

## **CONCEITOS E PROCEDIMENTOS EM ANÁLISE DINÂMICA DE CÓDIGO BASEADO EM SISTEMAS WINDOWS**

Adriano Mauro Cansian, Thiago Alves Siqueira, César Eduardo Atílio  
UNESP – Universidade Estadual Paulista  
{adriano,thiago,cesar}@acmesecurity.org

*Resumo - A Internet atualmente está sendo usada para a prática de crimes eletrônicos. Com isso, cresce a necessidade de que técnicas de perícia forense sejam utilizadas para coleta de evidências digitais, que possam ser usadas para apurar quem são os meliantes e gerar uma contra-medida contra cada tipo de ataque. Diante deste cenário, este artigo irá mostrar vários conceitos, técnicas e metodologias pertinentes à disciplina de perícia forense computacional, mais especificamente à análise dinâmica de código em ambientes Windows. Ao final, será feito um estudo de caso de um código malicioso que circulou na internet brasileira no início de 2003.*

### *I. Introdução*

Atualmente, a Internet é utilizada para os mais diversos fins. Informações confidenciais e valiosas trafegam pela rede. Este tipo de informação é um atrativo para meliantes, que, em posse delas, pode tirar proveito da vítima e trazer benefícios a si, sendo isso um crime. Dessa forma, técnicas de perícia forense computacional devem ser utilizadas para obtenção de informações de como isso ocorreu. Diante desta situação, este trabalho irá descrever conceitos e metodologias de perícia forense computacional, mais especificamente da análise dinâmica de código, voltadas ao sistema operacional Windows. Ao final, será feito um estudo de caso de um código malicioso que circulou na Internet brasileira no início de 2003.

### *II. Ciência forense computacional*

Para o escopo deste trabalho, é considerado e entendido que a Ciência Forense Digital consiste no uso de métodos científicos na preservação, coleta, validação, identificação, análise, interpretação, documentação e apresentação de evidência digital. Seu propósito é facilitar ou possibilitar posterior reconstrução de eventos criminais[1]. Para suportar os resultados de uma análise forense são necessários procedimentos e protocolos detalhados, documentados e revisados, aceitos pela comunidade científica relevante[2]. Para tal, a seguir, serão apresentados os procedimentos para uma análise dinâmica de uma evidência digital.

#### *A. Análise de programa*

Uma das maneiras, entre as várias existentes, para análise de programa, é o estudo dinâmico do artefato. Sua vantagem em relação a outros métodos é que ela

pode ser rápida e precisa. Sua desvantagem é que a saída da análise obtida é tudo que se possui como resultado.

### *B. Análise dinâmica*

O primeiro conceito a ser definido é o de código malicioso (artefato). No contexto em que estamos trabalhando, código malicioso é um termo geral que se refere a programas projetados para efetuar algum tipo de atividade não autorizada em sistemas computacionais. A análise dinâmica de artefatos é um processo minucioso, baseado na execução do código malicioso, observação e monitoração de todas as alterações causadas no sistema em tempo real. Os passos para a realização de uma análise serão explicitados ao longo do artigo.

## *III. Metodologia de análise*

### *A. Máquina que será usada na análise*

Um ambiente confiável deve ser preparado para a utilização no processo de análise. Deve-se levar em conta o equipamento e sistema operacional alvo do artefato e o que se deseja adquirir com a análise.

O ambiente de análise deve ser isolado de uma rede de produção, para que se evite possíveis ataques a esta. Isso pode ser feito por filtros de contenção de tráfego.

Determinado que um programa deve ser analisado é necessário coletá-lo e gerar um *hash* criptográfico[4]. Esta é uma técnica importante, pois atribui unicidade ao artefato. Assim, pode-se compará-lo com outros artefatos, e, caso esse *hash* seja semelhante, já se sabe que o código malicioso é o mesmo. Havendo alguma análise pronta, não é necessário uma nova, o que pode se determinar pelo *hash*.

### *B. Preparação do ambiente*

O ambiente de análise deve possuir o equipamento e o sistema operacional alvo do artefato. Deve-se utilizar um ambiente exclusivo para análise, afinal, após a mesma, estará impróprio para utilização como ambiente de produção. Este ambiente deve possuir todos os serviços requisitados pelo artefato. Dessa forma pode-se obter total performance do código, podendo cobrir todos os fluxos possíveis em seu código fonte.

Quaisquer alterações realizadas no sistema não devem passar despercebidas, e, para isso, deve-se utilizar as ferramentas corretas. Dependendo do sistema operacional, comportamentos diferentes são esperados. No sistema operacional Windows, por exemplo, espera-se que o registro do sistema seja alterado. Deve-se, então, monitorá-lo.

Além das alterações locais, podem ocorrer alterações remotamente. Inúmeros artefatos têm como objetivo disparar ataques remotos contra outros computadores. Deve-se, portanto, permitir que haja conexões externas, mas de forma controlada. Para isso, pode-se usar filtros no *firewall* [5] desta rede para bloqueá-las. Outra forma é uma rede isolada com os serviços requisitados.

Um analista de artefatos deve conhecer o maior número de ferramentas para análise para o maior número de sistemas operacionais possíveis, de maneira a ser eficiente e eficaz. Isto porque a análise dinâmica é um processo onde deve-se definir rapidamente as conseqüências causadas pelo artefato e as contra-medidas.

### *C. Evidências digitais*

O termo evidência digital refere-se a toda e qualquer informação digital capaz de determinar que um incidente ocorreu.

Dan Farmer e Wietse Venema introduziram um conceito denominado de ordem de volatilidade [6], que diz que o tempo de vida de uma evidência digital varia de acordo com o local onde ela está armazenada. O detalhamento de cada fonte de informação e das técnicas utilizadas para sua extração é apresentado a seguir.

### *D. Coleta de evidências*

A busca de evidências em um sistema computacional constitui-se de uma varredura minuciosa nas informações que nele residem, sejam dados em arquivos ou em memória, “deletados” ou não, cifrados ou possivelmente danificados [7].

Em uma investigação do comportamento de um programa, deve-se obter informações sobre o estado corrente do sistema. O estado da máquina comprometida não pode ser alterado. Assim, as informações devem ser gravadas em outra máquina ou a saída das ferramentas de coleta de informações direcionadas para a máquina forense. Para isso, uma boa ferramenta é o *netcat*<sup>1</sup>.

Na máquina forense que está recebendo o redirecionamento dos comandos, pode-se utilizar ferramentas diretamente de um CD-ROM. Garante-se dessa maneira que as ferramentas utilizadas para a análise não serão modificadas pelo artefato.

## *IV. Coletando informação volátil*

### *A. Processos*

Informação volátil pode ser descrita como a informação que representa o estado corrente do sistema. O primeiro item de interesse de um investigador é o conjunto de processos em execução no sistema, pois tais informações podem revelar evidências de atividades não autorizadas. Esse tipo de informação pode ser obtida pelo *Pslist.exe*<sup>2</sup>, que provê uma lista dos processos em execução, o número de identificação do processo (PID), e a quantidade de tempo de início de execução, além dos dados sobre a carga de processamento. *Fport.exe*<sup>3</sup> age do mesmo modo,

---

<sup>1</sup> O programa netcat e maiores detalhes sobre o mesmo podem ser encontrados na URL [http://www.atstake.com/research/tools/network\\_utilities/](http://www.atstake.com/research/tools/network_utilities/) (disponível em Maio de 2004).

<sup>2</sup> O programa pslist.exe e informações adicionais sobre o mesmo podem ser encontradas em <http://www.sysinternals.com/ntw2k/freeware/pslist.shtml> (disponível em Maio de 2004).

<sup>3</sup> O programa Fport.exe e informações adicionais sobre o mesmo podem ser encontradas em <http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/overview.htm> (disponível em Maio de 2004).

fornecendo além daquelas informações o caminho do comando executado e as portas TCP e UDP abertas referentes ao processo. *Listdlls.exe*<sup>4</sup> lista as DLL's<sup>5</sup> que estão em uso em cada processo e o caminho das DLL's carregadas.

### B. Conexões de rede

O estudo das conexões de rede provê informações valiosas acerca das conexões em andamento e dos processos aguardando uma conexão [8]. Assim, é possível determinar se existe alguma conexão não autorizada em andamento. Isto pode ser feito utilizando as ferramentas *netstat*<sup>6</sup> e *nbtstat*<sup>7</sup>.

Uma outra informação volátil consiste no tráfego enviado pelo sistema comprometido a outros hosts. Existem vários programas para Windows que utilizam as bibliotecas *wincap*<sup>8</sup>, comumente denominados de *sniffers*. O exemplo mais comum desse programas é o *Ethereal*<sup>9</sup>.

### V. Informação não volátil

Informação não volátil consiste das configurações do sistema que não mudam todo tempo, ou quando o sistema é reinicializado.

#### A. Arquivos

A fonte de informação onde o processo de análise forense geralmente mais se concentra é o sistema de arquivos. Uma das informações referentes aos arquivos que o investigador deve coletar são as marcas de tempo. Tais marcas correspondem aos tempos de última modificação no arquivo, último acesso e última mudança nas propriedades do arquivo (*MAC times*) [10]. Um exemplo de ferramenta que pode ser usado para este fim é a *afind.exe*<sup>10</sup>.

#### B. Registro

O registro do sistema é definido como uma base de dados hierárquica usada no Microsoft Windows 9x, CE, NT e 2000 para armazenar informação necessária para configurar o sistema, aplicações e dispositivos de *hardware*. Esta base de dados pode conter evidências valiosas acerca do comportamento do código

---

<sup>4</sup> O programa *Listdlls.exe* e informações adicionais podem ser encontradas em <http://www.sysinternals.com/ntw2k/freeware/listdlls.shtml> (disponível em Maio de 2004).

<sup>5</sup> Maiores informações sobre DLL em <http://support.microsoft.com/default.aspx?scid=kb;en-us;815065> (disponível em Maio de 2004).

<sup>6</sup> O programa *Netcat* e informações adicionais podem ser encontrados em [http://www.atstake.com/research/tools/network\\_utilities/](http://www.atstake.com/research/tools/network_utilities/) (disponível em Maio de 2004).

<sup>7</sup> Maiores informações sobre o *nbtstat* podem ser encontradas em <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/nbtstat.msp> (disponível em Maio de 2004).

<sup>8</sup> Maiores informações sobre o *Wincap* podem ser obtidas em <http://wincap.polito.it/> (disponível em Maio de 2004).

<sup>9</sup> A ferramenta *Ethereal* e maiores informações podem ser obtidas em <http://www.ethereal.com/> (disponível em Maio de 2004).

<sup>10</sup> O pacote que contém o *Afind* e maiores informações podem ser obtidos em <http://www.foundstone.com/> (disponível em Maio de 2004)

malicioso. Uma ferramenta nativa do sistema para coletar informação das entradas do registro é *Regedt32.exe* para Windows NT e o *Regedit.exe* para demais versões do Windows.

### *C. Documentação*

A documentação é um processo que ocorre paralelamente ao processo de análise. Todos os passos percorridos para execução da mesma deve ser minuciosamente relatado. Nisto, envolve a descrição das ferramentas utilizadas, modo de preparação do ambiente e razão pela qual o ambiente foi montado de tal maneira. Informações obtidas pela saída das ferramentas devem ser anexadas ao documento.

Além destes dados, deve-se ter a identificação do responsável pela análise, data da coleta do artefato, *hash* criptográfico do arquivo, forma como foi coletado o artefato (e-mail, honeypot etc) e razão pela qual foi considerado um artefato.

Como conclusão, a documentação deve possuir uma maneira para solucionar o problema causado pelo artefato analisado.

### *VI. Estudo de caso*

Abaixo, é apresentado um estudo de caso consistindo em uma análise dinâmica do código malicioso *certificado\_digital.exe*. Este programa foi distribuído através de correio eletrônico na Internet brasileira, em meados de maio de 2003. Trata-se de uma mensagem persuasiva de caráter cognitivo e um programa anexado. Este programa instala o artefato real MSNBC32.EXE no sistema onde foi executado.

#### *A. Análise Dinâmica do artefato de risco*

Análise: 12/2003

#### **certificado\_digital.exe**

Sistemas Afetados: Sistemas utilizando Microsoft Windows

Codinome: falso-verisign

Hash: 7e5776f9c965307d8033433cbdbc6bde

Tamanho do Arquivo: 334567 bytes

Dados do analista: xxxx

Dados do responsável pela análise: xxxx

Data e horário da análise: 13/05/2003 – 22:00 h.

#### *B. Motivações para análise*

- E-mail malicioso recebido por várias pessoas com informação aparentemente não autêntica;
- Tal atitude não condiz com o comportamento da empresa em questão ao enviar e-mails não solicitados;
- Um arquivo acompanhado de uma mensagem persuasiva, de origem não comprovada;
- Mensagens de listas de discussão relatando o comportamento

parcial do artefato;

- Mensagem cognitiva com erros graves de português;
- Não se distribui certificados digitais por e-mail.

#### *C. Ambiente de análise*

Máquina Forense: Sistema operacional: Windows 98 SE

Processador: AMD Athlon 800MHZ

Memória RAM: 128MB

Disco: 2 GB

Esta máquina é integrante de uma rede que possui mais 2 máquinas utilizando o sistema operacional Linux. Esta rede é limitada por um *firewall* (Zwicky e Cooper, 2000) utilizando IPTABLES (Stephens, 2002). O objetivo destas máquinas é monitorar e capturar a interação da máquina forense com o ambiente, quando o programa for executado. Estrategicamente, o filtro de contenção tem como principal objetivo conter qualquer tipo de tráfego malicioso destinado a algum computador externo do ambiente forense.

#### *D. Execução do artefato*

21:54:14 O programa é executado na máquina forense descrita acima.

A saída do programa pslist.exe forneceu a seguinte informação:

*Name (MSNBC32), Pid (1700), Elapsed Time (0:00:12.031)*

Arquivos criados ou modificados a partir da execução do artefato:

C:\WINDOWS\MSNBC32.EXE

C:\WINDOWS\Debug32

C:\faxsetup32.txt

C:\WINDOWS\Debug32\Clickxxx.jpg

C:\WINDOWS\SERVONE32.DLL

C:\WINDOWS\SERVTW032.DLL

C:\WINDOWS\ZIPDLL.DLL

C:\WINDOWS\SYSTEM.CB (Arquivo que contém informações a respeito do teclado).

Foi notificada a criação ou modificação destes arquivos de acordo com os atributos de tempo dos mesmos, conferindo com o horário de execução do programa. Esta verificação foi realizada inúmeras vezes para garantir que as mudanças possam ser atribuídas a execução do artefato.

Alterações no registro são realizadas para que o artefato execute automaticamente quando o sistema for reiniciado. Para isso, é adicionado um novo valor de *string* no registro com o nome MSNDC32.EXE, referenciando C:\WINDOWS\MSNBC32.EXE em MyComputer\HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\Current Version\Run

Quando o programa é inicializado automaticamente pelo Windows, o arquivo C:\faxsetup32.txt é criado. Outras possíveis localizações de MSNBC32.EXE no registro é RunServices, Run-, RunServices-, RunOnce.

### *E. Funcionamento do artefato*

Ao iniciar sua execução, o arquivo MSNBC32.EXE cria o arquivo C:\faxsetup32.txt, que irá armazenar um sumário com a especificação do sistema, os pressionamentos de teclas bem como os títulos das janelas de aplicativos ativos. Além disto ele correlaciona temporalmente os pressionamentos de teclas com fotos da região clicada com o *mouse*.

Os arquivos Clickxxx.jpg, gerados em C:\WINDOWS\Debug32, armazenam imagens de 25 x 25 pixels em 16 milhões de cores ao redor do cursor, quando o botão esquerdo do mouse é pressionado.

O programa captura imagens e o comportamento do sistema por aproximadamente 400 segundos. Após isso, o programa envia via correio eletrônico os arquivos faxsetup32.txt e todos Clickxxx.jpg, tendo como remetente zzz@zzz.zzz e destinatários xxx@xxx.xxx.xx e yy@yyy.yyy.yy. Segue abaixo o arquivo faxsetup32.txt gerado (Dados sanitizados):

#### Dados do Equipamento para Identificação

Sistema Operacional : Windows 95/98/Me

Nome do Computador : VWwin98

Versão do Sistema : xxx

Compilação do Sistema : xxxxxxxx

Sistema de Arquivo : FAT32

Nome do Disco Local : Disco Local

Serial do HD : xxxxxxxx

Data do Sistema : 5/13/03

Hora do Sistema : 10:07:55 PM

IP do Computador Local : 192.168.0.98

Janela Ativa - Servant Salamander

< Click1 >

< Click2 >

< Click3 >

<Tab> <Down> <F3> <Esc> <Down>

Janela Ativa - C:\WINDOWS\Debug32\Click2.jpg (25 x 25 x 16777216 colors - 100%) - PictView

Janela Ativa - Servant Salamander

<Esc> <Ctrl> <Alt>

< Click4 >

Janela Ativa - Microsoft Internet Explorer

< Click5 >

Janela Ativa - C:\WINDOWS\Debug32\Click9.jpg - Microsoft Internet Explorer

*F. Comportamento anômalo do sistema obtido através da execução do artefato*

- Sistema apresenta perda de desempenho;
- Falta de recursos ou de memória RAM;
- Cursor do mouse pisca intermitentemente.

#### *G. Recomendações para limpeza de máquinas comprometidas*

Os seguintes passos devem ser seguidos para limpeza de máquinas comprometidas:

Se a máquina puder ser formatada:

1. Desconectar a máquina comprometida de rede;
2. Fazer cópia de segurança de arquivos pessoais;
3. Formatar a máquina comprometida;
4. Trocar todas as senhas que foram inseridas no sistema a partir do instante em que o artefato foi executado, incluindo senhas e números de cartões de crédito;

Outra opção é excluir as entradas adicionadas pelo artefato no registro e os arquivos por ele criados, citados na análise. É altamente recomendável reinstalar o sistema, modificando algumas de suas características, como por exemplo a versão do sistema operacional, o nome da máquina e o endereço IP da mesma na rede.

#### *H. Conclusão da análise*

Frente a este comportamento, presumiu-se que um dos objetivos do programa é capturar senhas ou informações pessoais inseridas através de uma interface do tipo teclado virtual, muito usado atualmente em sistemas *Netbanking*, e também de informações digitadas a partir do teclado. Os teclados virtuais nada mais são do que uma interface gráfica que permite a simulação do pressionamento de teclas, através do uso de cliques do mouse.

### *VII. Conclusão*

O estudo apresentado neste trabalho representa um esforço no sentido de suprir a necessidade de um melhor entendimento de como se obter e utilizar evidências eletrônicas armazenadas em computadores. A discussão acerca dos vários procedimentos envolvidos em uma análise dinâmica, detalhando metodicamente cada etapa de uma análise, fornece um guia prático para aqueles que estão iniciando na área forense computacional.

### *IX. Referências*

- [1] G. Palmer, "A Road Map for Digital Forensic Research," Digital Forensic Research Workshop (Dfrws), *Report* 2001.
- [2] M. Noblett, M. Pollitt, L. Presley, "Recovering and Examining Computer Forensic Evidence," *Forensic Science Communications*, Number 4, Volume 2, U.S. Department of Justice, FBI.
- [3] W. Venema, "Finding the purpose of an unknown program," *Strangers In the Night*, Dr. Dobb's Journal, 2000.
- [4] B. Schneier, "*Applied Cryptography*", John Wiley & Sons, New York, 1996.
- [5] J. C. Stephens, "Iptables". Disponível:  
<http://www.sns.ias.edu/~jns/security/iptables/index.html>

- [6] D. Farmer, W. Venema, "Computer forensics analysis class handouts".  
Disponível: <http://www.fish.com/forensics/class.html>
- [7] M. Abdalla, P. Geus, "Forense Computacional: Procedimentos e Padrões",  
Simpósio de Segurança da Informação, 2001, São José dos Campos, Anais SSI2001.
- [8] W. G. Kruse II, J. G. Heiser. "Computer Forensics: Incident Response  
Essentials", Addison-Wesley, Reading, Massachusetts, 2002.
- [9] A. Silberschatz, P. Galvin, "Operating System Concepts", John Wiley & Sons,  
New York, 5 Edição.
- [10] E. Casey, "Handbook of Computer Crime Investigation", Academic Press, San  
Diego, Califórnia, 2000.