Esteganografia em Páginas Web

Bruno Bastos Guimarães, Luciano Porto Barreto

Departamento de Ciência da Computação / Laboratório de Sistemas Distribuídos

Universidade Federal da Bahia

Campus de Ondina, CEP: 40170-110, Salvador-BA, Brasil

{brunobg,lportoba}@ufba.br

1. Introdução

Existem atualmente diversas técnicas conhecidas de escrita oculta de informações ou esteganografia [Katzenbeisser and Petitcolas 2000]. A maioria destas utiliza arquivos de mídia (*eg*, GIF, JPEG, MP3) para ocultar a informação em trechos do arquivo nos quais a funcionalidade ou uso do arquivo não são comprometidos. Atualmente, existem diversos aplicativos estenográficos disponíveis, bem como as ferramentas de esteganoanálise que visam reverter esse processo.

Outra técnica interessante, chamada de esteganografia lingüística [Bergmair 2004], consiste em produzir textos que contenham informações ocultas. Entretanto, a dificuldade na produção de textos com semânica condizente a interlocutores humanos facilitam sua detecção, ainda que o conteúdo da mensagem seja preservado. Uma variante dessa técnica, ainda pouco explorada, transforma a mensagem em programas escritos em determinada linguagem de programação através de um mecanismo de geração. Alguns trabalhos nessa área incluem a camuflagem em código binário de programas executáveis [El-Khalil and Keromytis 2004] ou, no caso de arquivos HTML, utilização da ordem de aparição dos atributos para ocultar a informação.

De fato, ludibriar pessoas com códigos de programa é mais simples do que com linguagem natural. Ainda que o código produzido seja inócuo, pode-se levar tempo considerável para descobrir tal farsa em virtude do tamanho e da adequação técnica de geração. Além disso, se o código gerado for compilável, aumenta-se a probabilidade deste passar despercebido por um computador que analisasse os dados trafegados.

2. A Ferramenta WebHide

Nossa proposta consiste em codificar um conjunto de caracteres em código JavaScript que será inserido em uma ou mais páginas web. Com esse objetivo, construímos a ferramenta *WebHide* que, a partir de um texto de entrada e uma árvore de diretórios contendo páginas web, insere código JavaScript em uma ou mais páginas.

A produção do código JavaScript considera que cada bit ou grupo de bits do texto de entrada possui uma palavra (ou *token*) equivalente em JavaScript. Para isso, construímos um dicionário de codificação que efetua essa associação. Por exemplo, o os bits `00` poderiam produzir o texto `IF`. Para permitir uma gama maior de variações do código resultante e assegurar que o mesmo seja sintaticamente correto, utilizamos um conjunto de regras gramaticais para dirigir o mecanismo de tradução. Além dos bits de entrada, escolhemos uma regra gramatical, chamada de *estilo*, cujos *tokens* guiam o processo de tradução. Assim, a depender do estilo corrente, a geração de código sofre

transformações. Dessa forma, um mesmo texto de entrada pode produzir programas distintos, o que aumenta a robustez da ferramenta. Vale ressaltar que é desnecessário implementar todo o conjunto de regras gramaticais de JavaScript.

Uma das vantagens da utilização da ferramenta é permitir que o código JavaScript gerado seja inserido em uma ou mais páginas HTML. O fato do texto de entrada ser codificado em JavaScript reduz a chance de busca pela informação, pois os navegadores web não apresentam, por definição, o código JavaScript das páginas web. Ainda assim, o código JavaScript produzido é inócuo e, portanto, não produz resultados inapropriados ou perda de desempenho.

Algumas melhorias na ferramenta incluem enriquecer os estilos com trechos de programas válidos para aumentar a variedade e veracidade do código gerado. Outra extensão consiste em agregar a funcionalidade de *WebHide* a um navegador web ou site, por meio de *scripts*, a fim de permitir que as senhas de usuários fiquem registradas nas próprias páginas do servidor. Dessa forma, os usuários poderiam recuperar suas senhas diretamente do servidor através da descoberta de suas senhas nos trechos JavaScript das páginas do servidor. Isso dispensaria que os usuários guardassem suas senhas em seus navegadores ou as deixassem por descuido em navegadores de terceiros. Por fim, outra maneira de tornar o método mais seguro é criptografar ou comprimir o texto e, depois, ocultar o texto cifrado.

Referências

Katzenbeisser, S. and Petitcolas, F. A. (2000) “Information Hiding Techniques for Steganography and Digital Watermarking”. Artech House, Inc., 2000.

Cole, E. (2003) “Hiding in Plain Sight: Steganography and the Art of Covert Communication”. New York, NY, USA: John Wiley & Sons, Inc., 2003.

Bergmair, R. (2004) “Towards Linguistic Steganography: A Systematic Investigation of Approaches, Systems and Issues”. Technical Report, University of Derby. November, 2004.

El-Khalil, R.; Keromytis, A. D. (2004) “Hydan: Hiding information in program binaries”. In: Proceedings of the 6th International Conference on Information and Communications Security (ICICS). [S.l.: s.n.], 2004. p. 187–199.