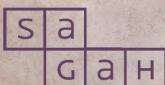


INTEGRAÇÃO DE APLICAÇÕES

Fabiano Berlinck Neumann



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS

Segurança em *webservices*

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Definir segurança da informação.
- Esclarecer sobre os riscos dos dados e comunicação.
- Propor soluções de segurança para tráfego e armazenamento de dados.

Introdução

Cada vez mais, há aplicações que precisam se comunicar entre si, e, à medida que se dá a inovação tecnológica, crescem os problemas de comunicação e segurança. Ter os dados de cartão de crédito espionados na hora em que o cliente os informa para uma compra em um *e-commerce* pode constituir um problema sério para uma empresa e seus clientes.

Para que consigamos propor soluções de segurança para tráfego e armazenamento de dados, é necessário conhecer os riscos dos dados e comunicação, bem como de conceitos de segurança da informação, além de protocolos, ferramentas e tecnologias passíveis de utilizar para alcançar melhores níveis de proteção dos dados.

Neste capítulo, você conhecerá a respeito da segurança da informação, dos requisitos que precisam ser atendidos para garantir a segurança em transferência de dados na *web*, dos riscos dos dados e comunicação, como Denial of Service (DoS), SQL Injection e Man-in-the-middle, bem como soluções de segurança [p. ex., criptografia, Single Sign-on e Virtual Private Network (VPN)].

1 Segurança da informação

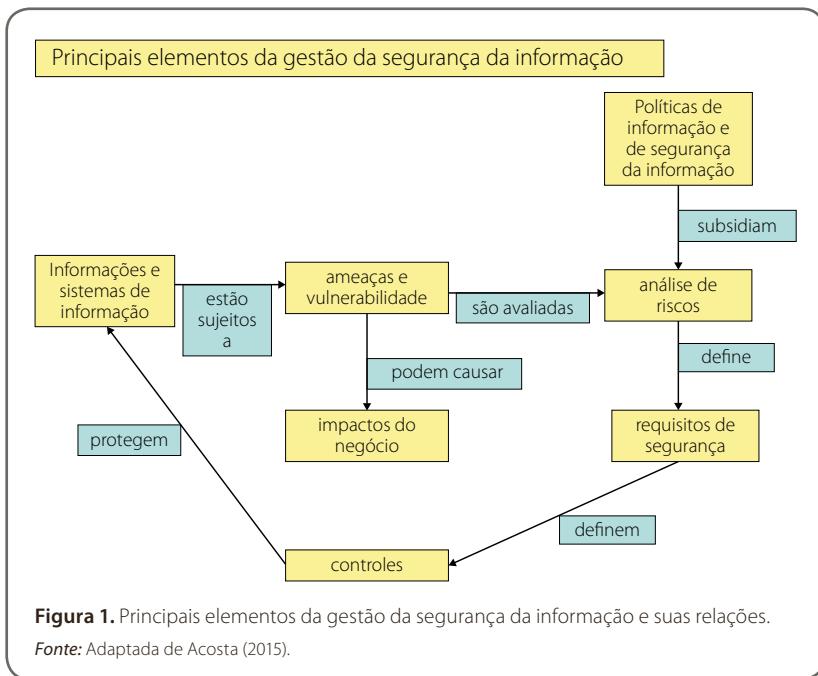
Com o crescimento cada vez maior da quantidade de soluções tecnológicas, em razão da evolução dos dispositivos tecnológicos e dos meios de comunicação, as empresas se tornam cada vez mais dependentes da comunicação entre sistemas, aplicativos e dispositivos, os quais, entre outros benefícios, podem melhorar e automatizar seus processos internos.

De acordo com a Mega Sistemas Corporativos (2018), comumente as grandes empresas utilizam um conjunto robusto de ferramentas digitais, que se comunicam entre si. Cada ferramenta tem um propósito, como sistemas de comunicação de equipes, sistemas de gestão de tarefas, pessoas e vendas, sistemas contábeis, repositórios de código, ferramentas de testes e de atendimento ao cliente, etc.

Muitas vezes, a comunicação entre aplicações distintas se dá de maneira transparente para os usuários. Segundo Vargas (2001), em sistemas distribuídos, as soluções são formadas por dispositivos autônomos e *software* que fornecem abstração como se fossem uma única máquina. Nesse tipo de sistema, a comunicação, ou troca de mensagens, ocorre por meio dos objetos de dados, cuja estrutura e aplicação são definidas pelas soluções de *software* que os utilizarão.

De acordo com Durbano (2018), a segurança da informação dessas aplicações tem importância fundamental para as empresas, sobretudo para o setor de tecnologia da informação (TI), correspondendo ao conjunto de ações que tem como objetivo proteger um conjunto de dados e o seu valor, tanto para as pessoas quanto para as empresas. Ainda, ela se aplica a todos os aspectos de proteção de dados, e não apenas a sistemas computacionais.

Segundo Acosta (2015), os principais elementos da gestão da segurança da informação compreendem as políticas de informação e segurança da informação, a análise de riscos, os requisitos de segurança, os controles, as informações e os sistemas de informação e as ameaças e vulnerabilidades, como mostrado na Figura 1.



Para garantir a segurança das informações trocadas pela *web*, como no caso de usuários que realizam compras e fornecem informações para comércio eletrônico, quatro requisitos de segurança, listados a seguir, devem ser atendidos (GRAHAM *et al.*, 2005).

- Confidencialidade — garante a proteção da informação que está sendo trocada na comunicação *web*, para que não seja exposta para outras pessoas.
- Integridade — garantia de que uma mensagem chegará da mesma maneira que foi enviada, sem ser modificada.
- Autenticação — assegura o acesso aos aplicativos e a seus dados para quem puder comprovar a sua identidade (p. ex., com *login* e senha).
- Irretratabilidade — maneira de certificar que o remetente da mensagem não possa negar o envio e que o receptor não possa negar o recebimento da mensagem.

E um quinto requisito eventualmente útil para permitir que apenas pessoas apropriadas consigam acessar os dados é a autorização, que trata da proteção de recursos, uma vez que o usuário esteja autenticado na aplicação, para que apenas as entidades apropriadas consigam acessar determinados recursos mesmo que autenticados, como no caso de informações em painéis administrativos. Assim, os sistemas podem identificar a entidade e definir se ela conseguirá acessar determinado recurso.

Segundo Wu *et al.* (2013), em virtude da definição de uma arquitetura de *webservices*, unidades de negócios, clientes e parceiros podem conseguir trocar informações, uma capacidade de comunicação vital para o sucesso dos negócios. Assim, os *webservices* possibilitam a união de informações distribuídas de forma prática e econômica, independentemente das barreiras de sistemas operacionais, da plataforma de desenvolvimento e das linguagens de programação.

De maneira simplificada, *webservices* são funções de objetos expostos via HTTP usando mensagens SOAP puras. Em termos de protocolos, um *webservice* inclui os seguintes componentes ou camadas (Figura 2): Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) e Universal Description, Discovery and Integration (UDDI).

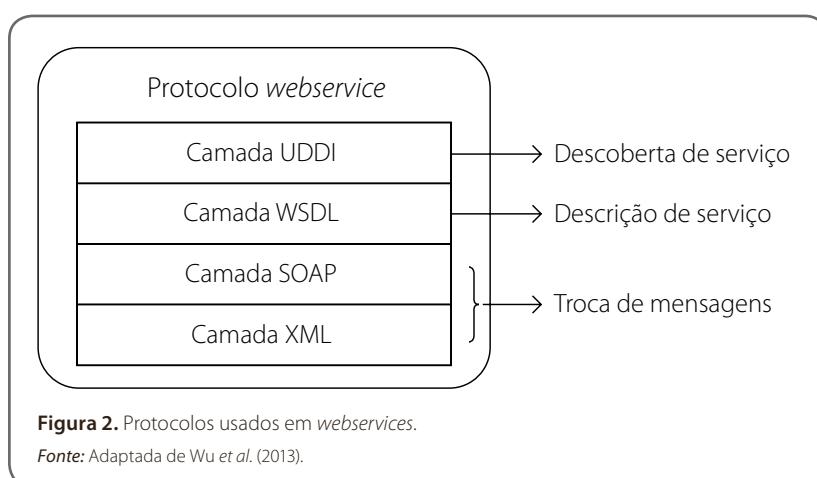


Figura 2. Protocolos usados em *webservices*.

Fonte: Adaptada de Wu *et al.* (2013).

O XML fornece um meio de troca de mensagens pela *web*, por meio de um documento que contém solicitações de informações e respostas entre sistemas diferentes. O SOAP é um padrão ou especificação de trocas de mensagens simples por documentos XML, que permite a comunicação eficaz entre remetentes e destinatários. A WSDL é responsável pela descrição completa para a comunicação entre aplicações. Já a UDDI representa uma forma de publicação e procura de *webservices* na rede, independentemente da plataforma, com registros baseados em XML.

Conforme Graham *et al.* (2005), o SOAP permite a interação mútua dos aplicativos. Para realizar a integração das aplicações entre os negócios, são utilizadas a WSDL e a UDDI. Essa integração e o uso dessas tecnologias de *webservices* possibilitam a criação de aplicações de baixo acoplamento, descentralizadas, com alcance que ultrapassa os limites das empresas. Com isso, desafios precisam ser solucionados em relação à segurança dos *webservices*.

De acordo com Wu *et al.* (2013), a segurança de *webservices* baseia-se em padrões XML abertos, aprovados pela W3C, grupo responsável por fornecer a base de segurança para aplicações que consomem *webservices*. Esses padrões independem da plataforma e promovem a interoperabilidade entre as aplicações. Além disso, a OASIS, grupo responsável por padrões abertos, publicou padrões que definem a forma de expansão de segurança para a comunicação por trocas de mensagens SOAP.

Com base nos critérios de segurança de informação apresentados, há desafios de segurança que precisam ser superados em cada um dos contextos, como apresentado a seguir.

2 Riscos dos dados e comunicação

São diversos os desafios e as complexidades relacionados aos *webservices* capazes de prejudicar os dados e a comunicação em termos de segurança (KUYORO *et al.*, 2012), já que as eventuais ameaças tendem a comprometer a confidencialidade, a integridade ou a disponibilidade de um *webservice* ou do sistema *back-end* exposto por um conjunto de *webservices*, afetando apenas *webservices* ou também *sites*.

Para Durbano (2018), as aplicações de *big data* têm se tornado cada vez mais comuns nas empresas, muitas das quais se comunicam por meio de *webservices*. Entretanto, esse uso apenas faz sentido se os dados estiverem

corretos e disponíveis: quando uma falha acontece, o servidor pode parar de receber informações, podendo comprometer a análise e gerar uma predição errada para determinada situação. Além disso, empresas trabalham com informações estratégicas transmitidas pela *web* e podem ser alvo de *hackers*.

Independentemente do tipo ou do tamanho da falha, ela pode provocar problemas para os negócios. Portanto, é importante que gestores invistam em segurança da informação por meio de ferramentas e técnicas que visem a melhorar a segurança dos próprios negócios, um trabalho que, se bem-feito, pode evitar ataques digitais, falhas humanas ou desastres tecnológicos.

Quando falamos de confidencialidade, um erro é capaz de causar a exposição de dados estratégicos de uma organização ou o vazamento de dados de clientes, que podem ser capturados por *hackers*. Essa falha de segurança costuma ficar evidenciada para o público, promovendo prejuízos financeiros e problemas para a imagem da empresa.

Já o risco que diz respeito à integridade das informações está relacionado a uma falha em arquivos importantes, que podem ser corrompidos com um erro no disco rígido. Se a empresa não tiver *backup*, as operações podem ficar comprometidas.

A indisponibilidade é outro risco associado aos dados e à comunicação, tornando-se essencial que os dados estejam acessíveis no momento em que forem requisitados, especialmente quando se trata da garantia de agilidade de processos das empresas. Um dos possíveis causadores de indisponibilidade são os ataques de sequestro de dados, nos quais *hackers* solicitam valores para resgate ou para não publicá-los.

Com o objetivo de garantir a segurança de dados, também é indispensável buscar garantir a autenticidade das informações, para assegurar que o objeto não sofreu mutações ao longo do processo. Assim, evitam-se fraudes, como no caso de roubo e clonagem de informações de cartões de crédito.

De modo geral, conforme Kuyoro *et al.* (2012), as principais ameaças à segurança que podem acontecer em qualquer aplicativo *web* são relacionadas a injeção de SQL, ataques de captura e reprodução, estouros de *buffer*, ataques de negação de serviço, tratamento inadequado de erros, acessos de bisbilhoteiros e sequestros de sessão.

Segundo Durbano (2018), os crimes cibernéticos crescem a cada ano, cujos responsáveis tendem a realizar invasões com o objetivo de capturar informações dos sistemas das empresas para obter vantagem, sobretudo para vazar informações, sequestrar dados, promover ataques distribuídos de negação de serviço (DDoS) ou ataques de negação de serviço (DoS), etc.

Os DoS são realizados para comprometer a disponibilidade do sistema, por exemplo, ao consumir recursos de aplicativos na *web* para que outros usuários não consigam mais acessá-los ou mesmo utilizar o aplicativo. Esses ataques podem ser cometidos ao enviar solicitações com buscas que resultam em grandes quantidades de dados, quando os *hackers* bloqueiam as contas de usuários ou causam a falha de todo o aplicativo por meio da sobrecarga do serviço com um grande número de solicitações; normalmente, essas abordagens são combinadas com outros ataques específicos de *webservices* para maximizar os danos.

Em relação às ameaças que envolvem *webservices*, pelo fato de as instruções SQL serem criadas no decorrer da execução do sistema, dinamicamente, surge uma oportunidade de violar a segurança por meio da injeção de SQL nas mensagens. Quando *hackers* conseguem quebrar a segurança, podem transmitir entradas para a instrução SQL, para que estas façam parte da instrução.

Com essa brecha, os *hackers* podem utilizar técnicas para acessar dados privilegiados e áreas protegidas sem dispor das credenciais adequadas, além de manipular o banco de dados, o que acontece por meio da inserção de valores ou caracteres especiais nas solicitações SOAP, nos envios de formulários ou nos parâmetros das URL.

Outra ameaça, relacionada à captura e à reprodução, diz respeito aos ataques intermediários nas mensagens transmitidas pela internet, situação em que os *hackers* capturam, manipulam e reproduzem uma solicitação SOAP, modificando-a antes que chegue ao seu destino.

Ainda em relação à ameaça de captura de dados, existem riscos de invasores bisbilhoteiros interceptarem mensagens SOAP, frequentemente transmitidas por meio de *webservices*, para ler e roubar informações. As informações roubadas que mais causam prejuízo para os usuários são as senhas e as informações do cartão de crédito. Assim, é importante manter a transmissão com segurança para que essas pessoas não autorizadas não consigam interceptar as mensagens.

No caso de aplicativos nativos, *hackers* conseguem ameaçar servidores por meio de *webservices* que contenham mensagens com tamanho de dados de entrada não verificados, ataques que podem acontecer por meio de solicitações SOAP ou envio de formulários *web*. Nos ataques de estouro de *buffer*, os *hackers* especificam mais dados em campos para gravar do que o tamanho em memória disponível para retê-los. Essas funções criadas por *hackers* para acessar informações não autorizadas podem resultar em falhas no aplicativo, em comprometimento do sistema ou no início de processos não autorizados.

De acordo com Kuyoro *et al.* (2012), existem ainda ameaças relacionadas ao tratamento inadequado de erros. Diversos servidores retornam detalhes da aplicação quando acontece um erro interno, os quais são úteis para a equipe de desenvolvimento para depurar o código. Entretanto, quando o aplicativo está em produção, essas informações não devem chegar aos usuários comuns, tendo em vista que podem incluir informações do código do projeto e expor possíveis vulnerabilidades.

Um problema de segurança associado a esse tipo de ameaça se dá quando o sistema retorna uma mensagem de erro para um usuário mal-intencionado, informando que a consulta SQL está incorreta. Com isso, o usuário recebe a informação de que as entradas podem ser utilizadas para gerar consultas ao banco de dados e consegue realizar a injeção de SQL indesejado (*SQL injection*), como exemplificado na Figura 3.

Figura 3. Exemplo de *SQL injection*.

Fonte: Moreira (2019, documento on-line).

Outro problema possível refere-se ao momento em que o sistema informa ao usuário que a senha está errada, situação em que o criminoso pode tentar outras senhas do mesmo usuário.

Para Durbano (2018), é importante se atentar ao cenário mundial atual e às suas necessidades de segurança, o que pode evitar graves prejuízos para as empresas e para a sociedade como um todo. Para isso, tornam-se necessários planejar e implementar medidas de segurança da informação alinhadas ao negócio de modo integral, evitando muitos dos eventuais problemas e determinando os planos de contingência que podem ser aplicados caso aconteça algo errado.

Os resultados para uma empresa que não investe em segurança podem ser desastrosos, assim como quando da utilização de soluções mais econômicas e menos eficientes ou quando algumas práticas deixam de ser implementadas. Se os dados ficarem inacessíveis para os usuários, os serviços podem ficar paralisados e o processo de correção do problema levar dias para ser solucionado.

De acordo com o serviço prestado, alguns minutos podem provocar grandes prejuízos financeiros. Além do acesso aos dados, outras fragilidades de segurança da organização são capazes de determinar a confiabilidade do mercado em relação à empresa, que pode perder os clientes por conta da exposição de dados sensíveis e sofrer processos judiciais em relação a essa falha de segurança das informações.

A seguir, mostraremos algumas propostas de soluções de segurança para tráfego e armazenamento de dados, que visam a proteger os dados das pessoas e das empresas.

3 Soluções de segurança

De acordo com Kuyoro *et al.* (2012), na década de 1990 o trabalho dos profissionais de segurança era relativamente simples, já que os dados sensíveis estavam localizados em bancos de dados monolíticos, com apenas alguns poucos caminhos de acesso aos dados, protegidos por mecanismos de controle de acesso.

Por muitos anos, utilizaram-se políticas, procedimentos e ferramentas para proteger bancos de dados legados e, com o surgimento de aplicações baseadas na *web*, especialmente em relação às aplicações de comércio eletrônico que acessam servidores na internet, a segurança se tornou ainda mais importante. Com essa evolução, surgiram tecnologias que amadureceram ao longo do tempo, como o Secure Socket Layer (SSL), os *firewalls*, a autenticação e a autorização pela *web*, que fortalecem a segurança entre os navegadores dos clientes e os servidores *web* das empresas.

O SSL, uma tecnologia ou protocolo utilizado para proteger contra ataques que invadam a segurança de *webservices*, criam um túnel seguro entre as máquinas cliente e servidor com base na técnica de criptografia de chave pública, comumente empregada para o envio seguro de mensagens. Para diversos aplicativos simples, esse tipo de tecnologia pode ser o suficiente; no caso de *webservices*, as mensagens podem ser criptografadas e assinadas para proteção da confidencialidade das informações e integridade dos dados.

Assim como o SSL, existe um conjunto de tecnologias com o objetivo de adicionar segurança aos *webservices*, como a extensão do SOAP WS-Security (WSS), que deve ser utilizada preferencialmente em conjunto com outros *webservices* e protocolos específicos de aplicações. O WSS determina como os dados de assinatura XML podem ser incluídos na mensagem SOAP: as especificações XML de assinatura e criptografia fornecem métodos-padrões para a assinatura digital e a criptografia de documentos que incluam mensagens SOAP, um processo utilizado tanto para documentos inteiros quanto para partes menores.

Outra tecnologia de segurança em *webservices* consiste na criptografia por meio de código XML, uma das maneiras de fornecer segurança de ponta a ponta para os aplicativos que exigem alteração segura dos dados, além de uma forma natural de lidar com requisitos de segurança em dados trocados entre aplicações. Esse tipo de criptografia não substitui o SSL, mas provê um mecanismo para os requisitos de segurança não cobertos pelo SSL.

Outras duas técnicas baseadas em XML são a Security Assertion Markup Language (SAML) e a eXtensible Access Controle Markup Language (XACML). A primeira compreende um protocolo para declarar informações de autenticação e autorização, além de fornecer atributos de um usuário final no formato XML e permitir adicionar informações a uma mensagem SOAP. Para possibilitar o *single sign-on* (SSO), servidores SAML podem ser acessados a fim de obter dados de autenticação e autorização, como visto na Figura 4.



Figura 4. Representação esquemática do *single sign-on*.

Fonte: Adaptada de Rodrigues (2017).

Caso o destinatário da mensagem SOAP confie no emissor dos dados SAML, o usuário também poderá ser autorizado para o *webservice*.

Já a XACML foi projetada para expressar regras de controle de acesso no formato XML.

E, mesmo que as duas técnicas não estejam explicitamente ligadas, podemos usá-las em conjunto, caso em que uma decisão de autorização expressa em uma declaração SAML pode ter sido baseada em regras expressas em XACML.

Dubin (2008) afirma que existem diversas opções de segurança baseadas em XML disponíveis quando falamos de *webservices*, as quais gerenciam criptografia, controle de acesso, autenticação, integridade e privacidade de dados necessários nos *webservices*. Para o autor, além do SAML, são outras possibilidades as assinaturas digitais em XML, o XML Key Management Specification (XKMS) e o serviço de mensagens ebXML, que não serão aprofundados neste capítulo.

Segundo Durbano (2018), aplicar as práticas de segurança não é o suficiente, exigindo-se a garantia de que não haverá brechas. Para isso, é necessário aplicar as melhores ações existentes na área, o que requer um acompanhamento contínuo das tendências e das evoluções da área, possibilitando a criação de soluções de proteção para os novos mecanismos de ataque. Ainda, é preciso manter os *software* e *drivers* dos dispositivos atualizados, estabelecer o controle de acesso aos colaboradores e políticas de segurança, alinhar os processos às políticas de segurança, fazer planos de contingência e empregar ferramentas de monitoramento e criptografia dos dados.

Para Graham *et al.* (2005), as técnicas de criptografia fornecem uma base para proteção de mensagens trocadas entre parceiros. A confidencialidade e a integridade podem ser garantidas com tecnologias de criptografia e a assinatura digital, respectivamente, categorizadas entre chaves ou criptografias simétricas e assimétricas.

A criptografia simétrica exige o uso da mesma chave para o processo de criptografia e descriptografia — quando uma pessoa pretende enviar uma mensagem para outra, a mensagem é criptografada com uma chave, e, quando o destinatário a recebe, também deve ter a mesma chave para que consiga obter a mensagem original. Considerando que a mesma chave é utilizada em ambos os pontos, esse tipo de criptografia é chamado de criptografia simétrica, bem como as chaves levam em geral o nome de chaves simétricas.

A criptografia assimétrica simplifica a distribuição de chaves, tendo em vista que possibilita tornar a chave de criptografia pública. Nesse caso, utilizamos duas chaves distintas em cada uma das pontas: uma chave pública e outra chave privada. A chave pública é utilizada para criptografar a mensagem, e a única maneira de descriptografar a mensagem se dá por meio da chave privada. Pelo reflexo da sua natureza assimétrica, essas chaves são chamadas de assimétricas.

Outra técnica bastante utilizada para melhorar a segurança na troca de mensagens é a utilização de VPN, com a qual se pode criar uma rede privada sobre a infraestrutura de uma rede pública, como observado na Figura 5.

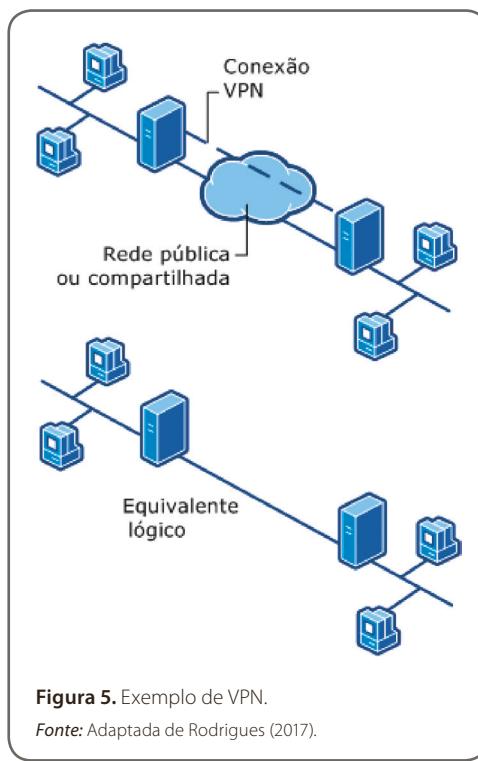


Figura 5. Exemplo de VPN.

Fonte: Adaptada de Rodrigues (2017).

Nesse tipo de rede, a criptografia nas informações e na comunicação pode aumentar a confidencialidade consideravelmente. O sistema de tunelamento permite a troca de mensagens sem que outras pessoas tenham acesso, ou que a mensagem esteja criptografada caso seja acessada.

De acordo com Cabral (2018), grandes empresas também utilizam a técnica de segmentação da rede por questões de segurança: a empresa cria compartimentos digitais e, caso uma máquina na rede ou a própria rede seja atacada, isso somente acontece no comportamento em questão, e não nos demais. Com isso, quando atacados, uma quantidade menor de dados fica exposta aos criminosos, minimizando os impactos. Além disso, também é importante proteger as redes com *firewalls* e outros mecanismos.

Neste capítulo, pudemos observar conceitos de segurança em *webservices*, além dos riscos que os dados e a comunicação sofrem, e eventuais soluções de segurança para tráfego e armazenamento de dados utilizadas para proteger os dados e a rede.



Referências

- ACOSTA, I. *Segurança da informação*. 2015. Disponível em: <https://slideplayer.com.br/slide/3550546/>. Acesso em: 11 jun. 2020.
- CABRAL, T. *Entenda a importância de segmentar a rede para mantê-la segura*. 2018. Disponível em: <https://blog.athenasecurity.com.br/segmentacao-de-rede/>. Acesso em: 11 jun. 2020.
- DUBIN, J. *What are the risks of connecting a web service to an external system via SSL?* 2008. Disponível em: <https://searchsecurity.techtarget.com/answer/What-are-the-risks-of-connecting-a-Web-service-to-an-external-system-via-SSL>. Acesso em: 11 jun. 2020.
- DURBANO, V. *Segurança da informação: o que é e 12 dicas práticas para garantir*. [2018]. Disponível em: <https://ecoit.com.br/seguranca-da-informacao/>. Acesso em: 11 jun. 2020.
- GRAHAM, S. et al. *Build web services with Java*. 2nd ed. Indianapolis: Pearson Education, 2005.
- KUYORO, S. O. et al. Security issues in web services. *International Journal of Computer Science and Network Security*, [s. l.], v.12, n.1, p. 23–27, 2012.
- MEGA SISTEMAS CORPORATIVOS. *Conheça os principais tipos de integração de sistemas*. 2018. Disponível em: <https://www mega.com.br/blog/conheca-os-principais-tipos-de-integracao-de-sistemas-9192/>. Acesso em: 11 jun. 2020.
- MOREIRA, C. *SQL injection: o que é e como funciona?* 2019. Disponível em: <https://camilamoreira.com.br/site/?p=626>. Acesso em: 11 jun. 2020.
- RODRIGUES, M. *Single Sign-On (SSO) server + laravel 5*. 2017. Disponível em: <https://medium.com/laravel-tips/single-sign-on-sso-server-laravel-5-6c7c70858c63>. Acesso em: 11 jun. 2020.

VARGAS, P. K. *Comunicação em sistemas distribuídos*. 2001. Disponível em: <https://www.cin.ufpe.br/~avmm/arquivos/provas%20software/resuminho2.pdf>. Acesso em: 11 jun. 2020.

WU, J. et al. A cross-layer security scheme of web-services-based communications for IEEE 1451 sensor and actuator networks. *International Journal of Distributed Sensor Networks*, [s. l.], v. 9, n. 3, p. 1–10, 2013.

Leituras recomendadas

MCLAUGHLIN, B. *Java & XML*. 2nd ed. Sebastopol: O'Reilly, 2001.

ORACLE. *Fusion middleware security and administrator's guide for web services*. 2020. Disponível em: https://docs.oracle.com/cd/E17904_01/web.1111/b32511/intro_ws.htm#WSSEC2935. Acesso em: 11 jun. 2020.

VARGAS, P. K. *Sistemas operacionais distribuídos e de redes*. 2000. Disponível em: <https://www.cin.ufpe.br/~avmm/arquivos/provas%20software/attachment.pdf>. Acesso em: 11 jun. 2020.



Fique atento

Os *links* para *sites da web* fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integralidade das informações referidas em tais *links*.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

