Uma Visão Geral do *Malware* Ativo no Espaço Nacional da Internet entre 2012 e 2015

Marcus F. Botacin[1], **André Grégio**[1,2], **Paulo Lício de Geus**[1]

[1] Instituto de Computação – Universidade Estadual de Campinas (Unicamp)
Av. Albert Einstein, 1251 – 13083-852 – Campinas – SP – Brasil

[2]Centro de Tecnologia da Informação Renato Archer (CTI/MCTI)
Rod. D. Pedro I (SP-65), KM 143,6 – 13069-901 – Campinas – SP – Brasil

{marcus, paulo}@lasca.ic.unicamp.br, andre.gregio@cti.gov.br

Abstract. *Malware is one of the main attack vectors to compromise computer systems. To be ahead of security mechanisms, malware authors diversify their creations by inserting evasive functions, applying obfuscation techniques, and modularizing them into distributed components. In addition, distinct trends can be observed in different countries, according to the type of users and other factors. In this paper, we analyze malware active in Brazilian cyberspace between 2012 and the first quarter of 2015. We evaluated over 20 thousand unique samples, presenting the results regarding static and dynamic analysis.*

Resumo. *Programas maliciosos são um dos principais vetores de ataque contra sistemas computacionais. Para estar um passo à frente de mecanismos de segurança, seus desenvolvedores inserem funções evasivas, técnicas de ofuscação e modularização em componentes distibuídos. Além disso, tendências podem ser observadas em países diferentes de acordo com o tipo de usuário, entre outros fatores. Neste artigo, são analisados exemplares de malware ativos no espaço brasileiro da Internet entre 2012 e início de 2015. Mais de 20 mil exemplares foram avaliados, provendo resultados de análise estática e dinâmica.*

1. Introdução

Programas maliciosos (*malware*) são os principais vetores de ataque contra sistemas computacionais. Nos últimos anos, a motivação maior dos ataques por *malware* é financeira: em 2012, a Federação Brasileira dos Bancos - FEBRABAN estimou perdas de R$ 1,4 bilhões relacionados com fraudes eletrônicas [FEBRABAN 2012]; em 2014, pesquisadores da RSA calcularam que o prejuízo com *malware* de boletos bancários no Brasil alcançou US$ 3,75 bilhões (mais de R$ 10 bilhões em valores atuais) [Marcus 2014]; de acordo com o FBI, entre abril de 2014 e junho de 2015, a atuação do *malware* Cryptowall—que criptografa arquivos da vítima e tenta praticar extorsão para pagamento de um "resgate" pela chave que os libera—causou aproximadamente US$ 18 milhões em perdas para os usuários atacados [The Register 2015].

Independente da motivação por trás do ataque, os exemplares de *malware* evoluem de acordo com peculiaridades da época na qual eles estão inseridos. Isso ocorre devido a diversos fatores, como a necessidade de continuidade do negócio por parte do desenvolvedor de *malware*, modificação ou sofisticação dos mecanismos de segurança, e o contexto do momento (para viabilizar a engenharia social). Observar e identificar tal evolução é um passo importante para ganho de conhecimento sobre a atuação de programas maliciosos, permitindo a descoberta de tendências ao longo do tempo.

Este artigo apresenta uma análise detalhada de mais de 20 mil exemplares de *malware* atuantes no espaço nacional da Internet entre 2012 e 2015, com a finalidade de prover informações acerca das decisões feitas por criadores de *malware* para seu encapsulamento (tipos de arquivo utilizados), das

características obtidas por meio da inspeção do binário (análise estática), do comportamento exibido durante a infecção (análise dinâmica), e de dados extraídos do tráfego de rede. A principal contribuição é prover um panorama sobre *malware* que compromete diariamente os sistemas de usuários brasileiros, causando prejuízos pessoais, em empresas privadas e em instituições públicas.

2. Trabalhos relacionados

A atuação de programas maliciosos é um problema que deve ser continuamente estudado, uma vez que tais programas evoluem e apresentam tendências distintas de comportamento em diferentes localidades geográficas. Esse tipo de pesquisa traz à luz facetas observadas em *malware* em diversas condições e pode auxiliar no desenvolvimento de contra-medidas específicas, aumentando a segurança geral dos usuários e sistemas conectados à Internet. Os trabalhos listados a seguir tratam de *malware* para ambientes *desktop* e móveis. Cabe ressaltar que não se tem notícia de um trabalho que leve em consideração o cenário nacional de forma abrangente, isto é, obtendo estatísticas provenientes de análise estática e dinâmica e durante um intervalo de tempo considerável.

Provos et al. [Provos et al. 2007] mostram os resultados de observação do comportamento de *malware* na Web—exemplares que visitam URLs ou que são obtidos a partir da visita a uma URL contaminada—durante 12 meses (mar/2006–2007). O estudo mostra que os usuários estão sujeitos a terem seus navegadores infectados ao visitar uma grande quantidade de *sites*, podendo inclusive tornarem-se parte de *botnets*. São identificados os mecanismos comumente usados para exploração de navegadores e injeção de conteúdo maliciosos em *sites*, e são discutidas a frequente mudança em binários maliciosos servidos aos usuários como forma de enganar os antivírus e a distribuição desses binários em várias URLs e domínios distintos. Bayer et al. [Bayer et al. 2009] analisaram dinâmicamente mais de 900 mil exemplares únicos de *malware* submetidos ao sistema Anubis[1] por aproximadamente 22 meses (fev/2007–dez/2008), considerando atividades relacionadas com sistema de arquivos, Registro do Windows, rede, aparecimento de janelas e detecção de *sandboxes*.

Abraham e Chengalur-Smith [Abraham and Chengalur-Smith 2010] estudam a ameaça de *malware* que ataca por engenharia social com base em dados de boletins e publicações de empresas de antivírus. Os autores discutem formas comuns de proliferação desse tipo de *malware*—e-mail, *sites* Web, programas de compartilhamento de arquivos e dispositivos de armazenamento portáteis—e as táticas utilizadas, tais como instigação da curiosidade ou ganância, indução de medo, entre outras, bem como o direcionamento dos alvos. Branco et al. [Branco et al. 2012] conduziram um estudo em 2012 sobre técnicas de anti-análise e evasão utilizadas por 4 milhões de exemplares de *malware* analisados estatica e dinamicamente e que, repetido em 2014 [Barbosa and Branco 2014], permitiu a comparação na evolução do comportamento dos exemplares em ambos os períodos.

Afonso et al. [Afonso et al. 2013] analisaram mais de 5 mil aplicações para Android obtidas de lojas (oficial e alternativas) que atendem o mercado brasileiro. O objetivo do trabalho foi buscar comportamento suspeito que pudesse indicar a presença de aplicações maliciosas, como a subversão de permissões, vazamento de informações do dispositivo móvel via rede ou SMS, e acesso a URLs de distribuição de propagandas. Lindorfer et al. [Lindorfer et al. 2014] introduzem o sistema Andrubis para análise de *malware* de Android e provêem estatísticas sobre a análise de mais de 1 milhão de aplicações únicas (≈40% maliciosas) submetidas por um período de 2 anos (jun/2012–jun/2014). O artigo baseou-se em análise estática e dinâmica das aplicações, mostrando as mudanças comportamentais observadas nos exemplares maliciosos e atividades mais frequentes, além de discutir as limitações das técnicas aplicadas.

[1] http://anubis.iseclab.org

3. Coleta de Dados e Testes

Para a análise, foram obtidos 33.811 exemplares, 21.359 deles únicos (com base no *hash* MD5 do binário)[2]. Anexos de e-mail e *crawling* de *links* contidos nessas mensagens e, principalmente, a cooperação com outros pesquisadores com acesso a exemplares em atividade no cenário nacional, serviram como fontes de *malware*. A coleta de dados foi realizada entre 01 de janeiro de 2012 e 31 de março de 2015 (1.186 dias). A Figura 1 exibe a quantidade total de *malware* obtido mensalmente entre 2012 e 2014 (31.745 exemplares). Em 2015, foram coletados 2.066 exemplares no primeiro trimestre. A Figura 2 mostra a distribuição das amostras com base em seu tipo de arquivo (discussão na Seção 4.1.1).

Todos os exemplares foram submetidos para análise estática, dinâmica e por antivírus, a fim de se obter informações presentes no binário por si só, bem como outras exibidas durante sua execução em ambiente controlado. Para a análise estática, foram utilizadas as ferramentas: `Pyew`[3] e `PEframe`[4], as quais permitem a extração de informações básicas do binário e a inspeção por alguns indícios potencialmente maliciosos; `Foremost`[5] para busca de cabeçalhos e *magic numbers* de arquivos embutidos no binário. Para análise dinâmica, utilizou-se sistema próprio [Botacin et al. 2014], ainda não disponível publicamente, que monitora as atividades de exemplares analisados, seus filhos e outros programas através dos quais o *malware* interage com o sistema de arquivos, Registro, processos e rede em sistemas operacionais Windows 7. A análise por antivírus consistiu em enviar os exemplares para o VirusTotal [6] e obter suas taxas de detecção e identificação.

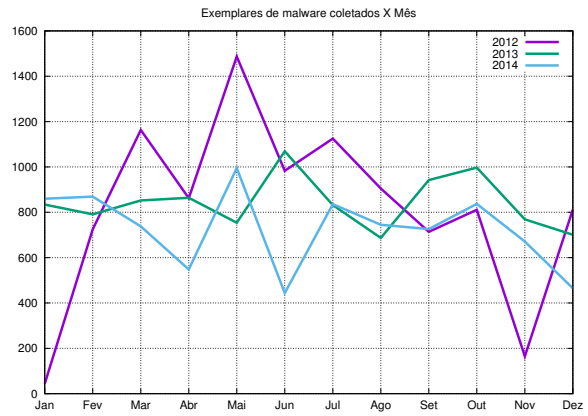

Figura 1. Coleta de amostras ao longo do período observado.

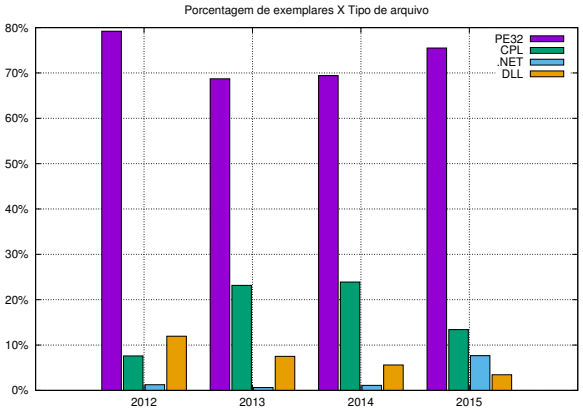

Figura 2. Evolução da distribuição de amostras por tipo de arquivo.

4. Processamento dos Dados

As análises realizadas dividem-se em duas categorias: estáticas e dinâmicas. Na primeira, observa-se os tipos de arquivos, formatos de distribuição e empacotamento usados pelos exemplares, e informações providas por antivírus. Já na análise dinâmica, na qual se obtém o traço de execução dos exemplares e o tráfego de rede correspondente, é feita a verificação se as funções mapeadas estaticamente (*statically linked*) são efetivamente executadas, se as técnicas de anti-análise foram efetivas e o que foi trafegado.

[2]A lista dos *hashes* está disponível em `http://justpaste.it/ltlh`.
[3]`https://github.com/joxeankoret/pyew`
[4]`https://github.com/guelfoweb/peframe`
[5]`http://foremost.sourceforge.net`
[6]`http://www.virustotal.com`

4.1. Informações Estáticas

A análise estática é a avaliação de um exemplar de *malware* sem sua execução. Esse tipo de análise se dá por meio da inspeção do binário e permite a extração de informações do cabeçalho, instruções, texto embutido, uso e tipo de mecanismo de ofuscação (*packer*), número de seções, funções mapeadas, códigos de exploração (*shellcodes*), entre outras.

4.1.1. Tipos de arquivo

A Figura 3 mostra a distribuição dos arquivos por extensão. Embora a maioria dos exemplares tenha a extensão "correta" (.exe, .dll, .cpl), ainda é possível observar a tentativa de ludibriação do usuário pelo uso de uma extensão secundária ou alternativa (e.g. `malware.exe.jpg`). A Figura 4 apresenta o formato real dos arquivos (com base na assinatura presente no cabeçalho). Nota-se uma frequência relativa do tipo PE (executáveis tradicionais) superior às ocorrências de ".exe", o que é esperado devido aos exemplares PE renomeados para outras extensões.

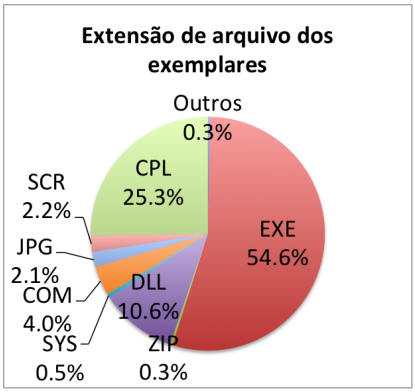

Figura 3. Distribuição por extensão.

Figura 4. Distribuição por tipo de arquivo.

Tradicionalmente, os tipos que têm sido mais usados na propagação de *malware* são executáveis (PE32) e bibliotecas (DLL). Entretanto, a Figura 2 mostra as recentes tendências de aumento das ameaças encapsuladas como arquivos CPL—bibliotecas executáveis do tipo "Painel de Controle"—entre 2012 e 2014, e o rápido crescimento de executáveis do tipo .NET em 2015, os quais podem ser programados para operar em múltiplas plataformas O aumento de *malware* encapsulado em CPL é um fenômeno nacional, identificado também por empresas de antivírus [Mercês 2014].

4.1.2. Chamadas de Função

As chamadas de função presentes em um *malware* podem ser obtidas diretamente do binário, na seção `.text` de seu cabeçalho. A presença destas funções não garante que elas serão executadas, seja pelo uso de um *packer*, evasão da análise pelo *malware*, ou por terem sido inseridas propositalmente no binário para atrapalhar a análise. No entanto, essas funções provêem algumas pistas sobre o comportamento dos exemplares analisados, dado que muitos não aplicam proteções contra análise (vide Seção 4.1.6).

Chamadas a funções que interagem com processos, Registros e o sistema de arquivos são, obviamente, comuns a programas maliciosos e legítimos. Porém, algumas sequências de chamadas são indicativos de atividades potencialmente maliciosas: `GetProcAddress` + `LoadLibrary` + `VirtualAlloc` (e `Free`) + `CreateThread` representa o procedimento para a injeção

de uma biblioteca (*DLL injection*) no escopo de um outro processo, técnica frequentemente usada por *malware* para infectar outros programas em execução no sistema alvo.

Outro exemplo interessante envolve chamadas às funções `Sleep`, `GetTickCount` e `IsDebuggerPresent`, relacionadas a técnicas de anti-análise. A primeira é frequentemente usada para esgotar o tempo de análise em *sandboxes* dinâmicas e nos emuladores dos antivírus. A contagem de *ticks* é utilizada para identificar sistemas de análise através da medição do *overhead* de tempo introduzido pelos mecanismos de inspeção/instrumentação. A função `IsDebuggerPresent` verifica se o programa está em execução em um *debugger*, de forma a tentar impedir a sua engenharia reversa. Na Tabela 1, mostra-se as chamadas de função mais encontradas nos exemplares analisados.

Tabela 1. Chamadas de função mapeadas estaticamente no conjunto de exemplares.

Função	**# Exemplares**	**Função**	**# Exemplares**
GetProcAddress	17317 (81,08%)	RegOpenKeyExA	7599 (35,58%)
LoadLibraryA	16713 (78,25%)	LoadLibraryExA	7045 (32,98%)
VirtualAlloc	15032 (70,38%)	ExitThread	6916 (32,38%)
VirtualFree	15007 (70,26%)	FindResourceA	6216 (29,10%)
Sleep	11812 (55,30%)	GetCurrentProcess	5983 (28,01%)
WriteFile	11545 (54,05%)	VirtualProtect	5448 (25,51%)
UnhandledExceptionFilter	11049 (51,73%)	WinExec	5111 (23,93%)
GetTickCount	9977 (46,71%)	CreateFileW	4910 (22,99%)
CreateThread	9214 (43,14%)	SetWindowsHookExA	4883 (22,86%)
GetCurrentProcessId	8521 (39,89%)	IsDebuggerPresent	3511 (16,44%)
GetWindowThreadProcessId	8142 (38,12%)	InternetCloseHandle	4405 (20,62%)
GetCommandLineA	7659 (35,86%)	InternetReadFile	3777 (17,68%)

4.1.3. Ofuscação

A fim de gerar novas variantes e passar incólumes por mecanismos de proteção, desenvolvedores de *malware* se utilizam de ferramentas chamadas *packers*. Essas ferramentas tinham por objetivo inicial a compressão dos arquivos, mas no caso de programas maliciosos começaram a ser utilizadas para ofuscá-los, tanto por compressão como por criptografia (simples ou de múltiplas camadas) [Jacob et al. 2013]. Com isso, impede-se a análise estática, a menos que se conheça o algoritmo para *unpacking*. O uso da ferramenta de detecção de *packers PEiD* (`http://www.aldeid.com/wiki/PEiD`) resultou na identificação de 57,95% de exemplares ofuscados. Os 42,05% restantes não estão ofuscados ou o *packer* utilizado não possui assinatura conhecida. Em contraponto com o cenário nacional, Martignoni et al. [Martignoni et al. 2007] estimam que ≈80% dos exemplares possuem *packers*. Na Tabela 2, apresenta-se uma comparação dos percentuais relativos a presença de *packers* encontrados no presente artigo e nos trabalhos de [Branco et al. 2012, Barbosa and Branco 2014].

Tabela 2. Uso de *packers* por *malware* ao longo do tempo. Comparação entre os resultados obtidos neste trabalho (T) entre 2012 e 2015 e por Branco (B) em 2012 e 2014 [Branco et al. 2012, Barbosa and Branco 2014].

	2012 (T)	**2012 (B)**	**2013 (T)**	**2014 (T)**	**2014 (B)**	**2015 (T)**
Uso de *packers*	49,28%	34,79%	56,59%	59,96%	37,53%	51,62%

A tabela 3 apresenta os tipos de *packers* encontrados nos exemplares coletados. Compiladores aplicam técnicas distintas de otimização na geração dos binários, deixando assinaturas que os fazem ser detectados como *packers*. Em relação à variedade, nota-se neste artigo que as três primeiras

ocorrências (Borland Delphi, UPX e Microsoft C/C++) correspondem a mais de 94% dos exemplares ofuscados. Comparando-se esses resultados com [Bayer et al. 2009], vê-se que os exemplares coletados internacionalmente possuem maior diversidade: as três ocorrências mais frequentes—UPX, Allaple e FSG—correspondem a 15,61%, 24,83% e 12,28%.

Tabela 3. Tipos de *packers* mais encontrados nos exemplares ofuscados.

Packer	**Exemplares**	***Packer***	**Exemplares**
Borland Delphi	45,90%	NsPack	0,95%
UPX	24,83%	PKLITE32	0,86%
Microsoft C/C++	23,50%	Enigma	0,67%
ASProtect	2,31%	Dev-C++	0,50%
Themida/WinLicense	2,11%	Thinstall	0,47%

Pode-se observar uma predominância de binários compilados com "Delphi", o que se deve a quantidade expressiva de arquivos do tipo CPL. Exemplares de *malware* CPL são majoritariamente criados nessa linguagem (devido à necessidade de janelas e formulários para roubo de credenciais) e têm por alvo principal os usuários de Internet Banking. Há também um número expressivo no uso de UPX, pois é uma ferramenta para compressão livre e de fácil acesso, mesmo para atacantes leigos. Por outro lado, analistas têm a vantagem de poder extrair o *packer*, facilitando a análise estática. Alguns *packers* mais sofisticados, como algumas versões de `Telock`, `Armadillo` e `ASProtect`, podem não executar em sistemas virtualizados ou emulados, inviabilizando a análise dinâmica do *malware* [Dinaburg et al. 2008].

A Tabela 4 mostra a variação geral dos tipos de *packers* entre o período de análise. Observa-se, nos dados de 2015, um aumento considerável de binários gerados a partir do VisualStudio (Microsoft C/C++), o que é compatível com o salto na quantidade de *malware* do tipo .NET visto na Figura 2.

Tabela 4. Evolução dos *packers* mais frequentemente encontrados por ano.

Packer	**2012**	**2013**	**2014**	**2015**
Borland Delphi	35,19%	51,19%	49,66%	43,39%
UPX	31,11%	25,95%	17,36%	8,53%
Microsoft C/C++	22,03%	17,62%	25,97%	44,01%
Themida/WinLicense	5,00%	—	—	—
ASProtect	2,61%	1,54%	2,90%	1,23%
PKLITE32	1,34%	—	—	—
Dev C++	—	1,10%	—	—
NsPack	1,08%	—	1,16%	—

4.1.4. Detecção por Antivírus

A submissão dos exemplares para o VirusTotal resultou em 21.359 retornos com taxa e rótulos de detecção. A porcentagem dos exemplares que foram detectados por pelo menos um antivírus foi de 98,57%. Apenas 307 arquivos (1,43%) não foram detectados por nenhum dos antivírus. A análise dos rótulos de detecção atribuídos ao conjunto de exemplares, considerando o antivírus AVG[7], resultou em mais da metade (54,66%) dos exemplares detectados como `PSW` (*password stealers*) ou `Downloader`. Esse tipo de *malware* utiliza a rede para enviar dados capturadas ou obter componentes. O terceiro rótulo mais atribuído (14,11%) foi `Generic`. Esse termo é muito utilizado quando a detecção é feita por heurística, mas não provê informação sobre o *malware*.

[7]A escolha baseou-se no fato de taxas de detecção alcançadas para o sistema operacional Windows 7 serem as mais altas em comparação com os outros antivírus, de acordo com o teste VB100 mais recente (`https://www.virusbtn.com/vb100/archive/test?order=29&id=207&tab=onDemand`).

4.1.5. *Malware* Modular

Alguns tipos de *malware* são compostos por diversos módulos, de forma a dificultar a chance de que sejam detectados. Esses tipos incluem os chamados *downloaders* e *droppers*: os primeiros são programas simples que, uma vez em execução no alvo, obtêm o real programa malicioso da Internet; os últimos são programas em geral ofuscados que trazem uma carga útil executável dentro de si e, ao serem executados, descarregam o *malware* embutido no sistema (decifrando-o ou descomprimindo-o).

A análise dos exemplares disponíveis revelou 6.145 URLs codificadas estaticamente, potencialmente *downloaders*: 52 referências ao Google, indicando teste de conectividade; 24 à nuvem da Amazon e 54 ao serviço Dropbox, incorrendo no *download* de conteúdo remoto. Para tentar identificar *droppers*, foi feita a extração de arquivos conhecidamente maliciosos embutidos nos binários analisados: 26 exemplares possuíam esse tipo de arquivo que, submetido ao VirusTotal, foi detectado como *malware*; 12 dos executáveis embutidos foram nomeados como "foto", podendo-se inferir que são partes de campanhas de engenharia social para atrair o usuário a acessá-los.

4.1.6. Anti-Análise

Existem exemplares de *malware* que implementam técnicas para detecção da execução em ambientes virtualizados ou emulados (conhecidas como anti-VM), pois tais ambientes são comumente utilizados em sistemas de análise dinâmica publicamente disponíveis. Assim, a aplicação de tais técnicas impede que a análise dinâmica do *malware* seja realizada adequadamente. Fez-se então uma busca pela presença de técnicas de anti-análise nos exemplares, com o objetivo de se identificar detectores de virtualização, tais como chamadas de função (e.g. *CPUID*, que retorna informações específicas do sistema), verificadores de bibliotecas ou *drivers* associados a virtualizadores, *shellcodes* de detecção, entre outras. A Tabela 5 mostra a quantidade de exemplares que aplicam determinada técnica para detecção de ambientes virtualizados.

Tabela 5. Técnicas anti-VM identificadas e exemplares que as implementam.

Técnica	**# de exemplares**	**Técnica**	**# de exemplares**
VMCheck.dll	2.729 (12,77%)	Detecção de VirtualBox	306 (1,43%)
VMware *trick*	843 (3,95%)	Bochs & QEmu CPUID *trick*	267 (1,25%)

Técnicas para identificar *debuggers* (*anti-Dbg*) incluem a chamada à função `IsDebuggerPresent`, identificada em 3.511 exemplares da coleção deste artigo. A Figura 5 ilustra a evolução na aplicação de técnicas anti-análise por *malware* entre 2012 e 2015, incluindo ofuscação por *packers*.

4.2. Informações Dinâmicas

A análise dinâmica consiste na execução dos exemplares maliciosos em um ambiente controlado (*sandbox*), de modo a obter informações do comportamento exibido por *malware* durante a infecção e a superar algumas limitações da análise estática. Neste tipo de análise, pode-se verificar as chamadas de função efetivamente executadas, os parâmetros passados e a relação entre elas. A análise dinâmica dos exemplares foi realizada em ambiente emulado com o sistema operacional Windows 7, por ser o sistema operacional com maior penetração de mercado atualmente[8] ($\approx$60%), utilizando um *driver* de *kernel* para monitorar os subsistemas de Registro, processos e sistema de arquivos.

Todos os exemplares foram submetidos para análise, tendo uma taxa de sucesso de 98,55%, isto é, de exemplares que exibiram comportamento durante a execução no sistema de análise dinâmica utilizado.

[8]http://www.netmarketshare.com/

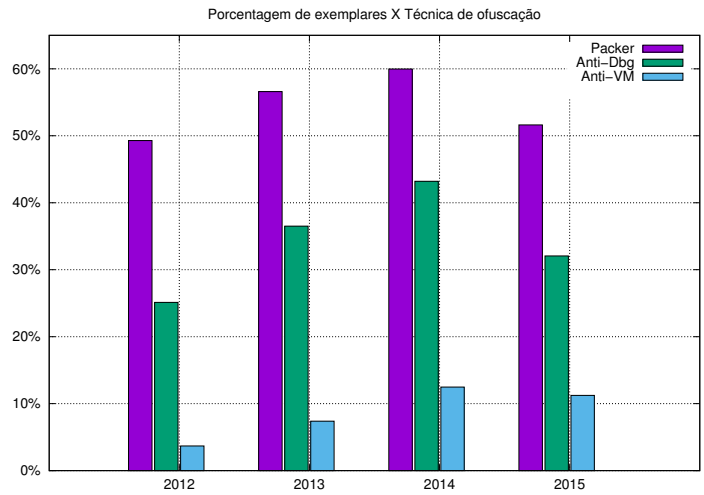

Figura 5. Evolução do uso de técnicas de anti-análise por exemplares (um exemplar pode implementar mais de uma técnica ao mesmo tempo).

Eventuais falhas na execução de *malware* podem ser explicadas por programas corrompidos, uso de técnicas de anti-análise, recursos não encontrados ou indisponíveis (e.g. falha no *download* de um arquivo fundamental para continuidade da infecção ou falta de componente no sistema operacional alvo), etc.

4.2.1. Comportamento Suspeito

Durante a análise dinâmica, algumas atividades suspeitas podem ser observadas, dentre as quais destaca-se: a modificação no arquivo de `hosts`, que pode levar o usuário a *sites* clonados; a remoção de arquivos que, se aplicada ao próprio *malware* pode indicar uma tentativa de evitar a perícia forense de máquinas comprometidas; a modificação de arquivos, possibilitando a inserção de novas funções em bibliotecas ou infecção de outros programas no sistema alvo; a instalação de um *plugin* (BHO) no navegador, infectando-o. A Tabela 6 compara as atividades extraídas das análises realizadas com as mostradas em [Bayer et al. 2009]. As diferenças encontradas enfatizam não só as diferentes épocas de análise, mas a situação brasileira, cuja infestação de exemplares voltados para ganhos financeiros causa um tráfego de rede muito maior (para exfiltração de informações) e menos interação com os subsistemas do alvo (pela necessidade de estímulos do usuário com o *malware* para provisão de informações).

Tabela 6. Comportamentos observados em comparação com [Bayer et al. 2009]

Porcentagem de exemplares		
Comportamento	**Este artigo**	**Bayer et al.**
Modificação no arquivo de *hosts*	0,11%	1,97%
Criação de arquivo	26,23%	70,78%
Remoção de arquivo	13,71%	42,57%
Modificação em arquivo	17,37%	79,87%
Instalação de BHO no IE	1,26%	1,72%
Tráfego de rede	98,82%	55,18%
Criação de chave no Registro	33,67%	64,71%
Criação de Processo	18,79%	52,19%

Um tipo de atividade relevante efetuada sobre o Registro é a inserção de caminhos de programas como valores de chaves, a fim de executá-los durante a inicialização do sistema. Este tipo de comportamento,

também chamado de persitência, é frequentemente utilizado por *keyloggers* ou outros programas espiões para permitir que o *malware* continue atuando após desligamento ou *reboot*. Nos testes realizados, pudemos identificar 38.160 diferentes ocorrências de escritas neste tipo de chave.

4.2.2. Tráfego de Rede

O tráfego de rede é capturado durante a execução de cada *malware* no sistema de análise dinâmica utilizado. A Tabela 7 mostra a distribuição do tráfego, bem como uma comparação com [Bayer et al. 2009].

Tabela 7. Informações extraídas do tráfego de rede deste artigo (T) e de [Bayer et al. 2009]

Porcentagem de exemplares					
Tipo de tráfego	**2012 (T)**	**2013 (T)**	**2014 (T)**	**2015 (T)**	**Bayer et al.**
TCP	40,87%	41,24%	56,19%	65,10%	45,74%
UDP	52,76%	54,74%	52%	58,79%	27,34%
ICMP	1,28%	1,70%	1,33%	1,18%	7,58%
DNS	52,69%	54,73%	51,98%	58,79%	24,53%
HTTP	38,63%	39,69%	52,03%	58,96%	20,75%
SSL	5,30%	5,62%	4,64%	7,99%	0,23%

Outros protocolos também foram avaliados e produziram os seguintes resultados: SMTP (**2012**: 0,21%; **2013**: 0,01%; **2014**: 0,06%; **2015**: 0,17%); MS-SQL (**2012**: 1,40%; **2013**: 0,24%; **2014**: 2,16%; **2015**: 1,77%); FTP (**2012**: 0,21%; **2013**: 0,04%; **2014**: 0,29%; **2015**: 0%). Os resultados de rede, além de compatíveis com os rótulos de antivírus da seção 4.1.4, confirmaram as expectativas sobre a presença considerável de *downloaders* (Seção 4.1.5). A análise mais detalhada do tráfego mostra que os exemplares exibiram o comportamento indicado pela análise estática. Tais comportamentos estão de acordo com os resultados mostrados em [Rossow et al. 2013], bem como o uso de serviços públicos para armazenamento (Amazon e Dropbox). Ressalta-se que o conjunto de exemplares deste trabalho contém 11 das 17 famílias estudadas por Rossow et al.

5. Considerações Finais

Este artigo apresentou informações obtidas da análise estática e dinâmica sobre de mais de 20 mil exemplares únicos de *malware* coletados no espaço da Internet brasileira. Dado que não há na literatura uma visão abrangente do *malware* nacional, os resultados providos permitem a observação de tendências atuais (ameaças .CPL e aumento nas ameaças .NET). Os indícios de crescimento na quantidade de *malware* .NET puderam ser vistos nos dados de 2015. Assim, embora o cenário para este ano ainda não esteja completo, registrou-se um pico de 7% de ocorrências dessas ameaças no primeiro trimestre (contra $\approx 1\%$ nos outros anos). Ademais, a quantidade de exemplares coletadas somente em 2015 já é superior a qualquer um dos períodos completos dos anos anteriores.

Mostrou-se ainda que, se por um lado boa parte das ameaças brasileiras ainda se mostra "desprotegida" (*unarmored*) em relação a aplicação de ofuscação, alguns exemplares apresentam estruturas mais complexas e modulares, como o uso de técnicas de anti-análise e criptografia. Desta forma, a disseminação da implementação de técnicas de anti-análise torna-se um provável próximo passo dos atacantes atuantes no cenário nacional. Por fim, os rótulos de antivírus relacionados a exemplares que enviam ou realizam o *download* de dados corroboram as informações apresentadas sobre o tráfego de rede e a ocorrência de *malware* modular.

6. Agradecimentos

Os autores agradecem o apoio recebido do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq via Projeto MCTI/CNPq/Universal-A 14/2014 (Processo 444487/2014-0).

Referências

Abraham, S. and Chengalur-Smith, I. (2010). An overview of social engineering malware: Trends, tactics, and implications. *Technology in Society*, 32(3):183 – 196.

Afonso, V. M., Grégio, A. R. A., Ellery, E. H. P., Junquera, G. B., Schick, G. A. K., Dahab, R., and de Geus, P. L. (2013). Tendências do Mercado Nacional: procurando malware em aplicações Android . In *Anais do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, SBSeg'13. SBC.

Barbosa, G. N. and Branco, R. R. (2014). Prevalent characteristics in modern malware. `http://www.kernelhacking.com/rodrigo/docs/blackhat2014-presentation.pdf`.

Bayer, U., Habibi, I., Balzarotti, D., Kirda, E., and Kruegel, C. (2009). A view on current malware behaviors. In *Proceedings of the 2Nd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, LEET'09, pages 8–8, Berkeley, CA, USA. USENIX Association.

Botacin, M., Afonso, V., de Geus, P., and Grégio, A. R. A. (2014). Monitoração de comportamento de malware em sistemas operacionais windows nt 6.x de 64 bits. In *SBSeg 2014 - Artigos Completos / Full Papers*, Belo Horizonte (MG).

Branco, R. R., Barbosa, G. N., and Neto, P. D. (2012). Scientific but not academical overview of malware anti-debugging, anti-disassembly and anti- vm technologies. `http://www.kernelhacking.com/rodrigo/docs/blackhat2012-paper.pdf`.

Dinaburg, A., Royal, P., Sharif, M., and Lee, W. (2008). Ether: Malware analysis via hardware virtualization extensions. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, CCS '08, pages 51–62, NY, USA. ACM.

FEBRABAN (2012). FEBRABAN dá dicas de segurança eletrônica. `https://febraban.org.br/Noticias1.asp?id_texto=1886`.

Jacob, G., Comparetti, P. M., Neugschwandtner, M., Kruegel, C., and Vigna, G. (2013). A static, packer-agnostic filter to detect similar malware samples. In *Proceedings of the 9th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, DIMVA'12, pages 102–122, Berlin, Heidelberg. Springer-Verlag.

Lindorfer, M., Neugschwandtner, M., Weichselbaum, L., Fratantonio, Y., van der Veen, V., and Platzer, C. (2014). ANDRUBIS - 1,000,000 Apps Later: A View on Current Android Malware Behaviors. In *Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*.

Marcus, E. (2014). RSA Uncovers Boleto Fraud Ring in Brazil. `https://blogs.rsa.com/rsa-uncovers-boleto-fraud-ring-brazil/`.

Martignoni, L., Christodorescu, M., and Jha, S. (2007). Omniunpack: Fast, generic, and safe unpacking of malware. In *In Proceedings of the Annual Computer Security Applications Conference (ACSAC*.

Mercês, F. (2014). Cpl malware - malicious control panel items. `http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-cpl-malware.pdf`.

Provos, N., McNamee, D., Mavrommatis, P., Wang, K., and Modadugu, N. (2007). The ghost in the browser analysis of web-based malware. In *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, HotBots'07, pages 4–4, Berkeley, CA, USA. USENIX Association.

Rossow, C., Dietrich, C., and Bos, H. (2013). Large-scale analysis of malware downloaders. In *Proceedings of the 9th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, DIMVA'12, pages 42–61. Springer.

The Register (2015). Feds count Cryptowall cost: $18 million says FBI. `http://www.theregister.co.uk/2015/06/24/feds_count_cryptowall_cost_18_million_says_fbi/`.