Um Survey de Propostas de Melhoria de Segurança em Web Services

Jamille S. Madureira
Instituto Federal de Sergipe
Universidade Federal de Sergipe
Sergipe, Brazil
jamillemadureira@gmail.com

Quelita A. D. da S. Ribeiro
Universidade Federal de Sergipe
Sergipe, Brazil
quelita.diniz@gmail.com

Telmo O. de Jesus
Instituto Federal de Alagoas
Universidade Federal de Sergipe
Sergipe, Brazil
jtelmooliveira@yahoo.com.br

Michel S. Soares
Universidade Federal de Sergipe
Sergipe, Brazil
mics.soares@gmail.com

ABSTRACT

The most effective way to implement applications with high security is to be in line with well-known principles, standards and practices. The negative impact of a security failure can compromise confidential data, allowing unauthorized access and compromise the reliability of the company that is providing the service. Designing secure Web services involves understanding to what threats the provided services are exposed. Therefore, there is a need to develop secure Web Services. In this work, mechanisms of proposals were studied for the improvement of web services security. Security issues have been identified and categorized for each proposed mechanism. As a result, it was observed that the aspects authentication, integrity and confidentiality have been addressed in most studies. The results also indicate that researchers have given an emphasis on improving the SOAP (Simple Object Access Protocol) protocol.

Keywords

Security; Web Services; SOAP, REST, WSDL

1. INTRODUÇÃO

Ao longo dos últimos anos, a utilização da Internet como um meio para a troca de informações tem aumentado significativamente. O compartilhamento de informações e recursos tornaram-se essenciais para a realização de negócios e prestação de serviços em muitos setores, como governo, educação e negócios em geral [11].

Web Services oferecem suporte para uma nova geração de aplicações corporativas, como serviços bancários on-line, motores de busca, computação em nuvem, entre outros [9] [3]. Em geral, um Web Service fornece serviços a clientes [12]. Web Services têm feito os arquitetos de software projetarem, desenvolverem e implantarem aplicações web complexas para diversas organizações. Nesse contexto, há uma necessidade de oferecer Web Services seguros para os clientes a fim de permitir que somente usuários autorizados acessem serviços e recursos [3]. Existem diversas propostas de melhorias na segurança de Web Services que fornecem os meios necessários que envolvem aspectos relacionados a autenticação, autorização, confidencialidade e integridade, entre outros [9] [17].

No artigo [15] é apresentado um estudo sobre técnicas de detecção de ataques XML (*Extensible Markup Language*) em mensagens SOAP. As técnicas de detecção são investigadas, as limitações são descritas e foram discutidas as medidas de prevenção e mitigação de ataques XML. As medidas consistem em política de Web Services (WS-*Policy*), estrutura da mensagem SOAP, abordagens baseadas em *string*, entre outras. Porém, os autores afirmam que todas as técnicas apresentadas são ineficientes ou acarretam outros problemas, como desempenho.

Em [1] é apresentado um estudo sobre as preocupações de segurança associadas a Web Services. Os autores ressaltam que Web Services permitem diversas formas de ataques, principalmente através da porta 80 dos *firewalls*, como ataques de negação de serviço, ataques de repetição e detecção de vírus. Ao final, são mostrados os padrões de segurança que estão sendo utilizados, como XML *Encryption*, XML *Signature*, SAML (*Security Assertion Markup Language*), WS-*Security* e UDDI (*Universal Description, Discovery and Integration*).

No artigo [7] foi realizado um levantamento sobre padrões de segurança para serviços Web colocando em perspectiva um em relação ao outro. Para cada padrão de segurança foi mostrado a sua utilização, vantagens, desvantagens e o relacionamento entre eles. Para alguns padrões são utilizados diagramas UML (*Unified Modeling Language*) para mostrar uma descrição mais precisa do funcionamento dos mesmos.

O tema segurança em Web Services, é, portanto, relevante e tem atraído a atenção de diversos pesquisadores, conforme literatura. Esse trabalho tem por objetivo fazer um survey de propostas de melhorias na segurança de Web Services em

EATIS '16 Cartagena das Índias, Bolívar, Colômbia

DOI:

SOAP, REST *(Representational State Transfer)* e WSDL *(Web Services Description Language)*.

2. REFERENCIAL TEÓRICO

Web Services são utilizados na integração de sistemas e na comunicação entre aplicações diferentes, sendo uma importante ferramenta para a indústria de software. O objetivo é permitir que uma coleção de serviços de software seja acessível via protocolos padronizados, cujas funcionalidades podem ser descobertas e integradas a aplicativos com simplicidade [20]. Web Services permitem que as organizações compartilhem dados sem a necessidade de conhecer os softwares por trás do *firewall* [17]. No serviço web, os participantes são classificados em três grupos: Prestador de serviço, clientes de serviço e de registro do serviço.

2.1 SOAP, REST e WSDL

Foram adotadas diversas propostas para realizar trocas de mensagens por meio de Web Services entre clientes e provedores de serviços, mas foi o protocolo SOAP que surgiu como padrão de fato. Definido pelo consórcio W3C, o SOAP é um protocolo de comunicação baseado em XML para a troca de mensagens, independente de linguagem, que trabalha com diversos sistemas operacionais e sobre protocolos de aplicação já consolidados, como HTTP (*Hyper Text Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), FTP (*File Transfer Protocol*), entre outros [6].

REST é um padrão arquitetural para a construção de aplicações web com o intuito de evitar a complexidade e a sobrecarga de processamento dos protocolos de serviços da web [2]. Para isto, o REST utiliza somente HTTP. REST é baseado no estilo "sem estado", em que o servidor não armazena nenhuma informação de contexto. Toda informação necessária para atender a uma requisição deve estar contida nela mesma. Isto torna o servidor mais simples, pois ele não precisa considerar o contexto atual para tomar decisões, uma vez que toda informação necessária será enviada a ele a cada requisição [8].

WSDL é uma linguagem em XML, extensível, usada para descrever as interfaces de Web Services de forma independente de tecnologia [6]. Um documento WSDL é independente de linguagem e de plataforma e tem por objetivo descrever quais são os serviços oferecidos, mostrar como os clientes e provedores irão processar as requisições e indicar em qual formato o serviço deve enviar as informações para um cliente [6].

Para que um Web Service seja utilizado, é necessário que os potenciais usuários tenham conhecimento de sua interface, da sua semântica, da sua chamada e a sua localização [13]. O objetivo da UDDI é a definição dos conjuntos de serviços que suporta a descrição do negócio e de como o serviço está disponível e das interfaces técnicas utilizadas para acesso a estes serviços [13]. O UDDI está baseado em um conjunto de padrões (HTTP, XML, XML *Schema* e SOAP) e provê uma infra-estrutura fundamental e interoperável para ambientes de software baseados em Web Services disponíveis ao público ou expostos internamente nas organizações. Um registro UDDI oferece um mecanismo para classificar, catalogar e gerenciar Web Services para que os mesmos possam ser descobertos e utilizados [13].

2.2 Segurança em Web Services

Web Services possibilitam o acesso a usuários externos a recursos, com o risco de permitir que intrusos ataquem os aplicativos internos e bancos de dados. A preocupação com segurança em Web Services envolve aspectos de autenticação, autorização, confidencialidade e integridade, entre outros [17].

Segurança de Web Services é baseada em vários aspectos importantes [21] [4] [22]:

- Autenticação: verifica a identidade de um usuário, processo ou dispositivo, muitas vezes como um pré-requisito para permitir o acesso aos recursos em um sistema de informação;
- Autorização: é a permissão para usar um recurso de computador, concedido por um proprietário aplicativo ou sistema;
- Integridade: garante que os dados não sejam alterados de forma não autorizada durante o armazenamento, durante o processamento, ou em trânsito;
- Não-repúdio: depois que é fornecido o comprovante de entrega ao remetente da informação e ao destinatário, é fornecida a identidade do remetente, então não é permitido posteriormente negar o processamento da informação;
- Confidencialidade: preserva as restrições autorizadas no acesso à informação e divulgação, incluindo os meios para proteger a privacidade pessoal e informações proprietárias;
- Privacidade: restringe acesso ao assinante, de acordo com a legislação e a política da organização.

3. METODOLOGIA

O objetivo desse trabalho é realizar um *Survey* de propostas de melhorias na segurança de Web Services com o foco em SOAP, REST e WSDL. Os artigos científicos foram consultados na biblioteca digital *IEEExplore* em outubro de 2015, utilizando as seguintes expressões:

- *Document Title restful OR Document Title soap OR Document Title wsdl AND Document Title security*;
- *Document Title rest OR Document Title SOAP OR Document Title WSDL AND Document Title security.*

Para seleção dos artigos, os critérios adotados foram que os artigos deveriam estar na língua inglesa e conter uma proposta de melhoria na segurança de Web Services para SOAP, REST e WSDL.

Ao realizar a pesquisa, foram encontrados 11 resultados para a primeira *string* de busca e 13 para a segunda, totalizando 24 artigos que abordassem o tema. Para seleção dos artigos, os critérios adotados foram que a data de publicação dos trabalhos fosse nos últimos 10 anos, e o artigo deveria conter uma proposta de melhoria na segurança de Web Services para SOAP, REST ou WSDL. Após aplicar o critérios de escolha foram selecionados nove artigos, que estão apresentados na tabela 1.

Table 1: Trabalhos Selecionados

Ano	Autores	Título	Proposta
2009	Peng, D. et al	An extended username Token-based approach for REST-style Web Service security authentication	REST
2009	Gruschka, N.;Iacono, L.	Vulnerable cloud: SOAP message security validation revisited	SOAP
2010	Gruschka, N. et al	A design pattern for event-based processing of security-enriched SOAP messages	SOAP
2011	Jamil, D. et al	Security issues in cloud computing and countermeasures	SOAP
2011	Shahgholi, N. et al	A new SOA security framework defending Web Services against WSDL attacks	WSDL
2011	Mirtalebi, A.;Khayyambashi, M.	Enhancing security of Web Service against WSDL threats	WSDL
2012	Serme, G. et al.	Enabling Message Security for RESTful services	REST
2014	Charles, J.;Kumar, S.	Design of a secure architecture for context-aware Web Services using access control mechanism	SOAP
2014	De Backere, F. et al.	Design of a Security Mechanism for RESTful Web Service Communication through Mobile Clients	REST

4. RESUMO DOS TRABALHOS SELECIONADOS

Em [16], os autores propuseram uma abordagem baseada em *UsernameTokenext* estendida para Web Services de estilo REST. A tecnologia *UsernameToken* é um serviço Web que recebe um nome de usuário e senha exclusivos para cada usuário, que pode acessar os recursos somente após se autenticar. Como a senha é armazenada no servidor, um atacante pode invadi-lo e receber a senha do usuário. Quando o cliente inicialmente define uma senha para o domínio, o servidor irá calcular um valor de *hash* para o usuário e guardá-lo para o banco de dados. Na proposta dos autores, uma senha secundária WS-*Security UsernameToken* é adicionada para o cabeçalho HTTP. Após a adição de senha secundária, o invasor não pode continuar a operação porque o servidor terá o valor de *hash* salvo no banco de dados para calcular a senha secundária correspondente, e em seguida, comparar com senha secundária enviada pelo cliente. A fim de obter permissão para acessar recursos, o atacante tem que quebrar uma criptografia de dois níveis. Assim, a proposta aumenta a dificuldade de descoberta da senha, e resiste ao ataque de detecção de forma eficaz. Outra vantagem da abordagem é que os prestadores de serviços podem personalizar a sua própria autenticação, melhorando a sua flexibilidade e segurança.

No trabalho [9] é discutida a vulnerabilidade de Web Services no *Amazon Elastic Compute Cloud* (EC2) em ataques XML. Além disso, são retratadas as etapas de verificação necessárias para validar uma solicitação de mensagem SOAP. O artigo contém um guia de boas práticas para detectar ataques de XML *Signature Wapping* e ataques de negação de serviços em mensagens SOAP. A solução dos autores é baseada em validar as mensagens recebidas em conformidade com a política de segurança e XML *Schema*. Embora as considerações de segurança mostrassem que a solução proposta reduz os ataques *signature wapping*, uma prova prática está faltando.

No trabalho [10] é apresentado um padrão de projeto de software para aplicações de processamento XML baseadas em eventos SAX (*Simple API for XML*). O padrão proposto usa o estilo *pipeline* e oferece um *gateway* de segurança. O padrão de segurança é dividido em quatro módulos. O módulo de validação de esquema verifica as mensagens SOAP em conformidade com o esquema XML. O módulo de segurança é capaz de processar extensões de segurança como definido pela especificação WS-*Security*. Com base nos resultados do módulo de segurança, o módulo de aplicação de política verifica a conformidade com a política de segurança. O módulo de controle de acesso verifica se o autor da mensagem tem permissão para acessar o recurso solicitado. Os eventos são transmitidos ao longo do SAX e manipulados por cada módulo. O artigo contém como exemplo um *gateway* de segurança implementado em um sistema chamado *CheckWay*.

Em [12], é retratado que a computação em nuvem é um novo conceito de tecnologia que utiliza a Internet e servidores remotos, a fim de manter dados e aplicativos de computação. Embora a computação em nuvem ofereça muitas vantagens para os usuários, também tem vários problemas de segurança. Os autores mencionam quatro ataques que envolvem XML *Signature*, navegador, *malware* na nuvem e negação de serviços e oferecem possíveis soluções para os problemas, que podem ser vistas na tabela 2.

Table 2: Soluções de ataques em mensagens SOAP para computação em nuvem. Adaptada de [12]

Ataques	Soluções
XML Signature	Combinar assinaturas WS-Security + XML Signature e chave de verificação de assinaturas fornecida com certificado X.509
Navegador	WS-Security nos navegadores web
Malware na nuvem	Verificação de integridade de segurança
Negação de serviço	Firewall ou detecção de intrusão (IDS)

O artigo [19] contém uma proposta de *framework* para proteger Web Services contra ataques WSDL. O *framework* utiliza uma proposta de encriptação assimétrica contendo os componentes PKI (*Public Key Infrastructure*) e XKMS (*XML Key Management Specification*). PKI é uma infraestrutura de chaves públicas que tem como objetivo gerenciar

Table 3: Aspectos de segurança abordados

Aspectos	Temas		
	REST	**SOAP**	**WSDL**
Autenticação	Peng, D. et al [16]	Jamil, D.,et al. [12] Charles, J.;Kumar, S.[3] Gruschka, N.; Iacono, L. [9]	Shahgholi, N. et al. [19]
Autorização		Charles, J.;Kumar, S.[3]	
Integridade		Jamil, D.,et al [12] Charles, J.;Kumar, S. [3] Gruschka, N. et al [10]	Mirtalebi, A.;Khayyambashi, M. [14]
Privacidade		Charles, J.;Kumar, S.[3]	
Confidencialidade	Serme, G. et al. [18]	Gruschka, N. et al [10]	Mirtalebi, A.;Khayyambashi, M. [14]
Não-repúdio	De Backere, F. et al. [5]	Jamil, D.,et al [12] Charles, J.; Kumar, S.[3]	

chaves e certificações. O XKMS define protocolos para o registro, localização e validação de informações de chaves públicas. O propósito do XKMS é facilitar o gerenciamento da PKI, abstraindo sua complexidade fornecendo uma interface entre a aplicação e o PKI. Os autores ressaltam que não é necessário aplicar esta estratégia de segurança em todos os arquivos WSDL. Devem ser utilizados em Web Services críticos, ou seja, quando se tem uma função essencial e importante podendo interromper a operação da organização.

No artigo [14] é apresentado um modelo para encriptar documentos WSDL para melhorar a segurança de Web Services. Os autores destacam duas ameaças com relação à WSDL: exploração do conteúdo WDSL e a modificação de parâmetros. Na primeira ameaça, os ataques podem ser realizados em informações importantes como tipos e mensagens e na segunda, os ataques podem acontecer por meio da adulteração dos parâmetros conseguindo acesso a informações não autorizadas. O modelo proposto é composto de dois módulos: o *Encryptor Service* que tem como papel encriptar o documento WSDL e o *Key Generator Service* que tem por finalidade a geração e gerenciamento das chaves que são necessárias nos algoritmos de encriptação.

Em [18], os autores explicam que a segurança e a confiabilidade de aplicativos em nuvem exigem forte confiança no protocolo de comunicação usado para acessar recursos da web. Nesse trabalho, os autores propõem um protocolo de segurança REST para fornecer serviço de comunicação seguro, equivalente a WS-*Security*. A solução proposta minimiza a sobrecarga de processamento para os consumidores de serviços, sem interferir na composição de serviços já em vigor, mantendo a confidencialidade das mensagens. A vantagem da abordagem é esconder a complexidade para os consumidores, sem poluição em parâmetros de solicitação. Os autores realizaram uma avaliação de desempenho, considerando cenários heterogêneos para comparar diferentes mecanismos de segurança entre eles, e o comportamento do servidor de aplicação quando se tratar de serviços RESTful contra Web Services baseados em SOAP. Os resultados mostraram que os serviços RESTful são processados de forma mais eficiente a partir de qualquer ponto de vista, o que é inerente à finalidade do serviço.

Em [3], os problemas de segurança são identificados e são propostos métodos contra as ameaças à segurança. A ênfase é o desafio de projetar modelos de privacidade e controle de acesso para dispositivos móveis. A proposta de melhoria é baseada em uma arquitetura de segurança para serviços sensíveis ao contexto web para a autenticação de credenciais, autorizações, controle de acesso, integridade, não-repúdio entre o consumidor e proprietário. A transmissão segura de informações de contexto fica entre o solicitante (cliente) e o fornecedor (servidor) com as especificações de segurança e processo de registro função do usuário.

Em [5], os autores afirmam que a segurança continua a ser um grande problema para RESTful Web Services, pois muitos dos mecanismos de segurança atuais violam os princípios RESTful e não são capazes de lidar com as vantagens de escalabilidade que REST oferece. Assim, é proposto um mecanismo de segurança personalizado, usando apenas um mínimo de elementos não RESTful. Comparando esta aplicação com uma solução totalmente baseada em TLS (*Transport Layered Security*), fica claro que este método supera TLS, tanto no aspecto de mensagens, quanto no tratamento de sobrecarga. Referenciais de escalabilidade revelaram que um grande número de pedidos simultâneos influenciam significativamente o desempenho. Devido à generalidade de REST, o mecanismo de segurança proposto pode ser adotado por uma grande variedade de outros serviços Web RESTful.

5. ANÁLISE DOS RESULTADOS

A fim de analisar as propostas de melhoria em segurança de Web Services encontradas nesta pesquisa, os trabalhos foram categorizados de acordo com os aspectos de segurança descritos na Seção 3.2 e os temas abordados na Seção 3.1. Na Tabela 3 são apresentados os trabalhos encontrados, relacionando os temas aos aspectos de segurança.

Por meio da Tabela 3 é possível afirmar que uma das maiores preocupações em segurança de Web Services diz respeito à autenticação, considerando que este é um dos aspectos de segurança em que foram encontradas propostas que envolvem SOAP, REST e WSDL. Os trabalhos [16], [12], [3], [9] e [19] contêm propostas de melhorias que abordam esse aspecto. WS-Security é utilizado em [16] e [12] para auxiliar a compor os mecanismos de segurança propostos pelos autores. A especificação WS-Security [16] fornece segurança em relação à troca de mensagens com integridade, confidencialidade ou autenticação. A vantagem de usar o WS-Security é que as mensagens são protegidas, mesmo se a mensagem percorra por vários serviços ou intermediários [17] [16].

Para o aspecto da confidencialidade, também foram encontradas propostas que envolvem os três temas desta pesquisa. Em [18], é recomendada uma melhoria voltada para REST. Em [10], existe uma proposta para SOAP, e em [14]

a proposta é voltada para a linguagem WSDL.

Ainda de acordo com a Tabela 3, outra preocupação recorrente diz respeito ao critério da integridade. Propostas de melhoria são encontradas em [12], [3], [10] e [14]. Porém, nenhuma das propostas é direcionada para REST.

Os autores Charles e Kumar [3] contemplaram os aspectos de autenticação, autorização, integridade, privacidade e não-repúdio. Pode ser visto que a proposta deles foi a mais ampla na cobertura dos aspectos em SOAP para melhoria na segurança de Web Services.

Pode ser observado que SOAP possui propostas para todos os aspectos de segurança, pois é um protocolo que é utilizado independentemente de linguagem, sistemas operacionais e por protocolos de aplicação como HTTP, SMTP, FTP [6], enquanto o REST utiliza somente HTTP [2].

Os autores [9], [12] e [18] propuseram melhorias para segurança de Web Services em computação em nuvem. Destas propostas, as duas primeiras são relativas à autenticação e a última diz respeito à confidencialidade.

6. CONCLUSÃO

A segurança em Web Services é fundamental nas inúmeras aplicações existentes, como computação em nuvem, dispositivos móveis, navegadores web, entre outros. Todavia, ainda não existe um padrão para que todos os aspectos de segurança sejam contemplados. Neste trabalho foi realizado um *survey* de propostas de melhorias na segurança em Web Services.

A segurança em SOAP possui padrões e especificações que já são utilizadas, como o WS-Security que suporta, integra e unifica vários modelos, mecanismos e tecnologias de segurança em uso no mercado. O SOAP é indicado para estruturas que exigem alta segurança e aplicações complexas.

O REST, por usar como base o protocolo HTTP, fica dependente dos mecanismos de segurança desse protocolo para disponibilizar um Web Service seguro, e tem como vantagem uma resposta mais rápida às requisições. Pela sua simplicidade e facilidade de entendimento, geralmente qualquer cliente ou servidor com suporte a HTTP/HTTPs pode fazer o uso dessa tecnologia, porém a falta de padrões maduros e a baixa segurança são os principais pontos fracos. Em situações onde não é necessária alta padronização e uma segurança elevada, REST funciona de maneira adequada.

A segurança em WSDL é um dos problemas críticos, pois expõe ao ambiente externo a interface de um Web Service. Desta forma, devem ser aplicados mecanismos de segurança para bloquear o acesso de agentes não-autorizados. Nos trabalhos encontrados, as soluções propostas envolveram PKI, XKMS e encriptação principalmente com relação a ameaças de exploração de conteúdo e manipulação de parâmetros.

Como trabalho futuro, sugere-se a implantação de algumas propostas a fim de avaliar o funcionamento delas na prática, já que somente um dos trabalhos apresenta um estudo de caso que aplica o mecanismo proposto. Além disso, é fundamental avaliar o desempenho da aplicação ao implantar tais mecanismos. Outro trabalho seria uma pesquisa relacionada à segurança de Web Services em nível das camadas de rede, transporte e aplicação avaliando as vantagens e desvantagens de cada trabalho.

7. REFERENCES

[1] Balasubramanian, N., and Ruba, A. Security: a Major Threat for Web Services. In *International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)* (2012), IEEE, pp. 104–109.

[2] Bianco, P., Kotermanski, R., and Merson, P. F. Evaluating a Service-oriented Architecture. In *Technical Report CMU/SEI-2007-TR-015* (2007), Software Engineering Institute (SEI) - Carnegie Mellon University (CMU), pp. 1–79.

[3] Charles, P. J., and Kumar, S. Design of a Secure Architecture for Context-aware Web Services Using Access Control Mechanism. In *International Conference on Contemporary Computing and Informatics (IC3I)* (2014), IEEE, pp. 780–784.

[4] Cleveland, F. M. Cyber Security Issues for Advanced Metering Infrasttructure (AMI). In *Power and Energy Society General Meeting -Conversion and Delivery of Electrical Energy in the 21st Century* (2008), IEEE, pp. 1–5.

[5] De Backere, F., Hanssens, B., Heynssens, R., Houthooft, R., Zuliani, A., Verstichel, S., Dhoedt, B., and De Turck, F. Design of a Security Mechanism for RESTful Web Service Communication Through Mobile Clients. In *Network Operations and Management Symposium (NOMS)* (2014), IEEE, pp. 1–6.

[6] de Mello, E. R., Wangham, M. S., da Silva Fraga, J., and Camargo, E. Segurança em Serviços Web. *Livro de Minicursos do VI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. Santos: SBC* (2006), 1–48.

[7] Fernandez, E. B., Ajaj, O., Buckley, I., Delessy-Gassant, N., Hashizume, K., and Larrondo-Petrie, M. M. A Survey of Patterns for Web Services Security and Reliability Standards. In *Future Internet* (2012), vol. 4, Molecular Diversity Preservation International, pp. 430–450.

[8] Fielding, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, 2000. AAI9980887.

[9] Gruschka, N., and Iacono, L. L. Vulnerable Cloud: SOAP Message Security Validation Revisited. In *International Conference on Web Services (ICWS)* (2009), IEEE, pp. 625–631.

[10] Gruschka, N., Jensen, M., and Iacono, L. L. A Design pattern for Event-based Processing of Security-enriched SOAP Messages. In *International Conference on Availability, Reliability, and Security (ARES'10)* (2010), IEEE, pp. 410–415.

[11] Hatala, M., Eap, T. M. T., and Shah, A. Federated Security: Lightweight Security Infrastructure for Object Repositories and Web Services. In *International Conference on Next Generation Web Services Practices (NWeSP)* (2005), IEEE, pp. 6–pp.

[12] Jamil, D., and Zaki, H. Security Issues in Cloud Computing and Countermeasures. *International Journal of Engineering Science and Technology (IJEST) 3*, 4 (2011), 2672–2676.

[13] Milanez, J., et al. *Modelo de Gerência para SPKI Através do XKMS*. PhD thesis, 2005.

[14] Mirtalebi, A., and Khayyambashi, M. R. Enhancing Security of Web Service Against WSDL

Threats. In *2nd International Conference on Emergency Management and Management Sciences (ICEMMS)* (2011), IEEE, pp. 920–923.

[15] Nasridinov, A., Byun, J.-Y., and Park, Y.-H. A Study on Detection Techniques of XML Rewriting Attacks in Web Services. *International Journal of Control and Automation 7*, 1 (2014), 391–400.

[16] Peng, D., Li, C., and Huo, H. An Extended UsernameToken-based Approach for REST-style Web Service Security Authentication. In *2nd International Conference on Computer Science and Information Technology (ICCSIT)* (2009), IEEE, pp. 582–586.

[17] Sachdeva, S., Mchome, S., and Bhal, S. Web Services Security Issues in Healthcare Applications. In *9th International Conference on Computer and Information Science (ICIS)* (2010), IEEE, pp. 91–96.

[18] Serme, G., De Oliveira, A. S., Massiera, J., and Roudier, Y. Enabling Message Security for RESTful Services. In *19th International Conference on Web Services (ICWS)* (2012), IEEE, pp. 114–121.

[19] Shahgholi, N., Mohsenzadeh, M., Seyyedi, M., and Qorani, S. H. A New SOA Security Framework Defending Web Services Against WSDL Attacks. In *Third International Conference on Privacy, Security, Risk and Trust (PASSAT) and Third Inernational Conference on Social Computing (SocialCom)* (2011), IEEE, pp. 1259–1262.

[20] Singhal, A. Web Services Security: Challenges and Techniques. In *8th International Workshop on Policies for Distributed Systems and Networks (POLICY'07)* (2007), IEEE, pp. 282–282.

[21] Singhal, A., Winograd, T., and Scarfone, K. Guide to Secure Web Services. *NIST Special Publication 800*, 95 (2007), 4.

[22] Zissis, D., and Lekkas, D. Addressing Cloud Computing Security Issues. In *Future Generation Computer Systems* (2012), vol. 28, Elsevier, pp. 583–592.