementação.

Qualquer que seja a regra do negócio colocado em *Web Services*, gerências de segurança devem ser claramente definidas para atender aos níveis de segurança padronizados e aceitáveis.

Podemos relacionar as seguintes gerências como necessárias e que devem ser detalhadamente descritas e implementadas a cada caso de *Web Service:* a) Gerência de Processo e Negócios – *Workflow*; b) Gerência de Autoatendimento aos Usuários Requisitantes – *Customer Self Care*; c) Gerência de Pré-Planejamento – *Pre-Planning*; d) Gerência de Contabilização – *Account Management & Billing;* e) Gerência de Interfaces e Conexão – *Information Buss*; f) Gerência de Configuração de Serviço – *Service Configuration*; g) Gerência de Aprovisionamento e Dimensionamento de *hardware*, *middleware*, *software* – *Provisioning*; h) Gerência de Eventos e Falhas – *Fault Management*; e i) Gerência de Nível de Serviço – *SLA Management*.

A não implementação de qualquer dessas gerências em níveis mínimos provoca a perda de controle e da capacidade da gestão do serviço e de sua capacidade de acessibilidade e disponibilidade. São recorrentes a queda do serviço e o dispêndio de tempo e recursos. Para retomar ou estabelecer essas gerências, somente depois de tratar do problema ou vulnerabilidade apresentada. Os custos e profundidade com que cada uma das gerências é implementada definem a relação custo x benefício de cada mecanismo. A segurança ideal exigiria custos mais elevados do que o valor do ativo informacional ou patrimonial a se proteger.

Em termos de interoperabilidade, destacamos os itens a seguir, não ordenados em grau de importância, principalidade ou custos, como necessários de serem analisados, avaliados, medidos e implementados para cada *Web Services* e, se possível, separados por WSDLs. Esses itens podem e devem ser compartilhados com outros serviços que não são exclusivamente *Web Services,* conferindo interoperabilidade entre eles. A replicação e diversidade de versões e modelos devem ser evitadas, visto que proporcionarão maior instabilidade para as gerências de segurança e disponibilidade de serviços *Web*.

3.3.1 Acessibilidade e Disponibilidade

Um *Web Service* seguro deve ser capaz de dar completo acesso e disponibilidade da informação a todos os requisitantes de serviço autorizados. No caso de informação pública, e considerando que todos os usuários e cidadãos são, potencialmente, requisitantes autorizados, deve-se implementar um conjunto de defesas que eliminem ou mitiguem ataques de indisponibilidade sofridos através de técnicas conhecidas.

Os direitos de acesso devem ser claros e definidos para que sejam separados os ataques intencionais de não intencionais, e para que sejam identificados os responsáveis por ataques ou erros de procedimento. Leitura, escrita, execução, adição, modificação e exclusão são exemplos de direitos a serem especificados a determinados usuários ou grupo de usuários, distintamente e inequivocamente. Sem essas especificações, os trabalhos de rastreamento, identificação, auditoria e outros ficam comprometidos, o que prejudica a segurança como um todo.

3.3.2 Administração da Segurança

Um *Web Service* seguro, assim como qualquer outro serviço disponibilizado por meios de comunicação digitais, deve fornecer mecanismos para administrar a segurança de seus

níveis de serviço. Consiste em uma gerência separada e, preferencialmente, desvinculada de todos os recursos de infraestrutura e funcionamento do próprio serviço. A gerência deve ter controle externo para verificação da disponibilidade do serviço fora da rede. A indisponibilidade do serviço não pode afetar a gerência de segurança e a capacidade de reconfiguração e recomposição do serviço.

3.3.3 Auditabilidade

Um *Web Service* seguro deve permitir que todos seus processos e componentes sejam auditados, implementar mecanismos e trilhas de auditoria que não sejam alteráveis pelos operadores e outras gerências de serviço, ser o primeiro serviço a ser ligado e o último a ser desligado, prover informações para sistemas e aplicações de auditorias externas e ter mecanismos que inibam completamente a alteração de qualquer configuração ou componente do serviço sob auditoria.

3.3.4 Autenticação das Partes

Entendendo Autenticação como o processo que verifica a capacidade de determinado requisitante, de identificar quem ele diz ser ou representar.

A maioria dos serviços colocados à disposição via *Web* exige uma identificação do usuário para que os modelos de disponibilidade e autorização possam ser cumpridos.

Um *Web Service* deve possibilitar que a identificação e autenticação entre as partes que irão trocar dados e informações tenham protocolos preestabelecidos. Normalmente, a utilização de uma Infraestrutura de Chaves Públicas provê, com as características inerentes ao Certificado Digital[10], os requisitos para autenticação das partes de maneira segura e confiável.

Da mesma forma que o requisitante tem que provar que é quem diz ser, o fornecedor da informação deve mostrar que não é um clone do *site* que o requisitante deseja. Orientações aos usuários quanto ao uso dos navegadores em *sites* seguros devem ter capítulo especial nos manuais e nos próprios *sites*, posto que a mudança de comportamento dos usuários será mais segura quanto maior for o nível de informação disponível. Armadilhas em serviços *Web* iniciam-se com a pouca atenção do requisitante sobre em qual ambiente ele navega ou opera.

3.3.5 Autorização

Um *Web Service* deve ter ação de autorização associada e subsequente à autenticação das partes. A autorização permite, a partir de algum serviço ou pessoa identificada e autenticada, acesso a atividades e processos específicos, limitados ou amplos.

O SAML (Security Assertion Markup Language) [7] é uma norma emergente para a troca

10. Propriedades do Certificado Digital – Identificação; Autenticidade; Não Repúdio; Privacidade; Integridade; Confidencialidade – que podem ser utilizadas juntas ou separadamente.

de informação sobre autenticação e autorização. O SAML soluciona um importante problema para as aplicações da próxima geração, que é a possibilidade de utilizadores transportarem seus direitos entre diferentes *Web Services*. Isso tem importância para aplicações que pretendem integrar um número de *Web Services* para formar uma aplicação unificada a partir de aplicações compostas ou unitárias.

Sucedem e complementam a linguagem SAML os padrões definidos WS-Security [8] e WS-Trust [9] referenciados no e-PING versão 2010.

3.3.6 Confidencialidade

Um *Web Service* que troca informações e dados entre equipamentos e aplicações ou entre o servidor e usuários finais deve garantir que o meio em que aquela mensagem ou informação, quando estiver sendo transferida, seja imune, o máximo possível, a quebras de sigilo e confidencialidade relativas ao seu conteúdo.

Confidencialidade, nesse contexto, é a garantia de que as informações serão mantidas em sigilo, com acesso limitado às pessoas competentes e autorizadas para conhecê-la e obtê-la.

3.3.7 Integridade

Em *Web Services* que trocam informações é necessária e mandatória a garantia que os dados/informações trocados sejam entregues na íntegra, de acordo com a base original. Mais do que proteger o conteúdo, o servidor e o solicitante da informação devem ter a certeza de que o conjunto de *bits* existente, e mantido na base de dados original, corresponde ao conjunto de *bits* entregue ao solicitante e esteja de acordo com o requisitado.

Integridade é um atributo determinante para garantir que a informação não será, acidentalmente ou maliciosamente, alterada, substituída ou destruída entre o solicitante e o servidor da mesma. É atributo complementar da Confidencialidade, pois deve ser implementado para todas as trocas de informação, sejam elas confidenciais ou não.

3.3.8 Não Repúdio e Autoria

Um *Web Service* deve prover mecanismos que garantam a autoria de ações efetuadas através desses serviços. A autoria deve ter elementos capazes de identificar, *a posteriori*, quem realizou determinada requisição, por quanto tempo, com quais recursos e com quais objetivos, de maneira persistente e por vezes transitória. Junto à Autoria, o Não Repúdio é condição básica para a gestão de segurança de um *Web Service*.

Não Repúdio é um método ou regra pela qual o requisitante ou o remetente de dados requisita ou fornece dados com o comprovante de entrega e autoria. O destinatário, requisitante da informação, tem a garantia de identidade do remetente, de modo que, mais tarde, o remetente não poderá negar ter sido o originador dos dados.

3.3.9 Política de Segurança

Um *Web Service* deve ter clara a publicação de uma Política de Segurança. É fundamentado nesta política que os requisitantes e fornecedores de informação devem atuar. A política geral de segurança deve ser publicada a todos os possíveis usuários, dentro e fora da rede de uso do *Web Service*, de tal forma que a punição ou até mesmo a criminalização possa ser tipificada e qualificada.

Uma Política de Segurança pode ser representada pela "Política de Uso", pelas regras de funcionamento, e deve apresentar, sempre que possível, a aceitação explícita dos requisitantes às regras e Política através de mecanismos de "COMPREENDO"; "CONCORDO"; "ACEITO OS TERMOS E CONDIÇÕES" e outros.

São procedimentos aplicáveis à formalização de Acordos de Cooperação entre as partes, no qual as linhas gerais da Política de Segurança sejam formalizadas, entendidas e aceitas.

3.3.10 Portas e Protocolos

Um *Web Service* deve ter seu detalhamento não publicizado, mas de conhecimento entre as partes autorizadas que trocarão informações predeterminadas, item que detalhe e descreva as especificações de portas, protocolos e respectivas especificações operacionais de cada serviço.

As portas destinadas a cada Serviço ou Aplicação são importantes para elevar ou diminuir requisitos de serviços. Entende-se, aqui, como serviços aqueles de baixo nível e tratados por equipamentos de rede com nenhuma intervenção do usuário e que influenciam na interoperabilidade entre cliente e servidor de *Web Services* por estes terem, entre si, diversos componentes de rede que filtram e separam esse tipo de aplicação.

3.3.11 Sincronizador de Tempo

Um *Web Service*, assim como qualquer outro serviço de rede diferente e com mecanismos de acesso, identificação e autorização heterogêneos, deve prover um sincronizador de tempo que permita equiparar os relógios e cronômetros, dando ao fornecedor da informação a capacidade de serializar as requisições e identificar cada uma das ações e requisições efetuadas. Esse sincronismo possibilita e facilita a auditabilidade e cessão de informações para ambientes externos ao fornecedor da informação.

Virtual Private Network (VPN)

Serviços disponibilizados via *Web Service* devem se apropriar dos conceitos e implementações de VPNs como se cada *Web Service* ou WSDL fosse uma rede de comunicação exclusiva. Isso significa que as abordagens utilizadas em VPNs devem servir de modelo para as implementações de *Web Services*. São opções de implementação a serem adotadas: a) *Gateways* de redes internas com Endereçamento IPs privativos; b) Circuitos Públicos Virtuais; c) Segmentação de Tráfego; d) IPSec; e) Acesso de Usuários Remotos Identificáveis; f) Usuários remotos não identificáveis.

4. CONCLUSÕES

Os protocolos e tecnologias associadas a serviços denominados *Web Services* tiveram uma explosão com o advento da denominada *Web* 2.0, forçando alguns ISP a implementarem aplicações com níveis de segurança aquém do necessário.

O aumento do uso de computadores com capacidade de processamento nas pontas exigiu elevação e mudanças em procedimentos e conceitos de uso e proteção da informação. A *Web* 2.0 e os *Web Services* exigem mudança comportamental e adaptação de ferramentas para melhoria na proteção dos dados e informação, que são o principal ativo de qualquer organização.

As etapas de avaliação de riscos, análise de custos x benefícios que considerem a probabilidade de ocorrência do risco, os valores envolvidos com a possível perda de dados, os custos de recuperação dos dados e da confiabilidade do depositário dessas informações, associados aos custos de implantação e obtenção dos resultados, devem ser reproduzidos e detalhados para *Web Services*.

Tem sido necessária a interoperabilidade das aplicações e a padronização do gerenciamento e das ferramentas de segurança. Serviços disponíveis na *Web* não precisam de ferramentas distintas em ambientes distintos. A elaboração de modelos e fluxos de funcionamento lógicos, publicados em repositórios abertos, não vulnerabiliza a informação ou processo e possibilita a ampliação da qualidade da informação do serviço prestado. Portais e *sites* de Governo Eletrônico devem apresentar esses modelos e fluxos claramente, pela extrema necessidade de interoperabilidade funcional e não funcional, com respectiva integração e interoperabilidade de tecnologias e ferramentas.

Etapas como Classificação das Informações e Dados, Planos de Tecnologias Utilizadas, Planos de Contingência, Planos de Recuperação de Nível de Serviço, Elaboração de Política de Segurança, Contratos de Níveis de Serviço, Acordos de Cooperação Tecnológica devem ser consolidados antes de serem disponibilizados os serviços em produção, a fim de garantir um mínimo de funcionalidade e disponibilidade.

A responsabilidade sobre níveis de segurança em *Web Services* será sempre compartilhada, com responsabilidades e atribuições claramente determinadas. Eventos não previstos e indesejáveis devem ser evitados em *Web Service*s, devido à gravidade e a consequências que causam.

A inobservância dos mínimos detalhes tratados neste artigo levam à ruptura dos serviços e consequente quebra da interoperabilidade entre sistemas e ambientes heterogêneos. O mau uso ou negligência na aplicação de regras obrigatórias tornam vulneráveis não somente o *Web Service* específico e malprojetado, como também provocam a vulnerabilidade de dados e informações limítrofes que antecedem os focos das falhas de segurança.

A adoção de mecanismos de segurança mais fortes, em *Web Service*s, além de proporcionar a elevação da interoperabilidade, provê a estrutura de TI das organizações de eficiente plataforma de segurança, preparada para novas customizações e integrações.

SIGLAS

ABNT – Associação Brasileira de Normas Técnicas
API – Application Programming Interface
DdoS – Distributed Denial of Service
DNS – Domain Name Server
DOS – Denial of Service
DTD – Document Type Definition
e-MAG – Modelo de Acessibilidade de Governo Eletrônico
e-PING – Padrões de Interoperabilidade de Governo Eletrônico
HTML – HyperText Markup Language
HTTP – HyperText Transfer Protocol
HTTPS – HyperText Tranfer Protocol Secure
ICP – Infraestrutura de Chaves Públicas
ISO – International Organization for Standardization
ISP – Internet Service Provider – Provedor de Serviços na Internet
ITI – Instituto Nacional de Tecnologia da Informação
MP – Ministério do Planejamento, Orçamento e Gestão
NSA – National Security Agency
OASIS – Organization for the Advancement of Structured Information Standards
PKI – Public Key Infrastructure
PNG - Portable Network Graphics
REST – Representational Status Transfer
RFC – Request for Comments
RPC – Remote Procedure Call
SLTI – Secretaria de Logística e Tecnologia da Informação
SOA – Service Oriented Architecture
SOAP – Simple Object Acess Protocol
SSL – Secure Socket Layer
TLS – Transport Layer Secure
UDDI – Universal Description, Discovery and Integration
URL – Uniform Resource Locator
VPN – Virtual Private Network
XML – eXtensible Markup Language
XSD – XML Schema Definition
W3C – World Wide Web Consortium
WSDL – Web Service Definition Language
WSIA – Workplace Safety and Insurance Act

GLOSSÁRIO

Arquitetura – Arquitetura de *software* de programas, sistemas computacionais ou estruturas de sistemas que se tornam uma abstração durante um ou mais processos em operação.

Artefato – Um componente constituído de informações digitais que podem ser de tamanhos variados ou compostos de outros artefatos. São exemplos de artefatos: um documento no formato XML, uma imagem no formato PNG, uma mensagem.

Assinatura Digital – Um valor calculado com um algoritmo criptográfico e anexado a um objeto de dados de tal forma que os destinatários dos dados possam usar a assinatura para verificar a origem e a integridade dos dados. [RFC 2828]

Certificado Digital – É um arquivo digital que contém um conjunto de informações referentes à entidade para o qual o certificado foi emitido (seja uma empresa, pessoa física ou computador), mais a chave pública referente à chave privada que se acredita ser única e de posse exclusiva da entidade proprietária do certificado.

Componente – Um componente é um objeto de *software*, com capacidade para interagir com outros componentes, por englobar algumas funcionalidades ou um conjunto de funcionalidades. Deve ter uma interface bem definida e obedecer a um comportamento prescrito comum a todos os componentes similares e dentro de uma arquitetura.

Controle de Acesso – Proteção de recursos computacionais e informacionais contra acessos não autorizados através do uso de política de segurança e procedimentos implementados que permitem o acesso somente a autorizações validadas. [RFC 2828]

Framework – Conjunto de classes específicas para determinada tarefa. Ao ser usado, o trabalho criado e implementado possibilita o reuso e cria facilidades na produção de novos serviços e produtos com economia de tempo e recursos.

ICP – Infraestrutura de Chaves Públicas – Modelo de distribuição de dados e informações baseado em Criptografia Assimétrica que disponibiliza um par de Chaves (Privada e Pública) e utiliza-se de atributos de Certificados Digitais para manter informações seguras.

Load Balance – É uma técnica utilizada na computação para distribuir o trabalho entre vários processos, computadores, discos ou outros recursos.

PKI – (acrônimo do inglês Public Key Infrastructure) – ver ICP.

QoS – (acrônimo do inglês Quality of Service) – Ver **Qualidade de Serviço.**

Qualidade de Serviço - **QoS** – Qualidade de Serviço é um conjunto de obrigações publicizadas e aceitas entre a entidade prestadora do serviço de informações e os clientes ou requisitantes de informações.

Sessão – Interação de determinada duração entre entidades do sistema, muitas vezes envolvendo um usuário, requisitante de informação, seja ele um equipamento, um

programa ou interface, caracterizada pela manutenção de um estado de comunicação durante a interação. [Glossário WSIA]

SOA – (acrônimo do inglês Service-Oriented Architecture) – É um conceito no qual aplicativos ou rotinas são disponibilizadas como serviços em uma rede de computadores (externas e internas) de forma independente e se comunicando através de protocolos, linguagens e tecnologias em padrões abertos.

SOAP – (acrônimo do inglês Simple Object Acess Protocol) – É um protocolo para intercâmbio de mensagens estruturadas em uma plataforma descentralizada e distribuída, utilizando tecnologias baseadas em XML. É um dos protocolos usados na forma de disponibilização de serviços através da *Web*.

UDDI – (acrônimo do inglês Universal Description, Discovery and Integration) – É um protocolo aprovado como padrão pela OASIS e especifica um método para publicizar e descobrir descrições de diretórios de serviços e objetos em uma Arquitetura Orientada a Serviços (ver SOA).

VPN – (acrônimo do inglês Virtual Private Network) – É uma rede de comunicações privada normalmente utilizada por uma empresa ou um conjunto de empresas e/ou instituições, construída sobre a infraestrutura de uma rede de comunicações pública.

XML – (acrônimo do inglês eXtensible Markup Language) – É uma recomendação do W3C para gerar linguagem de marcação para necessidades especiais. É um subtipo de Linguagem Padronizada de Marcação Genérica (SGML) capaz de descrever diversos tipos de dados com o propósito de facilitar o compartilhamento de dados e informações.

WSDL – (acrônimo do inglês Web Services Description Language) – É uma linguagem que serve para descrever as interfaces de serviços baseados em XML que inclui a estrutura do conteúdo e o protocolo de transporte utilizado na interface.

REFERÊNCIAS

[1]CHAVES, M. H. P. C.; HOEPERS, C.; STEDING-JESSEN, K. **Honeypots e Honeynets**: Definições e Aplicações. Disponível em: <http://www.cert.br/docs/whitepapers/honeypots-honeynets/>. Acesso em: 15 fev. 2010.

[2]GARFINKEL, Simson; SPATFORD, Gene. **Comércio & Segurança na Web**. São Paulo: Market Press, 1999.

[3]IBM. **Business Rule Standards** – Interoperability and Portability. Disponível em: <http://www.w3.org/2004/12/rules-ws/paper/113/>. Acesso em: 10 mar. 2010.

[4]_____. IBM. **Web Services Handbook** – Development and Deployment. IBM. com/Redbooks, 2005. Disponível em: <http://www.redbooks.ibm.com/redbooks/pdfs/sg246461.pdf>.

[5]KEYSER, Chris. **Composite Applications**: The New Paradigm Disponível em: <http://msdn.microsoft.com/en-us/architecture/bb266335.aspx>. Acesso em: 15 mar. 2010.

[6]OASIS. **Technical Standard:** Service Oriented Infrastructure Reference Framework. By: The Open Group., 2001.

[7]_____. 2005.

[8]_____. 2007.

[9]ROSANOVA, Dan. Why UDDI is Important. In: **CIO**. Disponível em: <http://advice.cio.com/dan_rosanova/why_uddi_is_important?page=0%2C0>. Acesso em: 10 fev. 2010.

[10]SILVA, L. S. **Virtual Private Network** – VPN. São Paulo: Novatec Editora, 2003.

[11] SPYMAN, Hacking. **Manual Completo do Hacker**. 4 ed. Rio de Janeiro: Editora Book Express, 2001.

[12]TSAI, W. T. **Service-Oriented System Engineering**: A New Paradigm (SOSE'05), 2005.

[13]VILELLA, R. M. Conteúdo, Usabilidade e Funcionalidade: Três Dimensões para avaliação de portais estaduais de Governo Eletrônico na Web. In: **iP – Informática Pública**. Belo Horizonte: Prodabel/PBH, 2003. – v. 5, n. 1 (jan-jun 2003).

[14]W3C. **W3C Recomendation** - Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. Disponível em: <http://www.w3.org/TR/wsdl20/>. Acesso em: 10 mar. 2010.

[15]_____. **W3C Recomendation** – SOAP Version 1.2 Part 1: Messaging Framework. 2nd ed. Disponível em: <http://www.w3.org/TR/soap12-part1/>. Acesso em: 10 mar. 2010.

[16]_____. **W3C Ubiquitous Web Domain** – Extensible Markup Language (XML). Disponível em:<http://www.w3.org/XML/>. Acesso em: 10 mar. 2010.

[17]_____. **W3C Ubiquitous Web Domain** – XML Schema. Disponível em: <http://www.w3.org/XML/Schema>. Acesso em: 10 mar. 2010.

[18]_____. **W3C Workgroup Note** – Web Services Architecture. Disponível em: <http://www.w3.org/TR/ws-arch/>. Acesso em: 10 mar. 2010.

[19]_____.. **W3C Workgroup Note** – Web Services Glossary. Disponível: <http://www.w3.org/TR/ws-gloss/>. Acesso em: 10 mar. 2010.