Detecção de Desfiguração de Sites por Similaridade em Imagens

Henry Mendes de Jesus [1], José Eduardo Malta de Sá Brandão[2]

[1]UNICEUB – Brasília – DF – Brasil

[2] Instituto de Pesquisa Econômica Aplicada - IPEA – Brasília – DF – Brasil.

{hmjbetah@gmail.com,je.brandao@ipea.gov.br}

Abstract. *This paper proposes a new approach for web page defacement detection. It's based on image analysis, identifying the symmetry between the initial state of the web page and the current state. This work differs from the others, due to its simplicity, maintaining effectiveness.*

Resumo. *Esse trabalho propõe uma nova abordagem de detecção de desfiguração de páginas web, baseada na análise de imagens, identificando a simetria entre o estado inicial da página web e o estado atual. Esse trabalho difere dos demais, devido à sua simplicidade, mantendo a eficácia.*

Introdução

Um dos ataques mais comuns na Internet é a desfiguração de página web, que de acordo com Kanti et al. (2011), "ocorre quando o intruso de uma forma maliciosa altera a página Web substituindo o seu conteúdo com material ofensivo ou provocativo". Além de ter seu sistema web comprometido, a reputação da empresa é afetada, causando muitas vezes prejuízos financeiros. Portanto, é de suma importância que haja mecanismos apropriados para detecção de ataques deste tipo.

Nesse artigo é proposto um novo mecanismo de detecção de ataques de desfiguração de páginas web, baseado na comparação de imagens obtidas a partir do sítio web monitorado.

A seguir é apresentada uma revisão da literatura contendo as principais técnicas de detecção de ataques e os principais trabalhos relacionados à proposta. A terceira seção descreve o mecanismo de detecção por similaridade de imagens, sua implementação e os resultados obtidos. Na última seção são relatadas as conclusões.

2. Técnicas de Detecção

Existem diversas maneiras de detectar desfiguração de websites, cada qual com um grau de eficiência. A seguir é apresentado um resumo das principais técnicas de detecção presentes na literatura.

2.1. Comparação por *Hash*

Uma das técnicas mais utilizadas é a **comparação por *hash.*** Segundo Oriyano (2014), o *hashing* permite que qualquer mudança nos dados seja facilmente detectada, mesmo que em uma pequena proporção, pois o resultado será um *hash* completamente diferente do original. Um exemplo desse tipo de técnica pode ser observado no trabalho de Kanti et al. (2011). O artigo apresenta uma forma de detectar a desfiguração de páginas através de comparação de *hashes* das páginas web obtidos por meio dos algoritmos MD5 ou SHA. A proposta de Kanti et al. (2011) avalia somente do conteúdo de texto, ignorando as imagens contidas na página web. A proposta apresentada nesse artigo também usa a comparação por *hash* para analisar texto e imagens de forma única e simplificada.

2.2. Árvore DOM (HTML)

O Modelo de Objetos de Documento (*DOM - Document Object Model*) é análogo a uma árvore de objetos que compõem uma página HTML (Philippe et al. 2017). Também é uma biblioteca que permite aos desenvolvedores manipular a estrutura de um documento de hipertexto. Uma forma de conseguir detectar uma mudança, em uma página de Internet, é armazenar a estrutura e os objetos, comparando com uma versão recente. Uma varredura é feita sentido *top-down* na Árvore DOM, verificando cada objeto entre os dois arquivos. No final do processo podem ser definidas métricas, como número de elementos alterados, para verificar se houve realmente uma desfiguração.

Um exemplo de aplicação dessa técnica está no trabalho de Kumar, Rishi e Singh (2013), que teve como objetivo detectar mudanças nos componentes de uma página web. A solução proposta extrai componentes que compõem uma página web e verifica a integridade deles comparando suas versões originais. Eles ficam armazenados em um repositório de componentes web. O mecanismo é composto por quatro módulos, Sistema de Monitoração, Extração de Componentes web, Pré-processamento de Textos e Imagens, Verificação de Integridade e Repositório de Componentes web. Seu modo de detecção de páginas desfiguradas é a forma de uso da técnica de *polling*. Essa técnica é empregada periodicamente em certos grupos de dados para verificação de mudanças. Dessa forma, é verificada a integridade dos dados com algoritmos de *hash*. A proposta de Kumar, Rishi e Singh (2013) precisa armazenar muitos dados sobre os objetos monitorados, divergindo da proposta do presente artigo, que necessita armazenar poucos dados, tornando a comparação mais simples.

2.3. Comparar a diferença das requisições HTML

Outra técnica consiste em comparar o conteúdo HTML de duas requisições ao mesmo site. Como pode haver conteúdo dinâmico em cada requisição, deve-se estabelecer um limite de aceitação de mudanças. Um exemplo dessa técnica é apresentado por Verma e Sayyad (2015) a fim de impedir que uma página desfigurada seja entregue ao cliente. Para isso, eles utilizaram o servidor web Apache. O mecanismo utiliza a função *handler* do Apache para receber o conteúdo da requisição HTTP, gerando um *hash* para os arquivos relacionados ao website no servidor, que é armazenado em um banco de dados. Os *hashes* armazenados do banco de dados são comparados com os gerados em tempo de execução. Da mesma forma que o trabalho de Kanti et al. (2011), a técnica não avalia alterações em imagens.

2.4. Detecção de páginas desfiguradas em um Sistema Especialista

Nesse artigo, para resolver o problema da alta taxa de falsos positivos, os Davanzo et al. (2011) propõem uma forma Sistema Especialista. Na etapa de treinamento foram analisados três meses de amostras de 320 tipos de desfiguração de páginas web coletadas do site ZoneH.org. Dessa maneira, foi desenvolvido um protótipo de framework para realizar o processo de detecção. Em seu modo de operação, a URL do site a ser detectado é utilizada como entrada para diversos tipos de sensores, que capturam e classificam os elementos da página web na forma de vetores com valores numéricos. Esses vetores servem de entrada para a comparação utilizando a base de conhecimento.

2.5. Palavras Comuns

Quando ocorre um caso de desfiguração de páginas é comum que os atacantes deixem assinaturas. Geralmente com termos conhecidos. A técnica de detecção consiste na busca por palavras ou termos específicos, como "*Hacked*" ou "Hackeado", a partir de um dicionário. Porém, pode haver textos bem mais elaborados. Como ocorre com as técnicas analisadas anteriormente, essa também não detecta alterações em imagens.

2.6. Detecção de desfigurações pelo reconhecimento de imagens

A técnica desenvolvida por Borgolte e Kruegel (2015) propõe uma forma de detecção por meio da Inteligência Artificial. O sistema baseado em uma rede neural aprende, analisando as assinaturas que os atacantes deixam em uma página desfigurada. Isso é possível por meio de arquivos de imagem gerados através de um dos bancos de dados mais famosos de páginas desfiguradas: THC.org. Para o aprendizado as imagens obtidas são divididas em partes menores. Elas não podem ser muito pequenas para não aumentar as chances de falsos positivos. Elas não podem ser muito grandes, pois aumentaria de uma forma considerada o tempo de aprendizado. Os métodos utilizados para detecção seriam inicialmente pela captura dos logotipos utilizados por grupos de atacantes, erros tipográficos ou gramaticais, *leetspeak*, combinação de letras e combinação de cores. Portanto, a proposta de Borgolte e Kruegel (2015) busca assinaturas de ataques, enquanto a proposta do presente artigo identifica as alterações da página web, sem a necessidade de conhecer os ataques.

3. Detecção por Simetria de Imagens

A proposta apresentada nesse trabalho adota uma nova abordagem de análise de imagens, identificando a simetria entre o estado inicial da página web e o estado atual. Esse trabalho difere dos demais, devido à sua simplicidade. Não é necessário armazenar ou analisar todo o conteúdo da página web e a técnica permite avaliar páginas estáticas e dinâmicas, conforme pode ser observado a seguir.

3.1. Modelo de detecção

A técnica consiste em transformar toda a página web em uma "fotografia", como seria observada pelo usuário em seu browser. Em seguida a imagem gerada é dividida em

partes menores e é calculado, cada uma das partes, um valor *hash*, que é armazenado em uma base de dados.

Conforme ilustrado na Figura 1, periodicamente são feitas novas capturas. Para cada captura é gerada uma nova imagem e uma nova subdivisão, usando os mesmos parâmetros da imagem original. Os valores *hash* de cada parte são comparados com os dados originais. O valor percentual da alteração da página inicial é obtido por meio da divisão do número de partes (*hashes*) diferentes, pelo total de partes da imagem. É considerado um ataque quando o valor percentual de alteração da página extrapola o indicador limite estabelecido pelo administrador.

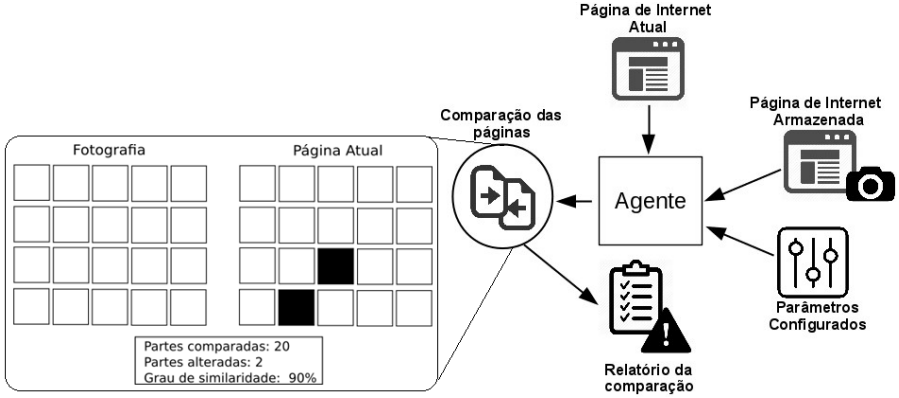

Figure 1. Mecanismo de detecção por similaridade de imagens

Além da comparação das partes da imagem, o modelo prevê uma verificação das resoluções da imagem completa. Nessa comparação, utilizam-se suas alturas e larguras como pontos de referência. Caso esses parâmetros sejam diferentes de um limite estabelecido é gerado o alerta. Essa técnica é bastante útil na identificação de ataques em páginas estáticas. Mesmo as páginas dinâmicas costumam possuir limites de tamanho, principalmente na largura.

Cabe ressaltar que a curacidade dos resultados depende das ferramentas adotadas para a geração e divisão das imagens das páginas. Todas as imagens deverão ser obtidas usando as mesmas ferramentas, com os mesmos parâmetros de configuração, para evitar distorções nos dados obtidos.

3.2. Implantação e testes

O mecanismo proposto foi implementado em um protótipo totalmente funcional, chamado de ***DefaceSpy***. O protótipo utilizou ferramentas *Open Source* para redução de custos. Foi implementado na linguagem de programação *PHP* e alguns módulos em *Shell Script.* O processo gráfico de criação de arquivos de imagem foi feita pela ferramenta *wkhtmltoimage*[1]. A divisão da imagem gerada, em partes, é feita pela

[1] https://wkhtmltopdf.org

ferramenta *ImageMagik*[2]. O protótipo foi implementado no sistema operacional Debian GNU/Linux 8.7.0 64 bits.

Para validar o protótipo foram feitos testes de desfiguração total e parcial em um ambiente controlado. Conforme ilustrado na Figura 2, a imagem da esquerda corresponde à página original, enquanto à imagem da direita corresponde à página capturada para análise. Observa-se uma alteração parcial na página.

Figure 2. Comparação de páginas

A Figura 3 apresenta o texto com o resultado da análise de uma desfiguração parcial da página de teste, demonstrando a eficácia da proposta.

```
################ DefaceSpy Versao:0.0.1 ################
[*] Dimensoes da Pagina Atual x Snapshot: OK
[*] Total de partes obtidas (Pagina Atual): 176
[*] Total de partes obtidas (Snapshot): 176
[*] Resultado da comparacao Snapshot e a versão Atual
>> Limite toleravel de alteracoes na pagina: 90%
>> Total de Partes do snapshot: 176
>> Total de Partes da versão Atual: 176
Similaridade entre o Snapshot e a versão Atual: 54.55%
Resultado: Similaridade abaixo da tolerancia de 90%.

ATENCAO! PAGINA POSSIVELMENTE DESFIGURADA.

[*] Gerado relatorio: defacespy_07052017130705.html
```

Figure 3. Resultado da análise nos testes

[2] http://www.imagemagick.org/script/index.php

Além desse teste, também foram realizados testes com alterações da resolução da página e com a desfiguração total. Em todos houve a identificação dos ataques.

3.2. Considerações

O protótipo implementado e os resultados dos testes demonstraram a efetividade do modelo proposto.

Uma questão a ser tratada no futuro diz respeito à definição do indicador de similaridade em páginas dinâmicas. Para cada caso deve-se realizar ajustes, dependendo da periodicidade das atualizações de página e do grau de modificação. O uso de mecanismos de aprendizado de máquina pode ser bastante útil para a automatização do cálculo do indicador.

Uma das limitações dessa proposta é a impossibilidade de identificar alterações não visíveis da página, como a infecção de scripts para ataques aos clientes. Para esses casos seria necessário incluir outras técnicas de detecção que analisassem também o código HTML da página. A associação de outras técnicas deverá ser abordada em trabalhos futuros.

Conclusão

O presente trabalho analisou a literatura contendo os principais métodos de detecção de desfiguração de páginas web e propôs um novo e simplificado mecanismo automatizado de detecção de ataques, baseado na simetria de imagens.

O modelo proposto foi implantado em um protótipo totalmente funcional, que demonstrou a eficácia da proposta.

Pretende-se, em trabalhos futuros, desenvolver técnicas automatizadas para o cálculo dos limites do indicador de similaridade, possivelmente utilizando inteligência artificial.

Referências

Borgolte, K., Kruegel, C.; Vigna, G. Meerkat: “Detecting Website Defacements through Image-based Object Recognition”. In: USENIX Security Symposium. 2015. p. 595-610.

Davanzo, G., Medvet, E., Bartoli, A (2011). “Anomaly detection techniques for a web defacement monitoring servisse”. Expert Systems with Applications, v. 38, n. 10, p. 12521-12530.

Gurjwar, K.R., Sahu, D.R., Tomar, D.S. (2013). “An approach to reveal website defacement”. International Journal of Computer Science and Information Security, v. 11, n. 6, p. 73.

Kanti, T., Richariya, V., Richariya, V. (2011). “Implementing a Web browser with Web defacement detection techniques”. World of Computer Science and Information Technology Journal (WCSIT), v. 1, n. 7, p. 307-310.

Oriyano, J.M. (2014). CEH v8: Certified Ethical Hacker Version 8 Study Guide. Wiley & Sons, USA.

Philippe, H., Lauren, W. (2017). “What is the Document Object Model?”, https://www.w3.org/TR/DOM-Level-2- Core/introduction.html.

Verma, R.K.; Sayyad, S. (2015). “Implementation of Web Defacement Detection”. International Journal of Innovations in Engineering and Technology (IJIET), v.6, p. 134-141.

