ANÁLISE DA SEGURANÇA DA INFORMAÇÃO EM UM FRAMEWORK PARA DESENVOLVIMENTO DE VOLUNTEERED GEOGRAPHIC INFORMATION

Sivoney Pinto Dias[1], Fábio de Oliveira Sales[2], Helder Guimarães Aragão[3]

[1] Especialista em Governança de TI pela UNIASSELVI, Técnico em Programação e Suporte de TI, EMBASA, Salvador, BA, sivoneypdias@gmail.com
[2] Especialista em Engenharia de Software pela faculdade Ruy Barbosa Devry, Analista de Saneamento, EMBASA, Salvador, BA, fosales@gmail.com
[3]Mestre em Sistemas e Computação, Professor Adjunto, Centro Universitário Estácio da Bahia, Salvador, BA, helderaragao@gmail.com

RESUMO: *Volunteered Geographic Information* confere aos sistemas Web características colaborativas que envolvem dados espaciais ou geográficos. Este tipo de contribuição voluntária tem crescido significativamente nos últimos anos. Em função da crescente demanda por sistemas Web com VGI, foram desenvolvidos alguns *frameworks* com o objetivo de facilitar a implementação deste tipo de sistema. Uma característica fundamental, que estes *frameworks* deveriam levar em consideração em sua implementação, é a segurança da informação. Neste contexto, o presente artigo visa avaliar o nível de segurança da informação de um destes *frameworks* disponíveis para a implementação de sistemas Web com VGI. O *framework* avaliado foi o ClickOnMap. A avaliação foi feita em etapas e com ferramentas específicas da área de segurança da informação. No final do artigo, são apresentados os resultados desta avaliação.

PALAVRAS-CHAVE: clickonmap, OWASP, VGI

INTRODUÇÃO: A produtividade de uma empresa está, atualmente, fortemente ligada ao uso da Internet. Em razão disso, a dependência das organizações pelos recursos tecnológicos em ambiente Web tem aumentado significativamente. Para manter a continuidade nos negócios, o mundo corporativo precisa proteger suas informações. A informação é um ativo valioso e isso desperta o interesse de vários agentes ameaçadores em obtê-la de forma ilícita. Existem diversas ferramentas e aplicações utilizadas na Web, que contêm falhas de segurança. Estas falhas podem estar nos recursos computacionais ou, até mesmo, na implementação dos sistemas (GHODDOSI, 2012). Estes problemas de vulnerabilidade representam danos e prejuízos às empresas. Os prejuízos que os atacantes provocam nem sempre são financeiros. Uma organização, que depende de voluntários engajados na contribuição de determinado assunto de um *site*, pode ter a sua reputação devastada, caso não sejam tomadas medidas mínimas de segurança (OWASP, 2016). Recentemente, com o surgimento da Web 2.0, que torna o usuário final um produtor de conteúdo, a preocupação com a segurança da informação exerce um papel fundamental. Neste contexto, o presente artigo aborda a segurança da informação em ambientes de contribuição Voluntária de Informação Geográfica ou *Volunteered Geographic Information* (VGI). Os aspectos de segurança foram avaliados em um *framework* chamado ClickOnMap, que é voltado para a construção de sites colaborativos com informações geográficas. O objetivo deste trabalho, portanto, é analisar os riscos de segurança em aplicações Web de mapeamento colaborativo, bem como apresentar algumas ferramentas disponíveis para que o desenvolvedor possa identificar se um determinado *framework* para o desenvolvimento de VGI é efetivamente seguro.

MATERIAL E MÉTODOS: Segundo Goodchild (2007), um grande número de pessoas utiliza a Web para criar, reunir e disseminar informação geográfica de forma voluntária. Wikimapia e OpenStreetMap são exemplos de sites que possibilitam a criação de mapas colaborativos pelos usuários. Estes sites foram criados com uma estrutura que permite ao usuário descobrir lugares e editar o mapa marcando pontos, atribuindo-os informações pertinentes. A Web 2.0 introduziu essa forma de contribuição, tornando o usuário final, além de consumidor, um produtor de informações. Esta nova forma de interação chegou aos Sistemas de Informações Geográficas (SIG), que são sistemas capazes

de manipular dados geográficos. Goodchild (2007) chamou este fenômeno especial de geração de conteúdo geográfico na Web, produzido pelo usuário, de Informação Geográfica Voluntária ou *Volunteered Geographic Information* (VGI). Para o desenvolvimento de um site VGI, é necessário o uso de alguns recursos tecnológicos, destacando-se as *geotags*. As *geotags* permitem ao usuário inserir informações em um determinado ponto do mapa.

Visando auxiliar o desenvolvimento dos sites VGI com recursos de *geotags* (Figura 1), alguns *frameworks* foram desenvolvidos. Dentre estes *frameworks*, pode-se destacar o ClickOnMap. O ClickOnMap foi criado para auxiliar no desenvolvimento de características VGI em um *geobrowser*, baseado na API do GoogleMaps e no modelo DM4VGI (*Dynamic Metadata for VGI*) (SOUZA et al., 2014).

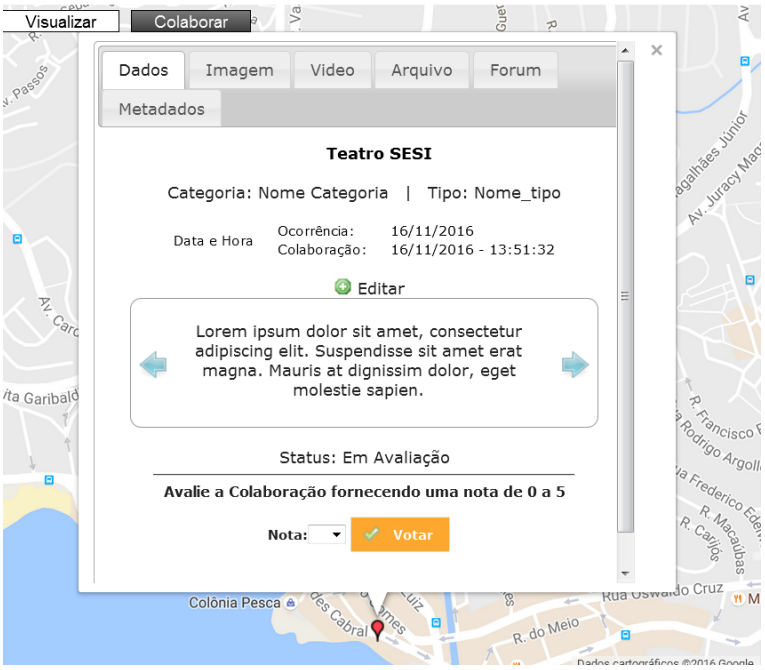

Figura 1– Exemplo de uma *geotag* no *geobrowser* criado a partir do ClickOnMap.

O ClickOnMap pode ser personalizado para coletar informações relacionadas com a contribuição do usuário de acordo com algumas categorias e tipos de assuntos informados pelo colaborador. Os assuntos são diversos, passando por questões urbanas (infraestruturas, segurança e entretenimento) e ambientais (desastres, incêndios florestais e inundações). O sistema permite aceitar uma colaboração anônima ou identificada, dependendo da política de uso do *site*. Por exemplo, em um sistema que coleta alerta sobre crimes, o voluntário pode fazer uma denúncia de forma anônima. Por outro lado, em um sistema de avaliação da qualidade de atendimento em um determinado estabelecimento, o usuário pode ter sua colaboração identificada (SOUZA et al., 2014).

Para a construção de um VGI é fundamental implementar a segurança de informação. Isto porque colaborações indevidas, ou identificações de colaborações com erros, podem causar transtornos no uso do *site* VGI. Segundo Fernandes e Abreu (2008), o *software* inseguro, construído sem critérios mínimos de segurança, aumenta os riscos de ataques diversos. Com o objetivo de alertar desenvolvedores, gestores e organizações sobre as consequências das mais importantes vulnerabilidades de segurança de aplicações Web, a Fundação OWASP criou o Projeto Top 10. *Open Web Application Security Project* (OWASP) é uma comunidade aberta, dedicada a capacitar as organizações a desenvolver, adquirir e manter aplicações confiáveis. Ela disponibiliza gratuitamente normas e ferramentas de teste de segurança (OWASP, 2016).

O impacto que os atacantes podem causar aos sistemas Web e, em particular, aos *sites* com VGI é muito grande (OWASP, 2016). Pode-se imaginar os transtornos causados por uma manipulação de

dados ilícita em uma ferramenta como o *Waze*, que possui informações de trânsito. Um invasor pode gerar alertas fictícios referentes aos engarrafamentos ou incluir rotas falsas, expondo o usuário a diversos riscos de segurança. No ano de 2015, foi noticiado que os policiais de Miami, nos Estados Unidos, incluíram informações falsas sobre radares e *blitzes* no *Waze* (EXAME, 2017).

Algumas ferramentas foram desenvolvidas visando auxiliar os testes de segurança em aplicações. Pode-se destacar o SQLMap, que é uma ferramenta de código aberto para teste de penetração automatizado. Ela permite detectar e explorar falhas de injeção de SQL. Um ataque de injeção de SQL, ou SQL *injection* em Inglês, consiste em inserir um comando SQL no sistema Web, por parte do usuário, através de um campo de dados. Com a execução desse tipo de exploração, o atacante consegue obter dados confidenciais do banco de dados, modificar os dados e executar operações na condição de um administrador no banco de dados (SQLMAP, 2016).

Neste artigo, apresenta-se o resultado da análise da segurança da informação do ClickOnMap. Esta análise foi executada em quatro etapas. Na primeira etapa, efetuou-se a configuração do ambiente de testes através do tutorial disponibilizado no site do projeto com o nome de “Site VGI” (CLICKONMAP, 2016). Na segunda etapa, realizou-se o mapeamento das vulnerabilidades através da ferramenta automatizada Zed Attack Proxy (ZAP), que é uma das soluções gratuitas de segurança da informação desenvolvidas pela Fundação OWASP. A ZAP é mantida ativamente por centenas de voluntários internacionais (OWASP, 2016). Basicamente, a ferramenta ZAP realiza uma busca por falhas de segurança em uma determinada aplicação Web, seguindo a categorização dos riscos segundo o OWASP TOP 10.

Na terceira etapa, utilizou-se os dados contidos no relatório gerado pela ferramenta ZAP Proxy, especificamente da categoria “A1 - Injeção de Código”, que corresponde à inserção de código para executar alguma instrução não desejada pela aplicação. Este relatório possui os endereços e parâmetros vulneráveis a ataques. De posse dessas informações, foi possível executar o SQLMap para o endereço do ambiente de testes.

Por fim, na última etapa, após utilizar as opções do SQLMap para descobrir qual o tipo e o nome do banco de dados, bem como os nomes de tabelas, foi executada a instrução a seguir visando obter todos os dados da tabela “usuário” do banco de dados chamado “clickonmap”:

sqlmap -u "http://nomedoservidor/sitevgi/autentica_outros.php?login=anonimo6@anonimo.com.br" --batch --dump -T usuario -D clickonmap

RESULTADOS E DISCUSSÃO: De acordo com a definição descrita sobre os dez riscos de segurança em aplicações no OWASP TOP 10 - 2013, “as falhas de injeção de SQL ocorrem quando dados não confiáveis são enviados para um interpretador como parte de um comando ou consulta”.

Uma das formas de descobrir as falhas de Injeção de SQL é examinando o código fonte da aplicação. O objetivo é verificar se instruções com dados não confiáveis poderão ser enviadas ao interpretador por meio do navegador (OWASP, 2016). Na Figura 2, pode-se observar um trecho de código do ClickOnMap, no qual um atacante pode inserir, via requisição HTTP (*Hypertext Transfer Protocol*), um valor para o parâmetro ‘login’ não previsto pela aplicação.

```
// trecho de código retirado do arquivo autentica_outros.php
$query = "SELECT * FROM usuario WHERE endEmail = '$login' ";
$result = mysql_query($query, $connection);
```

Figura 2–Trecho do código do ClickOnMap que permite o recebimento dados não confiáveis na construção da consulta SQL

Uma alternativa para corrigir a falha de segurança mostrada na Figura 2 é a utilização de consultas dinâmicas com a implementação de instruções preparadas (*Prepared Statements*). Desta forma, não é possível enviar dados ao interpretador via endereço de página no navegador, evitando a injeção de SQL (W3SCHOOLS.COM, 2017).

Na Figura 3, está um exemplo do trecho de código minimizando os riscos da vulnerabilidade apontada no trecho de código da Figura 2.

```
// $pdo é uma instância da classe PDO de acesso ao banco de dados
$query = "SELECT * FROM usuario WHERE endEmail = :login";
$stmt = $pdo->prepare($query);
$stmt->bindParam(':login', $login);
$ result = $stmt->execute();
```

Figura 3 –Exemplo do uso de *Prepared Statements* visando evitar o SQL *injection*.

Outra forma de encontrar falhas de injeção de código é realizando uma busca dinâmica automatizada, através de ferramentas como ZAP Proxy e *Skipfish* (SKIPFISH, 2017). A Figura 4 mostra um destaque do resultado da execução de uma instrução para recuperar todos os dados da tabela usuário do "Site VGI". Pode-se observar que algumas informações sensíveis podem ser obtidas: os nomes e os respectivos e-mails dos voluntários do sistema colaborativo. Desta forma, o atacante teria uma lista de contatos de email, que poderia ser utilizada facilmente para divulgação de e-mails falsos ou *spams*.

```
[14:38:47] [INFO] retrieved: "classeUsuario","int(11)"
[14:38:47] [INFO] fetching entries for table 'usuario' in database 'clickonmap'
[14:38:47] [INFO] the SQL query used returns 50 entries
[14:38:47] [INFO] retrieved: "1","1","admin@sitevgi.com","26 - 65","Administrador","0.00","e10adc3949
[14:38:48] [INFO] retrieved: "2","2","sivoneypdias@gmail.com","26 - 64","Sivoney Dias","15.00","e10ad
[14:38:48] [INFO] retrieved: "2","3","fosales@gmail.com","26 - 64","Fábio Sales","5.00","e10adc3949ba
[14:38:48] [INFO] retrieved: "2","4","helderaragao@gmail.com","26 - 64","Helder Aragão","15.00","3a5e
[14:38:48] [INFO] retrieved: "2","5","anonimo5@anonimo.com.br","26 - 64","Anonimo5","15.00","3a5ef988
[14:38:48] [INFO] retrieved: "2","6","anonimo6@anonimo.com.br","Não informado","","0.00","","C"
```

Figura 4 –Resultado do escaneamento feito com o SQLMap.

O atacante com a lista de e-mails pode, ainda, disseminar golpes na Internet. O e-mail é um dos principais caminhos para tentar induzir uma pessoa a acessar páginas com mensagens contendo links para códigos maliciosos, que redirecionam para *sites* falsos de comércio eletrônico ou *Internet Banking*. Além disso, uma lista de e-mail pode ser utilizada para divulgação de boatos indesejáveis (CERT.BR, 2016).

CONCLUSÕES: Os sistemas Web que implementam VGI compartilham informações associadas a uma posição geográfica, impondo novas formas de colaboração na Internet. Em razão disso, alguns *frameworks* foram desenvolvidos visando facilitar a construção de *sites* com VGI. Entretanto, observou-se neste artigo que um *framework* específico para desenvolvimento de sites VGI, chamado ClickOnMap, não levou em consideração aspectos de segurança da informação em sua implementação. Vale ressaltar que falhas em segurança de informação podem causar danos irreparáveis aos sistemas colaborativos.

Finalmente, pode-se observar nos resultados deste trabalho, que características básicas de segurança da informação encontradas em Sistemas de Informação tradicionais não foram desenvolvidas no ClickOnMap. Espera-se que este trabalho possa servir de alerta aos desenvolvedores de *frameworks* ou tecnologias para a construção de VGI.

Para trabalhos futuros ficam: *i*) avaliar outras questões de segurança da informação no ClickOnMap como, por exemplo, falhas na gestão da sessão do usuário e *ii*) avaliar aspectos de segurança da informação em outros frameworks para VGI.

REFERÊNCIAS:

CERT.BR: cartilha de segurança para internet. Disponível em: <http://cartilha.cert.br/seguranca/>. Acesso em: 31 outubro 2016.

CLICKONMAP. Disponível em: <http://www.dpi.ufv.br/projetos/clickonmap/>. Acesso em: 17 outubro 2016.

EXAME. Disponível em: <http://exame.abril.com.br/tecnologia/policiais-enchem-waze-de-informacoes-falsas-para-tentar-despistar-motoristas/>. Acesso em: 23 maio 2017.

FERNANDES, Aguinaldo Aragon; ABREU, Vladimir Ferraz de. **Implantando a governança de TI: da estratégia à gestão dos processos e serviços**. 2 ed. Rio de Janeiro: Brasport, 2008.

GHODDOSI, Nader. **Gestão da segurança da informação**. Indaial: Uniasselvi, 2012.

GOODCHILD, Michael Frank. **Citizens as Sensors: the world of volunteered geography**. GeoJournal, v. 69, p. 211-221, 2007.

OWASP Top 10 - 2013 Brazilian Portuguese. Disponível em: <https://storage.googleapis.com /google-code-archive-downloads/v2/code.google.com/owasptop10/OWASP_Top_10_-_2013 _Brazilian_Portuguese.pdf>. Acesso em: 17 outubro 2016.

SKIPFISH - Google Code Archive. Disponível em <https://code.google.com/archive/p/skipfish/>. Acesso em: 22 maio 2017.

SOUZA, Wagner Dias de; LISBOA-FILHO, Jugurta; CÂMARA, Jean Henrique de Sousa; VIDAL FILHO, Jarbas Nunes; OLIVEIRA, Alcione de Paiva. **ClickOnMap: A Framework to Develop Volunteered Geographic Information Systems with DynamicMetadata**. ICCSA 2014, Part II, LNCS 8580, pp. 532–546, 2014.

SQLMap: automatic SLQ injection and database takeover tool. Disponível em: <http://sqlmap.org/>. Acesso em: 31 outubro 2016.

W3SCHOOLS - PHP Prepared Statements. Disponível em: <https://www.w3schools.com/php/php_mysql_prepared_statements.asp>. Acesso em: 20 maio 2017.

WAZE - Aplicativo gratuito de trânsito e navegação baseado em mapas da comunidade. Disponível em: <https://www.waze.com/pt-BR>. Acesso em: 15 maio 2017.