

Detecção de técnicas de anti-forense em programas maliciosos

Vitor Falcão da Rocha*, Paulo Lício de Geus, André Grégio, Marcus Botacin.

Resumo

Programas maliciosos (*malware*) são ameaças persistentes à segurança, evoluindo constantemente para evitar a detecção e análise dinâmica, através de técnicas de anti-forense. Para acompanhar esse ritmo de evolução, é imprescindível a automatização de processos relacionados a análise de *malware*. Neste trabalho foram estudadas técnicas de anti-forense empregadas por programas maliciosos, além de métodos de detecção dessas técnicas. Com o conhecimento adquirido, criou-se um sistema capaz de detectar a presença dessas técnicas em arquivos binários.

Palavras-chave:

Anti-forense, *malware*, segurança computacional.

Introdução

Uma das ameaças atuais mais graves à segurança dos sistemas computacionais e seus usuários é o ataque por programas maliciosos (*malware*), que subvertem a operação legítima de um sistema afetando sua integridade, confidencialidade e disponibilidade. Ataques por *malware* são responsáveis por evasão de informações, roubo de credenciais, forja de identidade, armazenamento de conteúdo ilícito, lançamento de ataques a terceiros, entre outras atividades danosas para a vítima. Para que esses programas maliciosos possam atingir seus objetivos com êxito, são utilizadas técnicas de anti-forense. Essas técnicas evitam que mecanismos de segurança e especialistas humanos analisem os programas maliciosos e seus artefatos, dificultando assim a mitigação dessas ameaças. O presente projeto tem por objetivo o estudo dessas técnicas, bem como o estudo de mecanismos para detecção dessas técnicas em *malware*. As técnicas estudadas podem ser divididas em 3 grandes categorias [Branco et. al 2012], onde uma lida com anti-*debugging*, que são técnicas que dificultam a análise dinâmica do *malware*, outra com anti-*disassembly*, que dificulta a obtenção do código de montagem e anti-VM, que detecta ou compromete máquinas virtuais. O trabalho desenvolvido implementou métodos de detecção para técnicas das três categorias e gerou como resultado um sistema automatizado de análise de *malware*, que é capaz de detectar, a partir do código de montagem, a presença dessas técnicas em programas maliciosos.

Resultados e Discussão

A base do sistema é o PyBFD [PyBFD 2016], que é uma ferramenta *open-source*, em Python, que serve como uma interface para o *GNU Binutils libopcodes* e *libbfd*, e nos permite obter o código de montagem de um binário como objetos Python. O sistema é organizado em módulos independentes, onde o módulo principal é responsável por fazer a desmontagem do binário (*disassembly*) e em seguida executar os detectores de técnicas de anti-forense. Essa forma de organizar o sistema permite que esteja seja altamente extensível, pois basta criar um novo módulo para adicionar um detector para uma nova técnica de anti-forense. Testes iniciais com uma amostra de 70 mil exemplares mostraram que 4,3% dos exemplares aplicavam alguma técnica não detectada por outras soluções de análise, como Pyew [Pyew 2016] e PEframe [PEframe 2016].

Dentre as técnicas detectadas estavam a de *garbage bytes* e *anti-VMWare*, que são capazes de atrapalhar o *disassembly* do programa malicioso a ser analisado [Branco et. al 2012]. Destaca-se que o processo de implementação dos detectores é contínuo, sendo esperado que a taxa de detecção do sistema aumente com o desenvolvimento de novos detectores. As listagens 1 e 2 ilustram, respectivamente, um exemplo de detector e de um trecho de código detectado.

Listagem 1. Trecho de código de detecção anti-VM.

```
if instruction == 'in':  
    print "\"VMWareINInstruction\" Detected! Section: <  
%s> Address: 0x%X" % (section.name, address)
```

Listagem 2. Detecção de técnica anti-VM.

```
"VMWareINInstruction" Detected! Section: <.text>  
Address: 0x6090114D
```

Conclusões

É imprescindível a automatização de processos relacionados às análises, para que seja possível acompanhar o ritmo no qual novas ameaças surgem e assim mitigá-las. Nesse sentido, a solução proposta se mostra valiosa, pois é capaz de analisar rapidamente o binário e indicar não somente a presença, como a localização (endereço) da técnica de anti-forense. Adicionalmente, a solução será integrada ao projeto Behemot [Botacin et. al 2014] de modo a constituir uma solução completa de análise de *malware*, com capacidade de detecção de técnicas anti-forense.

Agradecimentos

Ao suporte do CNPq através do processo PIBIC 122796/2015-2.

Branco, R.; Barbosa, G. e Neto, P.; *Scientific but not academical overview of malware anti-debugging, anti-disassembly and anti-VM technologies*. In: BlackHat, 2012, Las Vegas, NV.

Botacin, M.; Afonso, V.; Geus, P. L. e Grégio, A.; *Monitoração de comportamento de malware em sistemas operacionais Windows NT 6.x de 64 bits*. In: Anais do XIV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, 2014, Belo Horizonte, MG.

Russ, F.; Muniz, S.; <https://github.com/Groundworkstech/pybfd>; 14/07/2016

Koret, J.; <https://github.com/joxeankoret/pyew>; 14/07/2016.

Amato, G.; <https://github.com/guelfoweb/peframe>; 14/07/2016.