Nível de conhecimento de desenvolvedores sobre segurança em aplicações web: Pesquisa e análise

Pablo V. Costa, Willian I. Gonçalves, Eliezer D. Gonçalves, Nilson M. Lazarin

[1]Bacharelado em Sistemas de Informação – Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (Cefet/RJ) – Nova Friburgo, RJ – Brazil

{vcpablo,w.goncalves91}@gmail.com

{eliezer.goncalves,nilson.lazarin}@cefet-rj.br

Abstract. *The use of web applications grows in Brazil along with a big number of virtual attacks once this kind of application are usually more vulnerable to malicious users that intend to compromise the quality, authenticity, privacy and availability of the information. Considering this context, it is assumed that web developers must be prepared to deal with these attacks. This paper shows the result of a survey research made through an interview with web developers that tries to measure their level of knowledge about some of the main types of web application attacks and their ability to identify, analyze and correct such threats in vulnerable pieces of source code.*

Resumo. *O crescimento do uso de aplicações web no Brasil está acompanhado de um grande número de ataques virtuais uma vez que tais tipos de programas tendem a ser mais vulneráveis a usuários mal-intencionados que buscam comprometer a qualidade, autenticidade, privacidade e disponibilidade de informações. Tendo isso em mente, entende-se que os desenvolvedores web devem estar preparados para lidar com estes ataques. Este artigo apresenta o resultado de um survey realizado através da entrevista com desenvolvedores web visando mensurar seu nível de conhecimento sobre alguns dos principais tipos de ataque a aplicações web e sua capacidade de identificar, analisar e corrigir tais ameaças em trechos de código-fonte vulneráveis.*

1. Introdução

Em 2015 57,5% dos domicílios brasileiros já tinham acesso à internet e até 2021 haverá um crescimento médio de 12.4%, ao ano, das vendas realizadas pela internet, atingindo um total de R$ 85 bilhões [Oliveira 2016] [IBGE 2016]. O crescimento do acesso à internet e do comércio eletrônico no Brasil é acompanhado pelo aumento dos ataques a sites e lojas virtuais. Em 2017, a NETSCOUT Arbor registrou um total de 264.900 ataques DDoS (Ataques de Negação de Serviço) no Brasil [Digital 2018].

No mundo todo foram registrados 7,5 milhões de ataques, colocando o Brasil entre os cinco primeiros alvos deste tipo de ataque [Digital 2018]. De acordo com o estudo Norton Cyber Security Report [Norton 2017], em 2017, crimes cibernéticos que incluem, entre outros tipos, ataques a aplicações web, causaram prejuízos de US$ 22 bilhões ao Brasil, sendo que cada vítima perdeu uma média 34 horas com as consequências dos ataques.

O objetivo deste artigo é apresentar um panorama sobre o nível de conhecimento e prática dos desenvolvedores acerca da implementação de técnicas de segurança para prevenção de ataques a aplicações web. São apresentados conceitos de segurança da informação, algumas das principais ameaças a aplicações web e, ao final, uma análise baseada nos resultados de uma pesquisa realizada através de um *survey* aplicado a desenvolvedores web.

2. Revisão da Literatura

Segurança da Informação é definida como a "*proteção contra o uso ou acesso não autorizado à informação e preservação da disponibilidade, integridade e confidencialidade desta mesma informação*"[Morris and Thompson 1979]. Segundo [David Kim 2014], é possível dividir, categoricamente, a segurança da informação em três principais pilares:

- **Confidencialidade**: A informação deve poder ser visualizada somente por usuários autorizados;
- **Disponibilidade**: A informação deve estar disponível e acessível a usuários autorizados sempre que a solicitarem;
- **Integridade**: A informação deve poder ser alterada somente por usuários autorizados.

De acordo com [David Kim 2014], é possível organizar os tipos de ataques, tanto a servidores quanto a aplicações web relacionando-os com os três princípios básicos de segurança da informação:

Ameaças de Negação ou Destruição: nesta categoria, os ataques têm por objetivo tornar determinado recurso indisponível ou inutilizável e, portanto, está diretamente relacionada à quebra do princípio da disponibilidade da informação. Pode-se destacar nesta categoria o ataque *Buffer Overflow*.

Ameaças de Alteração / Divulgação: nesta categoria, os ataques visam modificar ou divulgar dados sem autorização, comprometendo o resultado final de determinada operação em um sistema, violando, portanto, o princípio da integridade e confidencialidade da informação. Nesta categoria enquadram-se os ataques *SQL Injection*, *Credential Stuffing* e *Session Hijacking*.

2.1. Buffer Overflow

Um ataque é caracterizado como Buffer Overflow quando determinada ação envia ao programa um volume de dados maior do que este está preparado para tratar ou em um formato que aplicação não consegue manipular. Isto pode resultar no corrompimento de dados, travamento do programa ou até mesmo na execução de código indesejado, hospedado nas instruções que excedem a memória alocada inicialmente pela rotina original, fazendo com que o invasor consiga executar um programa malicioso dentro do ambiente do escopo de uma aplicação ou serviço [Foster James et al. 2005].

Por definição, este tipo de ataque não acontece em linguagens interpretadas como Java, Python ou PHP. Porém, analisando o conceito a partir de outra perspectiva, pode-se assumir que determinadas falhas de segurança em aplicações web, independente da linguagem utilizada, podem tornar tais aplicações vulneráveis a consequências muito similares às de um ataque de buffer overflow. Por exemplo, se o comprimento de um texto,

submetido à aplicação pelo usuário, exceder o limite estabelecido no banco de dados, quando a aplicação não está preparada para tratar esse caso.

Essa situação pode ser evitada através da implementação de uma validação, tanto no lado do cliente quanto no lado do servidor, de acordo com algumas regras de negócio, a fim de assegurar que os dados serão persistidos apenas se obedecerem à tais regras e recusados caso contrário.

2.2. SQL Injection

SQL Injection é um ataque no qual código SQL é inserido ou concatenado em parâmetros de entrada enviados pelo usuário à aplicação e, imediatamente após, enviado para um servidor back-end para interpretação e execução [Clarke 2012]. A injeção de SQL em uma aplicação pode trazer graves consequências a uma entidade ou organização, tendo em vista que operações podem ser realizadas diretamente na base de dados.

São diversas as formas de se realizar SQL Injection em uma aplicação. Segundo [Stephanos Mavromoustakos 2016] existem nada mais que doze diferentes formas de executar este tipo de ataque, dentre eles as chamadas tautologias que "*injetam código em um ou mais comandos condicionais que sempre são resolvidos como verdadeiro*" e que "*são utilizadas para burlar uma página de autenticação e extrair dados*". Por esse motivo, este ataque pode ser classificado também como uma ameaça de divulgação e destruição.

Basicamente, uma aplicação web se torna vulnerável ao ataque de SQL Injection quando o desenvolvedor não se preocupa em validar, tratar e encapsular os valores de entrada, informados pelos usuários, e que servirão de parâmetros para consultas ao banco de dados.

2.3. Credential Stuffing

Ataques do tipo *Credentials Stuffing* se caracterizam pela obtenção de acesso com um usuário válido, a partir da tentativa de descoberta dos dados de autenticação do mesmo, de forma fraudulenta, realizada por scripts de execução de força bruta [Rahman and Tomar 2018].

Normalmente o invasor realiza tentativas de acesso ao sistema de forma automatizada, utilizando inúmeras combinações de usuário e senha até que uma delas seja válida. Muito embora somente 0.2% das tentativas tenham chances de ser bem-sucedidas, caso isso aconteça, os dados do usuário ficam à mercê do invasor, além da possibilidade de que as credenciais roubadas possuam acesso de nível elevado no sistema sob ataque, o que potencializa as consequências dessa falha de segurança.

Existem inúmeras técnicas que, combinadas, podem mitigar o ataque de Credentials Stuffing. Uma delas é registrar e limitar o número de tentativas com credenciais inválidas e um IP, independente do navegador [OWASP 2017a].

2.4. Session Hijacking

Christian Schifflet define este ataque como um "*método que um atacante pode utilizar para acessar a sessão de um outro usuário*"[Schfillet 2009]. Em sua definição mais básica, uma sessão é um período de tempo acordado em que o estado conectado do cliente e do servidor é vetado e autenticado. Isso significa que, tanto o servidor quanto o cliente,

sabem (ou pensam que sabem) quem são, e com base nesse conhecimento, eles podem confiar que os dados enviados de qualquer forma acabarão nas mãos da parte apropriada.

De acordo com a [OWASP 2014], um ataque de Session Hijacking caracteriza-se pela exploração de brechas de segurança no mecanismo de sessões de uma aplicação web e comprometem o *token* da sessão ao roubar ou forjar um código válido para ganhar acesso não autorizado ao servidor web.

Para prevenir formas comuns de Session Hijacking, alguns cuidados básicos podem ser tomados quando se está programando do lado do servidor. Dentre eles, a ativação do *HttpOnly Flag* no momento da criação de um cookie no servidor, garantirá que o mesmo não poderá ser manipulado via JavaScript, após ser gravado no navegador do cliente. Desta forma, uma injeção de código JavaScript malicioso (*Cross-Site Scripts*) não poderá modificar o conteúdo dos cookies [OWASP 2017b].

3. Abordagem Metodológica

Para realizar a pesquisa quantitativa e qualitativa, utilizou-se um *survey* como ferramenta de entrevista de forma on-line, contendo perguntas fechadas e abertas. O público-alvo foi composto por desenvolvedores web alcançados por meio de grupos em redes sociais, listas de e-mail da SBC e contatos dos autores que atuam no mercado de trabalho. Optou-se por uma construção cuidadosa do *survey* contendo um misto de perguntas de múltipla escolha com perguntas dissertativas de modo ampliar as possibilidades de análise das respostas. Não foi realizada análise estatística. O *survey* foi dividido em cinco seções, conforme descrito abaixo:

Seção 1: Aproxima o entrevistado através de perguntas destinadas a identificar seu perfil educacional e profissional. Dessa forma, perguntou-se sobre: formação acadêmica; tempo de experiência em desenvolvimento web; linguagens de programação com maior intimidade e conhecimento.

Seção 2: Indaga sobre conhecimento de técnicas de implementação de segurança em aplicações web e a relação do entrevistado com estas técnicas no que diz respeito ao seu prévio conhecimento e opinião sobre possíveis motivos da não implementação de tais técnicas.

Seção 3: Apresenta quatro ameaças a aplicações web (*Buffer Overflow*, *SQL Injection*, *Credentials Stuffing* e *Session Hijacking*), realizando uma breve explicação destas e de como preveni-las. Nesta etapa o entrevistado é indagado sobre a ciência destes ataques com base em sua experiência e conhecimento.

Seção 4: Desafia o entrevistado a relacionar alguns trechos de códigos vulneráveis (escritos na linguagem de programação de domínio do entrevistado, informada na seção 1) às ameaças apresentadas na seção 3. Pode-se observar na Figura 1 o *survey* na linguagem PHP. A partir do momento em que o entrevistado declara conseguir identificar a vulnerabilidade no trecho de código, requere-se que o mesmo proponha melhorias, via comentário ou exemplo de código, para o trecho de código vulnerável apresentado.

Seção 5: Solicita a análise do entrevistado quanto a relevância das perguntas e desafios respondidos até então, bem como sua clareza. Nesta seção o entrevistado ainda é questionado sobre: outras ameaças que poderiam ter sido abordadas neste estudo; se o mesmo foi preparado para enfrentar situações que demandam conhecimento

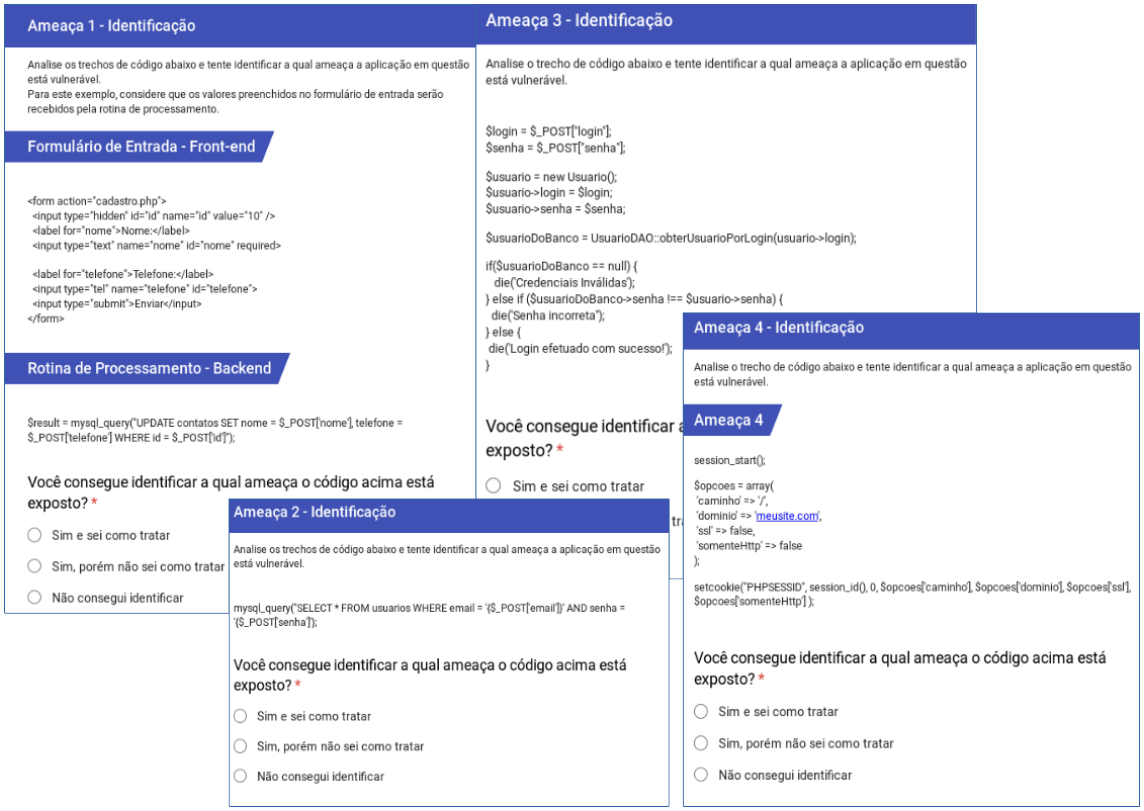


Figura 1. Perguntas sobre identificação de vulnerabilidades em linguagem PHP

sobre técnicas de implementação de contra-ataques web; e sua avaliação no que diz respeito à preocupação da empresa onde atua e de respectivos clientes quanto a importância da segurança em projetos de desenvolvimento.

Como ameaças à validade, é importante considerar que, por se tratar de uma abordagem 100% on-line é possível que os resultados aqui apresentados não representem fielmente a realidade, considerando a possibilidade de consulta a diversos meios para a obtenção de respostas, bem como a possibilidade de alteração de respostas fornecidas a perguntas já respondidas durante a execução do *survey*, dada as limitações da ferramenta utilizada. Uma vez enviado o formulário, o entrevistado não foi capaz de responder às perguntas novamente ou alterar as respostas enviadas.

4. Resultados

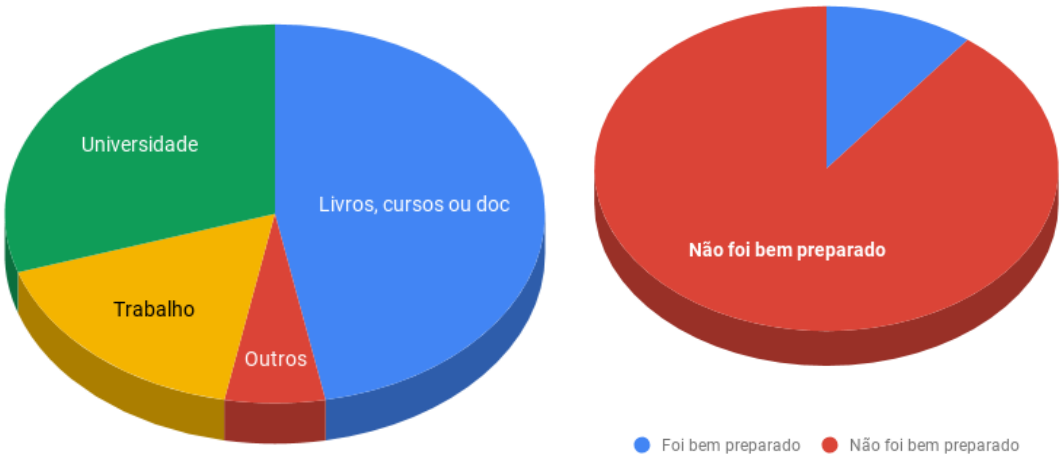
Os resultados obtidos do *survey*, detalham o cenário atual da comunidade desenvolvedora no Brasil através de uma pequena amostra de entrevistados do ramo. Foram obtidas 60 respostas da pesquisa. Dentre os entrevistados, 38% (23) tinham mais de 10 anos de experiência, 15% (9) tinham entre 6 e 10 anos de experiência e 20% (12) tinham entre 3 e 6 anos de experiência. Portanto, 73% das respostas foram obtidas de profissionais do nível pleno ou sênior.

Em relação à formação acadêmica, 57% (34) afirmaram ter nível superior. Pôde-se concluir através de uma análise quantitativa que:

- Os maiores motivos para a não implementação de técnicas de prevenção de ataques são: falta de conhecimento, assinalada em (92%) das respostas; prazos muito

apertados, assinalada em (70%) das respostas e baixa prioridade assinalada em (47%) das respostas, conforme 2(c);

- De acordo com 90% dos entrevistados, a formação acadêmica não fornece uma preparação adequada para lidar com segurança, conforme Figura 2(b). Um total de 50% dos entrevistados considera o ensino insuficiente, enquanto que 37,5% afirmam que o assunto não foi abordado em sua formação acadêmica;
- Entre os que afirmam conhecer sobre implementação de técnicas de segurança, 47% aprenderam através de livros, cursos ou documentações oficiais, enquanto 17% aprenderam no exercício da profissão. Apenas 30% afirmaram ter aprendido durante a formação acadêmica conforme Figura 2(a).



(a) Meios de aprendizado sobre implementação de técnicas de segurança

(b) Acreditam ter sido bem preparados na formação acadêmica

(c) Principais motivos para a não implementação de segurança

Figura 2. Resultados quantitativos

Além disso, a tabulação dos dados da parte qualitativa do estudo, composta por

perguntas abertas, exibe também dados significativos quanto ao conhecimento dos entrevistados em relação a brechas de segurança comuns em código já escrito em sua linguagem de programação mais conhecida. Através da apresentação de trechos de código que, propositalmente, continham falhas de segurança, foi perguntado se o entrevistado identificava essas falhas e os ataques que poderiam explorá-las, conforme Figura 1. A seguir, foi solicitada uma proposta de melhoria, que pudesse eliminar a brecha na segurança. Os resultados mostraram que:

- Ataques de Buffer Overflow foram facilmente identificados. Porém, as soluções propostas foram mais aplicáveis a ataques de SQL Injection, e não ao ataque de Buffer Overflow;
- Ataques de SQL Injection foram facilmente identificados e as soluções para tratá-lo se mostraram bem conhecidas;
- Ataques de Credential Stuffing não foram facilmente identificados. Além disso, poucas das soluções propostas foram realmente eficazes;
- Ataques de Session Hijacking também não foram facilmente identificados. Porém, a grande maioria das soluções propostas foram eficazes.

Tabela 1. Identificação de ameaças

	Buffer Overflow	SQL Injection	Credential Stuffing	Session Hijacking
Não identificou	4%	11%	30%	39%
Identificou	96%	89%	70%	61%
Identificou e corrigiu corretamente	10%	69%	10%	62%
Identificou, mas corrigiu errado	78%	18%	75%	7%
Identificou, mas não tentou corrigir	12%	13%	15%	31%

5. Considerações Finais

Após a análise dos resultados, verificou-se que grande parte dos entrevistados afirma não ter sido bem preparado em sua formação acadêmica para lidar com ameaças a aplicações web e que a grande maioria dos desenvolvedores entrevistados aprendeu a implementar técnicas de segurança e a se preocupar com estes aspectos no próprio mercado de trabalho com a prática do dia a dia.

Foi possível identificar também que, dentre os tipos de ataque abordados na pesquisa, o *Credential Stuffing* foi, assim como *Session Hijacking*, o tipo de ataque menos identificado e, dentre estes dois, este foi o que teve menos propostas de melhoria. Já o *SQL Injection* se mostrou ser o mais conhecido da maioria dos entrevistados. Tanto que, na análise de código vulnerável a *Buffer Overflow*, algumas propostas de melhoria levaram também em consideração a prevenção de *SQL Injection*, por se tratar de uma situação bastante próxima em termos de implementação.

Possíveis trabalhos futuros podem vir a investigar a existência de disciplinas voltadas a segurança de aplicações web na grade dos cursos voltados a sistemas de informação nas universidades brasileiras; pode-se ainda, analisar o nível de conhecimento de programadores web sobre outros tipos de ameaças; por fim, até mesmo a criação de um objeto de inspeção de código visando identificar brechas de segurança como ferramenta de auxílio a desenvolvedores.

Referências

Clarke, J. (2012). *SQL Injection Attacks and Defense*.

David Kim, M. G. S. (2014). *Fundamentos de Segurança de Sistemas de Informação*.

Digital, C. (2018). Brasil foi alvo do maior ataque DDoS do mundo. http://www.convergenciadigital.com.br/cgi/cgilua.exe/sys/start.htm?UserActiveTemplate=site&infoid=47296&sid=18.

Foster James, C., Vitaly, O., Nish, B., and Niels, H. (2005). Buffer overflow attacks: Detect, exploit, prevent.

IBGE, C. d. T. e. R. (2016). *Pesquisa nacional por amostra de domicílios : síntese de indicadores 2015*. IBGE.

Morris, R. and Thompson, K. (1979). Password security: A case history. *Communications of the ACM*, 22(11):594–597.

Norton (2017). Norton Cyber Security Insights Report global Results. http://now.symassets.com/content/dam/norton/global/pdfs/norton_cybersecurity_insights/NCSIR-global-results-US.pdf.

Oliveira, F. (2016). Vendas na internet no Brasil devem dobrar até 2021, indica Google. http://www1.folha.uol.com.br/mercado/2016/10/1823568-vendas-na-internet-devem-dobrar-ate-2021-indica-google.shtml.

OWASP (2014). Session hijacking attack. https://www.owasp.org/index.php/Session_hijacking_attack.

OWASP (2017a). Credential Stuffing Prevention Cheat Sheet. https://www.owasp.org/index.php/Credential_Stuffing_Prevention_Cheat_Sheet.

OWASP (2017b). HttpOnly. https://www.owasp.org/index.php/HttpOnly.

Rahman, R. U. and Tomar, D. S. (2018). *Security Attacks on Wireless Networks and Their Detection Techniques*, pages 241–270. Springer Singapore, Singapore.

Schfillet, C. (2009). *Essential PHP Security*. O'Reilly Media.

Stephanos Mavromoustakos, Aakash Patel, K. C. P. C. S. P. (2016). Causes and prevention of sql injection attacks in web applications. *University of Windsor*.