Malware em JAVA para Acessar Múltiplos Sistemas de Arquivos de Computadores Remotamente

Thiago Henrique de Almeida Espinhara[1]

[1]Universidade Federal Rural de Pernambuco (UFRPE)
Av. Bom Pastor – Boa vista – Garanhuns-PE

thiagohe@gmail.com

Abstract: *In this work we implemented a tool like malware in JAVA programming language to be able to obtain remote access to the infected computer and the application server control over the file system of the client application. We begin with a literature review on the state of malware today, types, their purposes and the strategies used for defense. Next is presented a methodology for the implementation of the malware that invades the file system, listing the features that will be available. Finally the results.*

Resumo: *Neste trabalho foi implementada uma ferramenta do tipo malware na linguagem de programação JAVA para que se conseguisse obter acesso remoto do computador infectado, tendo a aplicação servidor um controle sobre o sistema de arquivos da aplicação cliente. Inicialmente é feita uma revisão bibliográfica sobre o estado dos malwares na atualidade, os tipos, seus propósitos e as estratégias utilizadas para defesa. Em seguida é apresentada uma metodologia para a implementação do malware que invade o sistema de arquivos, listando as funcionalidades que estarão disponíveis. Por fim os resultados obtidos.*

1. Introdução

Os vírus de computadores ou *malwares* estão presentes no nosso cotidiano, atuando como pragas e com um crescimento considerável a cada dia. A cada novo momento estão sendo utilizadas diversas estratégias distintas de ataque a computadores, de forma que os antivírus modernos não utilizam apenas vacinas para um determinado vírus, mas sim analisam o comportamento dos programas em execução em busca de anomalias.

A maioria das contaminações ocorre pela ação do usuário, executando arquivos infectados de diversas formas: recebido como anexo de um e-mail, propagado como um link em redes sociais, através de *pen-drives* ou dispositivos de entrada e saída, etc. Outra causa relevante diz respeito a falhas de segurança do sistema operacional. Daí a importância de mantê-lo sempre atualizado, para que as possíveis vulnerabilidades sejam corrigidas. Vírus mais elaborados tem hora programada para entrar em ação, ficando assim ocultos para o usuário.

Este trabalho objetiva implementar um *malware* na linguagem de programação JAVA, utilizando uma estratégia para acesso remoto não autorizado à arquivos da família de sistemas operacionais Windows. Dessa forma, esse vírus busca a quebra dos princípios da confiabilidade e integridade dos arquivos computador alvo.

2. Malwares

Malware, abreviação de software mal-intencionado, é um software programado para interromper o funcionamento do computador, coletar informações confidenciais ou obter acesso a sistemas de computador particular. Ele pode aparecer sob a forma de código, scripts ou de outro software [5]. *Malware* é um termo geral utilizado para designar uma variedade de formas de softwares hostil ou intruso [7].

Malware inclui vírus, *ransomware*, *worms*, *trojan horses*, *rootkits*, *keyloggers*, *dialers*, *spyware*, *adware*, *BHOs* malicioso, software de segurança desonestos e outros programas maliciosos, a maioria das ameaças de *malware* ativos são geralmente *worms* ou *trojans* em vez de vírus [7]. Existe uma distinção entre *malware* e softwares defeituosos, que são softwares legítimos, mas contém erros prejudiciais que não foram corrigidos antes de sua liberação. No entanto, *malwares* geralmente estão disfarçados como software genuíno, e podem vir de um site oficial da empresa, na forma de um programa útil ou atraente que tem o *malware* prejudicial embutido nele, juntamente com o software de rastreamento adicional que reúne as estatísticas convenientes.

Anti-vírus, anti-*malware* e firewalls são vastamente utilizados pelos usuários em todos os níveis: pequenas e grandes organização, nível de nação e global. Tais softwares podem ajudar a proteger os computadores contra ataques, ajudando a identificar e prevenir a propagação.

2.1 Propósitos

Muitos programas maliciosos, incluindo o primeiro *worm* de Internet, foram escritos como experiências ou brincadeiras. Hoje, os *malwares* são utilizados principalmente para roubar informações confidenciais de importância pessoal, financeira ou de negócios por hackers com intenções prejudiciais.

Malwares são muitas vezes usados de forma ampla contra o governo ou sites corporativos para coletar informações ou para interromper sua operação em geral. No entanto, o *malware* é muitas vezes usado contra indivíduos para obter informações pessoais, como senhas, números de cartões bancários ou de crédito, e assim por diante. Computadores pessoais utilizando rede de computador correm riscos consideráveis. Estes são frequentemente prevenidos por vários tipos de firewalls, software antivírus e hardware de rede.

Desde 2003, a maioria dos vírus e *worms* foram concebidos para assumir o controle de computadores dos usuários para a exploração do mercado negro [6]. São os chamados "computadores zumbis", que são utilizados para enviar SPAM, para hospedar dados de contrabando, como pornografia infantil [1] ou se engajar em ataques distribuídos de negação de serviço como uma forma de extorsão [8].

Outra categoria estritamente para fins lucrativos de *malware* surgiu, chamado de *spyware*. Estes programas são projetados para monitorar a navegação na web dos usuários, exibir propagandas não solicitadas ou redirecionar receitas de marketing da filial para o criador do *spyware*. Os programas de *spyware* não se espalham como vírus, em vez disso eles são geralmente instalados através da exploração de falhas de segurança. Eles também podem ser embalados em conjunto com o software instalado pelo usuário, tais como aplicações *peer-to-peer*.

2.2 Estratégias dos antimalwares

Como os ataques de *malware* se tornaram mais frequentes, a atenção com este tipo de praga começou a mudar. Atualmente existem diversos programas que foram desenvolvidos especificamente para o combate a *malwares*.

Um componente específico do anti-vírus ou do software anti-*malware* comumente referido como o scanner *on-access* ou em tempo real, conecta profundamente o núcleo do sistema operacional ou funções do *kernel* de uma forma similar a como um *malware* que tente operar, embora com a permissão do usuário informado para proteger o sistema. Toda vez que o sistema operacional acessa um arquivo, o scanner *on-access* verifica se o arquivo é um arquivo "legítimo" ou não. Se o arquivo é considerado um *malware* pelo scanner, a operação de acesso será interrompida, o arquivo será tratado pelo scanner em modo pré-definido (como o programa anti-vírus foi configurado durante a instalação) e o usuário será notificado. Isso pode retardar consideravelmente o sistema operacional dependendo de quão bem o scanner foi programado. O objetivo é interromper quaisquer operações que o *malware* possa tentar fazer no sistema antes que elas ocorram, incluindo as atividades que podem explorar *bugs* ou desencadear um comportamento inesperado no sistema operacional.

Os programas anti-*malware* podem combater os *malwares* de duas maneiras:

1. Eles podem fornecer proteção em tempo real contra a instalação de software de *malware* em um computador. Este tipo de proteção contra *malware* funciona da mesma maneira como a de proteção antivírus em que o software anti-*malware* verifica todos os dados de rede de entrada para o software de *malware* e bloqueia quaisquer ameaças que se depara.
2. Programas de software anti-*malware* podem ser usados exclusivamente para detecção e remoção de *malwares* que já foram instalados em um computador. Este tipo de software anti-*malware* verifica o conteúdo do Registro do Windows, arquivos do sistema operacional e programas instalados em um computador e fornecem uma lista de todas as ameaças encontradas, permitindo ao usuário escolher quais arquivos apagar ou manter, ou para comparar esta lista para uma lista de componentes de *malware* conhecidos, removendo arquivos que combinam.

3. *Malware* para Acessar o Sistema de Arquivos

Neste trabalho, foi implementada uma ferramenta para obter acesso remoto não autorizado ao Sistema de Arquivos de diversos Sistemas Operacionais, com o objetivo de comprovar vulnerabilidades de confidencialidade e integridade de arquivos ou diretórios. O requisito básico para o seu funcionamento é que o computador da vítima tenha uma JRE (*Java Runtime Environment)* instalada. O que não é um grande problema, pois as versões atuais dos principais Sistemas Operacionais trazem junto consigo o pacote do Java previamente instalado. O StuffedBiscuit possui duas aplicações: Cliente e Servidor. A aplicação Cliente é o módulo que deve ser executado no computador da vítima, possuindo o código-malicioso. Esta aplicação funciona

basicamente como um cliente-escravo, esperando os comandos enviados pelo servidor, executando-os e devolvendo as respostas. A aplicação Servidor é executada no computador em que se quer obter o controle sobre as máquinas clientes que se conectaram. A ferramenta tem a possibilidade de ter multiconexões, assim, diversas instâncias de conexões com as vítimas (clientes) são mantidas e podem ser acessadas de acordo com a vontade do invasor.

De alguma forma a aplicação cliente precisa ser executada no computador da vítima. Diferentes técnicas podem ser utilizadas para conseguir isto. Após a inicialização do cliente, automaticamente este se conecta ao servidor através de um número de IP (previamente conhecido por ele) e uma porta de comunicação (1234), utilizando para isto a classe java.net.Socket. Neste momento, a aplicação Servidor deve estar pronta para receber novas conexões. Caso novos clientes venham a se conectar após a primeira conexão, o Servidor tem a capacidade de aceitar e manter todas as novas conexões, dando a possibilidade do usuário escolher qual Cliente quer invadir. É importante destacar que esta aplicação é multiplataforma, ou seja, independente do Sistema Operacional sob o qual a aplicação Cliente está executando. Isto por se tratar de um programa Java, que executa sob a Máquina Virtual.

Consumada uma conexão entre as aplicações, o usuário do servidor pode então interagir com o cliente através de comandos, com formato similar aos do MS-DOS do Windows, previamente especificados, que estarão sendo executados remotamente no computador da vítima. Assim, a aplicação servidor pode acessar o sistema de arquivos do Cliente sem que nenhum mecanismo de defesa (firewall, antivírus) possa detectar.

3.1 Funcionalidades Implementadas

O usuário da aplicação Servidor pode interagir com o computador Cliente (vítima) utilizando os comandos descritos na Tabela 1. Estes comandos podem ser executados através de um Console um de cada vez. Quando executados quebram princípios de segurança de confidencialidade e integridade dos dados.

Tabela 1. Comandos Implementados pela Aplicação Servidor

Comando	**Descrição**
dir	Lista todos os diretórios e arquivos indexando-os com um número (para facilitar a usabilidade).
cd *diretório*	Acessa o diretório através do número a ele indexado.
cd..	Retorna ao diretório anterior.
len *arquivo*	Mostra o tamanho em KB do arquivo especificado.

get *arquivo*	Realiza o download do arquivo especificado e salva-o no mesmo diretório da aplicação servidor.
del *arquivo*	Deleta o arquivo especificado.
del all	Deleta todos os arquivos, pastas e subpastas do diretório atual.
ping *inicio fim*	Realiza um ping remoto de acordo com a faixa de IPs passada como parâmetro.
list	Lista todos os computadores conectados.
con *ip*	Conecta ao IP especificado, alternando o controle.
exit	Termina a conexão e execução das aplicações Cliente e Servidor.

3.2 Arquitetura do sistema

A arquitetura do sistema baseia-se basicamente no modelo Cliente-Servidor (Master/Slave), onde existe a comunicação entre Cliente e Servidor através da Internet ou qualquer Rede de Computador que utilize o protocolo TCP/IP. Para o StuffedBiscuit em particular, a aplicação Cliente trata-se de um cliente-escravo, onde seu objetivo é realizar ações recebidas pelo Servidor e enviar-lhe a resposta adequada. Isto pode ser visualizado na Figura 1, onde o módulo Slave do Cliente recebe as informações do módulo de Rede. A comunicação em rede do sistema utiliza Socket, trazendo um elo bidirecional de comunicação. No Servidor, o usuário pode interagir com o sistema através de comandos em uma Interface que utiliza o Console (com aparência similar a um prompt de comando do Windows). Estes comandos são passados para o Controlador Master, que os processa e envia-os para o módulo de Rede do Servidor, responsável por transmitir o comando ao Cliente e aguardar a resposta. Quando a resposta é recebida, o resultado do comando é exibido na tela do console para o usuário.

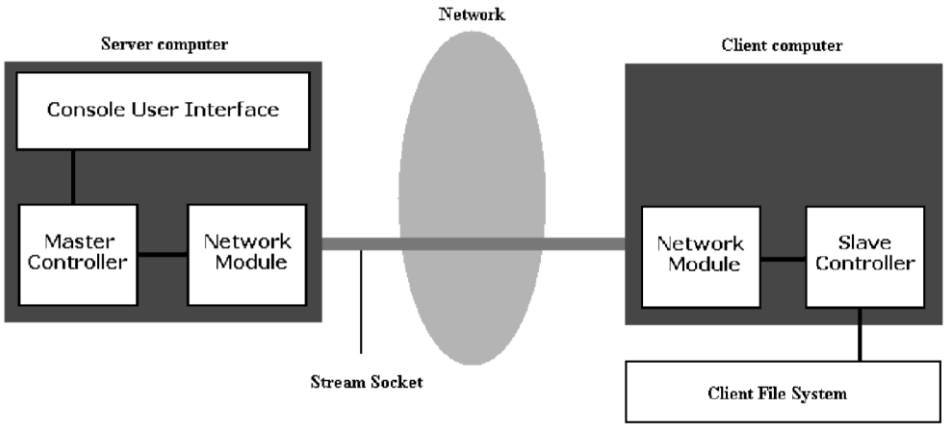

Figura 1. Arquitetura do Sistema.

3.3 Resultados

Foi observado que a aplicação cliente é de extremo risco para os usuários que possuem a máquina virtual do Java instalado, independente do Sistema Operacional que esteja em uso, demonstrando uma falha de segurança gravíssima em Java. Para constatar tal afirmação, foram realizados dois experimentos: o primeiro com máquinas com o Windows 7 e o segundo com Linux Ubuntu. Os resultados foram surpreendentes, onde foi possível obter acesso remoto aos arquivos dos usuários sem que nenhum antivírus, *firewall* ou restrições de acesso dos Sistemas Operacionais impedissem o funcionamento. Assim, mesmo o vírus executando sob usuários sem poder de administrador, o acesso e a modificação aos dados foi possível.

A funcionalidade mais perigosa implementada foi a denominada *del all*, que quando executada excluía todos os arquivos e pastas do diretório atual. A Figura 2 ilustra uma ação dessa função. As figuras 2.a e 2.c são as capturas de tela da aplicação servidor antes e depois da execução do comando, respectivamente. As figuras 2.b e 2.d são as capturas de tela do cliente (computador da vítima) antes e depois da ação executada pelo servidor.

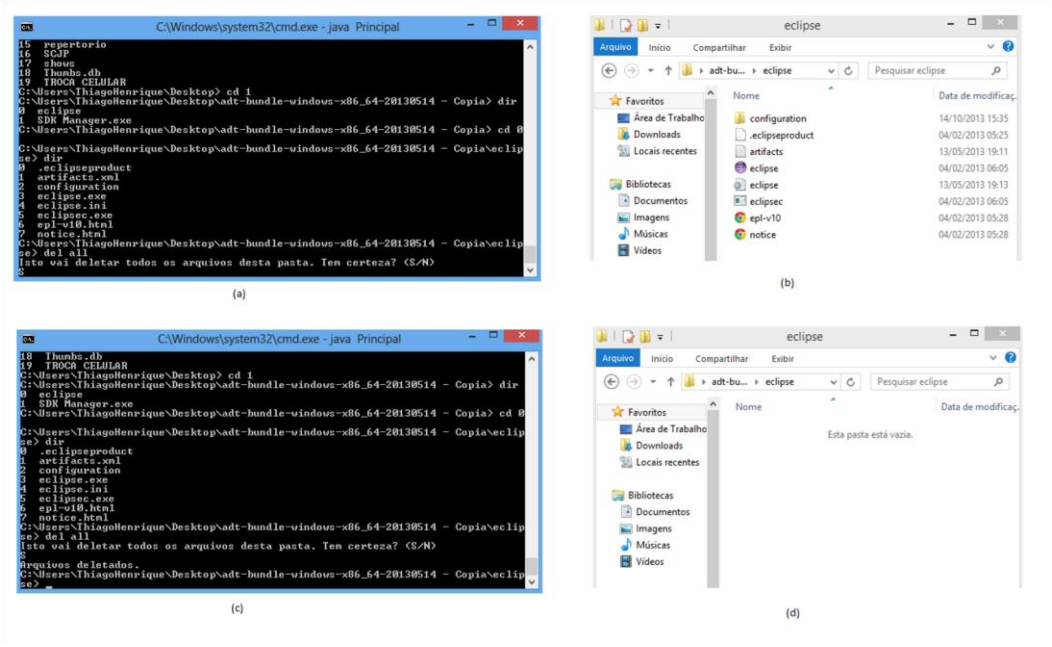

Figura 2. Ação da Funcionalidade *del all*.

4. Considerações Finais

O código produzido para a realização deste projeto foi desenvolvido para fins de pesquisa acadêmica. A intenção é de se estudar vulnerabilidades nos Sistemas Operacionais, através das redes de computadores, utilizando a linguagem de programação JAVA, e não para utilização em atividades criminosas, como previsto na Lei 12.737.

Observou-se que a aplicação do vírus não teve o seu funcionamento barrado por nenhum tipo de proteção, como antivírus ou *firewall* de ambos os Sistemas Operacionais testados: Windows e Linux. Esta não detecção se dá provavelmente por

dois motivos: (a) A aplicação não abre portas no computador da vítima (para entrada), fazendo uma conexão TCP através de um Socket (referenciando portas de saída). O bloqueio de pacotes de saída é mais complicado porque, embora muitos sistemas adotem convenções padrão para numeração de portas, eles não são obrigados a fazê-lo [13]. Para alguns serviços importantes, como FTP (*File Transfer Protocol*), esta atribuição é feita dinamicamente. (b) Todo programa JAVA executa sobre sua Máquina Virtual, o que pode dificultar na identificação do vírus. Desta forma, a portabilidade do vírus em relação ao Sistema Operacional ficou garantida, por utilizar JAVA. Ou seja, através deste é possível ter acesso simultâneo ao Sistema de Arquivos de quaisquer Sistema Operacional, executando os comandos *list* e *con* da Tabela 1 para realizar a migração.

Referências

[1] Child Porn: Malware's Ultimate Evil". 2009.

[2] Continuing Business with Malware Infected Customers". Gunter Ollmann. 2008.

[3] F-Secure Reports Amount of Malware Grew by 100% during 2007" (Press release). F-Secure Corporation. 2007.

[4] F-Secure Quarterly Security Wrap-up for the first quarter of 2008". F-Secure.

[5] ICS-CERT. "An Undirected Attack Against Critical Infrastructure". Us-cert.gov. <http://ics-cert.us-cert.gov/pdf/undirected_attack0905.pdf>. Acessado em 24 de maio de 2013.

[6] Malware Revolution: A Change in Target". 2007.

[7] Microfost. TechNet library. "Defining Malware: FAQ". <http://technet.microsoft.com/en-us/library/dd632948.aspx>. Acessado em 24 de maio de 2013.

[8] PC World - Zombie PCs: Silent, Growing Threat.

[9] Symantec Internet Security Threat Report: Trends for July–December 2007 (Executive Summary) (PDF) XIII. Symantec Corp. 2008.

[10] New Research Shows Remote Users Expose Companies to Cybercrime". Webroot. 2013.

[11] Symantec names Shaoxing, China as world's malware capital". Engadget. 15/04/2010.

[12] Rooney, Ben (23/05/2011). "Malware Is Posing Increasing Danger". Wall Street Journal.

[13] Tanenbaum, A. S. Redes de computadores. Tradução Vamdemberg D. de Souza.

13ª edição. Elsevier. Rio de Janeiro. p. 827. 1994.